

---

# Amazon Elastic Compute Cloud

## Linux インスタンス用ユーザーガイド



## Amazon Elastic Compute Cloud: Linux インスタンス用ユーザーガイド

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

Amazon EC2 とは .....	1
Amazon EC2 の機能 .....	1
Amazon EC2 の使用を開始する方法 .....	1
関連サービス .....	2
Amazon EC2 へのアクセス .....	3
Amazon EC2 の料金表 .....	3
PCI DSS コンプライアンス .....	4
インスタンスと AMI .....	4
インスタンス .....	5
AMI .....	6
リージョン、アベイラビリティゾーン、および ローカルゾーン .....	7
リージョン、アベイラビリティゾーン、および ローカルゾーン の概念 .....	7
利用できるリージョン .....	10
リージョンとエンドポイント .....	11
リージョン、アベイラビリティゾーン、および ローカルゾーン の確認 .....	11
リソースのリージョンの指定 .....	13
ローカルゾーン の有効化 .....	14
ローカルゾーン の無効化 .....	15
アベイラビリティゾーンまたは ローカルゾーン でのインスタンスの起動 .....	15
別のアベイラビリティゾーンへのインスタンスの移行 .....	15
ルートデバイスボリューム .....	16
ルートデバイストレージの概念 .....	16
ルートデバイスタイプによる AMI の選択 .....	18
インスタンスのルートデバイスタイプの判別 .....	19
永続的ルートデバイスボリュームへの変更 .....	19
セットアップ .....	22
AWS にサインアップする .....	22
キーペアを作成する .....	22
セキュリティグループの作成 .....	24
ご利用開始にあたって .....	26
概要 .....	26
前提条件 .....	27
ステップ 1: インスタンスを起動する .....	27
ステップ 2: インスタンスに接続 .....	28
ステップ 3: インスタンスをクリーンアップする .....	28
次のステップ .....	29
ベストプラクティス .....	30
チュートリアル .....	32
LAMP サーバーをインストールする (Amazon Linux 2) .....	32
ステップ 1: LAMP サーバーを準備する .....	33
ステップ 2: LAMP サーバーをテストする .....	36
ステップ 3: データベースサーバーをセキュリティで保護する .....	37
ステップ 4: (オプション) phpMyAdmin をインストールする .....	38
トラブルシューティング .....	41
関連トピック .....	41
LAMP サーバーをインストールする (Amazon Linux AMI) .....	42
ステップ 1: LAMP サーバーを準備する .....	42
ステップ 2: LAMP サーバーをテストする .....	46
ステップ 3: データベースサーバーをセキュリティで保護する .....	48
ステップ 4: (オプション) phpMyAdmin をインストールする .....	49
トラブルシューティング .....	52
関連トピック .....	53
チュートリアル: WordPress ブログのホスティング .....	53
前提条件 .....	54

WordPress のインストール .....	54
次のステップ .....	60
ヘルプ! パブリック DNS 名が変更されたため、ブログが壊れました .....	61
チュートリアル: Amazon Linux 2 に SSL/TLS を設定する .....	62
前提条件 .....	62
ステップ 1: サーバーでの TLS の有効化 .....	63
ステップ 2: CA 署名証明書の取得 .....	65
ステップ 3: セキュリティ設定のテストと強化 .....	70
トラブルシューティング .....	72
Certificate Automation: Amazon Linux 2 での Let's Encrypt と Certbot の使用 .....	73
チュートリアル: Amazon Linux に SSL/TLS を設定する .....	77
前提条件 .....	77
ステップ 1: サーバーでの TLS の有効化 .....	78
ステップ 2: CA 署名証明書の取得 .....	80
ステップ 3: セキュリティ設定のテストと強化 .....	84
トラブルシューティング .....	86
Certificate Automation: Amazon Linux での Let's Encrypt と Certbot の使用 .....	87
チュートリアル: アプリケーションの可用性の向上 .....	90
前提条件 .....	91
アプリケーションのスケーリングと負荷分散 .....	91
ロードバランサーをテストする .....	93
Amazon マシンイメージ .....	94
AMI の使用 .....	94
独自の AMI の作成 .....	94
AMI の購入、共有、販売 .....	95
AMI の登録解除 .....	95
Amazon Linux 2 および Amazon Linux AMI .....	95
AMI タイプ .....	95
起動許可 .....	96
ルートデバイスのストレージ .....	96
仮想化タイプ .....	98
Linux AMI の検索 .....	100
Amazon EC2 コンソールを使用した Linux AMI の検索 .....	100
AWS CLI を使用した AMI の検索 .....	101
Systems Manager を使用して最新の Amazon Linux AMI を検索する .....	101
クイックスタート AMI の検索 .....	101
共有 AMI .....	102
共有 AMI を見つける .....	103
AMI を一般公開する .....	105
特定の AWS アカウントと AMI を共有する .....	106
ブックマークの使用 .....	107
共有 Linux AMI のガイドライン .....	108
有料 AMI .....	112
ご自分の AMI を販売する .....	113
有料 AMI を見つける .....	113
有料 AMI の購入 .....	114
インスタンスの製品コードを取得する .....	114
有料サポートの利用 .....	115
有料およびサポート対象の AMI の請求書 .....	115
AWS Marketplace サブスクリプションの管理 .....	115
Amazon EBS-Backed Linux AMI の作成 .....	116
Amazon EBS-Backed AMI の作成の概要 .....	116
インスタンスからの Linux AMI の作成 .....	117
スナップショットからの Linux AMI の作成 .....	119
Instance Store-Backed Linux AMI の作成 .....	119
Instance Store-Backed AMI の作成プロセスの概要 .....	120
前提条件 .....	120

AMI ツールを設定する .....	121
Instance Store-Backed インスタンスから AMI を作成する .....	124
Amazon EBS-Backed AMI への変換 .....	131
AMI ツールリファレンス .....	134
EBS-Backed AMI での暗号化の利用 .....	151
インスタンスの起動シナリオ .....	151
イメージコピーのシナリオ .....	154
AMI のコピー .....	155
Instance Store-Backed AMI をコピーするアクセス許可 .....	156
リージョン間のコピー .....	157
アカウント間のコピー .....	158
暗号化とコピー .....	158
AMI のコピー .....	159
保留中の AMI コピー操作を中止する .....	160
請求情報の取得 .....	161
AMI 請求情報フィールド .....	161
プラットフォーム詳細および使用状況オペレーション請求コード .....	161
プラットフォーム詳細および請求情報の表示 .....	162
Linux AMI の登録解除 .....	163
Amazon EBS-Backed AMI のクリーンアップ .....	163
Instance Store-Backed AMI をクリーンアップする .....	164
Amazon Linux .....	165
Amazon Linux の入手可能性 .....	166
Amazon Linux インスタンスへの接続 .....	166
Amazon Linux イメージの特定 .....	166
AWS コマンドラインツール .....	167
パッケージリポジトリ .....	168
Extras Library (Amazon Linux 2) .....	170
参照のためのソースパッケージへのアクセス .....	171
cloud-init .....	171
Amazon Linux 通知にサブスクライブする .....	173
Amazon Linux 2 を仮想マシンとしてオンプレミスで実行する .....	174
ユーザー提供カーネル .....	176
HVM AMI (GRUB) .....	176
AMI の準仮想化 (PV-GRUB) .....	177
インスタンス .....	183
インスタンスタイプ .....	183
利用可能なインスタンスタイプ .....	184
ハードウェア仕様 .....	186
AMI 仮想化タイプ .....	186
Nitro ベースのインスタンス .....	187
ネットワーキング機能とストレージ機能 .....	187
インスタンス制限 .....	190
汎用インスタンス .....	190
コンピュート最適化インスタンス .....	231
メモリ最適化インスタンス .....	236
ストレージ最適化インスタンス .....	245
高速コンピューティングインスタンス .....	253
インスタンスタイプの検索 .....	266
インスタンスタイプを変更する .....	267
レコメンデーションの取得 .....	271
インスタンス購入オプション .....	274
インスタンスのライフサイクルの決定 .....	275
オンデマンドインスタンス .....	276
リザーブドインスタンス .....	279
スケジュールされたインスタンス .....	317
スポットインスタンス .....	320

Dedicated Hosts .....	395
ハードウェア専有インスタンス .....	425
オンデマンドキャパシティー予約 .....	431
インスタンスのライフサイクル .....	443
インスタンスの作成 .....	444
インスタンスの停止と起動 (Amazon EBS-Backed インスタンスのみ) .....	445
インスタンスの休止 (Amazon EBS Backed インスタンスのみ) .....	445
インスタンスの再起動 .....	446
インスタンスのリタイア .....	446
インスタンスの削除 .....	446
再起動、停止、休止、終了の違い .....	447
起動する .....	448
接続 .....	505
停止と起動 .....	529
休止 .....	532
再起動 .....	542
リタイア .....	543
終了 .....	545
復旧 .....	551
インスタンスの設定 .....	552
一般的な設定シナリオ .....	553
ソフトウェアの管理 .....	553
ユーザーの管理 .....	559
プロセッサのスタート制御 .....	561
時刻の設定 .....	567
CPU オプションの最適化 .....	571
ホスト名の変更 .....	583
動的な DNS のセットアップ .....	586
起動時にコマンドを実行 .....	588
インスタンスマタデータとユーザーデータ .....	593
Elastic Inference .....	623
インスタンスの特定 .....	623
インスタンスアイデンティティドキュメントの調査 .....	623
システム UUID の確認 .....	623
モニタリング .....	625
自動モニタリングと手動モニタリング .....	626
自動モニタリングツール .....	626
手動モニタリングツール .....	627
モニタリングのベストプラクティス .....	627
インスタンスのステータスのモニタリング .....	628
インスタンスステータスのチェック .....	628
予定されているイベント .....	633
CloudWatch を使用したインスタンスのモニタリング .....	642
詳細モニタリングを有効化 .....	642
利用可能なメトリクスのリスト表示 .....	644
メトリクスの統計情報を取得する .....	654
メトリクスをグラフ化 .....	662
アラームの作成 .....	662
インスタンスを停止、終了、再起動、または復旧するアラームを作成する .....	663
CloudWatch イベントによる Amazon EC2 の自動化 .....	672
メモリとディスクのメトリクスのモニタリング .....	673
CloudWatch エージェント .....	673
CloudWatch モニタリングスクリプト .....	673
AWS CloudTrail を使用した API コールのログ作成 .....	681
CloudTrail での Amazon EC2 と Amazon EBS に関する情報 .....	681
Amazon EC2 および Amazon EBS のログファイルエントリの概要 .....	682
EC2 Instance Connect を介して接続するユーザーを監査する .....	683

ネットワーク .....	685
インスタンスの IP アドレッシング .....	685
プライベート IPv4 アドレスと内部 DNS ホスト名 .....	685
パブリック IPv4 アドレスと外部 DNS ホスト名 .....	686
Elastic IP アドレス (IPv4) .....	687
Amazon DNS サーバー .....	687
IPv6 アドレス .....	687
インスタンスの IP アドレスの使用 .....	688
複数の IP アドレス .....	693
自分の IP アドレスを使用する .....	701
要件 .....	701
AWS アカウントにアドレス範囲を持ち込むための準備 .....	702
AWS で使用するためのアドレス範囲のプロビジョニング .....	703
AWS からアドレス範囲の公開 .....	704
アドレス範囲のプロビジョニング解除 .....	704
Elastic IP アドレス .....	705
Elastic IP アドレスの基本 .....	705
Elastic IP アドレスの操作 .....	706
電子メールアプリケーションでの逆引き DNS の使用 .....	712
Elastic IP アドレスの制限 .....	712
ネットワークインターフェイス .....	713
ネットワークインターフェイスの基本 .....	713
各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数 .....	714
ネットワークインターフェイスのシナリオ .....	724
ネットワークインターフェイスの設定に関するベストプラクティス .....	726
ネットワークインターフェイスでの作業 .....	727
リクエスマネージド型のネットワークインターフェイス .....	736
拡張ネットワーキング .....	737
拡張ネットワーキングのタイプ .....	738
インスタンスでの拡張ネットワーキングの有効化 .....	738
拡張ネットワーキング: ENA .....	738
拡張ネットワーキング: インテル 82599 VF .....	751
ENA のトラブルシューティング .....	757
Elastic Fabric Adapter .....	763
EFA の基本 .....	764
サポートされたインターフェイスとライブラリ .....	765
サポートされるインスタンスタイプ .....	765
サポート対象の AMI .....	765
EFA の制限事項 .....	765
EFA および MPI の開始方法 .....	765
EFA および NCCL の開始方法 .....	772
EFA の使用 .....	788
EFA のモニタリング .....	791
プレイスメントグループ .....	791
クラスタープレイスメントグループ .....	792
パーティションプレイスメントグループ .....	793
スプレッドプレイスメントグループ .....	793
プレイスメントグループのルールと制限 .....	794
プレイスメントグループの作成 .....	795
プレイスメントグループでのインスタンスの起動 .....	796
プレイスメントグループのインスタンスを説明する .....	798
インスタンスのプレイスメントグループの変更 .....	799
プレイスメントグループを削除する .....	800
ネットワーク MTU .....	801
ジャンボフレーム (9001 MTU) .....	801
パス MTU 検出 .....	802
2 つホスト間のパス MTU の確認 .....	802

Linux インスタンス上の MTU の確認および設定 .....	803
トラブルシューティング .....	804
Virtual Private Cloud .....	804
Amazon VPC ドキュメント .....	804
EC2-Classic .....	804
サポートされるプラットフォームの検出 .....	805
EC2-Classic で利用可能なインスタンスタイプ .....	806
EC2-Classic と VPC の違い .....	806
EC2-Classic と VPC との間でのリソースの共有とアクセス .....	811
ClassicLink .....	812
EC2-Classic から VPC への移行 .....	825
セキュリティ .....	836
インフラストラクチャセキュリティ .....	836
ネットワークの隔離 .....	837
物理ホストでの隔離 .....	837
ネットワークトラフィックの制御 .....	837
耐障害性 .....	838
データ保護 .....	838
保管時の暗号化 .....	839
転送中の暗号化 .....	839
Identity and Access Management .....	839
インスタンスへのネットワークアクセス .....	840
Amazon EC2 のアクセス許可属性 .....	840
IAM および Amazon EC2 .....	840
IAM ポリシー .....	842
IAM ロール .....	888
ネットワークアクセス .....	897
キーペア .....	899
Amazon EC2 を使用してキーペアを作成する .....	901
独自のパブリックキーを Amazon EC2 にインポートする .....	902
キーペアのパブリックキーを取得する (Linux) .....	903
キーペアのパブリックキーを取得する (Windows) .....	904
インスタンスからキーペアのパブリックキーを取得する .....	904
キーペアのフィンガープリントの確認 .....	905
キーペアの削除 .....	906
インスタンスのキーペアの追加または交換 .....	907
プライベートキーを紛失した場合の Linux インスタンスへの接続 .....	907
セキュリティグループ .....	911
セキュリティグループのルール .....	912
デフォルトのセキュリティグループ .....	914
カスタムのセキュリティグループ .....	915
セキュリティグループを操作する .....	915
セキュリティグループのルールのリファレンス .....	919
更新管理 .....	926
コンプライアンス検証 .....	926
ストレージ .....	928
Amazon EBS .....	929
Amazon EBS の機能 .....	930
EBS ボリューム .....	931
EBS スナップショット .....	970
EBS のデータサービス .....	1003
EBS ボリュームと NVMe .....	1027
EBS 最適化 .....	1031
EBS パフォーマンス .....	1044
EBS CloudWatch メトリクス .....	1060
EBS CloudWatch イベント .....	1066
インスタンスストア .....	1076

インスタンスストアの存続期間 .....	1077
インスタンスストアボリューム .....	1078
インスタンスストアボリュームを追加する .....	1083
SSD インスタンスストアボリューム .....	1086
インスタンスストアアップボリューム .....	1087
ディスクパフォーマンスの最適化 .....	1090
ファイルストレージ .....	1091
Amazon EFS .....	1091
Amazon FSx .....	1095
Amazon S3 .....	1095
Amazon S3 および Amazon EC2 .....	1096
インスタンスピリューム数の制限 .....	1097
Linux 固有のボリュームの制限 .....	1097
Windows 固有のボリュームの制限 .....	1097
インスタンスタイプの制限 .....	1098
帯域幅と容量 .....	1098
デバイスの名前付け .....	1098
使用できるデバイス名 .....	1099
デバイス名に関する考慮事項 .....	1099
ロックデバイスマッピング .....	1100
ロックデバイスマッピングの概念 .....	1100
AMI ロックデバイスマッピング .....	1103
インスタンスロックデバイスマッピング .....	1105
リソースとタグ .....	1110
リソースの場所 .....	1110
リソース ID .....	1111
長い ID の使用 .....	1112
長い ID 設定に対するアクセスの制御 .....	1115
リソースのリスト表示とフィルタリング .....	1116
高度な検索 .....	1116
コンソールを使用してリソースをリスト表示する .....	1117
コンソールを使用してリソースをフィルタリングする .....	1118
CLI および API を使用した一覧表示とフィルタリング .....	1119
リソースにタグを付ける .....	1120
タグの基本 .....	1120
リソースにタグを付ける .....	1121
タグの制限 .....	1124
請求用のリソースにタグを付ける .....	1125
コンソールでのタグの処理 .....	1125
CLI または API でのタグの操作 .....	1128
サービス制限 .....	1130
現在の制限を表示する .....	1131
制限の引き上げのリクエスト .....	1131
ポート 25 を使用した E メール送信に関連する制限 .....	1132
使用状況レポート .....	1132
トラブルシューティング .....	1133
起動の問題のトラブルシューティング .....	1133
インスタンス制限の超過 .....	1133
インスタンス容量の不足 .....	1134
インスタンスがすぐに削除される .....	1134
インスタンスへの接続 .....	1135
インスタンスへの接続エラー: 接続タイムアウト .....	1136
エラー: キーをロードできません ... Expecting: ANY PRIVATE KEY .....	1138
エラー: ユーザーキーがサーバーによって認識されない .....	1138
エラー: Host key not found、Permission denied (publickey)、または Authentication failed, permission denied (ホストキーが見つかりません、権限の拒否 (publickey)、または認証失敗、権限の拒否) .....	1140

エラー: Unprotected Private Key File (保護されていないプライベートキーファイル) .....	1141
エラー: プライベートキーの先頭は「----BEGIN RSA PRIVATE KEY----」、末尾は「----END RSA PRIVATE KEY----」にする必要があります .....	1142
エラー: Server refused our key または No supported authentication methods available (サーバーはキーを拒否しましたまたは利用可能なサポートされる認証方法はありません) .....	1142
ブラウザを使用して接続できない .....	1142
インスタンスに対して Ping を実行できない .....	1143
エラー: サーバーによる予期しないネットワーク接続の閉鎖 .....	1143
インスタンスの停止 .....	1143
代わりのインスタンスの作成 .....	1144
インスタンスの削除 .....	1145
インスタンスの削除の遅延 .....	1145
表示されているインスタンスを削除する .....	1145
インスタンスを自動的に起動または終了する .....	1145
失敗したステータスチェック .....	1146
ステータスチェック情報の確認 .....	1146
システムログの取得 .....	1147
Linux ベースのインスタンスに関するシステムログエラーのトラブルシューティング .....	1148
メモリ不足: プロセスの終了 .....	1148
エラー: mmu_update failed (メモリ管理の更新に失敗しました) .....	1149
I/O エラー (ブロックデバイス障害) .....	1150
I/O エラー: ローカルでもリモートディスクでもありません (破損した分散ブロックデバイス) .....	1151
request_module: runaway loop modprobe (古い Linux バージョンでレガシーカーネル modprobe がループしている) .....	1152
「FATAL: kernel too old」および「fsck: No such file or directory while trying to open /dev」(カーネルと AMI の不一致) .....	1153
「FATAL: Could not load /lib/modules」または「BusyBox」(カーネルモジュールの欠如) .....	1153
エラー: 無効のカーネル (EC2 と互換性のないカーネル) .....	1155
fsck: No such file or directory while trying to open... (ファイルシステムが見つからない。) .....	1156
General error mounting filesystems (マウント失敗) .....	1157
VFS: Unable to mount root fs on unknown-block (ルートファイルシステム不一致) .....	1159
Error: Unable to determine major/minor number of root device... (ルートファイルシステム/デバイス不一致) .....	1160
XENBUS: Device with no driver... .....	1161
... days without being checked, check forced (ファイルシステムのチェックが必要です) .....	1162
fsck died with exit status... (デバイスが見つからない) .....	1162
GRUB プロンプト (grubdom>) .....	1163
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (\アードコードされた MAC アドレス) .....	1165
SELinux ポリシーを読み込めません。Machine is in enforcing mode. Halting now. (SELinux の誤設定) .....	1166
XENBUS: Timeout connecting to devices (Xenbus タイムアウト) .....	1167
到達できないインスタンスのトラブルシューティング .....	1168
インスタンスの再起動 .....	1168
インスタンスコンソール出力 .....	1168
接続できないインスタンスのスクリーンショットの取得 .....	1169
ホストコンピュータに障害が発生した場合のインスタンスの復旧 .....	1170
間違ったボリュームで起動する .....	1171
Linux 用 EC2Rescue .....	1172
Linux 用 EC2Rescue のインストール .....	1172
(省略可能) Linux 用 EC2Rescue の署名を検証します。 .....	1173
Linux 用 EC2Rescue の使用 .....	1175
EC2Rescue モジュールを開発する .....	1177
診断割り込みの送信 .....	1182
サポートされるインスタンスタイプ .....	1182
前提条件 .....	1182
診断割り込みの送信 .....	1185

ドキュメント履歴 .....	1186
----------------	------

# Amazon EC2 とは

Amazon Elastic Compute Cloud (Amazon EC2) は、アマゾン ウェブ サービス (AWS) クラウドでサイズが変更できるコンピューティングキャパシティーを提供します。Amazon EC2 の使用により、ハードウェアに事前投資する必要がなくなり、アプリケーションをより早く開発およびデプロイできます。Amazon EC2 を使用して必要な数 (またはそれ以下) の仮想サーバーを起動して、セキュリティおよびネットワーキングの設定と、ストレージの管理を行います。Amazon EC2 では、要件変更や需要増に対応して迅速に拡張または縮小できるため、サーバートラフィック予測が不要になります。

クラウドコンピューティングの詳細については、「[クラウドコンピューティングとは](#)」を参照してください。

## Amazon EC2 の機能

Amazon EC2 には次の機能があります。

- インスタンスと呼ばれる仮想コンピューティング環境
- サーバーに必要なビットをパッケージ化した(オペレーティングシステムおよび追加のソフトウェアを含む)、Amazon Machine Image (AMI) と呼ばれる、インスタンス用に事前に設定されたテンプレート。
- インスタンスタイプと呼ばれる、インスタンス用の CPU、メモリ、ストレージ、ネットワーキングキャパシティーのさまざまな構成
- キーペアを使用したインスタンス用の安全なログイン情報 (AWS はパブリックキーを保存し、ユーザーはプライベートキーを安全な場所に保存します)。
- インスタンスストアボリュームと呼ばれる、インスタンスを停止または終了するときに削除される一時データ用のストレージボリューム
- Amazon EBS ボリュームと呼ばれる、Amazon Elastic Block Store (Amazon EBS) を使用したデータ用の永続的ストレージボリューム
- リージョンおよびアベイラビリティーゾーンと呼ばれる、インスタンスや Amazon EBS ボリュームなどのリソース用の複数の物理的な場所
- セキュリティグループを使用してインスタンスに到達可能で、プロトコル、ポート、ソース IP 範囲を指定できるファイアウォール
- Elastic IP アドレスと呼ばれる、動的クラウドコンピューティング用の静的な IPv4 アドレス
- タグと呼ばれ、作成して Amazon EC2 リソースに割り当てることができるメタデータ
- 残りの AWS クラウドから論理的に分離され、ユーザー独自のネットワークにオプションで接続できる、仮想プライベートクラウド (VPC) と呼ばれる仮想ネットワーク

Amazon EC2 の機能の詳細については、「[Amazon EC2 の製品ページ](#)」を参照してください。

AWS でのウェブサイトの実行の詳細については、「[ウェブホスティング](#)」を参照してください。

## Amazon EC2 の使用を開始する方法

まず、Amazon EC2 を使用するようにセットアップする必要があります。セットアップが終了したら、Amazon EC2 の使用開始チュートリアルを完了する準備が整います。Amazon EC2 の機能について詳細情報が必要なときは、技術ドキュメントを参照できます。

### 起動と実行

- [Amazon EC2 でのセットアップ \(p. 22\)](#)

- [Amazon EC2 Linux インスタンスの開始方法 \(p. 26\)](#)

## 基礎

- [インスタンスと AMI \(p. 4\)](#)
- [リージョンとアベイラビリティーボード \(p. 7\)](#)
- [インスタンスタイプ \(p. 183\)](#)
- [タグ \(p. 1120\)](#)

## ネットワークとセキュリティ

- [Amazon EC2 のキーペア \(p. 899\)](#)
- [セキュリティグループ \(p. 911\)](#)
- [Elastic IP アドレス \(p. 705\)](#)
- [Amazon EC2 と Amazon VPC \(p. 804\)](#)

## ストレージ

- [Amazon EBS \(p. 929\)](#)
- [インスタンスストア \(p. 1076\)](#)

## Linux インスタンスの使用

- [『AWS Systems Manager ユーザーガイド』の「AWS Systems Manager Run Command」](#)
- [チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする \(p. 32\)](#)
- [チュートリアル: Amazon Linux 2 に SSL/TLS を設定する \(p. 62\)](#)
- [AWS の使用開始: Linux 向けウェブアプリケーションのホスティング](#)

AWS がお客様に最適かどうかご質問がある場合は、[AWS セールスまでお問い合わせください](#)。Amazon EC2 について質問がある場合は、[Amazon EC2 forum](#) をご利用ください。

## 関連サービス

インスタンスやボリュームなど、Amazon EC2 のリソースは Amazon EC2 を使用して直接プロビジョニングできます。また、AWS の他のサービスを使用して Amazon EC2 リソースをプロビジョニングすることもできます。詳細については、次のドキュメントを参照してください。

- [Amazon EC2 Auto Scaling ユーザーガイド](#)
- [AWS CloudFormation ユーザーガイド](#)
- [AWS Elastic Beanstalk 開発者ガイド](#)
- [AWS OpsWorks ユーザーガイド](#)

受信アプリケーショントラフィックを複数のインスタンスに自動的に分散するには、Elastic Load Balancing を使用します。詳細については、「[Elastic Load Balancing ユーザーガイド](#)」を参照してください。

インスタンスと Amazon EBS ボリュームの基本的な統計情報をモニタリングするには、Amazon CloudWatch を使用します。詳細については、「[Amazon CloudWatch ユーザーガイド](#)」を参照してください。

新しい Amazon EC2 インスタンスが起動する度に Lambda 関数をアクティベートするなど、アクションを自動化するには、Amazon CloudWatch Events を使用します。詳細については、[Amazon CloudWatch Events ユーザーガイド](#) を参照してください。

アカウントの Amazon EC2 API 宛ての呼び出し (AWS マネジメントコンソール、コマンドラインツール、その他のサービスによって行われる呼び出しを含む) をモニタリングするには、AWS CloudTrail を使用します。詳細については、「[AWS CloudTrail User Guide](#)」を参照してください。

クラウドで管理されたリレーショナルデータベースを取得するには、Amazon Relational Database Service (Amazon RDS) を使用してデータベースインスタンスを起動します。EC2 インスタンス上でデータベースをセットアップできますが、Amazon RDS には、ソフトウェアのパッチ処理、バックアップ、バックアップの保存など、データベース管理タスクを処理できるという利点があります。詳細については、「[Amazon Relational Database Service 開発者ガイド](#)」を参照してください。

仮想マシン (VM) イメージをローカル環境から AWS にインポートして、利用可能な状態の AMI またはインスタンスに変換するには、VM Import/Export を使用します。詳細については、「[VM Import/Export ユーザーガイド](#)」を参照してください。

## Amazon EC2 へのアクセス

Amazon EC2 には、Amazon EC2 コンソールというウェブベースのユーザーインターフェイスがあります。AWS アカウントにサインアップ済みの場合は、AWS マネジメントコンソールにサインインし、コンソールのホームページから [EC2] を選択することで、Amazon EC2 コンソールにアクセスできます。

コマンドラインインターフェイスを使用する場合は、以下の選択肢があります。

AWS コマンドラインインターフェイス (CLI)

一連のさまざまな AWS 製品用のコマンドを提供し、Windows、Mac、および Linux でサポートされています。開始するには、[AWS Command Line Interface ユーザーガイド](#) を参照してください。Amazon EC2 のコマンドの詳細については、『AWS CLI Command Reference』の「`ec2`」を参照してください。

AWS Tools for Windows PowerShell

PowerShell 環境でスクリプトを記述するユーザー向けに、さまざまな AWS 製品用のコマンドが用意されています。開始するには、「[AWS Tools for Windows PowerShell ユーザーガイド](#)」を参照してください。Amazon EC2 のコマンドレットに関する詳細は、「[AWS Tools for PowerShell Cmdlet Reference](#)」を参照してください。

Amazon EC2 はクエリ API を提供します。このリクエストは、HTTP 動詞 (GET または POST) とクエリパラメータ Action で記述する HTTP または HTTPS リクエストです。Amazon EC2 の API アクションの詳細については、『Amazon EC2 API Reference』の「[アクション](#)」を参照してください。

HTTP または HTTPS を介してリクエストを送信する代わりに、言語固有の API を使用してアプリケーションを構築することを希望する場合に備えて、AWS には、ソフトウェア開発者向けのライブラリ、サンプルコード、チュートリアル、その他のリソースが用意されています。これらのライブラリには、リクエストの暗号化署名、リクエストの再試行、エラーレスポンスの処理などのタスクを自動化する基本機能が用意されているので、開発を簡単に始められます。詳細については、「[AWS の SDK およびツール](#)」を参照してください。

## Amazon EC2 の料金表

AWS にサインアップすると、[Amazon EC2 無料利用枠](#)を利用して、AWS を無料で使い始めることができます。

Amazon EC2 では、インスタンス用に次の購入オプションが用意されています。

#### オンデマンドインスタンス

秒単位で使用するインスタンスに対して支払いを行い、長期的な確約や前払い金は不要です。

#### Savings Plans

1~3 年の期間、1 時間 につき USD で、定期的な使用量を守ることにより Amazon EC2 コストを削減できます。

#### リザーブドインスタンス

1~3 年の期間、インスタンスタイプとリージョンを含む特定のインスタンス設定を守ることにより Amazon EC2 コストを削減できます。

#### スポットインスタンス

未使用的 EC2 インスタンスをリクエストして、Amazon EC2 コストを大幅に削減できます。

Amazon EC2 の課金および特定の料金の詳細な一覧については、「[Amazon EC2 の料金](#)」を参照してください。

プロビジョニングされたサンプル環境の費用を計算するには、「[クラウドエコノミクスセンター](#)」を参照してください。

請求を表示するには、[AWS Billing and Cost Management コンソール](#)で請求およびコスト管理ダッシュボードに移動します。請求書には、料金の明細が記載された使用状況レポートへのリンクが記載されています。AWS アカウント請求の詳細については、「[AWS Account Billing](#)」を参照してください。

AWS の請求、アカウント、イベントについてご質問がある場合は、[AWS サポートにお問い合わせください](#)。

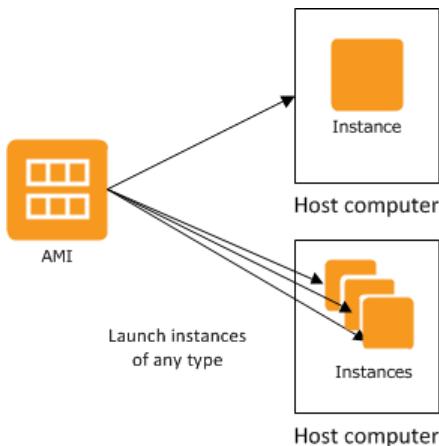
AWS 環境のコスト、セキュリティ、およびパフォーマンスを最適化できるサービスである Trusted Advisor の概要については、「[AWS Trusted Advisor](#)」を参照してください。

## PCI DSS コンプライアンス

Amazon EC2 は、マーチャントまたはサービスプロバイダーによるクレジットカードデータの処理、ストレージ、および伝送をサポートしており、Payment Card Industry (PCI) Data Security Standard (DSS) に準拠していることが確認されています。PCI DSS の詳細 (AWS PCI Compliance Package のコピーをリクエストする方法など) については、「[PCI DSS レベル 1](#)」を参照してください。

## インスタンスと AMI

Amazon マシンイメージ (AMI) は、ソフトウェア構成 (オペレーティングシステム、アプリケーションサーバー、アプリケーションなど) を記録したテンプレートです。AMI から、クラウドで仮想サーバーとして実行される AMI のコピーであるインスタンスを起動します。以下の図に示すように、1 つの AMI の複数のインスタンスを起動することができます。



インスタンスは、停止または終了させるか、エラーが発生するまで実行を続けます。インスタンスがエラーで終了した場合は、元の AMI から新しいインスタンスを起動できます。

## インスタンス

インスタンスとは、クラウドの仮想サーバーです。起動時の設定は、インスタンスを起動した際に指定した AMI のコピーです。

1 つの AMI から、複数の異なるタイプのインスタンスを起動することができます。インスタンスタイプとは本質的に、インスタンスに使用されるホストコンピュータのハードウェアを決定するものです。インスタンスタイプごとに異なる処理内容やメモリの機能が提供されます。インスタンスタイプは、インスタンス上で実行するアプリケーションやソフトウェアに必要なメモリの量と処理能力に応じて選択します。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

インスタンスの起動後は、通常のホストのように表示され、任意のコンピュータと同じように操作できます。インスタンスは完全に制御でき、sudo を使用して、ルート権限を必要とするコマンドを実行できます。

AWS アカウントでは、稼動できるインスタンスの数に制限があります。この制限の詳細、および増加を要求する方法については、Amazon EC2 の全般的なよくある質問の「[Amazon EC2 でいくつのインスタンスを稼動できますか](#)」を参照してください。

## インスタンストレージ

インスタンスのルートデバイスには、インスタンスの起動に使用されるイメージが含まれています。詳細については、「[Amazon EC2 ルートデバイスピリューム \(p. 16\)](#)」を参照してください。

インスタンスには、インスタンストアボリュームと呼ばれるローカルストレージボリュームを含めることができます。これはブロックデバイスマッピングによって起動時に設定できます。詳細については、「[ブロックデバイスマッピング \(p. 1100\)](#)」を参照してください。これらのボリュームがインスタンスに追加およびマッピングされたら、マウントして使用することができます。インスタンスが失敗、停止、または終了した場合、それらのボリュームのデータは失われます。したがって、これらのボリュームは一時データとして使用するのが最適です。重要なデータを安全に維持するには、複数のインスタンスにわたるレプリケーション方法を使用する必要があります。または、永続的なデータを Amazon S3 または Amazon EBS ボリュームに格納してください。詳細については、「[ストレージ \(p. 928\)](#)」を参照してください。

## セキュリティのベストプラクティス

- AWS Identity and Access Management (IAM) を使用して、ご利用のインスタンスを含め、AWS リソースへのアクセスを制御します。AWS アカウントで IAM ユーザーとグループを作成し、それぞれにセ

キュリティ認証情報を割り当て、それぞれが AWS のリソースとサービスに対して持つアクセスを制御できます。詳細については、「[Amazon EC2 の Identity and Access Management \(p. 839\)](#)」を参照してください。

- 信頼されたホストまたはネットワークのみがインスタンスのポートにアクセスできるように制限します。たとえば、ポート 22 の受信トラフィックを制限することで SSH アクセスを制限できます。詳細については、「[Linux インスタンスの Amazon EC2 セキュリティグループ \(p. 911\)](#)」を参照してください。
- セキュリティグループのルールを定期的に確認し、最小権限（—必要なアクセス許可のみを開く）の原則を適用してください。また、さまざまなセキュリティグループを作成して、異なるセキュリティ要件を持つ各インスタンスに対応することもできます。外部ログインが許可された基本となるセキュリティグループの作成を検討し、外部ログインが許可されていないグループで残りのインスタンスを管理してください。
- AMI から起動されるインスタンスについてはパスワードベースのログインを無効にしてください。パスワードは検知または解読される恐れがあり、セキュリティ上のリスクです。詳細については、「[ルートのパスワードベースのリモートログインを無効にする \(p. 109\)](#)」を参照してください。AMI の安全な共有の詳細については、「[共有 AMI \(p. 102\)](#)」を参照してください。

## インスタンスの停止、開始、および終了

### インスタンスの停止

インスタンスが停止されると、インスタンスは通常のシャットダウンを実行してから、`stopped` 状態に移行します。そのすべての Amazon EBS ボリュームはアタッチされたままになり、後でインスタンスを再び開始することができます。

インスタンスが停止状態にあるとき、インスタンスの使用分が追加で課金されることはありません。最低 1 分間分の使用料が課金されます。インスタンスが停止状態にあるときにインスタンスタイプを変更した場合、インスタンスを起動すると同時に新しいインスタンスタイプの料金が課金されます。ルートデバイスの使用を含め、インスタンスに関連する Amazon EBS の使用はすべて、Amazon EBS 料金で課金されます。

インスタンスが停止状態の場合は、Amazon EBS ボリュームをアタッチおよびデタッチできます。インスタンスから AMI を作成し、カーネル、RAM ディスク、インスタンスタイプを変更することもできます。

### インスタンスを終了した

インスタンスを終了すると、インスタンスは正常なシャットダウンを実行します。ルートデバイスピリュームはデフォルトで削除されますが、アタッチされた Amazon EBS ボリュームはデフォルトでは保持されます（各ボリュームの `deleteOnTermination` 属性の設定によって決まります）。インスタンスそのものも削除され、後でインスタンスを再度起動することはできません。

間違って終了しないようにするために、インスタンスの削除を無効にすることができます。この場合、インスタンスの `disableApiTermination` 属性は必ず `true` にします。インスタンスのシャットダウン時の動作を制御するには（Linux の `shutdown -h` や Windows の `shutdown` など）、`instanceInitiatedShutdownBehavior` インスタンス属性を必要に応じて `stop` または `terminate` に設定します。Amazon EBS ボリュームをルートデバイスに持つインスタンスはデフォルトで `stop` に設定されます。インスタンストアをルートデバイスに持つインスタンスはシャットダウンの結果として常に終了されます。

詳細については、「[インスタンスのライフサイクル \(p. 443\)](#)」を参照してください。

## AMI

Amazon ウェブ サービス (AWS) は、一般的な用途のための共通のソフトウェア設定を含む多くの [Amazon Machine Image \(AMI\) \(p. 100\)](#) を公開しています。加えて、AWS 開発者コミュニティのメンバーによって作成された、独自のカスタム AMI もあります。お客様自身でカスタム AMI を作成することもできます。必要なものがすべて含まれた新しいインスタンスを、すばやく簡単に起動できるようになります。

ます。たとえば、ウェブサイトまたはウェブサービスに使用する場合は、AMI に含まれるものとして、ウェブサーバー、関連する静的コンテンツ、動的ページ用のコードが考えられます。この AMI からインスタンスを起動すると、ウェブサーバーが起動し、アプリケーションはリクエストを受け付け可能な状態になります。

すべての AMI は、Amazon EBS-backed (AMI からインスタンスを起動するときのルートデバイスは Amazon EBS ボリュームである) と Instance-store backed (AMI からインスタンスを起動するときのルートデバイスは、Amazon S3 に格納されているテンプレートから作成されたインスタンストアボリュームである) のいずれかに分類されます。

AMI の説明に、ルートデバイスのタイプ (ebs または instance store) が明記されています。このことが重要であるのは、AMI のタイプによって、実行できる機能が大きく異なるからです。違いについての詳細は [ルートデバイスのストレージ \(p. 96\)](#) を参照してください。

## リージョン、アベイラビリティーゾーン、およびローカルゾーン

Amazon EC2 は、世界各地のロケーションでホスティングされています。これらのロケーションは、リージョン、アベイラビリティーゾーン、および ローカルゾーン で構成されています。リージョンはそれぞれ、地理的に離れた領域です。各リージョンには、アベイラビリティーゾーンと呼ばれる複数の独立した場所があります。ローカルゾーンを使用すると、コンピューティングやストレージなどのリソースを、エンドユーザーに近い複数の場所に配置できます。リソースは、お客様が特に指定しない限り、複数のリージョン間でレプリケートされることはありません。

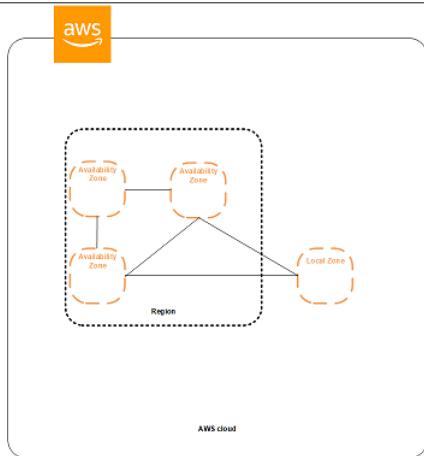
Amazon は、最新の高可用性のデータセンターを運用しています。しかし、非常にまれですが、同じ場所にあるインスタンスすべての可用性に影響する障害が発生することもあります。すべてのインスタンスを 1か所でホストしている場合、そのような障害が起きると、すべてのインスタンスが利用できなくなります。

### 目次

- [リージョン、アベイラビリティーゾーン、および ローカルゾーン の概念 \(p. 7\)](#)
- [利用できるリージョン \(p. 10\)](#)
- [リージョンとエンドポイント \(p. 11\)](#)
- [リージョン、アベイラビリティーゾーン、および ローカルゾーン の確認 \(p. 11\)](#)
- [リソースのリージョンの指定 \(p. 13\)](#)
- [ローカルゾーン の有効化 \(p. 14\)](#)
- [ローカルゾーン の無効化 \(p. 15\)](#)
- [アベイラビリティーゾーンまたは ローカルゾーン でのインスタンスの起動 \(p. 15\)](#)
- [別のアベイラビリティーゾーンへのインスタンスの移行 \(p. 15\)](#)

## リージョン、アベイラビリティーゾーン、および ローカルゾーン の概念

各リージョンは完全に独立しています。各アベイラビリティーゾーンは独立していますが、リージョン内のアベイラビリティーゾーンは低レイテンシーのリンクで接続されています。ローカルゾーンは、厳選したサービスをエンドユーザーのより近くに配置する AWS インフラストラクチャのデプロイです。ローカルゾーンは、リージョンの拡張であり、ご利用のリージョンとは別の場所に設定されます。AWS インフラストラクチャに広帯域幅のバックボーンを提供し、機械学習などのレイテンシーの影響を受けやすいアプリケーションに最適です。次の図に、リージョン、アベイラビリティーゾーン、および ローカルゾーン の関係を示します。



Amazon EC2 リソースは、グローバルなもの、リージョンに結び付けられたもの、アベイラビリティーゾーンに結び付けられたもの、または ローカルゾーン に結び付けられたもののいずれかです。詳細については、「[リソースの場所 \(p. 1110\)](#)」を参照してください。

## 目次

- [リージョン \(p. 8\)](#)
- [アベイラビリティーゾーン \(p. 8\)](#)
- [ローカルゾーン \(p. 9\)](#)
- [ネットワーク境界グループ \(p. 9\)](#)

## リージョン

各 Amazon EC2 リージョンは、他の Amazon EC2 リージョンと完全に分離されるように設計されています。これにより、最大限の耐障害性と安定性が達成されます。

リソースを表示すると、指定したリージョンに結び付けられているリソースのみが表示されます。これは、リージョンが相互に分離されており、リージョン間ではリソースが自動的にレプリケートされないためです。

インスタンスを起動するときは、同じリージョン内にある AMI を選択する必要があります。AMI が別のリージョンにある場合は、使用しているリージョンに AMI をコピーできます。詳細については、「[AMI のコピー \(p. 155\)](#)」を参照してください。

リージョン間のデータ転送には料金がかかることに注意してください。詳細については、「[Amazon EC2 料金表 - データ転送](#)」を参照してください。

## アベイラビリティーゾーン

インスタンスを起動するときに、アベイラビリティーゾーンを自分で選択するか、自動的に選択されるようになります。インスタンスを複数のアベイラビリティーゾーンに配布する場合は、1つのインスタンスで障害が発生したら別のアベイラビリティーゾーンのインスタンスが要求を処理するように、アプリケーションを設計できます。

また、Elastic IP アドレスを使用すると、あるアベイラビリティーゾーンのインスタンスの障害を、別のアベイラビリティーゾーンのインスタンスにアドレスをすばやく再マッピングすることによってマスクできます。詳細については、「[Elastic IP アドレス \(p. 705\)](#)」を参照してください。

アベイラビリティーゾーンは、リージョンコードとそれに続く文字識別子によって表されます (us-east-1a など)。リソースがリージョンの複数のアベイラビリティーゾーンに分散するように、アベイラビリティーゾーンは各 AWS アカウントの名前に個別にマップされます。たとえば、ご使用の AWS アカ

アカウントのアベイラビリティーゾーン `us-east-1a` は別の AWS アカウントのアベイラビリティーゾーン `us-east-1a` と同じ場所にはない可能性があります。

アカウント間でアベイラビリティーゾーンを調整するには、アベイラビリティーゾーンの一意で一貫性のある識別子である AZ ID を使用する必要があります。たとえば、`use1-az1` は、`us-east-1` リージョンの AZ ID で、すべての AWS アカウントで同じ場所になります。

AZ ID を表示すると、あるアカウントのリソースの場所を別のアカウントのリソースに対して決定できます。たとえば、AZ ID `use-az2` のアベイラビリティーゾーンにあるサブネットを別のアカウントと共有する場合、このサブネットは AZ ID が同じく `use-az2` であるアベイラビリティーゾーンのそのアカウントでも利用できます。各 VPC とサブネットの AZ ID は Amazon VPC コンソールに表示されます。詳細については、『Amazon VPC ユーザーガイド』の「[Working with VPC Sharing](#)」を参照してください。

アベイラビリティーゾーンが拡大すると、アベイラビリティーゾーンを拡張しにくくなる場合があります。その場合、ユーザーがアベイラビリティーゾーンに既にインスタンスを持っているのでない場合は、制約のあるアベイラビリティーゾーンでのインスタンスの起動を制限する場合があります。最終的に、制約のあるアベイラビリティーゾーンを新しいアカウントに対するアベイラビリティーゾーンのリストから削除することもあります。したがって、アカウントによってリージョン内で使用できるアベイラビリティーゾーンの数が異なる場合があります。

自分のアカウントで使用できるアベイラビリティーゾーンをリストできます。詳細については、「[リージョン、アベイラビリティーゾーン、および ローカルゾーン の確認 \(p. 11\)](#)」を参照してください。

## ローカルゾーン

ローカルゾーンは、ユーザーに近い場所に位置する、AWS リージョンの拡張です。インスタンスを起動するときに、ローカルゾーンのサブネットを選択できます。ローカルゾーンは、インターネットへの独自の接続を持ち、AWS Direct Connect をサポートしています。したがって、ローカルゾーンで作成したリソースは、非常に低いレイテンシーの通信を使用してローカルユーザーにサービスを提供できます。詳細については、「[AWS ローカルゾーン](#)」を参照してください。

ローカルゾーンは、リージョンコードと場所を示す識別子で表されます (`us-west-2-lax-1a`)。

ローカルゾーンを使用するには、最初にそれを有効にする必要があります。詳細については、「[ローカルゾーンの有効化 \(p. 14\)](#)」を参照してください。次に、ローカルゾーン内にサブネットを作成します。最後に、ローカルゾーンのサブネットで次のいずれかのリソースを起動し、アプリケーションとエンドユーザーを近づけます。

- Amazon EC2 インスタンス
- Amazon EBS ボリューム
- Amazon FSx ファイルサーバー
- Application Load Balancer

ローカルゾーンは、すべてのリージョンで利用できるわけではありません。ローカルゾーンをサポートするリージョンについては、「[the section called “利用できるリージョン” \(p. 10\)](#)」を参照してください。

アカウントで使用可能なローカルゾーンを一覧表示できます。詳細については、「[リージョン、アベイラビリティーゾーン、および ローカルゾーン の確認 \(p. 11\)](#)」を参照してください。

## ネットワーク境界グループ

ネットワーク境界グループは、AWS が IP アドレスをアドバタイズするアベイラビリティーゾーンまたはローカルゾーンの一意のセットです。ネットワーク境界グループから次のリソースを割り当てるできます。

- Amazon が提供する伸縮自在な IPv4 アドレス
- Amazon が提供する VPC の IPv6 アドレス

ネットワーク境界グループは、グループに対するアドレス数を制限します。IP アドレスは、ネットワーク境界グループ間を移動できません。

## 利用できるリージョン

アカウントにより、利用できるリージョンが決まります。例:

- AWS アカウントでは複数のリージョンが提供されるため、それぞれの要件に合った場所で Amazon EC2 インスタンスを起動できます。たとえば、ヨーロッパの顧客に近づけるため、または法的要件を満たすために、ヨーロッパでインスタンスを起動することができます。
- AWS GovCloud (米国西部) アカウントでは、AWS GovCloud (米国西部) リージョンにのみアクセスできます。詳細については、「[AWS GovCloud \(米国西部\) リージョン](#)」を参照してください。
- Amazon AWS アカウント (中国) では、北京および寧夏 リージョンにのみアクセスできます。詳細については、「[AWS in China](#)」を参照してください。

次の表に、AWS アカウントで提供されるリージョンのリストを示します。AWS GovCloud (米国西部) や中国リージョンなど、追加のリージョンは AWS アカウントから記述またはアクセスできません。2019 年 3 月 20 日より後に導入されたリージョンを使用するには、そのリージョンを有効にする必要があります。詳細については、AWS General Reference の「[AWS リージョンの管理](#)」を参照してください。

コード	名前	オプトインステータス	ローカルゾーン
us-east-2	米国東部 (オハイオ)	不要	いいえ
us-east-1	米国東部 (バージニア 北部)	不要	いいえ
us-west-1	米国西部 (北カリフォルニア)	不要	いいえ
us-west-2	米国西部 (オレゴン)	不要	はい - us-west-2-lax-1a ローカルゾーンを有効にする必要があります。
ap-east-1	アジアパシフィック (香港)	必須	いいえ
ap-south-1	アジアパシフィック (ムンバイ)	不要	いいえ
ap-northeast-3	アジアパシフィック (大阪: ローカル)	不要	いいえ
ap-northeast-2	アジアパシフィック (ソウル)	不要	いいえ
ap-southeast-1	アジアパシフィック (シンガポール)	不要	いいえ
ap-southeast-2	アジアパシフィック (シドニー)	不要	いいえ
ap-northeast-1	アジアパシフィック (東京)	不要	いいえ

コード	名前	オプトインステータス	ローカルゾーン
ca-central-1	カナダ (中部)	不要	いいえ
eu-central-1	欧州 (フランクフルト)	不要	いいえ
eu-west-1	欧州 (アイルランド)	不要	いいえ
eu-west-2	欧州 (ロンドン)	不要	いいえ
eu-west-3	欧州 (パリ)	不要	いいえ
eu-north-1	欧州 (ストックホルム)	不要	いいえ
me-south-1	中東 (バーレーン)	必須	いいえ
sa-east-1	南米 (サンパウロ)	不要	いいえ

詳細については、「[AWS グローバルインフラストラクチャ](#)」を参照してください。

リージョンごとのアベイラビリティーゾーンの数とマッピングは、AWS アカウント間で異なる場合があります。アカウントで使用可能なアベイラビリティーゾーンのリストを取得するには、Amazon EC2 コンソールまたはコマンドラインインターフェイスを使用できます。詳細については、「[リージョン、アベイラビリティーゾーン、および ローカルゾーンの確認 \(p. 11\)](#)」を参照してください。

## リージョンとエンドポイント

コマンドラインインターフェイスまたは API アクションを使用してインスタンスを操作するときは、そのリージョンエンドポイントを指定する必要があります。Amazon EC2 のリージョンおよびエンドポイントの詳細については、『[Amazon ウェブ サービス全般のリファレンス](#)』の「[リージョンとエンドポイント](#)」を参照してください。

AWS GovCloud (米国西部) のエンドポイントとプロトコルの詳細については、『[AWS GovCloud \(US\) User Guide](#)』の「[AWS GovCloud \(米国西部\) エンドポイント](#)」を参照してください。

## リージョン、アベイラビリティーゾーン、および ローカルゾーン の確認

Amazon EC2 コンソールまたはコマンドラインインターフェイスを使用して、アカウントで使用できるリージョン、アベイラビリティーゾーン、および ローカルゾーン を確認できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

コンソールを使用してリージョン、アベイラビリティーゾーン、および ローカルゾーン を確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、リージョンセレクターのオプションを表示します。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
リージョン、アベイラビリティーゾー  
ン、および ローカルゾーン の確認



3. ナビゲーションペインで、[EC2 ダッシュボード] を選択します。
4. アベイラビリティゾーンと ローカルゾーン は、[サービス状態] の [アベイラビリティゾーンのス  
テータス] に一覧表示されます。

AWS CLI を使用してリージョン、アベイラビリティゾーン、および ローカルゾーン を確認す  
るには

1. 次のように `describe-regions` コマンドを使用して、アカウントに対して有効になっているリージョン  
を記述します。

```
aws ec2 describe-regions
```

アカウントに対して無効になっているリージョンも含めてすべてのリージョンを記述するには、次  
のように `--all-regions` オプションを追加します。

```
aws ec2 describe-regions --all-regions
```

2. 次のように `describe-availability-zones` コマンドを使用すると、指定したリージョン内のアベイラビ  
リティゾーンと ローカルゾーン を確認できます。

```
aws ec2 describe-availability-zones --region region-name
```

3. 次のように `describe-availability-zones` コマンドを使用すると、オプトインのステータスに関係なく、  
アベイラビリティゾーンと ローカルゾーン を確認できます。

```
aws ec2 describe-availability-zones --all-availability-zones
```

AWS Tools for Windows PowerShell を使用してリージョン、アベイラビリティゾーン、およびローカルゾーンを確認するには

1. 次のように [Get-EC2Region](#) コマンドを使用して、アカウントのリージョンを記述します。

```
PS C:\> Get-EC2Region
```

2. 次のように [Get-EC2AvailabilityZone](#) コマンドを使用して、指定されたリージョン内のアベイラビリティゾーンを記述します。

```
PS C:\> Get-EC2AvailabilityZone -Region region-name
```

## リソースのリージョンの指定

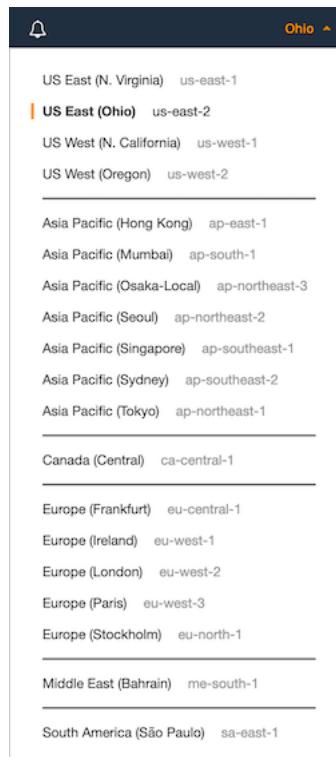
Amazon EC2 リソースを作成するたびに、リソースのリージョンを指定できます。リソースのリージョンは AWS マネジメントコンソール またはコマンドラインを使用して指定できます。

### Note

一部の AWS リソースは、リージョン、アベイラビリティゾーン、およびローカルゾーンによっては利用できない場合があります。特定のアベイラビリティゾーンでインスタンスを起動する前に、目的のリージョンまたはアベイラビリティゾーンで必要なリソースを作成できることを確認してください。

コンソールを使用してリソースのリージョンを指定するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーのリージョンセレクターを使用します。



## コマンドラインを使用してデフォルトのリージョンを指定するには

環境変数の値を、目的のリージョンエンドポイント (<https://ec2.us-east-2.amazonaws.com> など) に設定できます。

- AWS\_DEFAULT\_REGION (AWS CLI)
- Set-AWSDefaultRegion (AWS Tools for Windows PowerShell)

各コマンドで、--region (AWS CLI) または -Region (AWS Tools for Windows PowerShell) のコマンドラインオプションを使用することもできます。たとえば、--region us-east-2 と指定します。

Amazon EC2 のエンドポイントの詳細については、「[Amazon Elastic Compute Cloud Endpoints](#)」を参照してください。

## ローカルゾーン の有効化

リソースまたはサービスの ローカルゾーン を指定する前に、このゾーンを有効にする必要があります。

ローカルゾーン は、AWS マネジメントコンソール または AWS CLI を使用して有効にすることができます。

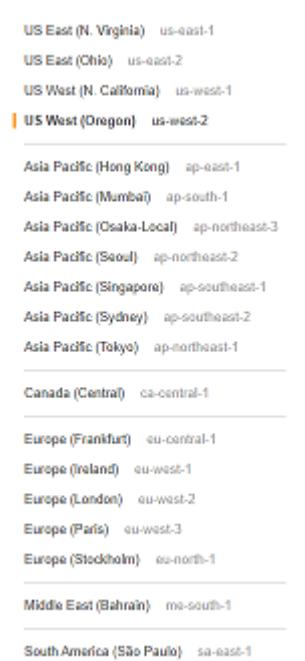
### Note

すべてのアベイラビリティゾーンはデフォルトで有効になっており、無効にすることはできません。

一部の AWS リソースは、一部のリージョンで利用できない場合があります。特定の ローカルゾーン でインスタンスを起動する前に、目的のリージョンまたは ローカルゾーン で必要なりソースを作成できることを確認してください。

## コンソールを使用して ローカルゾーン を有効にするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーのリージョンセレクターを使用して、リージョンを選択します。



3. ナビゲーションペインで、[EC2 ダッシュボード] を選択します。

4. [アベイラビリティーゾーンのステータス] で、[追加のローカルゾーンを有効にする] を選択します。
5. [ローカルゾーングループ] で、有効にする各 ローカルゾーン をオンにします。
6. [有効化] 確認ダイアログボックスに [Enable] と入力し、[OK] を選択します。

AWS CLI を使用して ローカルゾーン を有効にするには

- [modify-availability-zone-group](#) コマンドを使用します。

## ローカルゾーン の無効化

ローカルゾーン を無効にする場合は、[AWS サポート](#)に連絡する必要があります。

### Important

ローカルゾーン を無効にする前に、すべてのリソースを削除します。ローカルゾーン に残ったリソースには、料金が発生します。リソースを削除したら、「Disable Zone Group」というタイトルのケースを [AWS サポート](#) で作成します。

## アベイラビリティーゾーンまたは ローカルゾーン でのインスタンスの起動

インスタンスを起動するときは、インスタンスと特定のお客様を近づけるリージョン、または法的要件や他の要件を満たすリージョンを選択します。個別のアベイラビリティーゾーンでインスタンスを起動することにより、1つの場所で障害が発生しても、アプリケーションを保護することができます。

ローカルゾーン でインスタンスを起動することで、AWS インフラストラクチャの利点を享受しながら、レイテンシーの影響を受けやすいアプリケーションをエンドユーザーの近くで実行できます。

インスタンスを起動するときは、必要に応じて、使用しているリージョンでアベイラビリティーゾーンまたはローカルゾーン を指定できます。アベイラビリティーゾーンまたはローカルゾーン を指定しない場合は、アベイラビリティーゾーンが自動的に選択されます。初期インスタンスを起動するときには、デフォルトのアベイラビリティーゾーンを受け入れることをお勧めします。これにより、システムの状態と利用可能な機能に基づいて、最適なアベイラビリティーゾーンを選択できます。追加のインスタンスを起動する場合、アベイラビリティーゾーンを指定するのは、新しいインスタンスを実行中のインスタンスと近づけるか、分離することが必要な場合に限ります。

## 別のアベイラビリティーゾーンへのインスタンスの移行

必要に応じて、アベイラビリティーゾーン間でインスタンスを移行できます。たとえば、インスタンスのインスタンスタイプを変更しようとしていて、現在のアベイラビリティーゾーンでは新しいインスタンスタイプのインスタンスを起動できないとします。この場合、そのインスタンスタイプのインスタンスを起動できるアベイラビリティーゾーンにインスタンスを移行できます。

移行プロセスは、次の作業を伴います。

- 元のインスタンスからの AMI の作成
- 新しいアベイラビリティーゾーンでのインスタンスの起動
- 新しいインスタンスの設定の更新 (次の手順で示します)

別のアベイラビリティーゾーンにインスタンスを移行するには

1. インスタンスから AMI を作成します。手順は、オペレーティングシステムとインスタンスのルートデバイスピリュームの種類によって異なります。詳細については、使用しているオペレーティングシステムとルートデバイスピリュームに対応するドキュメントを参照してください。

- [Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)
  - [Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)
  - [Amazon EBS-backed Windows AMI の作成](#)
2. インスタンスのプライベート IPv4 アドレスを維持する必要がある場合は、現在のアベイラビリティーゾーンのサブネットを削除してから、新しいアベイラビリティーゾーンに元のサブネットと同じ IPv4 アドレス範囲のサブネットを作成する必要があります。サブネットを削除する前に、その中のすべてのインスタンスを終了する必要があります。したがって、サブネットのすべてのインスタンスから AMI を作成し、現在のサブネットのすべてのインスタンスを新しいサブネットに移動できるようにする必要があります。
  3. 新しいアベイラビリティーゾーンまたはサブネットを指定して、作成した AMI からインスタンスを起動します。インスタンスタイプは、元のインスタンスと同じにすることも、新しいインスタンスタイプを選択することもできます。詳細については、「[アベイラビリティーゾーンまたは ローカルゾーンでのインスタンスの起動 \(p. 15\)](#)」を参照してください。
  4. 元のインスタンスに Elastic IP アドレスが関連付けられていた場合は、それを新しいインスタンスに関連付けます。詳細については、「[Elastic IP アドレスの関連付け解除 \(p. 710\)](#)」を参照してください。
  5. 元のインスタンスが リザーブドインスタンス の場合は、予約のアベイラビリティーゾーンを変更します。(また、インスタンスタイプも変更する場合は、予約のインスタンスタイプも変更できます)。 詳細については、「[変更リクエストの送信 \(p. 311\)](#)」を参照してください。
  6. (オプション) 元のインスタンスを終了します。詳細については、「[インスタンスを削除する \(p. 547\)](#)」を参照してください。

## Amazon EC2 ルートデバイスピリューム

インスタンスを起動するときは、ルートデバイスピリュームに格納されているイメージを使用してインスタンスがブートされます。Amazon EC2 のサービス開始当初は、すべての AMI が「Amazon EC2 インスタンスストア backed」でした。つまり、AMI から起動されるインスタンスのルートデバイスは、Amazon S3 に格納されたテンプレートから作成されるインスタンスストアボリュームです。Amazon EBS の導入後は Amazon EBS を基にした AMI も導入されました。つまり、AMI から起動されるインスタンスのルートデバイスが、Amazon EBS スナップショットから作成される Amazon EBS ボリュームであるということです。

お客様は、「Amazon EC2 インスタンスストア backed」の AMI と「Amazon EBS backed」の AMI から選択できます。推奨されるのは「Amazon EBS backed」です。この AMI は起動が高速であり、永続的ストレージを使用しているからです。

Amazon EC2 でルートボリュームに使用するデバイス名の詳細については、「[Linux インスタンスでのデバイスの名前付け \(p. 1098\)](#)」を参照してください。

### トピック

- [ルートデバイストレージの概念 \(p. 16\)](#)
- [ルートデバイスタイプによる AMI の選択 \(p. 18\)](#)
- [インスタンスのルートデバイスタイプの判別 \(p. 19\)](#)
- [永続的ルートデバイスピリュームへの変更 \(p. 19\)](#)

## ルートデバイストレージの概念

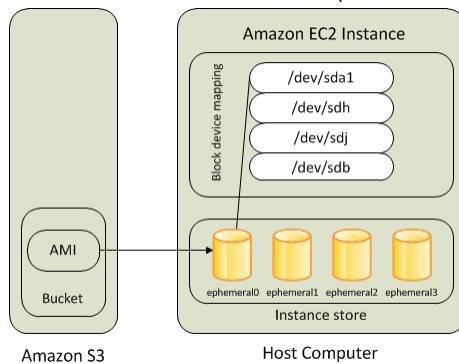
instance store-backed AMI または Amazon EBS-backed AMI のどちらからでもインスタンスを起動できます。AMI の説明にはそのタイプが含まれており、場所によってルートデバイスが ebs (Amazon EBS-

Backed の場合) または instance store (Instance store-Backed の場合) と表示されます。各タイプの AMI を使用して実行できることには大きな違いがあるため、タイプを区別できることは重要です。違いについての詳細は [ルートデバイスのストレージ \(p. 96\)](#) を参照してください。

#### instance store-backed のインスタンス

インスタンスストアをルートデバイスに使用するインスタンスでは自動的に、インスタンスストアボリュームを利用できるようになり、そのボリュームの 1 つがルートデバイスボリュームとなります。インスタンスを起動すると、インスタンスのブートに使用されるイメージがルートボリュームにコピーされます。インスタンスタイプによっては、オプションで追加のインスタンスストアボリュームを使用できることに注意してください。

インスタンスストアボリュームのデータはインスタンスが実行している間は維持されますが、インスタンスが終了すると (Instance store-Backed インスタンスは [Stop] アクションをサポートしていません)、またはインスタンスが失敗すると (基盤となるドライブに問題がある場合など)、削除されます。

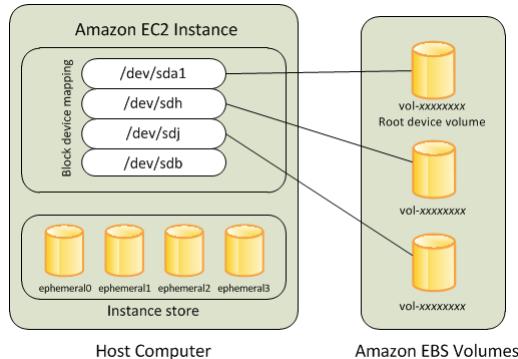


障害が発生したり終了されたりした instance store-backed インスタンスは復元できません。Amazon EC2 instance store-backed インスタンスの使用を予定している場合は、インスタンスストアのデータを複数のアベイラビリティーゾーンにまたがって分散させることを強くお勧めします。また、インスタンスストアボリュームからの重要なデータは永続的ストレージに定期的にバックアップする必要があります。

詳細については、「[Amazon EC2 インスタンスストア \(p. 1076\)](#)」を参照してください。

#### Amazon EBS-backed インスタンス

Amazon EBS をルートデバイスに使用するインスタンスには自動的に、Amazon EBS ボリュームがアタッチされます。Amazon EBS Backed インスタンスを起動するときに、AMI で参照されている Amazon EBS スナップショットごとに 1 つの Amazon EBS ボリュームが作成されます。インスタンスタイプによっては、Amazon EBS ボリュームまたはインスタンスストアボリュームをオプションで使用できます。



Amazon EBS-backed インスタンスは、停止後に再起動できます。アタッチされているボリュームに格納されているデータに影響を及ぼすこともありません。Amazon EBS-backed インスタンスが停止状態にあるときは、インスタンス - ボリューム関連の様々なタスクを実行できます。たとえば、インスタンスのプ

ロバティの変更、そのサイズの変更、あるいは使用しているカーネルを更新できます。また、デバッグなどの目的で別の実行中インスタンスにルートボリュームをアタッチすることもできます。

Amazon EBS-backed インスタンスに障害が発生した場合は、以下のいずれかの方法によってセッションを復元できます。

- 停止して再起動します(最初にこの方法を試してください)。
- 関連するすべてのボリュームのスナップショットを自動的に作成し、新しい AMI を作成します。詳細については、「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」を参照してください。
- 以下の手順に従って、ボリュームを新しいインスタンスにアタッチします。
  - ルートボリュームのスナップショットを作成します。
  - 作成したスナップショットを使用して新しい AMI を登録します。
  - 新しい AMI から新しいインスタンスを起動します。
  - 残りの Amazon EBS ボリュームを古いインスタンスからデタッチします。
  - Amazon EBS ボリュームを新しいインスタンスに再アタッチします。

詳細については、「[Amazon EBS ボリューム \(p. 931\)](#)」を参照してください。

## ルートデバイスタイプによる AMI の選択

インスタンスの起動時に指定する AMI によって、インスタンスのルートデバイスボリュームのタイプが決まります。

コンソールを使用して Amazon EBS-Backed AMI を選択するには

- Amazon EC2 コンソールを開きます。
- ナビゲーションペインで [AMIs] を選択します。
- フィルタの一覧から、イメージタイプ ([Public images] など) を選択します。検索バーで、[Platform] を選択してオペレーティングシステム ([Amazon Linux] など) を選択し、[Root Device Type] をクリックして [EBS images] を選択します。
- (オプション) 選択の参考になる追加情報を表示するには、[Show/Hide Columns] アイコンを選択し、表示する列を更新して、[Close] を選択します。
- AMI を選択し、その AMI ID を記録します。

コンソールを使用して instance store-backed AMI を選択するには

- Amazon EC2 コンソールを開きます。
- ナビゲーションペインで [AMIs] を選択します。
- フィルタの一覧から、イメージタイプ ([Public images] など) を選択します。検索バーで、[プラットフォーム] を選択してオペレーティングシステム ([Amazon Linux] など) を選択し、[ルートデバイスタイプ] を選択して [インスタンスストア] を選択します。
- (オプション) 選択の参考になる追加情報を表示するには、[Show/Hide Columns] アイコンを選択し、表示する列を更新して、[Close] を選択します。
- AMI を選択し、その AMI ID を記録します。

コマンドラインを使用して AMI のルートデバイスボリュームの種類を確認するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [describe-images \(AWS CLI\)](#)

- [Get-EC2Image](#) (AWS Tools for Windows PowerShell)

## インスタンスのルートデバイスタイプの判別

コンソールを使用してインスタンスのルートデバイスタイプを判別するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. 次のように、[Description] タブで [Root device type] の値を確認します。
  - 値が ebs の場合は Amazon EBS-Backed インスタンスです。
  - 値が instance store の場合、これは Instance store-Backed インスタンスです。

コマンドラインを使用してインスタンスのルートデバイスタイプを判別するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-instances](#) (AWS CLI)
- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

## 永続的ルートデバイスボリュームへの変更

デフォルトでは、Amazon EBS-backed AMI のルートデバイスボリュームは、インスタンスを終了すると削除されます。インスタンスの終了後もボリュームが永続化するように、デフォルトの動作を変更できます。デフォルトの動作を変更するには、ブロックデバイスマッピングを使用して、DeleteOnTermination 属性を false に設定します。

### トピック

- [インスタンスの起動時に永続化するためのルートボリュームの設定 \(p. 19\)](#)
- [実行中のインスタンスで永続化するためのルートボリュームの設定 \(p. 20\)](#)
- [ルートボリュームが永続化するように設定されていることの確認 \(p. 21\)](#)

## インスタンスの起動時に永続化するためのルートボリュームの設定

Amazon EC2 コンソールまたはコマンドラインツールを使用して、インスタンスの起動時に永続化するようにルートボリュームを設定できます。

コンソールを使用してインスタンスの起動時に永続化するようにルートボリュームを設定するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [インスタンス]、[インスタンスの作成] の順に選択します。
3. [Choose an Amazon Machine Image (AMI)] ページで、使用する AMI を選択し、[Select] を選択します。
4. ウィザードにしたがって [Choose an Instance Type] ページと [Configure Instance Details] ページを設定します。
5. [Add Storage] ページで、ルートボリュームの [Delete On Termination] の選択を解除します。
6. ウィザードの残りのページを完了した後、[Launch] を選択します。

AWS CLI を使用してインスタンスの起動時に永続化するようにルートボリュームを設定するには

[run-instances](#) コマンドを使用して、DeleteOnTermination 属性を false に設定するブロックデバイスマッピングを含めます。

```
$ aws ec2 run-instances --block-device-mappings file://mapping.json ...other parameters...
```

mapping.json で、以下を指定します。

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false  
    }  
}  
]
```

Tools for Windows PowerShell を使用してインスタンスの起動時に永続化するようにルートボリュームを設定するには

[New-EC2Instance](#) コマンドを使用して、DeleteOnTermination 属性を false に設定するブロックデバイスマッピングを含めます。

```
C:\> $ebs = New-Object Amazon.EC2.Model.EbsBlockDevice  
C:\> $ebs.DeleteOnTermination = $false  
C:\> $bdm = New-Object Amazon.EC2.Model.BlockDeviceMapping  
C:\> $bdm.DeviceName = "dev/xvda"  
C:\> $bdm.Ebs = $ebs  
C:\> New-EC2Instance -ImageId ami-0abcdef1234567890 -BlockDeviceMapping $bdm ...other  
parameters...
```

## 実行中のインスタンスで永続化するためのルートボリュームの設定

コマンドラインツールのみを使用して、実行中のインスタンスで永続化するようにルートボリュームを設定できます。

AWS CLI を使用して、実行中のインスタンスで永続化するようにルートボリュームを設定するには

[modify-instance-attribute](#) コマンドを使用して、DeleteOnTermination 属性を false に設定するブロックデバイスマッピングを含めます。

```
$ aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-mappings "[{\\"DeviceName\\": \"/dev/xvda\", \"Ebs\":{\\\"DeleteOnTermination\\\":false}}]"
```

AWS Tools for Windows PowerShell を使用して、実行中のインスタンスで永続化するようにルートボリュームを設定するには

[Edit-EC2InstanceAttribute](#) コマンドを使用して、DeleteOnTermination 属性を false に設定するブロックデバイスマッピングを含めます。

```
C:\> $ebs = New-Object Amazon.EC2.Model.EbsInstanceBlockDeviceSpecification  
C:\> $ebs.DeleteOnTermination = $false  
C:\> $bdm = New-Object Amazon.EC2.Model.InstanceBlockDeviceMappingSpecification  
C:\> $bdm.DeviceName = "/dev/xvda"  
C:\> $bdm.Ebs = $ebs
```

```
C:\> Edit-EC2InstanceAttribute -InstanceId i-1234567890abcdef0 -BlockDeviceMapping $bdm
```

## ルートボリュームが永続化するように設定されていることの確認

Amazon EC2 コンソールまたはコマンドラインツールを使用して、ルートボリュームが永続化するように設定されていることを確認できます。

Amazon EC2 コンソールを使用してルートボリュームが永続化するように設定されていることを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [インスタンス] を選択してから、インスタンスを選択します。
3. [Description (説明)] タブで、[ルートデバイス] のエントリを選択します。[合わせて削除] が `false` に設定されている場合、ボリュームは永続化するように設定されます。

AWS CLI を使用してルートボリュームが永続化するように設定されていることを確認するには

`describe-instances` コマンドを使用して、`BlockDeviceMappings` レスポンス要素の `DeleteOnTermination` 属性が `false` に設定されていることを確認します。

```
$ aws ec2 describe-instances --instance-id i-1234567890abcdef0
```

```
... "BlockDeviceMappings": [  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "Status": "attached",  
      "DeleteOnTermination": false,  
      "VolumeId": "vol-1234567890abcdef0",  
      "AttachTime": "2013-07-19T02:42:39.000Z"  
    }  
  }  
]
```

AWS Tools for Windows PowerShell を使用してルートボリュームが永続化するように設定されていることを確認するには

`Get-EC2Instance` コマンドを使用して、`BlockDeviceMappings` レスポンス要素の `DeleteOnTermination` 属性が `false` に設定されていることを確認します。

```
C:\> (Get-EC2Instance -InstanceId i-i-1234567890abcdef0).Instances.BlockDeviceMappings.Ebs
```

# Amazon EC2 でのセットアップ<sup>†</sup>

アマゾンウェブサービス (AWS) に既にサインアップしている場合は、Amazon EC2 をすぐに使用できます。Amazon EC2 コンソールを開き、[Launch Instance (インスタンスの起動)] を選択し、起動ウィザードの手順に従って最初のインスタンスを起動します。

AWS にまだサインアップしていない場合、または最初のインスタンスの起動についてサポートが必要な場合は、次のタスクを実行して、Amazon EC2 を使用するための設定を行ってください。

1. AWS にサインアップする (p. 22)
2. キーペアを作成する (p. 22)
3. セキュリティグループの作成 (p. 24)

## AWS にサインアップする

アマゾン ウェブ サービス (AWS) にサインアップすると、AWS アカウントが AWS 内のすべてのサービス (Amazon EC2 など) に自動的にサインアップされます。料金が発生するのは、実際に使用したサービスの分のみです。

Amazon EC2 については、お客様が利用された分のみのお支払いとなります。AWS の新規のお客様の場合、Amazon EC2 を無料で使い始めることができます。詳細については、「AWS 無料利用枠」を参照してください。

既に AWS アカウントをお持ちの場合は次のタスクに進んでください。AWS アカウントをお持ちでない場合は、次に説明する手順にしたがってアカウントを作成してください。

AWS アカウントを作成するには

1. <https://portal.aws.amazon.com/billing/signup> を開きます。
2. オンラインの手順に従います。

サインアップ手順の一環として、通話呼び出しを受け取り、電話のキーパッドを用いて確認コードを入力することが求められます。

## キーペアを作成する

AWS では公開キー暗号化を使用して、お客様のインスタンスのログイン情報の安全性を保護します。Linux インスタンスにはパスワードがありませんが、キーペアを使用することでインスタンスに安全にログインできます。インスタンスを起動するときにキーペアの名前を指定し、プライベートキーを指定して、SSH を使ってログインします。これにより、

キーペアをまだ作成していない場合は、Amazon EC2 コンソールを使用して作成できます。複数のリージョンでインスタンスを起動する予定がある場合は、各リージョンでキーペアを作成する必要があります。リージョンの詳細については、「[リージョン、アベイラビリティーゾーン、および ローカルゾーン \(p. 7\)](#)」を参照してください。

## キーペアを作成するには

1. AWS マネジメントコンソールにサインインをしたあと、<https://console.aws.amazon.com/ec2/> にある Amazon EC2 コンソールを開きます。
2. ナビゲーションバーで、キーペアを生成するリージョンを選択します。お客様は場所に関係なく、使用できるリージョンをどれでも選択できます。ただし、キーペアはリージョンに固有です。たとえば、米国東部（オハイオ）リージョンでインスタンスを起動する予定がある場合、米国東部（オハイオ）リージョンのインスタンス用にキーペアを作成する必要があります。



3. ナビゲーションペインの [NETWORK & SECURITY] で、[Key Pairs] を選択します。
4. [キーペアの作成] を選択します。
5. 次の作業を行います。
  - a. [Name (名前)] に、新しいキーペアのわかりやすい名前（お客様の名前など）を入力し、その後に -key-pair とリージョン名を続けます。たとえば、me-key-pair-useast2 とします。
  - b. [File format (ファイル形式)] で、プライベートキーを保存する形式を選択します。
    - OpenSSH で使用される形式でプライベートキーを保存するには、[pem] を選択します。
    - PuTTY (Windows から Linux インスタンスに接続できるツール) で使用される形式でプライベートキーを保存するには、[ppk] を選択します。
  - c. [Create] を選択します。
6. ブラウザによって秘密キーファイルが自動的にダウンロードされます。ベースファイル名はキーペアの名前として指定した名前となり、ファイル名の拡張子は .pem となります。プライベートキーファイルを安全な場所に保存します。

### Important

これは、プライベートキーを保存する唯一のチャンスです。インスタンスと対応するプライベートキーの起動時には、毎回インスタンスに接続するたびに、キーペアの名前を入力する必要があります。

- Mac または Linux コンピュータの SSH クライアントを使用して Linux インスタンスに接続する場合は、次のコマンドを使用してプライベートキーファイルの権限を設定すると、お客様以外のユーザーはそれを読み取ることができないようになります。

```
chmod 400 your_user_name-key-pair-region_name.pem
```

これらのアクセス権限を設定しないと、このキーペアを使用してインスタンスに接続できません。 詳細については、「[エラー: Unprotected Private Key File \(保護されていないプライベートキーファイル\) \(p. 1141\)](#)」を参照してください。

詳細については、「[Amazon EC2 のキーペア \(p. 899\)](#)」を参照してください。

## セキュリティグループの作成

セキュリティグループは、関連付けられたインスタンスのファイアウォールとして動作し、インバウンド トラフィックとアウトバウンド トラフィックの両方をインスタンスレベルでコントロールします。SSH を使用して IP アドレスからインスタンスに接続できるようにするためのルールをセキュリティグループに追加します。さらに、任意の場所からのインバウンドおよびアウトバウンドの HTTP アクセスおよび HTTPS アクセスを可能にするルールを追加できます。

複数のリージョンでインスタンスを起動する予定がある場合は、各リージョンでセキュリティグループを作成する必要があります。リージョンの詳細については、「[リージョン、アベイラビリティーゾーン、および ローカルゾーン \(p. 7\)](#)」を参照してください。

### 前提条件

ローカルコンピューターのパブリック IPv4 アドレスが必要です。Amazon EC2 コンソールのセキュリティグループエディタは、パブリック IPv4 アドレスを自動的に検出できます。別の方法として、インターネットブラウザで検索文字列として「私の IP アドレスは何ですか?」を使用するか、次のサービス: [Check IP](#) を使用することもできます。インターネットサービスプロバイダー (ISP) 経由で、またはファイアウォールの内側から静的 IP アドレスなしで接続する場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。

### 最小限の権限でセキュリティグループを作成するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

#### Tip

または、Amazon VPC コンソールを使用してセキュリティグループを作成することもできます。ただし、ここで説明する手順は、Amazon VPC コンソールと一致しません。したがって、前のセクションで Amazon VPC コンソールに切り替えた場合は、Amazon EC2 コンソールに戻ってここで説明する手順に従うか、『Amazon VPC 入門ガイド』の「[VPC のセキュリティグループをセットアップする](#)」の手順に従います。

- ナビゲーションバーで、セキュリティグループのリージョンを選択します。セキュリティグループはリージョンに固有であるため、キーペアを作成したリージョンと同じリージョンを選択する必要があります。



3. ナビゲーションペインで、[Security Groups] を選択します。
4. [Create Security Group] を選択します。
5. 新しいセキュリティグループの名前と説明を入力します。覚えやすい名前（ユーザー名など）を使用し、その後に \_SG\_ を続け、さらにリージョン名を続けます。たとえば、me\_SG\_uswest2 などです。
6. [VPC] リストで、リージョンのデフォルト VPC を選択します。
7. [Inbound] タブで、次のルールを作成し（新しいルールごとに [Add Rule] を選択）、最後に [Create] を選択します。
  - [Type] リストから [HTTP] を選択し、[Source] が [Anywhere] (0.0.0.0/0) に設定されていることを確認します。
  - [Type] リストから [HTTPS] を選択し、[Source] が [Anywhere] (0.0.0.0/0) に設定されていることを確認します。
  - [Type] リストから [SSH] を選択します。[Source] ボックスで [My IP] を選択すると、ローカルコンピューターのパブリック IPv4 アドレスが自動的にフィールドに入力されます。別の方法として、[Custom] を選択してコンピュータまたはネットワークのパブリック IPv4 アドレスを CIDR 表記で指定することもできます。CIDR 表記で個々の IP アドレスを指定するには、ルーティングサフィックス /32 を追加します（203.0.113.25/32 など）。会社が特定の範囲からアドレスを割り当てている場合、範囲全体（203.0.113.0/24 など）を指定します。

#### Warning

セキュリティ上の理由で、すべての IPv4 アドレス (0.0.0.0/0) からインスタンスへの SSH アクセスを許可することはお勧めしません。ただし、それがテスト目的で短期間の場合は例外です。

詳細については、「[Linux インスタンスの Amazon EC2 セキュリティグループ \(p. 911\)](#)」を参照してください。

# Amazon EC2 Linux インスタンスの開始方法

Linux インスタンスを起動、接続、使用して Amazon Elastic Compute Cloud (Amazon EC2) の使用を開始しましょう。インスタンスとは、AWS クラウドにある仮想サーバーです。Amazon EC2 を使用して、インスタンスで実行されるオペレーティングシステムとアプリケーションをセットアップし、設定することができます。

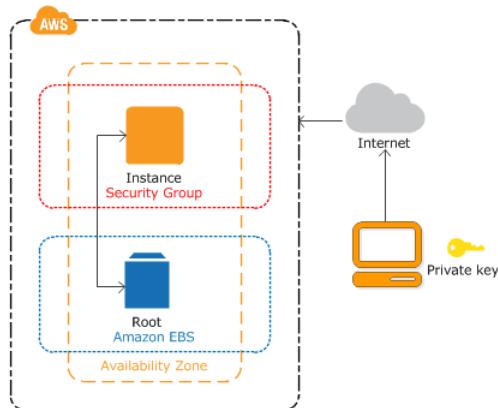
AWS にサインアップすると、[AWS 無料利用枠](#)を使って、Amazon EC2 を開始することができます。AWS アカウントを作成したのが過去 12 か月以内で、Amazon EC2 の無料利用枠を使い切っていない場合、無料利用枠内で利用できるオプションを選択することで、このチュートリアルでは一切費用がかかりません。それ以外の場合、インスタンスを起動したときから、インスタンスを削除するまで（このチュートリアルの最終タスク）、アイドル状態のままでも標準の Amazon EC2 使用料が発生します。

## 目次

- [概要 \(p. 26\)](#)
- [前提条件 \(p. 27\)](#)
- [ステップ 1: インスタンスを起動する \(p. 27\)](#)
- [ステップ 2: インスタンスに接続 \(p. 28\)](#)
- [ステップ 3: インスタンスをクリーンアップする \(p. 28\)](#)
- [次のステップ \(p. 29\)](#)

## 概要

インスタンスは Amazon EBS-backed インスタンスです（ルートボリュームが EBS ボリュームであることを意味します）。インスタンスが実行されるアベイラビリティゾーンは、指定するか、Amazon EC2 によって自動的に選択されます。インスタンスを起動するときは、キーペアとセキュリティグループを指定してインスタンスをセキュリティで保護しますインスタンスに接続するときは、インスタンスの起動時に指定したキーペアの秘密キーを指定する必要があります。



## タスク

このチュートリアルを完了するには、次のタスクを実行します。

1. インスタンスを起動する (p. 27)
2. インスタンスへの接続 (p. 28)
3. インスタンスのクリーンアップ (p. 28)

#### 関連チュートリアル

- Windows インスタンスを起動する場合は、『Windows インスタンスの Amazon EC2 ユーザーガイド』のチュートリアル「[Amazon EC2 Windows インスタンスの使用開始](#)」を参照してください。
- コマンドラインを使用する方法については、『AWS Command Line Interface ユーザーガイド』のチュートリアル「[Using Amazon EC2 through the AWS CLI](#)」を参照してください。

## 前提条件

開始する前に、必ず「[Amazon EC2 でのセットアップ \(p. 22\)](#)」の手順を完了してください。

## ステップ 1: インスタンスを起動する

以下の手順で説明しているように AWS マネジメントコンソールを使用して Linux インスタンスを起動できます。このチュートリアルは、初めてのインスタンスをすばやく起動できるように設計されています。そのため、可能なすべてのオプションを扱ってはいません。オプションの詳細については、「[Launching an Instance](#)」を参照してください。

#### インスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. コンソールダッシュボードで、[Launch Instance] を選択します。
3. [Choose an Amazon Machine Image (AMI) (Amazon マシンイメージ (AMI) を選択)] ページに、Amazon マシンイメージ (AMI) と呼ばれる基本設定のリストが表示されます。これは、インスタンスのテンプレートとして機能します。HVM バージョンの Amazon Linux 2 を選択します。これらの AMI は [Free tier eligible] と表示されていることに注意してください。
4. [Choose an Instance Type] ページで、インスタンスのハードウェア構成を選択できます。t2.micro タイプを選択します。これはデフォルトで選択されています。このインスタンスタイプは無料利用枠であることに注意してください。
5. [Review and Launch] を選択して、ウィザードが他の設定を完了できるようにします。
6. [Review Instance Launch] ページの [Security Groups] に、ウィザードで作成および選択したセキュリティグループが表示されます。このセキュリティグループを使用するか、または次のステップを使用して設定を行うときに作成したセキュリティグループを選択できます。
  - a. [Edit security groups] を選択します。
  - b. [Configure Security Group] ページで、[Select an existing security group] が選択されていることを確認します。
  - c. 既存のセキュリティグループのリストからセキュリティグループを選択してから、[Review and Launch] を選択します。
7. [Review Instance Launch] ページで、[Launch] を選択します。
8. キーペアの入力を求められたら、[Choose an existing key pair] を選択し、セットアップ中に作成したキーペアを選択します。

新しいキーペアを作成することもできます。[Create a new key pair] を選択し、キーペアの名前を入力してから、[Download Key Pair] を選択します。秘密キーファイルはこのときしか保存できないため、必ずダウンロードしてください。プライベートキーファイルを安全な場所に保存します。インスタン

スと対応するプライベートキーの起動時には、毎回インスタンスに接続するたびに、キーペアの名前を入力する必要があります。

Warning

[Proceed without a key pair] オプションは選択しないでください。キーペアを使用せずにインスタンスを起動すると、インスタンスに接続できません。

準備ができたら、確認チェックボックスをオンにし、[Launch Instances] を選択します。

9. インスタンスを起動することを知らせる確認ページが表示されます。[View Instances] を選択して確認ページを閉じ、コンソールに戻ります。
10. [Instances] 画面に起動のステータスが表示されます。インスタンスはすぐに起動します。インスタンスを起動した直後のステータスは pending です。インスタンスが開始されると、ステータスは running に変わり、インスタンスはパブリック DNS 名を取得します([Public DNS (IPv4)] 列が表示されない場合は、ページの右上隅にある [Show/Hide Columns] (歯車型のアイコン) を選択してから、[Public DNS (IPv4)] を選択します)。
11. インスタンスの準備ができるようになるまでには、数分かかる場合があります。インスタンスのステータスチェックが正常に終了したことを確認してください。この情報は [Status Checks] 列で確認できます。

## ステップ 2: インスタンスに接続

Linux インスタンスに接続するにはいくつかの方法があります。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。

Important

.pem ファイルがあるキーペアで起動し、コンピュータから SSH アクセスを許可するセキュリティグループで起動していない限り、インスタンスに接続することはできません。インスタンスに接続できない場合は、「[インスタンスへの接続に関するトラブルシューティング \(p. 1135\)](#)」を参照してください。

## ステップ 3: インスタンスをクリーンアップする

このチュートリアル用に作成したインスタンスを使用した操作が終了したら、インスタンスを終了してクリーンアップする必要があります。クリーンアップする前にこのインスタンスでやることがある場合は、「[次のステップ \(p. 29\)](#)」を参照してください。

Important

インスタンスを削除するということは、実質的には、そのインスタンスを削除するということです。いったん終了したインスタンスに再接続することはできません。

[AWS 無料利用枠](#)外でインスタンスを起動した場合は、インスタンスのステータスが `shutting down` または `terminated` になるとインスタンスの課金が停止します。後のためにインスタンスを維持したいが料金を発生させたくない場合は、インスタンスを停止して後で再び開始できます。詳細については、「[インスタンスの停止](#)」を参照してください。

インスタンスを終了するには

1. ナビゲーションペインで、[インスタンス] を選択します。インスタンスの一覧で、インスタンスを選択します。
2. [Actions]、[Instance State]、[Terminate] の順に選択します。
3. 確認を求めるメッセージが表示されたら、[Yes, Terminate] を選択します。

Amazon EC2 によって、インスタンスがシャットダウンおよび終了します。インスタンスの終了後、インスタンスはしばらくの間コンソールに表示されたままですが、エントリは削除されます。

## 次のステップ

インスタンスを起動した後で、次の演習の一部を行ってみていいかもしれません。

- Run Command を使用してリモートに EC2 インスタンスを管理する方法を説明します。詳細については、『AWS Systems Manager ユーザーガイド』の「[AWS Systems Manager Run Command](#)」を参照してください。
- 使用量が無料利用枠を超えた場合に通知する CloudWatch アラームの設定。詳細については、『AWS Billing and Cost Management ユーザーガイド』の「[請求アラームの作成](#)」を参照してください。
- EBS ボリュームの追加。詳細については、「[Amazon EBS ボリュームの作成 \(p. 949\)](#)」および「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。
- LAMP スタックのインストール。詳細については、「[チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする \(p. 32\)](#)」を参照してください。

# Amazon EC2 のベストプラクティス

このプラクティスのリストにより、Amazon EC2 の利点を最大限に活用できます。

## セキュリティとネットワーク

- ID フェデレーション、IAM ユーザー、IAM ロールを使用して、AWS リソースおよび API へのアクセスを管理します。AWS アクセス認証情報の作成、配布、ローテーション、および取り消しを行うための認証情報管理のポリシーおよび手順を確立します。詳細については、IAM ユーザーガイドの「[IAM のベストプラクティス](#)」を参照してください。
- セキュリティグループに対して、最小権限となるルールを適用します。詳細については、「[セキュリティグループのルール \(p. 912\)](#)」を参照してください。
- 定期的にインスタンスのオペレーティングシステムやアプリケーションに対してパッチ処理、更新、および保護を行います。Amazon Linux 2 または Amazon Linux AMI の更新の詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「[ソフトウェアの管理 \(Linux インスタンスの場合\)](#)」を参照してください。

## ストレージ

- データの永続性、バックアップ、および復元に対するルートデバイスタイプの影響について理解します。詳細については、「[ルートデバイスのストレージ \(p. 96\)](#)」を参照してください。
- オペレーティングシステム用およびデータ用として個別に Amazon EBS ボリュームを使用します。データのボリュームがインスタンス終了後も保持されることを確認します。詳細については、「[インスタンスの削除で Amazon EBS ボリュームを保持する \(p. 549\)](#)」を参照してください。
- インスタンスで一時データの格納に使用できるインスタンスストアを使用します。インスタンスを停止または終了すると、インスタンスストアに格納されたデータは削除されることに注意してください。データベースストレージにインスタンスストアを使用する場合は、耐障害性を確保するレプリケーション係数が設定されたクラスターがあることを確認します。
- EBS ボリュームとスナップショットを暗号化します。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

## リソース管理

- AWS リソースを追跡および識別するために、インスタンスマタデータおよびリソースのカスタムタグを使用します。詳細については、「[インスタンスマタデータとユーザーデータ \(p. 593\)](#)」および「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。
- Amazon EC2 の現在の制限を表示します。制限の引き上げに対するリクエストは、制限の引き上げが必要となる前に計画してください。詳細については、「[Amazon EC2 サービスの制限 \(p. 1130\)](#)」を参照してください。

## バックアップ & リストア

- 定期的に [Amazon EBS スナップショット \(p. 970\)](#) を使用して EBS ボリュームをバックアップし、インスタンスから [Amazon Machine Image \(AMI\) \(p. 94\)](#) を作成して、それ以降にインスタンスを起動するためのテンプレートとして設定を保存します。
- 複数のアベイラビリティゾーンにアプリケーションの重要なコンポーネントをデプロイし、データを適切にレプリケートします。
- インスタンスが再開したときに、動的な IP アドレスを処理するアプリケーションを設計します。詳細については、「[Amazon EC2 インスタンスの IP アドレッシング \(p. 685\)](#)」を参照してください。

- ・イベントを管理し、対応します。詳細については、「[Amazon EC2 のモニタリング \(p. 625\)](#)」を参照してください。
- ・フェイルオーバーを処理する準備が整っていることを確認します。基本的な解決策として、手動でネットワークインターフェイスをアタッチすることも、代替インスタンスに Elastic IP アドレスを関連付けることもできます。詳細については、「[Elastic Network Interface \(p. 713\)](#)」を参照してください。自動化されたソリューションとして Amazon EC2 Auto Scaling を使用できます。詳細については、「[Amazon EC2 Auto Scaling ユーザーガイド](#)」を参照してください。
- ・障害が発生した場合にインスタンスと Amazon EBS ボリュームを復元するプロセスを定期的にテストします。

# Linux を実行する Amazon EC2 インスタンスのチュートリアル

次のチュートリアルでは、Linux を実行する EC2 インスタンスを使用して一般的なタスクを実行する方法を示します。動画については、「[AWS 講習動画とラボ](#)」を参照してください。

## チュートリアル

- [チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする \(p. 32\)](#)
- [チュートリアル: Amazon Linux AMI を使用して LAMP ウェブサーバーをインストールする \(p. 42\)](#)
- [チュートリアル: Amazon Linux を使った WordPress ブログのホスティング \(p. 53\)](#)
- [チュートリアル: Amazon Linux 2 に SSL/TLS を設定する \(p. 62\)](#)
- [チュートリアル: Amazon Linux に SSL/TLS を設定する \(p. 77\)](#)
- [チュートリアル: Amazon EC2 のアプリケーションの可用性の向上 \(p. 90\)](#)

## チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする

次の手順では、Apache ウェブサーバーを PHP と MariaDB (コミュニティによって開発された MySQL の派生版) のサポートとともに Amazon Linux 2 インスタンスにインストールします (LAMP ウェブサーバーまたは LAMP スタックとも呼ばれます)。このサーバーを使用して静的ウェブサイトをホストしたり、データベースとの情報の読み取りと書き込みを行う動的な PHP アプリケーションをデプロイしたりできます。

### Important

Amazon Linux AMI で LAMP ウェブサーバーを設定するには、「[チュートリアル: Amazon Linux AMI を使用して LAMP ウェブサーバーをインストールする \(p. 42\)](#)」を参照してください。

Ubuntu または Red Hat Enterprise Linux インスタンスでの LAMP ウェブサーバーのセットアップは、このチュートリアルの範囲外です。その他のディストリビューションの詳細については、各ドキュメントを参照してください。Ubuntu の LAMP ウェブサーバーについては、Ubuntu コミュニティのドキュメントの [ApacheMySQLPHP](#) トピックを参照してください。

オプション: オートメーションを使用してこのチュートリアルを完了する

以下のタスクを行う代わりに AWS Systems Manager Automation を使用してこのチュートリアルを完了するには、Automation ドキュメントである [AWS Docs - Install a LAMP Server - AL2](#) を実行します。

### タスク

- [ステップ 1: LAMP サーバーを準備する \(p. 33\)](#)
- [ステップ 2: LAMP サーバーをテストする \(p. 36\)](#)
- [ステップ 3: データベースサーバーをセキュリティで保護する \(p. 37\)](#)
- [ステップ 4: \(オプション\) phpMyAdmin をインストールする \(p. 38\)](#)
- [トラブルシューティング \(p. 41\)](#)

- 関連トピック (p. 41)

## ステップ 1: LAMP サーバーを準備する

### 前提条件

このチュートリアルでは、インターネットからアクセス可能なパブリック DNS 名を持つ、Amazon Linux 2 を使用する新しいインスタンスをすでに起動していることを前提にしています。詳細については、「[ステップ 1: インスタンスを起動する \(p. 27\)](#)」を参照してください。また、セキュリティグループを設定して、SSH (ポート 22)、HTTP (ポート 80)、HTTPS (ポート 443) 接続を有効にしている必要があります。前提条件の詳細については、[Linux インスタンス用の受信トラフィックの認可 \(p. 897\)](#) を参照してください。

### Note

次の手順により Amazon Linux 2 で使用可能な最新の PHP バージョンがインストールされます。現在は PHP 7.2 です。このチュートリアルで説明されている以外の PHP アプリケーションを使用する場合は、PHP 7.2 と互換性を確認する必要があります。

### LAMP サーバーを準備するには

1. [インスタンスに接続します \(p. 28\)](#)。
2. すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。この処理には数分かかりますが、最新の更新とバグ修正を確実に適用することが重要です。

-y オプションを指定すると、確認メッセージを表示せずに更新をインストールします。インストール前に更新を検査する場合は、このオプションを省略できます。

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

3. lamp-mariadb10.2-php7.2 と php7.2 Amazon Linux Extras リポジトリをインストールして、LAMP MariaDB と Amazon Linux 2 PHP パッケージの最新バージョンを取得します。

```
[ec2-user ~]$ sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
```

### Note

sudo: amazon-linux-extras: command not found というエラーが表示された場合、インスタンスは Amazon Linux 2 AMI で起動されていません (おそらく、代わりに Amazon Linux AMI を使用しています)。次のコマンドを使用して、Amazon Linux のバージョンを表示できます。

```
cat /etc/system-release
```

Amazon Linux AMI で LAMP ウェブサーバーを設定するには、「[チュートリアル: Amazon Linux AMI を使用して LAMP ウェブサーバーをインストールする \(p. 42\)](#)」を参照してください。

4. これでインスタンスが最新状態になったので、Apache ウェブサーバー、MariaDB、PHP ソフトウェアパッケージをインストールできます。

yum install コマンドを使用すると、複数のソフトウェアパッケージと関連するすべての依存関係を同時にインストールできます。

```
[ec2-user ~]$ sudo yum install -y httpd mariadb-server
```

Note

次のコマンドを使用して、これらのパッケージの現在のバージョンを表示できます。

```
yum info package_name
```

- Apache ウェブサーバーを起動します。

```
[ec2-user ~]$ sudo systemctl start httpd
```

- systemctl コマンドを使用して、システムがブートするたびに Apache ウェブサーバーが起動するよう に設定します。

```
[ec2-user ~]$ sudo systemctl enable httpd
```

httpd が有効であることは、次のコマンドを実行して確認できます。

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

- インバウンド HTTP (ポート 80) 接続をインスタンスに許可するセキュリティルールを追加していない場合には、このルールを追加します。デフォルトでは、起動時に [launch-wizard-**N**] セキュリティグループがインスタンスに設定されます。このグループには SSH 接続を許可する単一のルールが含まれます。

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- [インスタンス] を選択し、該当するインスタンスを選択します。
- [セキュリティグループ] で [インバウンドルールの表示] を選択します。
- デフォルトのセキュリティグループに次のルールの一覧が表示されます。

```
Security Groups associated with i-1234567890abcdef0
Ports      Protocol      Source      launch-wizard-N
22         tcp          0.0.0.0/0    #
```

「[セキュリティグループへのルールの追加 \(p. 916\)](#)」の手順を使用して、次の値で新しいインバウンドセキュリティルールを追加します。

- [Type]: HTTP
- [Protocol]: TCP
- [Port Range]: 80
- [Source]: Custom

- ウェブサーバーをテストします。ウェブブラウザで、インスタンスのパブリック DNS アドレス (またはパブリック IP アドレス) を入力します。/var/www/html にコンテンツがない場合、Apache テストページが表示されます。インスタンスのパブリック DNS は、Amazon EC2 コンソールを使用して取得できます ([Public DNS] 列を確認します。この列が表示されない場合は、[Show/Hide Columns] (歯車型のアイコン) をクリックして、[Public DNS] を選択します)。

Apache テストページが表示されない場合、使用しているセキュリティグループに、HTTP (ポート 80) トラフィックを許可するルールが含まれていることを確認します。HTTP ルールをセキュリティグループに追加する方法については、[セキュリティグループへのルールの追加 \(p. 916\)](#) を参照してください。

Important

Amazon Linux を使用していない場合は、それらの接続を許可するようにインスタンスのファイアウォールを設定する必要があるかもしれません。ファイアウォールの設定方法の詳細については、ディストリビューション用のドキュメントを参照してください。

## Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

### If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](http://www.example.com), you should send e-mail to "webmaster@example.com".

### If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being displayed, follow the instructions in the file `/etc/httpd/conf/wELCOME.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



Apache httpd は、Apache ドキュメントルートと呼ばれるディレクトリに維持されるファイルを提供します。Amazon Linux Apache ドキュメントルートは `/var/www/html` であり、デフォルトでは `root` によって所有されます。

`ec2-user` アカウントがこのディレクトリで複数のファイルを操作することを許可するには、ディレクトリの所有権とアクセス許可を変更する必要があります。このタスクを行うには、複数の方法があります。このチュートリアルでは、`ec2-user` を `apache` グループに追加し、`apache` ディレクトリの所有権を `/var/www` グループに付与し、グループへの書き込み権限を割り当てます。

#### ファイルの許可を設定するには

1. ユーザー (この場合は `ec2-user`) を `apache` グループに追加します。

```
[ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

2. ログアウトし、再度ログインして新しいグループを選択し、メンバーシップを確認します。
  - a. ログアウトします (`exit` コマンドを使用するか、ターミナルウィンドウを閉じます)。

```
[ec2-user ~]$ exit
```

- b. `apache` グループのメンバーシップを検証するには、インスタンスに再接続して次のコマンドを実行します。

```
[ec2-user ~]$ groups  
ec2-user adm wheel apache systemd-journal
```

3. /var/www とそのコンテンツのグループ所有権を apache グループに変更します。

```
[ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```

4. グループの書き込み許可を追加して、これからサブディレクトにグループ ID を設定するには、/var/www とサブディレクトのディレクトリ許可を変更します。

```
[ec2-user ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
```

5. グループ書き込み許可を追加するには、/var/www とサブディレクトリのファイル許可を再帰的に変更します。

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

ここで、ec2-user (および apache グループの将来のメンバー) は、Apache ドキュメントルートでファイルを追加、削除、編集できるようになります。したがって、静的ウェブサイトや PHP アプリケーションなどのコンテンツを追加できます。

#### ウェブサーバーを保護するには (オプション)

HTTP プロトコルを実行するウェブサーバーは、送受信したデータのransportセキュリティを提供しません。ウェブブラウザを使用して HTTP サーバーに接続すると、閲覧した URL、受信したウェブページのコンテンツ、送信した HTML フォームの内容 (パスワードなど) はすべて、ネットワーク経路上のどちらでも傍受できるようになります。ウェブサーバーを保護するためのベストプラクティスとして、SSL/TLS 暗号化でデータを保護する HTTPS (HTTP Secure) のサポートをインストールしてください。

サーバーで HTTPS を有効にする方法については、「チュートリアル: Amazon Linux 2 に SSL/TLS を設定する (p. 62)」を参照してください。

## ステップ 2: LAMP サーバーをテストする

サーバーがインストールおよび実行されており、ファイルのアクセス許可が正しく設定されている場合、ec2-user アカウントは、インターネットから使用できる /var/www/html ディレクトリに PHP ファイルを作成できます。

#### LAMP サーバーをテストするには

1. Apache ドキュメントルートで PHP ファイルを作成します。

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

このコマンドを実行しようとしたときに「許可が拒否されました」というエラーが表示された場合は、ログアウトし、再度ログインして、[ファイルの許可を設定するには \(p. 35\)](#) で設定した正しいグループ許可を取得します。

2. ウェブブラウザで、作成したファイルの URL を入力します。この URL は、インスタンスのパブリック DNS アドレスにスラッシュとファイル名を追加したものです。例:

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

PHP 情報ページが表示されるはずです。

## PHP Version 7.2.0

System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64
Build Date	Dec 13 2017 03:34:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-datetime.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-domxml.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mysqlind.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-pspell.ini, /etc/php.d/20-session.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pspell.ini, /etc/php.d/30-session.ini, /etc/php.d/pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

### Note

このページが表示されない場合は、前のステップで `/var/www/html/phpinfo.php` ファイルが正しく作成されたことを確認します。次のコマンドで、必要なパッケージがすべてインストールされたことを確認することもできます。

```
[ec2-user ~]$ sudo yum list installed httpd mariadb-server php-mysqlnd
```

必要なパッケージのいずれかが出力に表示されていない場合は、`sudo yum install package` コマンドを使ってインストールします。また、`php7.2` と `1amp-mariadb10.2-php7.2` のエキストラが `amazon-linux-extras` のコマンド出力で有効になっていることを確認してください。

3. `phpinfo.php` ファイルを削除します。これは有用な情報であることもあります、セキュリティ上の理由から、インターネット上で公表しないでください。

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

これで、完全に機能する LAMP ウェブサーバーを設定しました。`/var/www/html` の Apache ドキュメントルートにコンテンツを追加する場合、そのコンテンツはインスタンスのパブリック DNS アドレスで表示できます。

## ステップ 3: データベースサーバーをセキュリティで保護する

MariaDB サーバーのデフォルトのインストールには、テストおよび開発に役立ついくつかの機能がありますが、実稼働サーバーでは無効にするか削除する必要があります。`mysql_secure_installation` コマンドを

使用すると、ルートパスワードを設定し、安全でない機能をインストールから削除する手順が案内されます。MariaDB サーバーを使用する予定がない場合でも、この手順を実行することが推奨されます。

MariaDB サーバーをセキュリティで保護するには

1. MariaDB サーバーを起動します。

```
[ec2-user ~]$ sudo systemctl start mariadb
```

2. mysql\_secure\_installation を実行します。

```
[ec2-user ~]$ sudo mysql_secure_installation
```

- a. プロンプトが表示されたら、ルートアカウントのパスワードを入力します。

- i. 現在のルートパスワードを入力します。デフォルトでは、ルートアカウントにはパスワードが設定されていません。Enter キーを押します。
- ii. 「y」と入力してパスワードを設定し、安全なパスワードを 2 回入力します。安全なパスワード作成の詳細については、「<https://identitysafe.norton.com/password-generator/>」を参照してください。このパスワードは必ず安全な場所に保管します。

#### Note

MariaDB のルートパスワードの設定は、データベースを保護するための最も基本的な手段にすぎません。データベース駆動型アプリケーションを構築またはインストールする必要がある場合、通常はそのアプリケーションのデータベースサービスユーザーを作成します。ルートアカウントは、データベース管理以外には使用しないでください。

- b. 「y」と入力して匿名ユーザーアカウントを削除します。
  - c. 「y」と入力してリモートルートログインを無効にします。
  - d. 「y」と入力してテストデータベースを削除します。
  - e. 「y」と入力して権限テーブルを再ロードし、変更を保存します。
3. (オプション) MariaDB サーバーをすぐに使用する予定がない場合は、これを停止します。再び必要になったときには再起動できます。

```
[ec2-user ~]$ sudo systemctl stop mariadb
```

4. (オプション) ブート時に毎回 MariaDB サーバーを起動させる場合は、次のコマンドを入力します。

```
[ec2-user ~]$ sudo systemctl enable mariadb
```

## ステップ 4: (オプション) phpMyAdmin をインストールする

phpMyAdmin は、EC2 インスタンスで MySQL データベースを表示して編集するために使用できる、ウェブベースのデータベース管理ツールです。Amazon Linux インスタンスで phpMyAdmin をインストールして設定するには、以下の手順に従ってください。

#### Important

Apache で SSL/TLS を有効にしていない場合、LAMP サーバーへのアクセスに phpMyAdmin を使用することは推奨されません。そのようにすると、データベース管理者のパスワードや他のデータは、インターネット上を安全ではない状態で送信されます。開発者によるセキュリティ関

連の推奨事項については、「[Securing your phpMyAdmin installation](#)」を参照してください。EC2 インスタンスでのウェブサーバーの保護に関する一般的な情報については、「[チュートリアル: Amazon Linux 2 に SSL/TLS を設定する \(p. 62\)](#)」を参照してください。

### phpMyAdmin をインストールするには

- 必要な依存ファイルをインストールします。

```
[ec2-user ~]$ sudo yum install php-mbstring -y
```

- Apache を再起動します。

```
[ec2-user ~]$ sudo systemctl restart httpd
```

- 再起動 php-fpm.

```
[ec2-user ~]$ sudo systemctl restart php-fpm
```

- /var/www/html で Apache ドキュメントルートに移動します。

```
[ec2-user ~]$ cd /var/www/html
```

- <https://www.phpmyadmin.net/downloads> で最新の phpMyAdmin リリース用のソースパッケージを選択します。ファイルディレクトリをインスタンスにダウンロードするには、次の例のようにリンクをコピーして wget コマンドに貼り付けます。

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

- phpMyAdmin フォルダを作成し、次のコマンドでパッケージを展開します。

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components 1
```

- `phpMyAdmin-latest-all-languages.tar.gz` Tarball を削除します。

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

- (オプション) MySQL サーバーが実行中ではない場合は、今すぐ起動します。

```
[ec2-user ~]$ sudo systemctl start mariadb
```

- ウェブブラウザで、phpMyAdmin のインストール URL を入力します。この URL は、インスタンスのパブリック DNS アドレス (または、パブリック IP アドレス) にスラッシュとインストールディレクトリを追加してものです。例:

```
http://my.public.dns.amazonaws.com/phpMyAdmin
```

phpMyAdmin ログインページが表示されます。



10. 前に作成した `root` ユーザー名と MySQL のルートパスワードを使って、phpMyAdmin インストールにログインします。

インストールは、サービス開始前に設定する必要があります。phpMyAdmin を設定するには、[手動で設定ファイルを作成する](#)、[設定コンソールを使用する](#)、またはその両方の方法を組み合わせることができます。

phpMyAdmin の使用に関する情報は、「[phpMyAdmin ユーザーガイド](#)」を参照してください。

## トラブルシューティング

このセクションでは、新しい LAMP サーバーの設定時に発生する可能性がある一般的な問題の解決案を提供します。

### ウェブブラウザを使用してサーバーに接続できません。

以下のチェックを行って、Apache ウェブサーバーが実行されていて、アクセス可能であるかどうかを確認します。

- ウェブサーバーが実行されていますか？

httpd が有効であることは、次のコマンドを実行して確認できます。

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

httpd プロセスが実行されていない場合は、[LAMP サーバーを準備するには \(p. 33\)](#) に記載されているステップを繰り返します。

- ファイアウォールは正しく設定されていますか？

Apache テストページが表示されない場合、使用しているセキュリティグループに、HTTP (ポート 80) トラフィックを許可するルールが含まれていることを確認します。HTTP ルールをセキュリティグループに追加する方法については、[セキュリティグループへのルールの追加 \(p. 916\)](#) を参照してください。

## 関連トピック

インスタンスへのファイルの転送、またはウェブサーバーへの WordPress ブログのインストールの詳細については、次のドキュメントを参照してください。

- [WinSCP を使用した Linux インスタンスへのファイルの転送 \(p. 523\)](#)
- [SCP を使用した Linux から Linux インスタンスへのファイルの転送 \(p. 509\)](#)
- [チュートリアル: Amazon Linux を使った WordPress ブログのホスティング \(p. 53\)](#)

このチュートリアルで使用されているコマンドおよびソフトウェアの詳細については、次のウェブページを参照してください。

- Apache ウェブサーバー: <http://httpd.apache.org/>
- MariaDB データベースサーバー: <https://mariadb.org/>
- PHP プログラミング言語: <http://php.net/>
- chmod コマンド: <https://en.wikipedia.org/wiki/Chmod>
- chown コマンド: <https://en.wikipedia.org/wiki/Chown>

ウェブサーバーのドメイン名の登録、または、既存のドメイン名をこのホストに移す方法についての詳細は、『Amazon Route 53 開発者ガイド』の「[Amazon Route 53 のドメインとサブドメインの作成と移行](#)」を参照してください。

# チュートリアル: Amazon Linux AMI を使用して LAMP ウェブサーバーをインストールする

次の手順では、Apache ウェブサーバーを PHP と MySQL のサポートとともに Amazon Linux インスタンスにインストールします (LAMP ウェブサーバーまたは LAMP スタックとも呼ばれます)。このサーバーを使用して静的ウェブサイトをホストしたり、データベースとの情報の読み取りと書き込みを行う動的な PHP アプリケーションをデプロイしたりできます。

## Important

Amazon Linux 2 で LAMP ウェブサーバーを設定するには、「[チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする \(p. 32\)](#)」を参照してください。

Ubuntu または Red Hat Enterprise Linux インスタンスでの LAMP ウェブサーバーのセットアップは、このチュートリアルの範囲外です。その他のディストリビューションの詳細については、各ドキュメントを参照してください。Ubuntu の LAMP ウェブサーバーについては、Ubuntu コミュニティのドキュメントの [ApacheMySQLPHP](#) トピックを参照してください。

**オプション: オートメーションを使用してこのチュートリアルを完了する**

以下のタスクを行う代わりに AWS Systems Manager Automation を使用してこのチュートリアルを完了するには、Automation ドキュメントである [AWS Docs - Install a LAMP Server - AL](#) を実行します。

## タスク

- [ステップ 1: LAMP サーバーを準備する \(p. 42\)](#)
- [ステップ 2: LAMP サーバーをテストする \(p. 46\)](#)
- [ステップ 3: データベースサーバーをセキュリティで保護する \(p. 48\)](#)
- [ステップ 4: \(オプション\) phpMyAdmin をインストールする \(p. 49\)](#)
- [トラブルシューティング \(p. 52\)](#)
- [関連トピック \(p. 53\)](#)

## ステップ 1: LAMP サーバーを準備する

### 前提条件

このチュートリアルでは、インターネットからアクセス可能なパブリック DNS 名を持つ、Amazon Linux AMI を使用する新しいインスタンスをすでに起動していることを前提にしています。詳細については、「[ステップ 1: インスタンスを起動する \(p. 27\)](#)」を参照してください。また、セキュリティグループを設定して、SSH (ポート 22)、HTTP (ポート 80)、HTTPS (ポート 443) 接続を有効にしている必要があります。前提条件の詳細については、[Linux インスタンス用の受信トラフィックの認可 \(p. 897\)](#) を参照してください。

Amazon Linux AMI を使用して LAMP ウェブサーバーをインストールして起動するには

1. [インスタンスに接続します \(p. 28\)](#)。
2. すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。この処理には数分かかりますが、最新の更新とバグ修正を確実に適用することが重要です。

`-y` オプションを指定すると、確認メッセージを表示せずに更新をインストールします。インストール前に更新を検査する場合は、このオプションを省略できます。

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

3. これでインスタンスが最新状態になったので、Apache ウェブサーバー、MySQL、PHP ソフトウェア パッケージをインストールできます。

Note

アプリケーションによっては、次の推奨のソフトウェア環境と互換性がない場合があります。これらのパッケージをインストールする前に、LAMP アプリケーションと互換性があることを確認してください。問題がある場合には、代替環境のインストールが必要になることがあります。詳細については、「[サーバーで実行するアプリケーションソフトウェアに、インストールされている PHP バージョンまたは他のソフトウェアとの互換性がありません。\(p. 52\)](#)」を参照してください。

yum install コマンドを使用すると、複数のソフトウェアパッケージと関連するすべての依存関係を同時にインストールできます。

```
[ec2-user ~]$ sudo yum install -y httpd24 php72 mysql57-server php72-mysqld
```

Note

No package *package-name* available エラーが表示された場合、インスタンスは Amazon Linux AMI で起動されていません(おそらく、代わりに Amazon Linux 2 を使用しています)。次のコマンドを使用して、Amazon Linux のバージョンを表示できます。

```
cat /etc/system-release
```

4. Apache ウェブサーバーを起動します。

```
[ec2-user ~]$ sudo service httpd start
Starting httpd: [ OK ]
```

5. chkconfig コマンドを使用して、システムがブートするたびに Apache ウェブサーバーが起動するように設定します。

```
[ec2-user ~]$ sudo chkconfig httpd on
```

chkconfig コマンドでは、それを使用してサービスを正常に有効にしたときに確認メッセージは一切表示されません。

httpd が有効であることは、次のコマンドを実行して確認できます。

```
[ec2-user ~]$ chkconfig --list httpd
httpd      0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

ここで、httpd は実行レベル 2、3、4、および 5 で on です(これらの値である必要があります)。

6. インバウンド HTTP(ポート 80)接続をインスタンスに許可するセキュリティルールを追加していない場合には、このルールを追加します。デフォルトでは、起動時に [Launch-Wizard-N] セキュリティグループがインスタンスに設定されます。このグループには SSH 接続を許可する単一のルールが含まれます。

- a. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- b. [インスタンス] を選択し、該当するインスタンスを選択します。
- c. [セキュリティグループ] で [インバウンドルールの表示] を選択します。
- d. デフォルトのセキュリティグループに次のルールの一覧が表示されます。

```
Security Groups associated with i-1234567890abcdef0
```

Ports	Protocol	Source	launch-wizard- <i>N</i>
22	tcp	0.0.0.0/0	#

「セキュリティグループへのルールの追加 (p. 916)」の手順を使用して、次の値で新しいインバウンドセキュリティルールを追加します。

- [Type]: HTTP
- [Protocol]: TCP
- [Port Range]: 80
- [Source]: Custom

7. ウェブサーバーをテストします。ウェブブラウザで、インスタンスのパブリック DNS アドレス (またはパブリック IP アドレス) を入力します。/var/www/html にコンテンツがない場合、Apache テストページが表示されます。インスタンスのパブリック DNS は、Amazon EC2 コンソールを使用して取得できます ([Public DNS] 列を確認します。この列が表示されない場合は、[Show/Hide Columns] (歯車型のアイコン) をクリックして、[Public DNS] を選択します)。

Apache テストページが表示されない場合、使用しているセキュリティグループに、HTTP (ポート 80) トラフィックを許可するルールが含まれていることを確認します。HTTP ルールをセキュリティグループに追加する方法については、[セキュリティグループへのルールの追加 \(p. 916\)](#) を参照してください。

**Important**

Amazon Linux を使用していない場合は、それらの接続を許可するようにインスタンスのファイアウォールを設定する必要があるかもしれません。ファイアウォールの設定方法の詳細については、ディストリビューション用のドキュメントを参照してください。

## Amazon Linux AMI Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly, but has not yet been configured.

### If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](http://www.example.com), you should send e-mail to ["webmaster@example.com"](mailto:webmaster@example.com).

The [Amazon Linux AMI](#) is a supported and maintained Linux image provided by [Amazon Web Services](#) for use on [Amazon Elastic Compute Cloud \(Amazon EC2\)](#). It is designed to provide a stable, secure, and high performance execution environment for applications running on [Amazon EC2](#). It also includes packages that enable easy integration with [AWS](#), including launch configuration tools and many popular AWS libraries and tools. [Amazon Web Services](#) provides ongoing security and maintenance updates to all instances running the [Amazon Linux AMI](#). The [Amazon Linux AMI](#) is provided at no additional charge to [Amazon EC2 users](#).

### If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and Amazon Linux AMI powered HTTP servers. Thanks for using Apache and the Amazon Linux AMI!



### Note

このテストページが表示されるのは、`/var/www/html` にコンテンツがない場合のみです。ドキュメントルートにコンテンツを追加すると、コンテンツはこのテストページではなく、インスタンスのパブリック DNS アドレスに表示されます。

Apache httpd は、Apache ドキュメントルートと呼ばれるディレクトリに維持されるファイルを提供します。Amazon Linux Apache ドキュメントルートは `/var/www/html` であり、デフォルトでは root によって所有されます。

```
[ec2-user ~]$ ls -l /var/www
total 16
drwxr-xr-x 2 root root 4096 Jul 12 01:00 cgi-bin
drwxr-xr-x 3 root root 4096 Aug 7 00:02 error
drwxr-xr-x 2 root root 4096 Jan 6 2012 html
drwxr-xr-x 3 root root 4096 Aug 7 00:02 icons
drwxr-xr-x 2 root root 4096 Aug 7 21:17 noindex
```

ec2-user アカウントがこのディレクトリで複数のファイルを操作することを許可するには、ディレクトリの所有権とアクセス許可を変更する必要があります。このタスクを行うには、複数の方法があります。

このチュートリアルでは、`ec2-user` を `apache` グループに追加し、`apache` ディレクトリの所有権を /var/www グループに付与し、グループへの書き込み権限を割り当てます。

ファイルの許可を設定するには

1. ユーザー (この場合は `ec2-user`) を `apache` グループに追加します。

```
[ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

2. ログアウトし、再度ログインして新しいグループを選択し、メンバーシップを確認します。

- a. ログアウトします (`exit` コマンドを使用するか、ターミナルウィンドウを閉じます)。

```
[ec2-user ~]$ exit
```

- b. `apache` グループのメンバーシップを検証するには、インスタンスに再接続して次のコマンドを実行します。

```
[ec2-user ~]$ groups
ec2-user wheel apache
```

3. /var/www とそのコンテンツのグループ所有権を `apache` グループに変更します。

```
[ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```

4. グループの書き込み許可を追加して、これからのサブディレクトにグループ ID を設定するには、/var/www とサブディレクトのディレクトリ許可を変更します。

```
[ec2-user ~]$ sudo chmod 2775 /var/www
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

5. グループ書き込み許可を追加するには、/var/www とサブディレクトリのファイル許可を再帰的に変更します。

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

ここで、`ec2-user` (および `apache` グループの将来のメンバー) は、Apache ドキュメントルートでファイルを追加、削除、編集できるようになります。したがって、静的ウェブサイトや PHP アプリケーションなどのコンテンツを追加できます。

#### (オプション) ウェブサーバーの保護

HTTP プロトコルを実行するウェブサーバーは、送受信したデータのトランスポートセキュリティを提供しません。ウェブブラウザを使用して HTTP サーバーに接続すると、閲覧した URL、受信したウェブページのコンテンツ、送信した HTML フォームの内容 (パスワードなど) はすべて、ネットワーク経路上のどれでも傍受できるようになります。ウェブサーバーを保護するためのベストプラクティスとして、SSL/TLS 暗号化でデータを保護する HTTPS (HTTP Secure) のサポートをインストールしてください。

サーバーで HTTPS を有効にする方法については、「チュートリアル: Amazon Linux に SSL/TLS を設定する (p. 77)」を参照してください。

## ステップ 2: LAMP サーバーをテストする

サーバーがインストールおよび実行されており、ファイルのアクセス許可が正しく設定されている場合、`ec2-user` アカウントは、インターネットから使用できる /var/www/html ディレクトリに PHP ファイルを作成できます。

## LAMP ウェブサーバーをテストするには

- Apache ドキュメントルートで PHP ファイルを作成します。

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

このコマンドを実行しようとしたときに「許可が拒否されました」というエラーが表示された場合は、ログアウトし、再度ログインして、[ステップ 1: LAMP サーバーを準備する \(p. 42\)](#) で設定した正しいグループ許可を取得します。

- ウェブブラウザで、作成したファイルの URL を入力します。この URL は、インスタンスのパブリック DNS アドレスにスラッシュとファイル名を追加したものです。例:

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

PHP 情報ページが表示されるはずです。

### PHP Version 7.2.0

<b>System</b>	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64
<b>Build Date</b>	Dec 13 2017 03:34:37
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc
<b>Loaded Configuration File</b>	/etc/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php.d
<b>Additional .ini files parsed</b>	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-dba.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mysqlind.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-session.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-mysqlind.ini, /etc/php.d/pdo_sqlite.ini
<b>PHP API</b>	20170718
<b>PHP Extension</b>	20170718
<b>Zend Extension</b>	320170718
<b>Zend Extension Build</b>	API320170718,NTS
<b>PHP Extension Build</b>	API20170718,NTS

このページが表示されない場合は、前のステップで `/var/www/html/phpinfo.php` ファイルが正しく作成されたことを確認します。次のコマンドで、必要なパッケージがすべてインストールされたことを確認することもできます。2 番目の列のパッケージのバージョンが、この出力例に一致する必要はありません。

```
[ec2-user ~]$ sudo yum list installed httpd24 php72 mysql57-server php72-mysqlind
Loaded plugins: priorities, update-motd, upgrade-helper
Installed Packages
httpd24.x86_64                               2.4.25-1.68.amzn1                                @amzn-
updates
mysql56-server.x86_64                           5.6.35-1.23.amzn1                                @amzn-
updates
php70.x86_64                                   7.0.14-1.20.amzn1                                @amzn-
updates
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
ステップ 3: データベースサー  
バーをセキュリティで保護する

php70-mysqlnd.x86\_64  
updates

7.0.14-1.20.amzn1

@amzn-

必要なパッケージのいずれかが出力に表示されていない場合は、`sudo yum install package` コマンドを使ってインストールします。

3. `phpinfo.php` ファイルを削除します。これは有用な情報であることもあります、セキュリティ上の理由から、インターネット上で公表しないでください。

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

## ステップ 3: データベースサーバーをセキュリティで保護する

MySQL サーバーのデフォルトのインストールには、テストおよび開発に役立ついくつかの機能がありますが、実稼働サーバーでは無効にするか削除する必要があります。`mysql_secure_installation` コマンドを使用すると、ルートパスワードを設定し、安全でない機能をインストールから削除する手順が案内されます。MySQL サーバーを使用する予定がない場合でも、この手順を実行することが推奨されます。

データベースサーバーをセキュリティで保護するには

1. MySQL サーバーを起動します。

```
[ec2-user ~]$ sudo service mysqld start
Initializing MySQL database:
...
PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
...
Starting mysqld: [ OK ]
```

2. `mysql_secure_installation` を実行します。

```
[ec2-user ~]$ sudo mysql_secure_installation
```

- a. プロンプトが表示されたら、ルートアカウントのパスワードを入力します。

- i. 現在のルートパスワードを入力します。デフォルトでは、ルートアカウントにはパスワードが設定されていません。Enter キーを押します。
- ii. 「Y」と入力してパスワードを設定し、安全なパスワードを 2 回入力します。安全なパスワード作成の詳細については、「<https://identitysafe.norton.com/password-generator/>」を参照してください。このパスワードは必ず安全な場所に保管します。

Note

MySQL のルートパスワードの設定は、データベースを保護するための最も基本的な手段にすぎません。データベース駆動型アプリケーションを構築またはインストールする必要がある場合、通常はそのアプリケーションのデータベースサービスユーザーを作成します。ルートアカウントは、データベース管理以外には使用しないでください。

- b. 「Y」と入力して匿名ユーザーアカウントを削除します。
- c. 「Y」と入力してリモートルートログインを無効にします。
- d. 「Y」と入力してテストデータベースを削除します。
- e. 「Y」と入力して権限テーブルを再ロードし、変更を保存します。

3. (オプション) MySQL サーバーをすぐに使用する予定がない場合は、これを停止します。再び必要になったときには再起動できます。

```
[ec2-user ~]$ sudo service mysqld stop
Stopping mysqld: [OK]
```

4. (オプション) ブート時に毎回 MySQL サーバーを起動させる場合は、次のコマンドを入力します。

```
[ec2-user ~]$ sudo chkconfig mysqld on
```

これで、完全に機能する LAMP ウェブサーバーを設定しました。`/var/www/html` の Apache ドキュメントルートにコンテンツを追加する場合、そのコンテンツはインスタンスのパブリック DNS アドレスで表示できます。

## ステップ 4: (オプション) phpMyAdmin をインストールする

phpMyAdmin をインストールするには

phpMyAdmin は、EC2 インスタンスで MySQL データベースを表示して編集するために使用できる、ウェブベースのデータベース管理ツールです。Amazon Linux インスタンスで phpMyAdmin をインストールして設定には、以下の手順に従ってください。

### Important

Apache で SSL/TLS を有効にしていない場合、LAMP サーバーへのアクセスに phpMyAdmin を使用することは推奨されません。そのようにすると、データベース管理者のパスワードや他のデータは、インターネット上を安全ではない状態で送信されます。開発者によるセキュリティ関連の推奨事項については、「[Securing your phpMyAdmin installation](#)」を参照してください。

### Note

Amazon Linux パッケージの管理システムは、現在のところ PHP 7 環境における phpMyAdmin の自動インストールをサポートしていません。このチュートリアルでは、phpMyAdmin を手動でインストールする方法を説明します。

1. SSH を使用して EC2 インスタンスにログインします。
2. 必要な依存ファイルをインストールします。

```
[ec2-user ~]$ sudo yum install php72-mbstring.x86_64 -y
```

3. Apache を再起動します。

```
[ec2-user ~]$ sudo service httpd restart
Stopping httpd: [OK]
Starting httpd: [OK]
```

4. `/var/www/html` で Apache ドキュメントルートに移動します。

```
[ec2-user ~]$ cd /var/www/html
[ec2-user html]$
```

5. <https://www.phpmyadmin.net/downloads> で最新の phpMyAdmin リリース用のソースパッケージを選択します。ファイルディレクトリをインスタンスにダウンロードするには、次の例のようにリンクをコピーして wget コマンドに貼り付けます。

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

6. phpMyAdmin フォルダを作成し、次のコマンドを使用してパッケージを展開します。

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components 1
```

7. *phpMyAdmin-latest-all-languages.tar.gz* Tarball を削除します。

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

8. (オプション) MySQL サーバーが実行中ではない場合は、今すぐ起動します。

```
[ec2-user ~]$ sudo service mysqld start
Starting mysqld: [ OK ]
```

9. ウェブブラウザで、phpMyAdmin のインストール URL を入力します。この URL は、インスタンスのパブリック DNS アドレス (または、パブリック IP アドレス) にスラッシュとインストールディレクトリを追加してものです。例:

```
http://my.public.dns.amazonaws.com/phpMyAdmin
```

phpMyAdmin ログインページが表示されます。



10. 前に作成した `root` ユーザー名と MySQL のルートパスワードを使って、phpMyAdmin インストールにログインします。

インストールは、サービス開始前に設定する必要があります。phpMyAdmin を設定するには、[手動で設定ファイルを作成する](#)、[設定コンソールを使用する](#)、またはその両方の方法を組み合わせることができます。

phpMyAdmin の使用に関する情報は、「[phpMyAdmin ユーザーガイド](#)」を参照してください。

## トラブルシューティング

このセクションでは、新しい LAMP サーバーの設定時に発生する可能性がある一般的な問題の解決案を提供します。

### ウェブブラウザを使用してサーバーに接続できません。

以下のチェックを行って、Apache ウェブサーバーが実行されていて、アクセス可能であるかどうかを確認します。

- ウェブサーバーが実行されていますか？

httpd が有効であることは、次のコマンドを実行して確認できます。

```
[ec2-user ~]$ chkconfig --list httpd
httpd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

ここで、httpd は実行レベル 2、3、4、および 5 で on です (これらの値である必要があります)。

httpd プロセスが実行されていない場合は、[ステップ 1: LAMP サーバーを準備する \(p. 42\)](#) に記載されているステップを繰り返します。

- ファイアウォールは正しく設定されていますか？

Apache テストページが表示されない場合、使用しているセキュリティグループに、HTTP (ポート 80) トライフィックを許可するルールが含まれていることを確認します。HTTP ルールをセキュリティグループに追加する方法については、[セキュリティグループへのルールの追加 \(p. 916\)](#) を参照してください。

### サーバーで実行するアプリケーションソフトウェアに、インストールされている PHP バージョンまたは他のソフトウェアとの互換性がありません。

このチュートリアルでは、最新バージョンの Apache HTTP Server、PHP、MySQL をインストールすることをお勧めします。追加の LAMP アプリケーションをインストールする前に、そのシステム条件を調べて、インストール済みの環境と互換性があることを確認してください。最新バージョンの PHP がサポートされていない場合は、以前サポートされていた設定にダウングレードできます (全体的に安全です)。PHP は、複数のバージョンを並行してインストールすることもできます。これにより、最小限の労力で特定の互換性の問題を解決できます。複数バージョンの PHP の優先順位を設定する方法については、「[Amazon Linux AMI 2016.09 Release Notes](#)」を参照してください。

#### ダウングレード方法

このチュートリアルの十分にテストされた前バージョンでは、次のコア LAMP パッケージを推奨しています。

- httpd24
- php56
- mysql55-server
- php56-mysqld

このチュートリアルの始まりの推奨に従って最新のパッケージをインストールした場合は、まずこのパッケージとその他の依存ファイルを次のようにアンインストールする必要があります。

```
[ec2-user ~]$ sudo yum remove -y httpd24 php72 mysql57-server php72-mysqlnd perl-DBD-MySQL57
```

次に、代替環境をインストールします。

```
[ec2-user ~]$ sudo yum install -y httpd24 php56 mysql55-server php56-mysqlnd
```

後で推奨環境にアップグレードすることを決定した場合は、まずカスタマイズされたパッケージと依存関係を削除する必要があります。

```
[ec2-user ~]$ sudo yum remove -y httpd24 php56 mysql55-server php56-mysqlnd perl-DBD-MySQL56
```

これで、前に説明した最新のパッケージをインストールできます。

## 関連トピック

インスタンスへのファイルの転送、またはウェブサーバーへの WordPress ブログのインストールの詳細については、次のドキュメントを参照してください。

- [WinSCP を使用した Linux インスタンスへのファイルの転送 \(p. 523\)](#)
- [SCP を使用した Linux から Linux インスタンスへのファイルの転送 \(p. 509\)](#)
- [チュートリアル: Amazon Linux を使った WordPress ブログのホスティング \(p. 53\)](#)

このチュートリアルで使用されているコマンドおよびソフトウェアの詳細については、次のウェブページを参照してください。

- Apache ウェブサーバー: <http://httpd.apache.org/>
- MySQL データベースサーバー: <http://www.mysql.com/>
- PHP プログラミング言語: <http://php.net/>
- chmod コマンド: <https://en.wikipedia.org/wiki/Chmod>
- chown コマンド: <https://en.wikipedia.org/wiki/Chown>

ウェブサーバーのドメイン名の登録、または、既存のドメイン名をこのホストに移す方法についての詳細は、『Amazon Route 53 開発者ガイド』の「[Amazon Route 53 のドメインとサブドメインの作成と移行](#)」を参照してください。

## チュートリアル: Amazon Linux を使った WordPress ブログのホスティング

次の手順では、お客様の Amazon Linux インスタンスで、WordPress ブログのインストール、構成を実行し、安全性を確立します。このチュートリアルは、WordPress ブログをホストするウェブサーバーを完全に制御する（これは従来のホスティングサービスでは一般的なことではありません）という点で、Amazon EC2 を使用するための優れた手引きになります。

サーバーに対するソフトウェアパッケージの更新とセキュリティパッチの維持は、お客様の責任となります。ウェブサーバー構成との直接的な対話操作を必要としない、より自動化された WordPress をインストールする場合、AWS CloudFormation サービスは、迅速に始められる WordPress テンプレートを提供します。詳細については、『AWS CloudFormation ユーザーガイド』の「[ご利用開始にあたって](#)」を参照し

てください。Windows インスタンスで WordPress ブログをホスティングする場合は、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Deploying a WordPress Blog on Your Amazon EC2 Windows Instance](#)」を参照してください。データベースが分離された高可用性のソリューションが必要な場合は、『AWS Elastic Beanstalk 開発者ガイド』の「[高可用性の WordPress ウェブサイトをデプロイする](#)」を参照してください。

#### Important

これらの手順は Amazon Linux で使用するためのものです。その他のディストリビューションの詳細については、各ドキュメントを参照してください。このチュートリアルの多くの手順は、Ubuntu インスタンスには使用できません。Ubuntu インスタンスでの WordPress のインストールについては、Ubuntu のドキュメントの「[WordPress](#)」を参照してください。

オプション: オートメーションを使用してこのチュートリアルを完了する

以下のタスクを行う代わりに AWS Systems Manager Automation を使用してこのチュートリアルを完了するには、Automation ドキュメントである [AWS Docs Hosting A WordPress Blog - AL](#) (Amazon Linux) か [AWS Docs Hosting A WordPress Blog - AL2](#) (Amazon Linux 2) のいずれかを実行します。

## 前提条件

このチュートリアルでは、Amazon Linux AMI の「[チュートリアル: Amazon Linux AMI を使用して LAMP ウェブサーバーをインストールする \(p. 42\)](#)」または Amazon Linux 2 の「[チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする \(p. 32\)](#)」のすべての手順に従って、PHP とデータベース (MySQL または MariaDB) をサポートする機能的なウェブサーバーを含む Amazon Linux インスタンスを起動したことを前提としています。このチュートリアルでは、セキュリティグループで HTTP および HTTPS トラフィックを許可するように設定する手順や、ウェブサーバー用にファイルアクセス許可が正しく設定されていることを確認する手順も示します。セキュリティグループへのルール追加の詳細については、[セキュリティグループへのルールの追加 \(p. 916\)](#) を参照してください。

Elastic IP アドレス (EIP) は、WordPress ブログのホストに使用しているインスタンスに関連付けることを強くお勧めします。これにより、インスタンスのパブリック DNS アドレスが変更されて、インストールが破損することを防止できます。ドメイン名を所有していてそのドメインをブログに使用する場合、EIP アドレスをポイントするようにドメイン名の DNS レコードを更新できます (これを行うには、ドメイン名レジストラに問い合わせてください)。実行中のインスタンスに関連付けられた EIP アドレスを無料で 1 つ取得できます。詳細については、「[Elastic IP アドレス \(p. 705\)](#)」を参照してください。

ブログのドメイン名がまだない場合は、Route 53 にドメイン名を登録し、インスタンスの EIP アドレスをドメイン名に関連付けることができます。詳細については、『Amazon Route 53 開発者ガイド』の「[Amazon Route 53 を使用したドメイン名の登録](#)」を参照してください。

## WordPress のインストール

インスタンスに接続して、WordPress インストールパッケージをダウンロードします。

WordPress インストールパッケージをダウンロードして解凍するには

1. wget コマンドを使って、最新の WordPress インストールパッケージをダウンロードします。次のコマンドを実行すると、最新リリースが必ずダウンロードされます。

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
```

2. インストールパッケージを解凍します。インストールフォルダは、wordpress という名前のフォルダに解凍されます。

```
[ec2-user ~]$ tar -xzf latest.tar.gz
```

## WordPress インストール用にデータベースユーザーとデータベースを作成するには

WordPress インストールは、ブログの投稿、ユーザーコメントなどの情報をデータベースに格納する必要があります。この手順を実行すると、ブログのデータベースを作成するのに役立ち、このデータベースに対して情報の読み取りや保存を許可されたユーザーにも有用です。

1. データベースサーバーを起動します。

- Amazon Linux 2

```
[ec2-user ~]$ sudo systemctl start mariadb
```

- Amazon Linux AMI

```
[ec2-user ~]$ sudo service mysqld start
```

2. データベースサーバーに root ユーザーとしてログインします。メッセージが表示されたら、データベース root パスワードを入力します。これは通常の root システムパスワードと異なることもあります。データベースサーバーのセキュリティ確保を実行していない場合は、空のときもあります。

データベースサーバーのセキュリティを確保していない場合、セキュリティ確保を行うことは重要です。詳細については、「[MariaDB サーバーをセキュリティで保護するには \(p. 38\)](#)」(Amazon Linux 2) または「[データベースサーバーをセキュリティで保護するには \(p. 48\)](#)」(Amazon Linux AMI) を参照してください。

```
[ec2-user ~]$ mysql -u root -p
```

3. MySQL データベースのユーザーとパスワードを作成します。WordPress インストールは、これらの値を使って、MySQL データベースと通信を行います。一意のユーザー名とパスワードを入力して、次のコマンドを入力します。

```
CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';
```

ユーザー用に強力なパスワードを作成してください。パスワードに一重引用符 (`) を使用しないでください。この文字は前述のコマンドを中断させるためです。安全なパスワードの作成の詳細については、「<http://www.pctools.com/guides/password/>」を参照してください。既存のパスワードを再利用しないでください。また、このパスワードは必ず安全な場所に保管してください。

4. データベースを作成します。wordpress-db など、データベースにはわかりやすい名前を使用します。

### Note

次のコマンドのデータベース名を囲む区切り記号は、「バックティック」と呼ばれています。バックティック (`) キーは通常、標準キーボードの Tab キーの上に配置されています。バックティックは必ずしも必要ではありませんが、データベース名では使用できない文字 (ハイフンなど) の代わりに使用できます。

```
CREATE DATABASE `wordpress-db`;
```

5. データベースに対して、以前作成した WordPress ユーザーに対する完全な権限を付与します。

```
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
```

6. すべての変更を有効にするため、データベース権限をフラッシュします。

```
FLUSH PRIVILEGES;
```

7. mysql クライアントを終了します。

exit

wp-config.php ファイルの作成と編集を行うには

WordPress インストールフォルダには、wp-config-sample.php という名前の構成ファイル例が格納されています。この手順では、このファイルをコピーして、特定の構成に合うように編集します。

1. wp-config-sample.php ファイルを wp-config.php という名前でコピーします。この操作を実行すると、新しい構成ファイルが作成され、元のファイルがバックアップとしてそのまま保持されます。

```
[ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

2. お好みのテキストエディタ (nano、vim など) を使って wp-config.php ファイルを編集し、インストール用の値を入力します。お好みのテキストエディタがない場合、nano が初心者に適しています。

```
[ec2-user ~]$ nano wordpress/wp-config.php
```

- a. DB\_NAME を定義する行を探して、database\_name\_here を Step 4 (p. 55) の WordPress インストール用にデータベースユーザーとデータベースを作成するには (p. 55) で作成したデータベース名に変更します。

```
define('DB_NAME', 'wordpress-db');
```

- b. DB\_USER を定義する行を探して、username\_here を Step 3 (p. 55) の WordPress インストール用にデータベースユーザーとデータベースを作成するには (p. 55) で作成したデータベースユーザーに変更します。

```
define('DB_USER', 'wordpress-user');
```

- c. DB\_PASSWORD を定義する行を探して、password\_here を Step 3 (p. 55) の WordPress インストール用にデータベースユーザーとデータベースを作成するには (p. 55) で作成した強力なパスワードに変更します。

```
define('DB_PASSWORD', 'your_strong_password');
```

- d. Authentication Unique Keys and Salts というセクションを見つけます。これらの KEY と SALT の値は、WordPress ユーザーがローカルマシンに保存したブラウザクッキーに対する暗号化レイヤーを提供します。基本的に、ここで長くてランダムな値を指定すると、サイトのセキュリティが向上します。<https://api.wordpress.org/secret-key/1.1/salt/> にアクセスして、ランダムに生成されるキーセット値を取得し、wp-config.php ファイルにコピーして貼り付けることができます。PuTTY 端末にテキストを貼り付けるには、テキストを貼り付ける場所にカーソルを置き、PuTTY 端末内でマウスを右クリックします。

セキュリティキーの詳細については、「[http://codex.wordpress.org/Editing\\_wp-config.php#Security\\_Keys](http://codex.wordpress.org/Editing_wp-config.php#Security_Keys)」にアクセスしてください。

#### Note

次の値はサンプル専用です。これらの値を実際のインストールには使わないでください。

```
define('AUTH_KEY', '#U$$_+[RXN8:b^-I_0(WU_+ c+WFkI~c]o]-bHw+');
Aj[wTwSiz<Qb[mghEXcRh-');
```

```
define('SECURE_AUTH_KEY', 'Zsz._P=l/|y.Lq)XjlkW$1y5Nj76E6EJ.AV0pCKZZB,*~r?6OP
$eJ7@;+(ndLg');
define('LOGGED_IN_KEY', 'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2jh^,8x>)Y |;(^[Iw]Pi
+LG#A4R?7N`YB3');
define('NONCE_KEY', 'P(g62HeZxEes/LnI^i=H,[XwK9I&[2s|:?ON}VJM%?;v2v]v+;
+^9eXUahg@::Cj');
define('AUTH_SALT', 'C$DpB4Hj[JK:{ql`sRVa:::7yShy(9A@5wg+^JJVb1fk%_-
Bx*M4(qc[Og%JT!h');
define('SECURE_AUTH_SALT', 'd!uRu#)+q#{f$Z?Z9uFPG.${+S{n-1M&%@~gL>U>NV<zpD-@2-
Es7Q1O-bp28EKV');
define('LOGGED_IN_SALT', 'j{00P*owZf)kVD+FVLn-->.|Y%Ug4#I^*LVd9QeZ^&XmK/e(76mic
+&W&+^OP');
define('NONCE_SALT', '-97r*V/cgxLmp?Zy4zUU4r99QQ_xGs2LTd%P; |
_e1ts)8_B/, .6[=UK<J_y9?JWG');
```

- e. ファイルを保存し、テキストエディタを終了します。

#### WordPress ファイルを Apache ドキュメントルートの下にインストールするには

- インストールフォルダの解凍、MySQL データベースとユーザーの作成、WordPress 構成ファイルのカスタマイズが終了したため、インストールファイルをウェブサーバーのドキュメントルートにコピーし、インストールスクリプトを実行して、インストールを終了する準備ができました。これらのファイルの場所は、ウェブサーバーの実際のルートで WordPress ブログを使用できるようにするかどうか ([my.public.dns.amazonaws.com](#) など)、またはルートの下のサブディレクトリやフォルダに格納するか ([my.public.dns.amazonaws.com/blog](#) など) によって異なります。
  - WordPress をドキュメントルートで実行する場合は、WordPress のインストールディレクトリのコンテンツを次のようにコピーします (ただし、ディレクトリ自体はコピーしません)。

```
[ec2-user ~]$ cp -r wordpress/* /var/www/html/
```

- WordPress をドキュメントルートの下の別のディレクトリで実行する場合、まず、そのディレクトリを作成してから、そこにファイルをコピーします。この例では、WordPress はディレクトリ blog から実行されます。

```
[ec2-user ~]$ mkdir /var/www/html/blog
[ec2-user ~]$ cp -r wordpress/* /var/www/html/blog/
```

#### Important

セキュリティ上の理由から、次の手順にすぐに進まない場合は、Apache ウェブサーバー (`httpd`) を直ちに停止してください。インストールを Apache ドキュメントルートの下に移動すると、WordPress インストールスクリプトは保護されなくなり、Apache ウェブサーバーが実行している場合、攻撃者はブログへのアクセス権を取得する可能性があります。Apache ウェブサーバーを停止するには、`sudo service httpd stop` コマンドを入力します。次の手順に移動する場合、Apache ウェブサーバーを停止する必要はありません。

#### WordPress がパーマリンクを使用できるようにするには

WordPress のパーマリンクが正しく機能するには Apache の `.htaccess` ファイルを使用する必要がありますが、Amazon Linux ではデフォルトで有効になっていません。Apache ドキュメントルートですべての上書きできるようにするには、次の手順を使用します。

- お好みのテキストエディタ (`nano` や `vim` など) で、`httpd.conf` ファイルを開きます。お好みのテキストエディタがない場合、`nano` が初心者に適しています。

```
[ec2-user ~]$ sudo vim /etc/httpd/conf/httpd.conf
```

2. <Directory "/var/www/html"> で始まるセクションを見つけます。

```
<Directory "/var/www/html">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksIfOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

3. 上のセクションの AllowOverride None 行を AllowOverride All に変更します。

Note

このファイルには複数の AllowOverride 行があります。必ず <Directory "/var/www/html"> セクションの行を変更してください。

```
AllowOverride All
```

4. ファイルを保存し、テキストエディタを終了します。

PHP グラフィック描画ライブラリをインストールするには

PHP 用の GD ライブラリを使用すると、イメージを変更することができます。ブログのヘッダーイメージをトリミングする必要がある場合は、次のようにこのライブラリをインストールします。

```
[ec2-user ~]$ sudo yum install php72-gd
```

Apache ウェブサーバーのファイル許可を修正するには

WordPress で使用できる機能の中には、Apache ドキュメントルートへの書き込み権限が必要なことがあります (管理画面を使った、メディアのアップロードなど)。まだ設定していない場合は、次のグループのメンバーシップおよびアクセス許可を適用します (プロセスの詳細は「[LAMP ウェブサーバーチュートリアル \(p. 42\)](#)」を参照)。

1. /var/www とそのコンテンツのファイル所有権を apache ユーザーに付与します。

```
[ec2-user ~]$ sudo chown -R apache /var/www
```

2. /var/www とそのコンテンツのグループ所有権を apache グループに付与します。

```
[ec2-user ~]$ sudo chgrp -R apache /var/www
```

3. /var/www およびそのサブディレクトリのディレクトリ許可を変更してグループの書き込み許可を設定し、将来のサブディレクトリにグループ ID を設定します。

```
[ec2-user ~]$ sudo chmod 2775 /var/www
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

4. /var/www およびそのサブディレクトリのファイル許可を繰り返し変更してグループの書き込み許可を追加します。

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

5. Apache ウェブサーバーを再起動して、新しいグループと許可を有効にします。

- Amazon Linux 2

```
[ec2-user ~]$ sudo systemctl restart httpd
```

- Amazon Linux AMI

```
[ec2-user ~]$ sudo service httpd restart
```

#### Amazon Linux 2 で WordPress インストールスクリプトを実行するには

WordPress をインストールする準備ができました。使用するコマンドは、オペレーティングシステムによって異なります。この手順のコマンドは、Amazon Linux 2 で使用するためのものです。Amazon Linux AMI で、この後の手順を使用します。

1. systemctl コマンドを使って、httpd サービスとデータベースサービスがシステムブート時に起動することを確認します。

```
[ec2-user ~]$ sudo systemctl enable httpd && sudo systemctl enable mariadb
```

2. データベースサーバーが実行中であることを確認します。

```
[ec2-user ~]$ sudo systemctl status mariadb
```

データベースサービスが実行されていない場合は、起動します。

```
[ec2-user ~]$ sudo systemctl start mariadb
```

3. Apache ウェブサーバー (httpd) が実行中であることを確認します。

```
[ec2-user ~]$ sudo systemctl status httpd
```

httpd サービスが実行されていない場合は、起動します。

```
[ec2-user ~]$ sudo systemctl start httpd
```

4. ウェブブラウザで WordPress ブログの URL を入力します (インスタンスのパブリック DNS アドレス、または blog フォルダに続くアドレス)。WordPress インストールスクリプトが表示されます。WordPress のインストールに必要な情報を入力します。[WordPress のインストール] を選択し

て、インストールを完了します。詳細については、WordPress のウェブサイトの「[インストールスクриプトの実行](#)」を参照してください。

Amazon Linux AMI で WordPress インストールスクリプトを実行するには

1. chkconfig コマンドを使って、httpd サービスとデータベースサービスがシステムブート時に起動することを確認します。

```
[ec2-user ~]$ sudo chkconfig httpd on && sudo chkconfig mysqld on
```

2. データベースサーバーが実行中であることを確認します。

```
[ec2-user ~]$ sudo service mysqld status
```

データベースサービスが実行されていない場合は、起動します。

```
[ec2-user ~]$ sudo service mysqld start
```

3. Apache ウェブサーバー (httpd) が実行中であることを確認します。

```
[ec2-user ~]$ sudo service httpd status
```

httpd サービスが実行されていない場合は、起動します。

```
[ec2-user ~]$ sudo service httpd start
```

4. ウェブブラウザで WordPress ブログの URL を入力します (インスタンスのパブリック DNS アドレス、または blog フォルダに続くアドレス)。WordPress インストールスクリプトが表示されます。WordPress のインストールに必要な情報を入力します。[WordPress のインストール] を選択して、インストールを完了します。詳細については、WordPress のウェブサイトの「[インストールスクриプトの実行](#)」を参照してください。

## 次のステップ

WordPress ブログをテストしたら、設定の更新を検討します。

カスタムドメイン名を使用する

EC2 インスタンスの EIP アドレスに関連付けられたドメイン名がある場合、EC2 パブリック DNS アドレスの代わりにその名前を使用するようにブログを設定できます。詳細については、「[http://codex.wordpress.org/Changing\\_The\\_Site\\_URL](http://codex.wordpress.org/Changing_The_Site_URL)」を参照してください。

ブログの構成

読者にパーソナライズされた体験を提供するため、さまざまな [テーマ](#) や [プラグイン](#) を使用するようにブログを設定できます。ただし、インストールプロセスで問題が発生してブログ全体が失われることがあります。インストール中に問題が発生した場合もブログを復元できるように、テーマやプラグインを貯蔵する前にインスタンスのバックアップ Amazon マシンイメージ (AMI) を作成しておくことを強くお勧めします。詳細については、「[独自の AMI の作成 \(p. 94\)](#)」を参照してください。

容量の拡大

WordPress ブログが人気になり処理能力やストレージを増やす必要がある場合は、次のステップを検討してください。

- 
- インスタンスストレージ領域を拡張する。詳細については、「[Amazon EBS Elastic Volumes \(p. 1003\)](#)」を参照してください。
  - MySQL データベースを [Amazon RDS](#) に移動して、サービスが持つ容易にスケールする機能を活用する。
  - より大きなインスタンスタイプに移行する。詳細については、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。
  - 追加インスタンスを追加する。詳細については、「[チュートリアル: Amazon EC2 のアプリケーションの可用性の向上 \(p. 90\)](#)」を参照してください。

#### WordPress の詳細

WordPress の詳細については、「<http://codex.wordpress.org/>」にある WordPress Codex ヘルプ文書を参照してください。インストールのトラブルシューティングについては、「[http://codex.wordpress.org/Installing\\_WordPress#Common\\_Installation\\_Problems](http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems)」にアクセスしてください。WordPress ブログのセキュリティ向上の詳細については、「[http://codex.wordpress.org/Hardening\\_WordPress](http://codex.wordpress.org/Hardening_WordPress)」にアクセスしてください。WordPress ブログを最新状態に維持する方法についての詳細は、「[http://codex.wordpress.org/Updating\\_WordPress](http://codex.wordpress.org/Updating_WordPress)」にアクセスしてください。

## ヘルプ! パブリック DNS 名が変更されたため、ブログが壊れました

WordPress のインストールは、EC2 インスタンスのパブリック DNS アドレスを使用して自動的に設定されます。インスタンスを停止および再開した場合、パブリック DNS アドレスが変更され (Elastic IP アドレスに関連付けられている場合を除く)、ブログが存在しなくなった (または別の EC2 インスタンスに割り当てられた) アドレスにあるリソースを参照することになるため、ブログは機能しなくなります。問題と考えられるいくつかの解決策の詳細については、「[http://codex.wordpress.org/Changing\\_The\\_Site\\_URL](http://codex.wordpress.org/Changing_The_Site_URL)」で説明されています。

WordPress のインストール時にこの状況が発生した場合、WordPress の wp-cli コマンドラインインターフェイスを使用する以下の手順でブログを復元できる可能性があります。

wp-cli を使用して WordPress のサイト URL を変更するには

- SSH を使って EC2 インスタンスに接続します。
- インスタンスの古いサイト URL と新しいサイト URL を書き留めます。古いサイト URL は、WordPress をインストールした時点での EC2 インスタンスのパブリック DNS 名と考えられます。新しいサイト URL は、EC2 インスタンスの現在のパブリック DNS 名です。古いサイト URL が不明な場合、次のコマンドで curl を使用して調べることができます。

```
[ec2-user ~]$ curl localhost | grep wp-content
```

古いパブリック DNS 名への参照が出力に表示されます。次に例を示します (古いサイト URL は赤色になっています)。

```
<script type='text/javascript' src='http://ec2-52-8-139-223.us-west-1.compute.amazonaws.com/wp-content/themes/twentyfifteen/js/functions.js?ver=20150330'></script>
```

- 次のコマンドを使って wp-cli をダウンロードします。

```
[ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

- 次のコマンドを使って、WordPress インストールの古いサイト URL を検索し、置き換えます。EC2 インスタンスの古いサイト URL と新しいサイト URL、および WordPress のインストールパス（通常は /var/www/html または /var/www/html/blog）を置き換えます。

```
[ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/path/to/wordpress/installation --skip-columns=guid
```

- ウェブブラウザで、WordPress ブログの新しいサイト URL を入力し、サイトが再び正しく動作していることを確認します。正しく動作していない場合は、詳細について [http://codex.wordpress.org/Changing\\_The\\_Site\\_URL](http://codex.wordpress.org/Changing_The_Site_URL) と [http://codex.wordpress.org/Installing\\_WordPress#Common\\_Installation\\_Problems](http://codex.wordpress.org/Installing_WordPress#Common_Installation_Problems) を参照してください。

## チュートリアル: Amazon Linux 2 に SSL/TLS を設定する

Secure Sockets Layer/Transport Layer Security (SSL/TLS) は、ウェブサーバーとウェブクライアントの間に、転送中のデータが傍受されないように保護する、暗号化されたチャネルを確立します。このチュートリアルでは、Amazon Linux 2 と Apache ウェブサーバーを使用して EC2 インスタンスに、SSL/TLS のサポートを手動で追加する方法を説明します。商業グレードのサービスを提供する予定がある場合、ここでは説明しませんが、[AWS Certificate Manager](#) をお勧めします。

歴史的経緯から、ウェブの暗号化は、単純に SSL と呼ばれることが少なくありません。ウェブブラウザでは今でも SSL がサポートされていますが、後継プロトコルである TLS プロトコルの方が攻撃を受けにくくなります。Amazon Linux 2 では、すべてのバージョンの SSL でサーバー側のサポートをデフォルトで無効にしています。[セキュリティ基準の本文](#) TLS 1.0 は安全ではないと考えてください。TLS 1.0 と TLS 1.1 はどちらも正式に IETF で [非推奨](#) になる予定です。このチュートリアルは、TLS 1.2 を有効にすることを前提としたガイダンスです。（新しい TLS 1.3 プロトコルはドラフト形式には含まれますが、現時点では Amazon Linux 2 でサポートされていません。）最新の暗号化基準の詳細については、「[RFC 7568](#)」および「[RFC 8446](#)」を参照してください。

このチュートリアルでは、現代のウェブ暗号化を単に TLS と呼びます。

### Important

これらの手順は Amazon Linux 2 で使用するためのものです。また、新しい Amazon EC2 インスタンスを使用して開始するものと仮定します。他のディストリビューションのインスタンスで LAMP ウェブサーバーをセットアップする場合や、古い既存のインスタンスを再利用する場合は、このチュートリアルの一部の手順を使用できないことがあります。Ubuntu の LAMP ウェブサーバーについては、Ubuntu コミュニティのドキュメントの [ApacheMySQLPHP](#) を参照してください。Red Hat Enterprise Linux については、Customer Portal のウェブサーバーに関するトピックを参照してください。

### コンテンツ

- [前提条件 \(p. 62\)](#)
- [ステップ 1: サーバーでの TLS の有効化 \(p. 63\)](#)
- [ステップ 2: CA 署名証明書の取得 \(p. 65\)](#)
- [ステップ 3: セキュリティ設定のテストと強化 \(p. 70\)](#)
- [トラブルシューティング \(p. 72\)](#)
- [Certificate Automation: Amazon Linux 2 での Let's Encrypt と Certbot の使用 \(p. 73\)](#)

## 前提条件

このチュートリアルを開始する前に、次のステップを完了してください。

- EBS-backed Amazon Linux 2 インスタンスを起動します。詳細については、「[ステップ 1: インスタンスを起動する \(p. 27\)](#)」を参照してください。
- インスタンスが以下の TCP ポートで接続を受け付けるようにセキュリティグループを設定します。
  - SSH (ポート 22)
  - HTTP (ポート 80)
  - HTTPS (ポート 443)

詳細については、「[Linux インスタンス用の受信トラフィックの認可 \(p. 897\)](#)」を参照してください。

- Apache ウェブサーバーをインストールします。手順については、「[チュートリアル: Amazon Linux 2 に LAMP ウェブサーバーをインストールする \(p. 32\)](#)」を参照してください。必要なのは httpd パッケージおよび対応する従属コンポーネントのみです。PHP および MariaDB に関する手順は無視してかまいません。
- ウェブサイトの識別と認証を行うため、TLS の公開鍵基盤 (PKI) ではドメインネームシステム (DNS) を使用します。EC2 インスタンスを使用してパブリックウェブサイトをホストするには、ウェブサーバーのドメイン名を登録するか、既存のドメイン名を Amazon EC2 ホストに移す必要があります。これについては、ドメイン登録および DNS ホスティングに関するサードパーティのサービスが多数存在します。[Amazon Route 53](#) を使用することもできます。

## ステップ 1: サーバーでの TLS の有効化

この手順では、自己署名のデジタル証明書を使用して、Amazon Linux 2 で TLS をセットアップします。

### Note

自己署名証明書はテスト用であり、本稼働環境では使用できません。インターネットに自己署名証明書を公開すると、サイトへの訪問者にセキュリティ警告が表示されます。

サーバーで TLS を有効にするには

1. インスタンスに接続 (p. 28) し、Apache が実行されていることを確認します。

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

返される値が「enabled」でない場合、Apache を起動し、システムブート時に毎回起動されるように設定します。

```
[ec2-user ~]$ sudo systemctl start httpd && sudo systemctl enable httpd
```

2. すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。この処理には数分かかりますが、最新の更新とバグ修正を確実に適用することが重要です。

### Note

-y オプションを指定すると、確認メッセージを表示せずに更新をインストールします。インストール前に更新を検査する場合は、このオプションを省略できます。

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

3. これでインスタンスが最新状態になったため、Apache モジュール mod\_ssl をインストールして TLS サポートを追加します。

```
[ec2-user ~]$ sudo yum install -y mod_ssl
```

次のファイルがインスタンスに作成されました。このファイルは、セキュアサーバーの設定とテスト用の証明書の作成に使用します。

• `/etc/httpd/conf.d/ssl.conf`

`mod_ssl` の設定ファイル。このファイルには、暗号化キーと証明書の場所、許可する TLS プロトコル、受け入れる暗号化アルゴリズムを Apache に指示するディレクティブが含まれています。

• `/etc/pki/tls/certs/make-dummy-cert`

サーバーホスト用の自己署名 X.509 証明書とプライベートキーを生成するためのスクリプト。この証明書は、TLS を使用するように Apache が正しくセットアップされているかどうかをテストする場合に役立ちます。アイデンティティは証明されないため、本稼働環境では使用しないでください。本稼働環境で使用すると、ウェブブラウザで警告が表示されます。

4. テスト用に自己署名のダミー証明書とキーを生成するためのスクリプトを実行します。

```
[ec2-user ~]$ cd /etc/pki/tls/certs  
sudo ./make-dummy-cert localhost.crt
```

`/etc/pki/tls/certs/` ディレクトリに新しいファイル `localhost.crt` が生成されます。指定されたファイル名は、`/etc/httpd/conf.d/ssl.conf` の `SSLCertificateFile` ディレクティブで割り当てたデフォルトの名前と一致します。

このファイルには、自己署名証明書と証明書のプライベートキーのいずれも含まれます。Apache では、証明書とキーを PEM 形式にする必要があります。これは、次の短縮化された例のように、"BEGIN" 行と "END" 行で囲まれた Base64 エンコードの ASCII 文字で構成されます。

```
-----BEGIN PRIVATE KEY-----  
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQD2KKx/8Zk94m1q  
3gQMZF9ZN66Ls19+3tHAgQ5Fpo9KJDhzLjOOC18u1PTcGmAah5kEitCEc0wzmNeo  
BCl0wYR6GOrGaKtK9Dn7CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vr  
GvwnKoMh3DlK44D9dX7IDua2PlYx5+eroA+1Lqf32ZSaAOobBIMIYTHigwbHMZoT  
...  
56tE7THvH7vOEf4/iUOsIrEzaMaJ0mqkmY1A70qQGQKBgBF3H1qNRNHuyMcPODFs  
27hDzPDinrquSEvoZIggkDMlh2irTiipJ/GhkvtPq0lv0fK/VXw8vSgeaBuhwJvS  
LXU9HvYqoU604FgD3nAyB9hI0BE13r1HjUvbjT7moH+RhnNz6eqqdscCS09VtRAo  
4QQvAqOa8UheYeoxLdWcHaLP  
-----END PRIVATE KEY-----  
  
-----BEGIN CERTIFICATE-----  
MIIEazCCA10gAwIBAgICWxQwdQYJKoZIhvcNAQELBQAwgbExCzABgNVBAYTAi0t  
MRIwEAYDVQQIDA1Tb21lU3RhGUxETAPBgNVBAcMCNvbwVWDaXR5MRkwFwYDVQQK  
DBBTb21lT3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb21lT3JnYW5pemF0aW9uYWxv  
bmlOMRkwFwYDVQQDDBBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhVv  
...  
z5rRUE/XzxRLBZooWZpNWTXJkQ3uFYH6s/sBwtHpKKZMzOvDedREjNKAvk4ws6F0  
CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vrGvwnKoMh3DlK44D9d1U3  
WanXWehT6FiSzvB4sTEXXJN2jdw8g+sHGNz8zC0sclknYhHrCVD2vnBlZJKSzvak  
3ZazhBxtQSukFMOnWPP2a0DMMFGYUHodOBQE8sBJxg==  
-----END CERTIFICATE-----
```

ファイル名および拡張子は利便性のためであり、機能には影響しません。たとえば、`cert.crt` または `cert.pem` などのファイル名で証明書を呼び出すことができます。ただし、`ssl.conf` ファイルの関連ディレクティブが同じ名前を使用している場合に限ります。

#### Note

デフォルトの TLS ファイルを独自にカスタマイズしたファイルに置き換える場合は、PEM 形式であることを確認してください。

5. /etc/httpd/conf.d/ssl.conf ファイルを開き、次の行をコメントアウトします。ダミーの自己署名証明書にも同じキーが含まれているためです。次のステップに進む前にこの行をコメントアウトしないと、Apache サービスは起動に失敗します。

```
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

6. Apache を再起動します。

```
[ec2-user ~]$ sudo systemctl restart httpd
```

#### Note

前述のとおり、TCP 443 番ポートが EC2 インスタンスでアクセス可能であることを確認してください。

7. Apache ウェブサーバーではポート 443 経由で HTTPS (セキュア HTTP) がサポートされるようになっています。これをテストするには、ブラウザの URL バーに、<https://> というプレフィックスを指定して、EC2 インスタンスの IP アドレスまたは完全修飾ドメイン名を入力します。

信頼されていない自己署名ホスト証明書を使用してサイトに接続しようとしているため、ブラウザには一連のセキュリティ警告が表示されることがあります。この警告を無視し、サイトに進みます。

サーバーで TLS を正しく設定できていれば、Apache のデフォルトのテストページが開きます。これで、ブラウザとサーバーの間でやり取りされるすべてのデータが暗号化されるようになります。

#### Note

サイト訪問者に対して警告画面が表示されないようにするには、暗号化だけではなく、サイト所有者のパブリック認証を行うための信頼された CA 署名証明書を取得する必要があります。

## ステップ 2: CA 署名証明書の取得

CA 署名証明書を取得するには、次の手順に従います。

- プライベートキーから証明書署名リクエスト (CSR) を作成します。
- 作成した CSR を認証機関 (CA) に送信します。
- 署名付きホスト証明書を入手する
- 証明書を使用するように Apache を設定します

自己署名 TLS X.509 ホスト証明書は、暗号化技術上は CA 署名証明書と同じです。これらの相違は数学的なものではなく、社会的なものです。CA では、最低でもドメイン所有権を検証してから申請者に証明書を発行することを保証しています。そのため、各ウェブブラウザには、ブラウザベンダーが信頼する CA のリストが含まれています。X.509 証明書は主に、プライベートサーバーキーに対応するパブリックキーと、このパブリックキーに暗号で関連付けられている CA による署名で構成されています。HTTPS 経由でブラウザがウェブサーバーに接続すると、サーバーは、信頼された CA のリストをブラウザが確認できるように、証明書を提示します。署名者がリストに含まれている場合や、他の信頼された署名者の信頼チェーンを通じてアクセス可能である場合、ブラウザはサーバーと、高速暗号化データチャネルのネゴシエーションを行い、ページをロードします。

証明書には、リクエストの確認作業が必要であり、一般的に費用がかかるため、各社を比較することをお勧めします。よく知られている CA のリストについては、[dmoztools.net](https://dmoztools.net) のサイトを参照してください。いくつかの CA では、基本レベル証明書が無料で提供されます。これらの CA で最も注目すべきは Let's Encrypt プロジェクトです。このプロジェクトでは、証明書の作成および更新プロセスの自動化もサポートしています。Let's Encrypt を CA として使用する方法の詳細については、「[Certificate Automation: Amazon Linux 2 での Let's Encrypt と Certbot の使用 \(p. 73\)](#)」を参照してください。

ホスト証明書の基盤にはキーがあります。2019 年時点で、[政府および業界グループ](#)は、2030 年まで、ドキュメントを保護するための RSA キーに 2048 ビットの最小キー(モジュロ)サイズを使用することを推奨しています。Amazon Linux 2 で OpenSSL によって生成されるデフォルトのモジュラスサイズは 2048 ビットです。つまり、CA 署名証明書に適しています。次の手順では、モジュラスサイズを大きくする、別の暗号化アルゴリズムを使用するなど、キーのカスタマイズが必要な場合のオプションのステップを提供しています。

CA 署名ホスト証明書を取得するための手順は、登録およびホスト済みの DNS ドメインを所有している場合を除き、使用しません。

### CA 署名証明書を取得するには

1. [インスタンスに接続](#)(p. 28)して、/etc/pki/tls/private/ に移動します。サーバーの TLS 用プライベートキーは、このディレクトリに格納されます。既存のホストキーを使用して CSR を生成する場合は、ステップ 3 に進みます。
2. (オプション) 新しいプライベートキーを生成します。キー設定のいくつかのサンプルを次に示します。生成されたキーのどれもウェブサーバーで機能しますが、実装されるセキュリティの強度とタイプはそれぞれ異なります。
  - 例 1: デフォルトの RSA ホストキーを作成します。結果として生成されるファイル **custom.key** が、2048 ビットの RSA プライベートキーです。

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key
```

- 例 2: これより大きなモジュラサイズを使用して、より強力な RSA キーを作成します。結果として生成されるファイル **custom.key** が、4096 ビットの RSA プライベートキーです。

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

- 例 3: パスワードで保護された 4096 ビット暗号化 RSA キーを作成します。結果のファイル、**custom.key** は、AES-128 暗号で暗号化された 4096 ビットの RSA プライベートキーです。

#### Important

キーを暗号化するとセキュリティを強化できますが、暗号化キーにはパスワードが必要であるため、暗号化に依存するサービスを自動的に開始することはできません。このキーを使用するたびに、SSH 接続でパスワード(前述の例では、"abcde12345")を指定する必要があります。

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out custom.key 4096
```

- 例 4: 非 RSA 暗号を使用してキーを作成します。RSA 暗号化は、2 つの大きな素数の積に基づくパブリックキーのサイズのために、比較的遅くなる可能性があります。ただし、非 RSA 暗号化方式を使用する TLS 用のキーを作成することも可能です。同等レベルのセキュリティを提供する場合は、楕円曲線の計算に基づいたキーのほうが小さく計算処理も高速です。

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

結果は、prime256v1(OpenSSL でサポートされる "名前付き曲線")を使用した 256 ビットの楕円曲線プライベートキーです。暗号化強度は([NIST](#) によると) 2048 ビットの RSA キーよりも優れています。

#### Note

すべての CA で、楕円曲線ベースのキーに対して RSA キーと同じレベルのサポートが提供されているわけではありません。

新しいプライベートキーには、制限の厳しい所有権とアクセス権を設定します(所有者 = root、グループ = root、所有者のみの読み取り/書き込み)。コマンドは次の例のようになります。

```
[ec2-user ~]$ sudo chown root:root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

上記のコマンドにより、次のような結果が得られます。

```
-rw----- root root custom.key
```

適切なキーを作成し、設定できたら、CSR を作成できます。

- 好みのキーを使用して CSR を作成します。次の例では **custom.key** を使用しています。

```
[ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
```

OpenSSL によりダイアログが開かれ、次の表に示されている情報の入力が求められます。基本的なドメイン検証済みホスト証明書については、[Common Name (共通名)] 以外のフィールドはすべてオプションです。

名前	説明	例
国名	2 文字の ISO 略称 (国名コード)。	US (= 米国)
州名	あなたが所属する組織の所在地の州または県。省略不可です。	ワシントン
市区町村	市など、組織の場所。	シアトル
組織名	組織の正式名称。組織名は、省略不可です。	Example Corp
部門名	組織に関する追加情報 (存在する場合)。	Example Dept
共通名	この値は、ユーザーがブラウザに入力する必要のあるウェブアドレスと正確に一致します。通常、これはプレフィックス付きのホスト名またはエイリアスによるドメイン名 ( <code>www.example.com</code> の形式) を意味します。自己署名証明書を使用し、DNS 解決なしでテストを行う場合、共通名の構成要素はホスト名のみになる場合があります。CA では、 <code>*.example.com</code> などのワイルドカード名を許容する、よりコストの高い証明書も用意されています。	<code>www.example.com</code>
E メールアドレス	サーバー管理者の E メールアドレス。	<code>someone@example.com</code>

最後に、OpenSSL により、オプションのチャレンジパスワードが求められます。このパスワードは CSR と、ユーザーと CA の間のトランザクションのみに適用されるため、このフィールドと、もう 1 つのオプションフィールドである、オプションの会社名については、CA の推奨事項に従ってください。CSR のチャレンジパスワードは、サーバー操作には影響しません。

結果として生成されるファイル **csr.pem** には、パブリックキー、パブリックキーのデジタル署名、入力したメタデータが含まれています。

- CA に CSR を送信します。この作業は通常、テキストエディタで CSR ファイルを開く動作と、内容をウェブフォームにコピーする動作で構成されています。このとき、証明書に適用する 1 つ以上のサブジェクト代替名 (SAN) を指定するように求められることがあります。共通名が **www.example.com** の場合、有効な SAN は **example.com** になります (逆も同様です)。サイトへの訪問者がこれら名前のいずれかを入力すると、エラーなしの接続が提示されます。CA のウェブフォームで許可される場合は、SAN のリストに共通名を含めます一部の CA では自動的に含められます。

リクエストが承認されると、CA によって署名された新しいホスト証明書が届きます。CA の信頼チェーンを完成するために必要な、追加の証明書が含まれている中間証明書ファイルをダウンロードするよう指示されることもあります。

Note

多様な用途向けに複数の形式のファイルを送信してくる CA もあります。このチュートリアルでは、PEM 形式の証明書ファイルのみ使用してください。PEM 形式のファイルには通常、**.pem** または **.crt** ファイル拡張子が使用されます (ただし、常にこれらの拡張子が使用されるわけではありません)。どのファイルを使用すべきかわからない場合は、テキストエディタでファイルを開き、以下の行で始まる 1 つ以上のブロックを含むファイルを見つけてください。

```
- - - - -BEGIN CERTIFICATE - - - - -
```

ファイルの末尾は次のような行になっている必要があります。

```
- - - - -END CERTIFICATE - - - - -
```

以下に示すように、コマンドラインでファイルをテストすることもできます。

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```

これらの行がファイルに表示されていることを確認してください。**.p7b**、**.p7c**、または類似のファイル拡張子で終了するファイルは使用しないでください。

- 新しい CA 署名証明書と任意の中間証明書を **/etc/pki/tls/certs** ディレクトリに配置します。

Note

EC2 インスタンスに新しい証明書をアップロードする方法は複数ありますが、最も簡単でわかりやすい方法は、テキストエディタ (**vi**、**nano**、またはメモ帳など) をローカルコンピュータとインスタンスの両方で開いて、両者の間でファイルの内容をコピーして貼り付けることです。EC2 インスタンス内でこれらの操作を実行する際には、root [sudo] アクセス許可が必要です。こうすることで、許可やパスに問題があるかどうかをすぐに確認できます。ただし、内容をコピーする際に行を追加したり、内容を変更したりしないでください。

**/etc/pki/tls/certs** ディレクトリの中から、ファイルの所有者、グループ、アクセス権の設定が制限の厳しい Amazon Linux 2 のデフォルト (所有者 = root、グループ = root、所有者のみの読み込み/書き込み可) と一致することを確認します。以下の例では、使用するコマンドを示しています。

```
[ec2-user certs]$ sudo chown root:root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

これらのコマンドによって、次の結果が得られます。

```
-rw----- root root custom.crt
```

中間証明書ファイルのアクセス権は、比較的厳しくありません（所有者 = root、グループ = root、所有者による書き込み可、グループによる読み取り可、その他による読み取り可）。以下の例では、使用するコマンドを示しています。

```
[ec2-user certs]$ sudo chown root:root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

これらのコマンドによって、次の結果が得られます。

```
-rw-r--r-- root root intermediate.crt
```

6. CSR の作成に使用したプライベートキーを /etc/pki/tls/private/ ディレクトリに配置します。

Note

EC2 インスタンスにカスタムキーをアップロードする方法は複数ありますが、最も簡単でわかりやすい方法は、テキストエディタ (vi、nano、メモ帳など) をローカルコンピュータとインスタンスの両方で開いて、両者の間でファイルの内容をコピーして貼り付けることです。EC2 インスタンス内でこれらの操作を実行する際には、root [sudo] アクセス許可が必要です。こうすることで、許可やパスに問題があるかどうかをすぐに確認できます。ただし、内容をコピーする際に行を追加したり、内容を変更したりしないでください。

/etc/pki/tls/private ディレクトリの中から、次のコマンドを使用してファイルの所有者、グループ、アクセス許可の設定が制限の厳しい Amazon Linux 2 のデフォルト（所有者 = root、グループ = root、所有者のみの読み込み/書き込み可）と一致することを確認します。

```
[ec2-user private]$ sudo chown root:root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

これらのコマンドによって、次の結果が得られます。

```
-rw----- root root custom.key
```

7. 新しい証明書とキーファイルに合わせるには、/etc/httpd/conf.d/ssl.conf を編集します。

- a. CA 署名のホスト証明書のパスとファイル名を Apache の SSLCertificateFile ディレクティブで指定します。

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

- b. 中間証明書ファイル（この例では intermediate.crt）を受け取ったら、Apache の SSLCACertificateFile ディレクティブを使用して、次のファイルのパスとファイル名を指定します。

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

Note

一部の CA では、ホスト証明書と中間証明書を組み合わせて 1 つのファイルを作成するため、この SSLCACertificateFile ディレクティブは必要ありません。CA が提供している手順を参照してください。

- c. プライベートキー（この例では `custom.key`）のパスとファイル名を Apache の `SSLCertificateKeyFile` ディレクトリで指定します。

```
SSLCertificateKeyFile /etc/pki/tls/private/custom.key
```

8. `/etc/httpd/conf.d/ssl.conf` を保存して、Apache を再起動します。

```
[ec2-user ~]$ sudo systemctl restart httpd
```

9. サーバーをテストするには、ブラウザの URL バーにドメイン名を入力し、プレフィックス `https://` を指定します。ブラウザによって、エラーが生成されることなく、HTTPS 経由でテストページがロードされます。

## ステップ 3: セキュリティ設定のテストと強化

TLS が運用可能になりパブリックに公開されたら、実際の安全性をテストする必要があります。セキュリティアップの詳細な分析を無料で行うことのできる [Qualys SSL Labs](#) などのオンラインサービスを使用すると簡単です。その結果に基づき、受け入れるプロトコル、優先する暗号化方式、除外する暗号化方式を制御することによって、デフォルトのセキュリティ設定を強化するかどうかを決定できます。詳細については、「[Qualys のスコアの計算方法](#)」を参照してください。

### Important

サーバーのセキュリティを確保するには、実際のテストが非常に重要です。小さな設定エラーによって、深刻なセキュリティ侵害やデータの損失が生じる可能性があります。調査や新たな脅威に応じて、推奨されるセキュリティ管理方法は常に変化するため、適切なサーバー管理を行うには、定期的なセキュリティ監査が不可欠です。

[Qualys SSL Labs](#) のサイトで、サーバーの完全修飾ドメイン名を `www.example.com` という形式で入力します。約 2 分後に、サイトに関するグレード（A から F）と、結果の詳細な内訳が届きます。以下の表は、Amazon Linux 2 でのデフォルトの Apache 設定およびデフォルトの Certbot 証明書と同じ設定を使用しているドメインのレポートをまとめたものです。

総合評価	B
証明書	100%
プロトコルサポート	95%
キー交換	70%
暗号強度	90%

概要は設定がほとんど正常であることを示していますが、詳細レポートでは、いくつかの潜在的な問題が指摘されています。重大度の高い順に以下に示します。

XRC4 暗号は、特定の古いブラウザでの使用がサポートされています。暗号は、暗号化アルゴリズムの計算の中核です。TLS データストリームの暗号化に使用される高速の暗号化方式である RC4 は、いくつかの重大な脆弱性を持つことで知られています。従来のブラウザをサポートするもともな理由がない限り、この暗号化方式を無効にする必要があります。

X旧バージョンの TLS がサポートされています。設定では TLS 1.0 (すでに廃止されています) と TLS 1.1 (廃止予定) がサポートされています。2018 年以降は、TLS 1.2 のみ推奨されています。

X前方秘匿性は完全にサポートされていません。前方秘匿性は、プライベートキーから派生した一時 (エフェメラル) セッションキーを使用して暗号化を行う、アルゴリズムの機能です。これは、攻撃者がウェ

ブリッジ接続の長期的なプライベートキーを所有していても、HTTPS データを復号できないことを意味します。

TLS 設定を修正し、将来への対応性を確保するには

1. 設定ファイル `/etc/httpd/conf.d/ssl.conf` を開き、行頭に # を付けて以下の行をコメントアウトしてください。

```
#SSLProtocol all -SSLv3
```

2. 次のディレクティブを追加します。

```
#SSLProtocol all -SSLv3
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```

このディレクティブにより、SSL バージョン 2、3、および TLS バージョン 1.0、1.1 が明示的に無効化されます。これで、サーバーでは、TLS 1.2 以外を使用した、クライアントとの暗号化された接続の受け入れが拒否されます。ディレクティブに含める指定が多くなるほど、サーバーの動作に対する設定内容が明確に伝わります。

#### Note

このようにして、TLS バージョン 1.0 および 1.1 を無効にすると、ごく一部の古くなったウェブブラウザによるサイトへのアクセスがブロックされるようになります。

許可された暗号のリストを変更するには

1. 設定ファイル `/etc/httpd/conf.d/ssl.conf` で、`SSLCipherSuite` ディレクティブを含むセクションを探し、行頭に # を付けて既存の行をコメントアウトします。

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

2. 明示的な暗号スイートと、前方秘匿性を優先し、安全でない暗号を禁止する暗号順序を指定します。ここで使用される `SSLCipherSuite` ディレクティブは、[Mozilla SSL Configuration Generator](#) の出力に基づいています。これは、お客様のサーバーで実行されている特定のソフトウェアに合わせて TLS 設定を調整します。(詳細については、Mozilla の有益なリソース「[Security/Server Side TLS](#)」を参照してください。) まず、以下のコマンドの出力を使用して、Apache と OpenSSL のバージョンを確認します。

```
[ec2-user ~]$ yum list installed | grep httpd
[ec2-user ~]$ yum list installed | grep openssl
```

たとえば、返された情報が Apache 2.4.34 および OpenSSL 1.0.2 である場合、これをジェネレーターに入力します。"最新" 互換性モデルを選択すると、`SSLCipherSuite` ディレクティブが作成されます。このディレクティブは、積極的にセキュリティを適用しますが、ほとんどのブラウザで使用できます。ソフトウェアで最新互換性モデルがサポートされていない場合は、ソフトウェアを更新するか、"中間" の構成を選択します。

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-
AES128-SHA256
```

選択された暗号化方式の名前には、ECDHE が含まれています (Elliptic Curve Diffie-Hellman Ephemeral の略語です)。ephemeral は前方秘匿性を示します。また、これらの暗号化方式では、RC4 はサポートされていません。

デフォルトや、内容が見えない簡単なディレクティブに依存するのではなく、暗号化方式の明示的なリストを使用することをお勧めします。

生成されたディレクティブを `/etc/httpd/conf.d/ssl.conf` にコピーします。

Note

ここでは読みやすくするために数行に分けて示していますが、このディレクティブは、`/etc/httpd/conf.d/ssl.conf` にコピーする際に、暗号化方式名の間をコロンのみ (スペースなし) で区切り、1 行に指定する必要があります。

- 最後に、次の行について、行頭の # を削除してコメント解除します。

```
#SSLHonorCipherOrder on
```

このディレクティブは、(この場合) 前方秘匿性をサポートするものも含めて、ランクの高い暗号化方式を優先するようサーバーに強制します。このディレクティブが有効になると、サーバーは、セキュリティの弱い暗号化方式に戻る前に、セキュリティが強力な接続を確立しようとします。

これらの手順がいずれも完了したら、変更内容を `/etc/httpd/conf.d/ssl.conf` に保存し、Apache を再起動します。

[Qualys SSL Labs](#) でドメインをもう一度テストすると、RC4 脆弱性やその他の警告は解決し、次のようなサマリレポートが出力されます。

総合評価	A
証明書	100%
プロトコルサポート	100%
キー交換	90%
暗号強度	90%

Important

OpenSSL の更新ごとに、新しい暗号化方式が導入され古い暗号化方式のサポートが削除されます。EC2 の Amazon Linux 2 インスタンスを最新の状態に維持し、セキュリティに関する [OpenSSL](#) からの告知に注意して、技術分野の報道でセキュリティ面の新しい脆弱性に関するレポートを警戒してください。詳細については、『クラシックロードバランサー用ユーザーガイド』の「[Elastic Load Balancing での事前定義された SSL のセキュリティポリシー](#)」を参照してください。

## トラブルシューティング

- パスワードを指定しないと Apache ウェブサーバーが起動しません。

これは、パスワードで保護された暗号化プライベート サーバー キーをインストールした場合は正常な動作です。

暗号化とパスワードの要件をキーから削除できます。デフォルトディレクトリに `custom.key` という暗号化プライベート RSA キーがあり、そのパスワードが `abcde12345` であるとすると、EC2 インスタンスで次のコマンドを実行し、このキーの非暗号化バージョンを生成してください。

```
[ec2-user ~]$ cd /etc/pki/tls/private/  
[ec2-user private]$ sudo cp custom.key custom.key.bak  
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out  
    custom.key.nocrypt  
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key  
[ec2-user private]$ sudo chown root:root custom.key  
[ec2-user private]$ sudo chmod 600 custom.key  
[ec2-user private]$ sudo systemctl restart httpd
```

パスワードが求められずに Apache が起動するようになります。

- `sudo yum install -y mod_ssl` を実行するとエラーが発生します。

SSL に必要なパッケージをインストールすると、次のようなエラーが表示されることがあります。

```
Error: httpd24-tools conflicts with httpd-tools-2.2.34-1.16.amzn1.x86_64  
Error: httpd24 conflicts with httpd-2.2.34-1.16.amzn1.x86_64
```

これは、通常 EC2 インスタンスが Amazon Linux 2 を実行していないことを意味します。このチュートリアルでは、公式の Amazon Linux 2 AMI から新しく作成されたインスタンスのみをサポートしています。

## Certificate Automation: Amazon Linux 2 での Let's Encrypt と Certbot の使用

[Let's Encrypt](#) 認証機関は、インターネット全体を暗号化するための Electronic Frontier Foundation (EFF) の取り組みの中核部分です。この目標を踏まえ、Let's Encrypt ホスト証明書は、手動による介入を最小限に抑えながら、作成、評価、インストール、保守されるように設計されています。証明書管理の自動化側面はウェブサーバー上で動作しているソフトウェアエージェントによって実行されます。インストールおよび設定されたエージェントは、Let's Encrypt と安全に通信して、Apache およびキー管理システム上で管理タスクを実行します。このチュートリアルでは、[Certbot](#) エージェントを使用します。このエージェントを使用することで、カスタマイズされた暗号化キーを証明書の基盤として提供するか、エージェント自身がそのデフォルト値を使用してキーを作成できます。また、「[Certbot を自動化するには \(p. 76\)](#)」で説明するように、手動による介入なしで定期的に証明書を更新するよう Certbot を設定することもできます。詳細については、Certbot の「[ユーザーガイド](#)」および「[マニュアルページ](#)」を参照してください。

Certbot は Amazon Linux 2 で公式にサポートされていませんが、ダウンロードすることができ、インストールすると正常に機能します。データを保護し、問題を回避するため、次のバックアップを作成してください。

- 開始する前に、Amazon EBS ルートボリュームのスナップショットを作成します。これにより、EC2 インスタンスの元の状態に復元することができます。EBS スナップショットの作成方法の詳細については、「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」を参照してください。
- 以下の手順では、Apache の操作を制御する `httpd.conf` ファイルを編集する必要があります。Certbot はこのファイルと他の設定ファイルに独自の自動変更を加えます。復元する必要が生じたときのために、`/etc/httpd` ディレクトリ全体のバックアップコピーを作成してください。

## インストールの準備

Certbot をインストールする前に、次の手順を完了します。

- 
- Extra Packages for Enterprise Linux (EPEL) 7 パッケージをダウンロードします。これは、Certbot が必要とする依存関係を提供するために必要です。

- ホームディレクトリ (/home/ec2-user) に移動します。次のコマンドを使って EPEL をダウンロードします。

```
[ec2-user ~]$ sudo wget -r --no-parent -A 'epel-release*.rpm' http://dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/
```

- 次のコマンドに示すようにリポジトリパッケージをインストールします。

```
[ec2-user ~]$ sudo rpm -Uvh dl.fedoraproject.org/pub/epel/7/x86_64/Packages/e/epel-release*.rpm
```

- 次のコマンドに示すように EPEL を有効にします。

```
[ec2-user ~]$ sudo yum-config-manager --enable epel*
```

次のコマンドを使って、EPEL が有効であることを確認できます。これにより、次のような情報が返されます。

```
[ec2-user ~]$ sudo yum repolist all

...
epel/x86_64           Extra Packages for Enterprise Linux 7 - x86_64
enabled: 12949+175
epel-debuginfo/x86_64   Extra Packages for Enterprise Linux 7 - x86_64
enabled:      2890
epel-source/x86_64     Extra Packages for Enterprise Linux 7 - x86_64
enabled:          0
epel-testing/x86_64    Extra Packages for Enterprise Linux 7 -
Testing - x86_64        enabled:    778+12
epel-testing-debuginfo/x86_64 Extra Packages for Enterprise Linux 7 -
Testing - x86_64 - Debug enabled:      107
epel-testing-source/x86_64 Extra Packages for Enterprise Linux 7 -
Testing - x86_64 - Source enabled:          0
...
```

- Apache のメイン設定ファイル /etc/httpd/conf/httpd.conf を編集します。「Listen 80」ディレクティブを見つけてその後ろに次の行を追加します。このとき、サンプルドメイン名を、実際の共通名およびサブジェクト代替名 (SAN) に置き換えます。

```
<VirtualHost *:80>
    DocumentRoot "/var/www/html"
    ServerName "example.com"
    ServerAlias "www.example.com"
</VirtualHost>
```

ファイルを保存して、Apache を再起動します。

```
[ec2-user ~]$ sudo systemctl restart httpd
```

## Certbot のインストールと実行

この手順は、Certbot を Fedora と RHEL 7 にインストールするための EFF ドキュメントに基づいています。Certbot のデフォルトの使用方法について説明します。これにより、2048 ビットの RSA キーに基づ

＜証明書が作成されます。カスタマイズされたキーを試してみたい場合は、まず、「[Let's Encrypt での ECDSA 証明書の使用](#)」を参照してください。

1. 次のコマンドを使用して Certbot パッケージと依存関係をインストールします。

```
[ec2-user ~]$ sudo yum install -y certbot python2-certbot-apache
```

2. Certbot を実行します。

```
[ec2-user ~]$ sudo certbot
```

3. "Enter email address (used for urgent renewal and security notices)" というプロンプトが表示されたら、連絡先住所を入力し、Enter キーを押します。
4. プロンプトが表示されたら Let's Encrypt のサービス利用規約に同意します。「A」と入力し、Enter キーを押します。

```
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: A
```

5. EFF のメーリングリストに登録するための承認で、「Y」または「N」と入力して Enter キーを押します。
6. Certbot に、VirtualHost ブロックで入力した共通名およびサブジェクト代替名 (SAN) が表示されます。

```
Which names would you like to activate HTTPS for?
-----
1: example.com
2: www.example.com
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
```

入力を空白にしたまま Enter キーを押します。

7. Certbot が証明書を作成して Apache を設定すると、次の出力が表示されます。その後、HTTP クエリを HTTPS にリダイレクトするどうかの確認が求められます。

```
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for example.com
http-01 challenge for www.example.com
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at /etc/httpd/conf/httpd-le-ssl.conf
Deploying Certificate for example.com to VirtualHost /etc/httpd/conf/httpd-le-ssl.conf
Enabling site /etc/httpd/conf/httpd-le-ssl.conf by adding Include to root configuration
Deploying Certificate for www.example.com to VirtualHost /etc/httpd/conf/httpd-le-ssl.conf

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
-----
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
```

```
-----  
Select the appropriate number [1-2] then [enter] (press 'c' to cancel):
```

訪問者が暗号化されていない HTTP 経由でサーバーに接続するには、「1」と入力します。HTTPS 経由の暗号化接続のみ受け入れる場合は、「2」と入力します。Enter を押して選択内容を送信します。

- Apache の設定が完了し、成功した旨とその他の情報が報告されます。

```
Congratulations! You have successfully enabled https://example.com and  
https://www.example.com
```

```
You should test your configuration at:  
https://www.ssllabs.com/ssltest/analyze.html?d=example.com  
https://www.ssllabs.com/ssltest/analyze.html?d=www.example.com  
-----
```

#### IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:  
/etc/letsencrypt/live/certbot.oneeyedman.net/fullchain.pem  
Your key file has been saved at:  
/etc/letsencrypt/live/certbot.oneeyedman.net/privkey.pem  
Your cert will expire on 2019-08-01. To obtain a new or tweaked  
version of this certificate in the future, simply run certbot again  
with the "certonly" option. To non-interactively renew \*all\* of  
your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot  
configuration directory at /etc/letsencrypt. You should make a  
secure backup of this folder now. This configuration directory will  
also contain certificates and private keys obtained by Certbot so  
making regular backups of this folder is ideal.

- インストールが完了したら、「[ステップ 3: セキュリティ設定のテストと強化 \(p. 70\)](#)」に記載されている手順に従って、サーバーのセキュリティのテストと最適化を行います。

## 証明書の更新の自動化設定

Certbot は、サーバーシステムを構成する不可視のエラー回復性のあるパートとなるように設計されています。デフォルトでは、90 日間の短い有効期限を持つホスト証明書を生成します。システムがコマンドを自動的に呼び出すように設定していない場合は、certbot コマンドを有効期限前に手動で再実行する必要があります。以下の手順は、cron ジョブを設定して Certbot を自動化する方法を示しています。

### Certbot を自動化するには

- テキストエディタで /etc/crontab を開き、次のような行を追加します。

```
39      1,13    *      *      *      root    certbot renew --no-self-upgrade
```

完了したらファイルを保存します。以下に、コマンドの各構成要素について説明します。

```
39 1,13 * * *
```

毎日、01:39 と 13:39 にコマンドが実行されるようにスケジュールします。ここで選択した値は任意ですが、Certbot 開発者は、コマンドを少なくとも毎日 2 回実行することを推奨しています。これにより、侵害されていることがわかつた証明書は必ずすぐに取り消されて置き換えられます。

```
root
```

コマンドは、root アクセス許可で実行されます。

```
certbot renew --no-self-upgrade
```

実行されるコマンド。renew サブコマンドを実行すると、Certbot は、以前に取得した証明書があれば確認し、有効期限が近づいているものを更新します。--no-self-upgrade フラグにより、Certbot が手動による介入なしで自動的にアップグレードされることを禁止しています。

2. cron デーモンを再起動します。

```
[ec2-user ~]$ sudo systemctl restart crond
```

## チュートリアル: Amazon Linux に SSL/TLS を設定する

Secure Sockets Layer/Transport Layer Security (SSL/TLS) は、ウェブサーバーとウェブクライアントの間に、転送中のデータが傍受されないように保護する、暗号化されたチャネルを確立します。このチュートリアルでは、Amazon Linux AMI と Apache ウェブサーバーを使用して EC2 インスタンスに、SSL/TLS のサポートを手動で追加する方法を説明します。商業グレードのサービスを提供する予定がある場合、ここでは説明しませんが、[AWS Certificate Manager](#) をお勧めします。

歴史的経緯から、ウェブの暗号化は、単純に SSL と呼ばれることが少なくありません。ウェブブラウザでは今でも SSL がサポートされていますが、後継プロトコルである TLS プロトコルの方が攻撃を受けにくくなります。Amazon Linux AMI は、デフォルトですべてのバージョンの SSL をサーバー側でサポートすることを無効にします。[セキュリティ基準の本文](#) TLS 1.0 は安全ではないと考えてください。TLS 1.0 と TLS 1.1 はどちらも正式に IETF で非推奨になる予定です。このチュートリアルは、TLS 1.2 を有効にすることを前提としたガイドです。(新しい TLS 1.3 プロトコルはドラフト形式には含まれますが、現時点では Amazon Linux 2 でサポートされていません。) 最新の暗号化基準の詳細については、「[RFC 7568](#)」および「[RFC 8446](#)」を参照してください。

このチュートリアルでは、現代のウェブ暗号化を単に TLS と呼びます。

### Important

これらの手順は Amazon Linux AMI で使用するためのものです。他のディストリビューションのインスタンスで LAMP ウェブサーバーをセットアップする場合、このチュートリアルの一部の手順は使用できません。Ubuntu の LAMP ウェブサーバーについては、Ubuntu コミュニティのドキュメントの [ApacheMySQLPHP](#) を参照してください。Red Hat Enterprise Linux については、Customer Portal のドキュメントの [ウェブサーバー](#) を参照してください。

### コンテンツ

- [前提条件 \(p. 77\)](#)
- [ステップ 1: サーバーでの TLS の有効化 \(p. 78\)](#)
- [ステップ 2: CA 署名証明書の取得 \(p. 80\)](#)
- [ステップ 3: セキュリティ設定のテストと強化 \(p. 84\)](#)
- [トラブルシューティング \(p. 86\)](#)
- [Certificate Automation: Amazon Linux での Let's Encrypt と Certbot の使用 \(p. 87\)](#)

## 前提条件

このチュートリアルを開始する前に、次のステップを完了してください。

- Amazon Linux AMI. を使用して EBS-backed インスタンスを起動します。詳細については、「[ステップ 1: インスタンスを起動する \(p. 27\)](#)」を参照してください。

- インスタンスが以下の TCP ポートで接続を受け付けるようにセキュリティグループを設定します。

- SSH (ポート 22)
- HTTP (ポート 80)
- HTTPS (ポート 443)

詳細については、「[Linux インスタンス用の受信トラフィックの認可 \(p. 897\)](#)」を参照してください。

- Apache ウェブサーバーをインストールします。手順については、「[チュートリアル: Amazon Linux への LAMP ウェブサーバーのインストール \(p. 42\)](#)」を参照してください。必要なのは http24 パッケージおよび対応する従属コンポーネントのみです。PHP および MySQL に関する手順は無視してかまいません。
- ウェブサイトの識別と認証を行うため、TLS の公開鍵基盤 (PKI) ではドメインネームシステム (DNS) を使用します。EC2 インスタンスを使用してパブリックウェブサイトをホストするには、ウェブサーバーのドメイン名を登録するか、既存のドメイン名を Amazon EC2 ホストに移す必要があります。これについては、ドメイン登録および DNS ホスティングに関するサードパーティのサービスが多数存在します。[Amazon Route 53](#) を使用することもできます。

## ステップ 1: サーバーでの TLS の有効化

この手順では、自己署名のデジタル証明書を使用して、Amazon Linux で TLS をセットアップします。

### Note

自己署名証明書はテスト用であり、本稼働環境では使用できません。インターネットに自己署名証明書を公開すると、サイトへの訪問者にセキュリティ警告が表示されます。

サーバーで TLS を有効にするには

- インスタンスに接続 (p. 28) し、Apache が実行されていることを確認します。

```
[ec2-user ~]$ sudo service httpd status
```

必要であれば、Apache を起動します。

```
[ec2-user ~]$ sudo service httpd start
```

- すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。この処理には数分かかりますが、最新の更新とバグ修正を確実に適用することが重要です。

### Note

-y オプションを指定すると、確認メッセージを表示せずに更新をインストールします。インストール前に更新を検査する場合は、このオプションを省略できます。

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

- これでインスタンスが最新状態になったため、Apache モジュール mod\_ssl をインストールして TLS サポートを追加します。

```
[ec2-user ~]$ sudo yum install -y mod24_ssl
```

次のファイルがインスタンスに作成されました。このファイルは、セキュアサーバーの設定とテスト用の証明書の作成に使用します。

/etc/httpd/conf.d/ssl.conf

mod\_ssl の設定ファイル。このファイルには、暗号化キーと証明書の場所、許可する TLS プロトコル、受け入れる暗号化アルゴリズムを Apache に指示する「ディレクティブ」が含まれています。

/etc/pki/tls/private/localhost.key

Amazon EC2 ホスト用に自動生成された 2048 ビットの RSA プライベートキー。インストール時には、自己署名ホスト証明書を生成するために OpenSSL によってこのキーが使用されます。また、このキーを使用して、認証局 (CA) に送信する証明書署名リクエスト (CSR) を生成することもできます。

/etc/pki/tls/certs/localhost.crt

サーバーホスト用に自動生成された、自己署名の X.509 証明書。この証明書は、TLS を使用するように Apache が正しくセットアップされているかどうかをテストする場合に役立ちます。

.key ファイルと .crt はどちらも PEM 形式であり、この短縮化された証明書の例のように、「BEGIN」行と「END」行で囲まれ Base64 でエンコードされた ASCII 文字で構成されます。

```
-----BEGIN CERTIFICATE-----
MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwgbExCzAJBgNVBAYTAi0t
MRIwEAYDVQQIDA1Tb21lU3RhGUxETAPBgNVBAcMCFNvbWVDaXR5MRkwFwYDVQQK
DBBtb21lT3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb21lT3JnYW5pemF0aW9uYWxv
bml0MRkwFwYDVQQDDBBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhV
...
z5rRUE/XzxRLBZ0oWZpNWTXJkQ3uFYH6s/sBwtHpkKZMzOvDedREjNKAvk4ws6F0
WanXWehT6FiSzvB4sTEXXJN2jdw8g+sHGnZ8zCOsclknYhHrCVD2vnBlZJKSzvak
3ZazhBxtQSukFMONWPP2a0DMMFGYUHo0BQE8sBJxg==
-----END CERTIFICATE-----
```

ファイル名および拡張子は利便性のためであり、機能には影響しません。証明書は、cert.crt ファイルの関連ディレクティブと同じ名前を使用している限り、cert.pem、ssl.conf、またはその他のファイル名で呼び出すことができます。

#### Note

デフォルトの TLS ファイルを独自にカスタマイズしたファイルに置き換える場合は、PEM 形式であることを確認してください。

- Apache を再起動します。

```
[ec2-user ~]$ sudo service httpd restart
```

- Apache ウェブサーバーではポート 443 経由で HTTPS (セキュア HTTP) がサポートされるようになっています。これをテストするには、ブラウザの URL バーに、**https://** というプレフィックスを指定して、EC2 インスタンスの IP アドレスまたは完全修飾ドメイン名を入力します。信頼されていない自己署名ホスト証明書を使用してサイトに接続しようとしているため、ブラウザには一連のセキュリティ警告が表示されることがあります。

この警告を無視し、サイトに進みます。サーバーで TLS を正しく設定できていれば、Apache のデフォルトのテストページが開きます。これで、ブラウザとサーバーの間でやり取りされるすべてのデータが安全に暗号化されるようになります。

サイト訪問者に対して警告画面が表示されないようにするには、暗号化だけではなく、サイト所有者のパブリック認証を行うための証明書を取得する必要があります。

## ステップ 2: CA 署名証明書の取得

CA 署名証明書を取得するには、次の手順に従います。

- プライベートキーから証明書署名リクエスト (CSR) を作成します。
- 作成した CSR を認証機関 (CA) に送信します。
- 署名付きホスト証明書を入手する
- 証明書を使用するように Apache を設定します

自己署名 TLS X.509 ホスト証明書は、暗号化技術上は CA 署名証明書と同じです。これらの相違は数学的なものではなく、社会的なものです。CA では、最低でもドメイン所有権を検証してから申請者に証明書を発行することを保証しています。そのため、各ウェブブラウザには、ブラウザベンダーが信頼する CA のリストが含まれています。X.509 証明書は主に、プライベートサーバーキーに対応するパブリックキーと、このパブリックキーに暗号で関連付けられている CA による署名で構成されています。HTTPS 経由でブラウザがウェブサーバーに接続すると、サーバーは、信頼された CA のリストをブラウザが確認できるように、証明書を提示します。署名者がリストに含まれている場合や、他の信頼された署名者の信頼チェーンを通じてアクセス可能である場合、ブラウザはサーバーと、高速暗号化データチャネルのネゴシエーションを行い、ページをロードします。

証明書には、リクエストの確認作業が必要であり、一般的に費用がかかるため、各社を比較することをお勧めします。よく知られている CA のリストについては、[dmoztools.net](#) のサイトを参照してください。いくつかの CA では、基本レベル証明書が無料で提供されます。なかでも最も注目すべきは [Let's Encrypt](#) プロジェクトです。このプロジェクトでは、証明書の作成および更新プロセスの自動化もサポートしています。Let's Encrypt を CA として使用する方法の詳細については、「[Certificate Automation: Amazon Linux での Let's Encrypt と Certbot の使用 \(p. 87\)](#)」を参照してください。

ホスト証明書の基盤にはキーがあります。2017 年時点で、[政府および業界グループ](#)は、2030 年まで、ドキュメントを保護するための RSA キーに 2048 ビットの最小キー (モジュロ) サイズを使用することを推奨しています。Amazon Linux で OpenSSL によって生成されるデフォルトのモジュラスサイズは 2048 ビットです。つまり、自動生成された既存のキーは、CA 署名証明書に適しています。モジュラスサイズを大きくする、別の暗号化アルゴリズムを使用するなど、キーのカスタマイズが必要な場合は、次に示す代替手順に従ってください。

CA 署名ホスト証明書を取得するための手順は、登録およびホスト済みの DNS ドメインを所有している場合を除き、使用しません。

CA 署名証明書を取得するには

1. [インスタンスに接続 \(p. 28\)](#)して、/etc/pki/tls/private/ に移動します。これは、サーバーの TLS 用プライベートキーが格納されているディレクトリです。既存のホストキーを使用して CSR を生成する場合は、ステップ 3 に進んでください。
2. (オプション) 新しいプライベートキーを生成します。キー設定のいくつかのサンプルを次に示します。生成されたキーのどれもウェブサーバーで機能しますが、セキュリティの実装方法 (および強度) はそれぞれ異なります。
  - 例 1: デフォルトの RSA ホストキーを作成します。結果として生成されるファイル **custom.key** が、2048 ビットの RSA プライベートキーです。

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key
```

- 例 2: これより大きなモジュラスサイズを使用して、より強力な RSA キーを作成します。結果として生成されるファイル **custom.key** が、4096 ビットの RSA プライベートキーです。

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

- 例 3: パスワードで保護された 4096 ビット暗号化 RSA キーを作成します。結果のファイル、**custom.key** は、AES-128 暗号で暗号化された 4096 ビットの RSA プライベートキーです。

Important

キーを暗号化するとセキュリティを強化できますが、暗号化キーにはパスワードが必要であるため、暗号化に依存するサービスを自動的に開始することはできません。このキーを使用するたびに、SSH 接続でパスワード（前述の例では、「abcde12345」）を指定する必要があります。

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out custom.key  
4096
```

- 例 4: 非 RSA 暗号を使用してキーを作成します。RSA 暗号化は、2 つの大きな素数の積に基づくパブリックキーのサイズのために、比較的遅くなる可能性があります。ただし、非 RSA 暗号化方式を使用する TLS 用のキーを作成することも可能です。同等レベルのセキュリティを提供する場合は、橿円曲線の計算に基づいたキーのほうが小さく計算処理も高速です。

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

結果は、prime256v1（OpenSSL でサポートされる「名前付き曲線」）を使用した 256 ビットの橿円曲線プライベートキーです。暗号化強度は（NIST によると）2048 ビットの RSA キーよりも優れています。

Note

すべての CA で、橿円曲線ベースのキーに対して RSA キーと同じレベルのサポートが提供されているわけではありません。

新しいプライベートキーには、制限の厳しい所有権とアクセス権を設定します（所有者 = root、グループ = root、所有者のみの読み取り/書き込み）。コマンドは次のようにになります。

```
[ec2-user ~]$ sudo chown root.root custom.key  
[ec2-user ~]$ sudo chmod 600 custom.key  
[ec2-user ~]$ ls -al custom.key
```

上のコマンドを実行すると、次のような結果になります。

```
-rw----- root root custom.key
```

適切なキーを作成し、設定できたら、CSR を作成できます。

- 好みのキーを使用して CSR を作成します。次の例では、**custom.key** を使用しています。

```
[ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
```

OpenSSL によりダイアログが開かれ、次の表に示されている情報の入力が求められます。基本的なドメイン検証済みホスト証明書については、[Common Name (共通名)] 以外のフィールドはすべてオプションです。

名前	説明	例
国名	2 文字の ISO 略称（国名コード）。	US (= 米国)

名前	説明	例
州名	あなたが所属する組織の所在地の州または県。省略不可です。	ワシントン
市区町村	市など、組織の場所。	シアトル
組織名	組織の正式名称。組織名は、省略不可です。	Example Corp
部門名	組織に関する追加情報(存在する場合)。	Example Dept
共通名	この値は、ユーザーがブラウザに入力する必要のあるウェブアドレスと正確に一致します。通常、これはプレフィックス付きのホスト名またはエイリアスによるドメイン名( <code>www.example.com</code> の形式)を意味します。自己署名証明書を使用し、DNS 解決なしでテストを行う場合、共通名の構成要素はホスト名のみになる場合があります。CAでは、 <code>*.example.com</code> などのワイルドカード名を許容する、よりコストの高い証明書も用意されています。	<code>www.example.com</code>
E メールアドレス	サーバー管理者の E メールアドレス。	<code>someone@example.com</code>

最後に、OpenSSLにより、オプションのチャレンジパスワードが求められます。このパスワードは CSR と、ユーザーと CA の間のトランザクションのみに適用されるため、このフィールドと、もう 1 つのオプションフィールドである、オプションの会社名については、CA の推奨事項に従ってください。CSR のチャレンジパスワードは、サーバー操作には影響しません。

結果として生成されるファイル `csr.pem` には、パブリックキー、パブリックキーのデジタル署名、入力したメタデータが含まれています。

- CA に CSR を送信します。この作業は通常、テキストエディタで CSR ファイルを開く動作と、内容をウェブフォームにコピーする動作で構成されています。このとき、証明書に適用する 1 つ以上のサブジェクト代替名(SAN)を指定するように求められることがあります。共通名が `www.example.com` の場合、有効な SAN は `example.com` になります(逆も同様です)。サイトへの訪問者がこれら名前のいずれかを入力すると、エラーなしの接続が提示されます。CA のウェブフォームで許可される場合は、SAN のリストに共通名を含めます一部の CA では自動的に含められます。

リクエストが承認されると、CA によって署名された新しいホスト証明書が届きます。CA の信頼チェーンを完成するために必要な、追加の証明書が含まれている中間証明書ファイルをダウンロードするよう指示されることもあります。

#### Note

多様な用途向けに複数の形式のファイルを送信してくる CA もあります。このチュートリアルでは、PEM 形式の証明書ファイルのみ使用してください。PEM 形式のファイルには通常、`.pem` または `.crt` 拡張子が使用されます(ただし、常にこれらの拡張子が使用されるわけではありません)。どのファイルを使用すべきかわからない場合は、テキストエディタでファイルを開き、以下の行で始まる 1 つ以上のブロックを含むファイルを見つけてください。

```
-----BEGIN CERTIFICATE-----
```

ファイルの末尾は次のようにになっている必要があります。

```
- - - - -END CERTIFICATE - - - - -
```

次のように、コマンドラインでファイルを確認することもできます。

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```

これらの行がファイルに表示されていることを確認してください。`.p7b`、`.p7c`、または類似のファイル拡張子で終了するファイルは使用しないでください。

- 新しい CA 署名証明書と任意の中間証明書を `/etc/pki/tls/certs` ディレクトリに配置します。

Note

EC2 インスタンスにカスタムキーをアップロードする方法は複数ありますが、最も簡単でわかりやすい方法は、テキストエディタ (`vi`、`nano`、メモ帳など) をローカルコンピュータとインスタンスの両方で開いて、両者の間でファイルの内容をコピーして貼り付けることです。EC2 インスタンス内でこれらの操作を実行する際には、`root [sudo]` アクセス許可が必要です。こうすることで、許可やパスに問題があるかどうかをすぐに確認できます。ただし、内容をコピーする際に行を追加したり、内容を変更したりしないでください。

`/etc/pki/tls/certs` ディレクトリの中から、次のコマンドを使用してファイルの所有者、グループ、アクセス許可の設定が制限の厳しい Amazon Linux のデフォルト (所有者 = `root`、グループ = `root`、所有者のみの読み込み/書き込み可) と一致することを確認します。

```
[ec2-user certs]$ sudo chown root.root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

上のコマンドを実行すると、次のような結果になります。

```
-rw----- root root custom.crt
```

中間証明書ファイルのアクセス権は、比較的厳しくありません (所有者 = `root`、グループ = `root`、所有者による書き込み可、グループによる読み取り可、その他による読み取り可)。コマンドは次のようになります。

```
[ec2-user certs]$ sudo chown root.root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

上のコマンドを実行すると、次のような結果になります。

```
-rw-r--r-- root root intermediate.crt
```

- カスタムキーを使用して CSR を作成し、ホスト証明書を取得した場合は、`/etc/pki/tls/private/` ディレクトリから古いキーを削除するか、名前を変更して、同ディレクトリに新しいキーをインストールします。

Note

EC2 インスタンスにカスタムキーをアップロードする方法は複数ありますが、最も簡単でわかりやすい方法は、テキストエディタ (`vi`、`nano`、メモ帳など) をローカルコンピュータとインスタンスの両方で開いて、両者の間でファイルの内容をコピーして貼り付けることです。EC2 インスタンス内でこれらの操作を実行する際には、`root [sudo]` 権限が必要です。こうすることで、許可やパスに問題があるかどうかをすぐに確認できます。ただし、内容をコピーする際に行を追加したり、内容を変更したりしないでください。

/etc/pki/tls/private ディレクトリの中から、ファイルの所有者、グループ、アクセス権の設定が制限の厳しい Amazon Linux のデフォルト (所有者 = root、グループ = root、所有者のみの読み込み/書き込み可) と一致することを確認します。コマンドは次のようにになります。

```
[ec2-user private]$ sudo chown root.root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

上のコマンドを実行すると、次のような結果になります。

```
-rw----- root root custom.key
```

7. 新しい証明書とキーファイルに合わせるには、/etc/httpd/conf.d/ssl.conf を編集します。
  - a. CA 署名のホスト証明書のパスとファイル名を Apache の SSLCertificateFile ディレクティブで指定します。

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

- b. 中間証明書ファイル (この例では intermediate.crt) を受け取つたら、Apache の SSLCACertificateFile ディレクティブを使用して、次のファイルのパスとファイル名を指定します。

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

#### Note

一部の CA では、ホスト証明書と中間証明書を組み合わせて 1 つのファイルを作成するため、このディレクティブは必要ありません。CA が提供している手順を参照してください。

- c. プライベートキーのパスとファイル名を Apache の SSLCertificateKeyFile ディレクティブで指定します。

```
SSLCertificateKeyFile /etc/pki/tls/private/custom.key
```

8. /etc/httpd/conf.d/ssl.conf を保存して、Apache を再起動します。

```
[ec2-user ~]$ sudo service httpd restart
```

9. サーバーをテストするには、ブラウザの URL バーにドメイン名を入力し、プレフィックス https:// を指定します。ブラウザによって、エラーが生成されることなく、HTTPS 経由でテストページがロードされます。

## ステップ 3: セキュリティ設定のテストと強化

TLS が運用可能になりパブリックに公開されたら、実際の安全性をテストする必要があります。セキュリティアップの詳細な分析を無料で行うことできる [Qualys SSL Labs](#) などのオンラインサービスを使用すると簡単です。その結果に基づき、受け入れるプロトコル、優先する暗号化方式、除外する暗号化方式を制御することによって、デフォルトのセキュリティ設定を強化するかどうかを決定できます。詳細については、「[Qualys のスコアの計算方法](#)」を参照してください。

#### Important

サーバーのセキュリティを確保するには、実際のテストが非常に重要です。小さな設定エラーによって、深刻なセキュリティ侵害やデータの損失が生じる可能性があります。調査や新たな脅威

に応じて、推奨されるセキュリティ管理方法は常に変化するため、適切なサーバー管理を行うには、定期的なセキュリティ監査が不可欠です。

Qualys SSL Labs のサイトで、サーバーの完全修飾ドメイン名を [www.example.com](http://www.example.com) という形式で入力します。約 2 分後に、サイトに関するグレード (A から F) と、結果の詳細な内訳が届きます。概要は設定がほとんど正常であることを示していますが、詳細レポートでは、いくつかの潜在的な問題が指摘されています。例:

XRC4 暗号は、特定の古いブラウザでの使用がサポートされています。暗号は、暗号化アルゴリズムの計算の中核です。TLS データストリームの暗号化に使用される高速の暗号化方式である RC4 は、いくつかの重大な脆弱性を持つことで知られています。従来のブラウザをサポートするもっともな理由がない限り、この暗号化方式を無効にする必要があります。

X旧バージョンの TLS がサポートされています。設定では TLS 1.0 (すでに廃止されています) と TLS 1.1 (廃止予定) がサポートされています。2018 年以降は、TLS 1.2 のみ推奨されています。

#### TLS 設定を修正するには

1. テキストエディタで設定ファイル `/etc/httpd/conf.d/ssl.conf` を開き、行頭に # を付けて以下の行をコメントアウトしてください。

```
#SSLProtocol all -SSLv3  
#SSLProxyProtocol all -SSLv3
```

2. 次のディレクティブを追加します。

```
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2  
SSLProxyProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```

これらのディレクティブにより、SSL バージョン 2、3、および TLS バージョン 1.0、1.1 が明示的に無効化されます。これで、サーバーでは、TLS 1.2 以外を使用した、クライアントとの暗号化された接続の受け入れが拒否されます。ディレクティブに含める指定が多くなるほど、サーバーの動作に対する設定内容が明確にわかりやすくなります。

#### Note

このようにして、TLS バージョン 1.0 および 1.1 を無効にすると、ごく一部の古くなったウェブブラウザによるサイトへのアクセスがブロックされるようになります。

#### 許可された暗号のリストを変更するには

1. 設定ファイル `/etc/httpd/conf.d/ssl.conf` を開き、`SSLCipherSuite` と `SSLProxyCipherSuite` を設定するためのコメントアウトされた例のセクションを見つけます。

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5  
#SSLProxyCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

これらの設定はそのままにして、その下に以下のディレクティブを追加します。

#### Note

ここでは読みやすくするために数行に分けて示していますが、これらの 2 つのディレクティブは 1 行に指定する必要があります。暗号化方式名はスペースで区切りません。

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-  
CHACHA20-POLY1305:  
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-  
SHA256:ECDHE-ECDSA-AES256-SHA384:
```

```
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES:!aNULL:  
eNULL:!EXPORT:!DES:  
!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA  
  
SSLProxyCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-  
ECDSA-CHACHA20-POLY1305:  
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-  
SHA256:ECDHE-ECDSA-AES256-SHA384:  
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES:!aNULL:  
eNULL:!EXPORT:!DES:  
!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA
```

これらの暗号化方式は、OpenSSL でサポートされている暗号化方式の長いリストのうち、一部です。これらは、以下の条件に応じて選択され、順序付けられています。

- 前方秘匿性のサポート
- Strength
- スピード
- 特定の暗号化方式、その後に暗号化方式のファミリー
- 許可されている暗号化方式、その後に拒否されている暗号化方式

ランクの高い暗号化方式の名前には、ECDHE が含まれています (Elliptic Curve Diffie-Hellman Ephemeral ; など)。ephemeral は前方秘匿性を示します。また、RC4 は現在、リストの最後に近い位置にあり、禁止された暗号化方式に含まれています。

デフォルトや、内容が見えない簡単なディレクティブに依存するのではなく、暗号化方式の明示的なリストを使用することをお勧めします。

#### Important

ここに示されている暗号化方式リストは、多数考えられるリストの 1 つに過ぎません。たとえば、前方秘匿性よりスピードを重視したリストが必要になることもあります。

古いクライアントをサポートする必要性が予測される場合は、DES-CBC3-SHA 暗号化スイートを許可することができます。

最後に、OpenSSL の更新ごとに、新しい暗号化方式が導入され古い暗号化方式が廃止されます。EC2 の Amazon Linux インスタンスを最新の状態に維持し、セキュリティに関する OpenSSL からの告知に注意して、技術分野の報道でセキュリティ面の新しい脆弱性に関するレポートを警戒してください。詳細については、『クラシックロードバランサー用ユーザーガイド』の「[Elastic Load Balancing での事前定義された SSL のセキュリティポリシー](#)」を参照してください。

2. 次の行について、"#" を削除してコメント解除します。

```
#SSLHonorCipherOrder on
```

このコマンドは、(この場合) 前方秘匿性をサポートするものも含めて、ランクの高い暗号化方式を優先するようサーバーに強制します。このディレクティブが有効になると、サーバーは、セキュリティの弱い暗号化方式に戻る前に、セキュリティが強力な接続を確立しようとします。

3. Apache を再起動します。[Qualys SSL Labs](#) でドメインをもう一度テストすると、RC4 の脆弱性がなくなったことがわかります。

## トラブルシューティング

- パスワードを指定しないと Apache ウェブサーバーが起動しません

これは、パスワードで保護された暗号化プライベートサーバーキーをインストールした場合は正常な動作です。

暗号化とパスワードの要件をキーから削除できます。デフォルトディレクトリに custom.key という暗号化プライベート RSA キーがあり、そのパスワードが abcde12345 であるとすると、EC2 インスタンスで次のコマンドを実行し、このキーの非暗号化バージョンを生成してください。

```
[ec2-user ~]$ cd /etc/pki/tls/private/  
[ec2-user private]$ sudo cp custom.key custom.key.bak  
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out  
custom.key.nocrypt  
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key  
[ec2-user private]$ sudo chown root.root custom.key  
[ec2-user private]$ sudo chmod 600 custom.key  
[ec2-user private]$ sudo service httpd restart
```

パスワードが求められずに Apache が起動するようになります。

## Certificate Automation: Amazon Linux での Let's Encrypt と Certbot の使用

Let's Encrypt 認証局は、インターネット全体を暗号化するための Electronic Frontier Foundation (EFF) の取り組みの中核部分です。この目標を踏まえ、Let's Encrypt ホスト証明書は、手動による介入を最小限に抑えながら、作成、評価、インストール、保守されるように設計されています。証明書管理の自動化側面はウェブサーバー上で動作しているエージェントによって実行されます。インストールおよび設定されたエージェントは、Let's Encrypt と安全に通信して、Apache およびキー管理システム上で管理タスクを実行します。このチュートリアルでは、Certbot エージェントを使用します。このエージェントを使用することで、カスタマイズされた暗号化キーを証明書の基盤として提供するか、エージェント自身がそのデフォルト値を使用してキーを作成できます。また、「[Certbot を自動化するには \(p. 76\)](#)」で説明するように、手動による介入なしで定期的に証明書を更新するよう Certbot を設定することもできます。詳細については、Certbot の「[ユーザーガイド](#)」または「[マニュアルページ](#)」を参照してください。

Certbot は Amazon Linux AMI で公式にサポートされていませんが、ダウンロードすることができ、インストールすると正常に機能します。データを保護し、問題を回避するため、次のバックアップを作成しておくことをお勧めします。

- 開始する前に、Amazon EBS ルートボリュームのスナップショットを作成します。これにより、EC2 インスタンスの元の状態に復元することができます。EBS スナップショットの作成方法の詳細については、「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」を参照してください。
- 以下の手順では、Apache の操作を制御する httpd.conf ファイルを編集する必要があります。Certbot はこのファイルと他の設定ファイルに独自の自動変更を加えます。復元する必要が生じたときのために、/etc/httpd ディレクトリ全体のバックアップコピーを作成してください。

Certbot をインストールして実行するには

- インスタンスで Fedora プロジェクトの Extra Packages for Enterprise Linux (EPEL) リポジトリを有効にします。Certbot のインストールスクリプトを実行するには、依存ファイルとして EPEL のパッケージが必要です。

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

- 次のコマンドを使用して、EFF から Certbot の最新リリースを EC2 インスタンスにダウンロードします。

```
[ec2-user ~]$ wget https://dl.eff.org/certbot-auto
```

3. ダウンロードしたファイルを実行可能にします。

```
[ec2-user ~]$ chmod a+x certbot-auto
```

4. ルート権限で、--debug フラグを使用してファイルを実行します。

```
[ec2-user ~]$ sudo ./certbot-auto --debug
```

5. "Is this ok [y/d/N]" というプロンプトが表示されたら「y」と入力し、Enter キーを押します。
6. "Enter email address (used for urgent renewal and security notices)" というプロンプトが表示されたら、連絡先住所を入力し、Enter キーを押します。
7. プロンプトが表示されたら Let's Encrypt のサービス利用規約に同意します。「A」と入力し、Enter キーを押します。

```
-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.1.1-August-1-2016.pdf. You must agree
in order to register with the ACME server at
https://acme-v01.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: A
```

8. 「Y」または「N」と入力し、Enter キーを押してすべての承認を完了すると、EFF のメーリングリストに登録されます。
9. 以下のプロンプトで、共通名(上記のドメイン名)とサブジェクトの別名(SAN)をカンマで区切って入力します。Enter キーを押します。この例では、入力した名前が提供されています。

```
No names were found in your configuration files. Please enter in your domain
name(s) (comma and/or space separated) (Enter 'c' to cancel): example.com
www.example.com
```

10. Apache のデフォルト設定を使用した Amazon Linux システムでは、次の例のような出力が表示され、入力した最初の名前にについて尋ねられます。「1」と入力して、Enter キーを押します。

```
Obtaining a new certificate
Performing the following challenges:
tls-sni-01 challenge for example.com
tls-sni-01 challenge for www.example.com

We were unable to find a vhost with a ServerName or Address of example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-----
1: ssl.conf | HTTPS | Enabled
-----
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
```

11. 次に、2 番目の名前にについて尋ねられます。「1」と入力して、Enter キーを押します。

```
We were unable to find a vhost with a ServerName or Address of www.example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-----
1: ssl.conf | HTTPS | Enabled
-----
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
Certificate Automation: Amazon Linux  
での Let's Encrypt と Certbot の使用

Press 1 [enter] to confirm the selection (press 'c' to cancel): **1**

この時点では、キーと CSR が作成されます。

```
Waiting for verification...
Cleaning up challenges
Generating key (2048 bits): /etc/letsencrypt/keys/0000_key-certbot.pem
Creating CSR: /etc/letsencrypt/csr/0000_csr-certbot.pem
```

12. Certbot にすべての必要なホスト証明書の作成を許可します。各名前の入力を求められたら、例のように、「1」と入力して、Enter キーを押します。

```
We were unable to find a vhost with a ServerName or Address of example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-----
1: ssl.conf | | HTTPS | Enabled
-----
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
Deploying Certificate for example.com to VirtualHost /etc/httpd/conf.d/ssl.conf

We were unable to find a vhost with a ServerName or Address of www.example.com.
Which virtual host would you like to choose?
(note: conf files with multiple vhosts are not yet supported)
-----
1: ssl.conf | example.com | HTTPS | Enabled
-----
Press 1 [enter] to confirm the selection (press 'c' to cancel): 1
Deploying Certificate for www.example.com to VirtualHost /etc/httpd/conf.d/ssl.conf
```

13. ウェブサーバーへの安全でない接続を許可するかどうかを選択します。例のようにオプション 2 を選択すると、サーバーへのすべての接続が暗号化されるか、拒否されます。

```
Please choose whether HTTPS access is required or optional.
-----
1: Easy - Allow both HTTP and HTTPS access to these sites
2: Secure - Make all requests redirect to secure HTTPS access
-----
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
```

Apache の設定が完了し、成功した旨とその他の情報が報告されます。

```
Congratulations! You have successfully enabled https://example.com and
https://www.example.com

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=example.com
https://www.ssllabs.com/ssltest/analyze.html?d=www.example.com
-----
IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at
  /etc/letsencrypt/live/example.com/fullchain.pem. Your cert will
  expire on 2017-07-19. To obtain a new or tweaked version of this
  certificate in the future, simply run certbot-auto again with the
  "certonly" option. To non-interactively renew *all* of your
  certificates, run "certbot-auto renew"
....
```

14. インストールが完了したら、「[ステップ 3: セキュリティ設定のテストと強化 \(p. 70\)](#)」に記載されている手順に従って、サーバーのセキュリティのテストと最適化を行います。

Certbot は、サーバーシステムを構成する不可視のエラー回復性のあるパートとなるように設計されています。デフォルトでは、90 日間の短い有効期限を持つホスト証明書を生成します。システムがコマンドを自動的に呼び出すように事前に設定していない場合は、certbot コマンドを手動で再実行する必要があります。以下の手順は、cron ジョブを設定して Certbot を自動化する方法を示しています。

#### 証明書の更新を自動化設定するには

1. Certbot が最初に正常に実行されたら、テキストエディタで /etc/crontab ファイルを開き、次のような行を追加します。

```
39      1,13    *      *      *      root    certbot renew --no-self-upgrade
```

完了したらファイルを保存します。以下に、各構成要素について説明します。

```
39 1,13 * * *
```

毎日、01:39 と 13:39 にコマンドが実行されるようにスケジュールします。ここで選択した値は任意ですが、Certbot 開発者は、コマンドを少なくとも毎日 2 回実行することを推奨しています。これにより、侵害されていることがわかった証明書は必ずすぐに取り消されて置き換えられます。

```
root
```

コマンドは、root 権限で実行されます。

```
certbot renew --no-self-upgrade
```

実行されるコマンド。renew サブコマンドを実行すると、Certbot は、以前に取得した証明書があれば確認し、有効期限が近づいているものを更新します。--no-self-upgrade フラグにより、Certbot が手動による介入なしで自動的にアップグレードされることを禁止しています。

2. cron デーモンを再起動します。

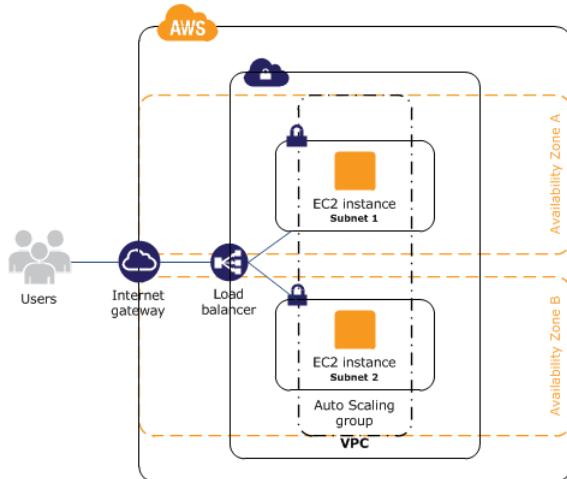
```
[ec2-user ~]$ sudo service crond restart
```

## チュートリアル: Amazon EC2 のアプリケーションの可用性の向上

単一の EC2 インスタンスで開始したアプリやウェブサイトが、時間が立つにつれてトラフィックが増加し、需要を満たすために複数のインスタンスが必要な時点まで来ているとします。AMI から複数の EC2 インスタンスを起動し、Elastic Load Balancing を使用してアプリケーションの受信トラフィックをこれらの EC2 インスタンスに分配できます。これにより、アプリケーションの可用性が向上します。複数のアベイラビリティーゾーンにインスタンスを置くことで、アプリケーションの耐障害性も向上します。片方のアベイラビリティーゾーンが停止すると、トラフィックがもう片方のアベイラビリティーゾーンにルーティングされます。

Amazon EC2 Auto Scaling を使用して、アプリケーション用に実行するインスタンスを常に最小数に維持できます。Amazon EC2 Auto Scaling はインスタンスやアプリケーションの不具合を検出し、自動的に置き換えて、アプリケーションの可用性を維持します。また、Amazon EC2 Auto Scaling を使用して、必要に応じて指定した条件に従って Amazon EC2 の容量を自動的にスケールできます。

このチュートリアルでは、Amazon EC2 Auto Scaling で Elastic Load Balancing を使用して、ロードバランサーの背後の正常な EC2 インスタンスの指定数を確実に維持します。トラフィックはロードバランサーにアクセスし、その後インスタンスにルーティングされるので、これらのインスタンスにパブリック IP アドレスが必要ないことに注目してください。詳細については、「[Amazon EC2 Auto Scaling](#)」および「[Elastic Load Balancing](#)」を参照してください。



## コンテンツ

- [前提条件 \(p. 91\)](#)
- [アプリケーションのスケーリングと負荷分散 \(p. 91\)](#)
- [ロードバランサーをテストする \(p. 93\)](#)

## 前提条件

このチュートリアルでは、以下を実行済みであることを前提としています。

1. 2つ以上のアベイラビリティゾーンに1つのパブリックサブネットを持つ Virtual Private Cloud (VPC) を作成した。
2. VPC のインスタンスを起動した。
3. インスタンスに接続し、カスタマイズした。たとえば、ソフトウェアやアプリケーションのインストール、データのコピー、追加の EBS ボリュームのアタッチを行うことができます。インスタンスでのウェブサーバーの設定については、「[チュートリアル: Amazon Linux AMI を使用して LAMP ウェブサーバーをインストールする \(p. 42\)](#)」を参照してください。
4. インスタンスが正しく設定されたことを確認するために、インスタンスでアプリケーションをテストした。
5. インスタンスからカスタム Amazon Machine Image (AMI) を作成した。詳細については「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」または「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」を参照してください。
6. (オプション) 不要になったインスタンスを削除した。
7. アプリケーションに必要な AWS へのアクセスを付与する IAM ロールを作成した。詳細については、「[IAM コンソールを使用して IAM ロールを作成するには \(p. 891\)](#)」を参照してください。

## アプリケーションのスケーリングと負荷分散

以下の手順を使用して、ロードバランサーの作成、インスタンスの起動設定の作成、複数のインスタンスを持つ Auto Scaling グループの作成、Auto Scaling グループへのロードバランサーの関連付けを行います。

### アプリケーションのスケーリングと負荷分散を行うには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [LOAD BALANCING] で [ロードバランサー] を選択します。

3. [Create Load Balancer] を選択します。
4. [アプリケーションロードバランサー] で [作成] を選択します。
5. [Configure Load Balancer] ページで、以下を実行します。
  - a. [Name] に、ロードバランサーの名前を入力します。たとえば、**my-1b** と指定します。
  - b. [Scheme] で、デフォルト値 [Internet-facing] を保持します。
  - c. [Listeners] では、デフォルトを保持します。これは、ポート 80 で HTTP トラフィックを受け付けるリスナーです。
  - d. [Availability Zones] で、インスタンスに使用する VPC を選択します。アベイラビリティゾーンを選択してから、そのアベイラビリティゾーンのパブリックサブネットを選択します。2 番目のアベイラビリティゾーンに対して、この操作を繰り返します。
  - e. [Next: Configure Security Settings (次へ: セキュリティ設定の構成)] を選択します。
6. このチュートリアルでは、セキュアリスナーを使用しません。[Next: Configure Security Groups (次へ: セキュリティグループの設定)] を選択します。
7. [Configure Security Groups (セキュリティグループの設定)] ページで、次の手順を完了します。
  - a. [Create a new security group (新しいセキュリティグループの作成)] を選択します。
  - b. セキュリティグループの名前と説明を入力するか、デフォルトの名前と説明を維持します。この新しいセキュリティグループには、リスナーに設定されたポートへのトラフィックを許可するルールが含まれます。
  - c. [Next: Configure Routing (次へ: ルーティングの設定)] を選択します。
8. [Configure Routing (ルーティングの設定)] ページで、以下を実行します。
  - a. [Target group] で、デフォルトの [New target group] を保持します。
  - b. [Name] に、ターゲットグループの名前を入力します。
  - c. [プロトコル] は HTTP、[ポート] は 80、[ターゲットの種類] はインスタンスで維持します。
  - d. [ヘルスチェック] は、デフォルトプロトコルとパスを保持します。
  - e. [Next: Register Targets] を選択します。
9. Amazon EC2 Auto Scaling を使用して EC2 インスタンスをターゲットグループに追加するため、[Register Targets] ページで [Next: Review] を選択して次のページに進みます。
10. [Review] ページで、[Create] を選択します。ロードバランサーが作成されたら、[Close] を選択します。
11. ナビゲーションペインの [AUTO SCALING] で、[起動設定] を選択します。
  - Amazon EC2 Auto Scaling を初めて使用する場合は、ウェルカムページを参照してください。[Create Auto Scaling group] を選択して [Auto Scaling グループの作成] ウィザードを起動してから、[Create launch configuration (起動設定の作成)] を選択します。
  - その他の場合は、[Create launch configuration] を選択します。
12. [Choose AMI] ページで、[My AMIs] タブを選択し、「[前提条件 \(p. 91\)](#)」で作成した AMI を選択します。
13. [Choose Type] ページでインスタンスタイプを選択してから、[Next: Configure details] を選択します。
14. [Configure details] ページで、以下を実行します。
  - a. [Name] に起動設定の名前を入力します (例: **my-launch-config**)。
  - b. [IAM role] で、「[前提条件 \(p. 91\)](#)」で作成した IAM ロールを選択します。
  - c. (オプション) 起動スクリプトを実行する必要がある場合は、[Advanced Details] を展開して [User data] にスクリプトを入力します。
  - d. [Skip to review] を選択します。
15. [Review] ページで、[Edit security groups (セキュリティグループの編集)] を選択します。既存のセキュリティグループを選択することも、新しいセキュリティグループを作成することもできます。このセキュリティグループは、ロードバランサーからの HTTP トラフィックおよびヘルスチェックを許

可する必要があります。インスタンスにパブリック IP アドレスがある場合は、インスタンスに接続する必要がある場合はオプションで SSH トラフィックを許可できます。完了したら、[Review] を選択します。

16. [Review] ページで、[Create launch configuration] を選択します。
17. プロンプトが表示されたら、既存のキーペアを選択するか、新しいキーペアを作成するか、またはキーペアなしで先に進みます。確認チェックボックスをオンにし、[Create launch configuration (起動設定の作成)] を選択します。
18. 起動設定を作成した後、Auto Scaling グループを作成する必要があります。
  - Amazon EC2 Auto Scaling の使用が初めてで Create Auto Scaling group (Auto Scaling グループの作成) ウィザードを使用している場合は、自動的に次のステップに移動します。
  - その他の場合は、[Create an Auto Scaling group using this launch configuration] を選択します。
19. [Configure Auto Scaling group details] ページで、以下を実行します。
  - a. [Group name] に、Auto Scaling グループの名前を入力します。たとえば、**my-asg** と指定します。
  - b. [Group size] で、インスタンスの値を入力します (例: 2)。各アベイラビリティゾーンのインスタンスをおよそ同数にすることが推奨されていることに注意してください。
  - c. [Network] から使用する VPC を選択し、[Subnet] から 2 つのパブリックサブネットを選択します。
  - d. [Advanced Details] で、[Receive traffic from one or more load balancers] を選択します。[Target Groups] からターゲットグループを選択します。
  - e. [Next: Configure scaling policies] を選択します。
20. Amazon EC2 Auto Scaling にグループを指定したサイズに維持させるため、[Configure scaling policies] ページで、[Review] を選択します。後からこの Auto Scaling グループを手動でスケールしたり、グループをスケジュールでスケールするように構成したり、オンデマンドでグループをスケールするように構成したりできることに注目してください。
21. [Review] ページで、[Create Auto Scaling group] を選択します。
22. グループが作成されたら、[Close] を選択します。

## ロードバランサーをテストする

クライアントがロードバランサーにリクエストを送信すると、ロードバランサーは、登録されたいずれかのインスタンスにリクエストをルーティングします。

### ロードバランサーをテストするには

1. インスタンスの準備が完了していることを確認します。[Auto Scaling Groups] ページから Auto Scaling グループを選択して、[Instances] タブを選択します。当初、インスタンスの状態は Pending です。状態が InService になると、使用する準備が整っています。
2. インスタンスがロードバランサーに登録されていることを確認します。[Target Groups] ページからターゲットグループを選択して、[Targets] タブを選択します。インスタンスの状態が initial である場合は、まだ登録中の可能性があります。インスタンスの状態が healthy になると、使用できる状態です。インスタンスの準備が整ったら、次のようにしてロードバランサーをテストできます。
3. [Load Balancers] ページからロードバランサーを選択します。
4. [Description] タブで、DNS 名を見つけます。この名前は次の形式です。

**my-lb-xxxxxxxxxx.us-west-2.elb.amazonaws.com**

5. ウェブブラウザで、ロードバランサーの DNS 名をアドレスバーに貼り付けて、Enter キーを押します。ウェブサイトが表示されることを確認します。

# Amazon マシンイメージ (AMI)

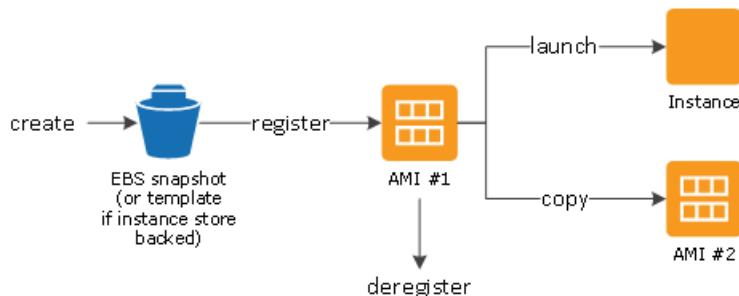
Amazon マシンイメージ (AMI) には、インスタンスの起動に必要な情報が用意されています。インスタンスを起動するときは、AMI を指定する必要があります。同じ設定で複数のインスタンスが必要な場合は、1 つの AMI から複数のインスタンスを起動できます。さまざまな設定のインスタンスが必要なときは、各インスタンスをそれぞれ異なる AMI から起動できます。

AMI には次が含まれています。

- 1 つまたは複数の EBS スナップショット、または、instance-store-backed AMI、インスタンスのルートボリュームのテンプレート (オペレーティングシステム、アプリケーションサーバー、アプリケーションなど)
- 起動許可 (AMI を使用してインスタンスを起動する権限を特定の AWS アカウントに与える)
- インスタンスの起動時にインスタンスにアタッチするボリュームを指定するブロックデバイスマッピング

## AMI の使用

次の図は AMI のライフサイクルをまとめたものです。AMI を作成し、登録したら、それを使用して新しいインスタンスを起動できます (AMI 所有者から起動許可を与えられた場合、AMI からインスタンスを起動することもできます)。AMI は同じリージョン内でコピーすることも、異なるリージョンにコピーすることもできます。不要になった AMI は登録を解除できます。



ご自分のインスタンスの基準に一致する AMI を検索できます。AWS が提供する AMI またはコミュニティが提供する AMI を検索できます。詳細については、「[AMI タイプ \(p. 95\)](#)」および「[Linux AMI の検索 \(p. 100\)](#)」を参照してください。

AMI からインスタンスを起動したら、インスタンスに接続できます。インスタンスに接続したら、そのインスタンスを他のサーバーとまったく同じように使用できます。インスタンスの起動、接続、使用に関する詳細については、[Amazon EC2 インスタンス \(p. 183\)](#) を参照してください。

## 独自の AMI の作成

既存の AMI からインスタンスを起動し、インスタンスをカスタマイズして、この更新された設定をカスタム AMI として保存できます。この新しいカスタム AMI から起動されるインスタンスには、AMI の作成時に追加したカスタマイズが含まれます。

AMI の作成プロセスは、インスタンスのルートストレージデバイスにより決まります。インスタンスのルートボリュームは、Amazon EBS ボリュームまたはインスタンスストアボリュームのどちらかです。ルートデバイスピリュームの詳細については、「[Amazon EC2 ルートデバイスピリューム \(p. 16\)](#)」を参照してください。

- Amazon EBS-Backed AMI を作成するには、「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」を参照してください。
- Instance Store-Backed AMI の作成については、「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」を参照してください。

AMI には分類や管理のために任意のタグを付けられます。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。

## AMI の購入、共有、販売

AMI を作成したら、自分がそれを使用できるようにプライベートとして保存したり、AWS アカウントの指定リストと共有したりできます。コミュニティで利用できるように、カスタム AMI を公開することもできます。安全で信頼性が高く、便利な AMI を作成して、一般公開する手順はきわめて単純で、いくつかのシンプルなガイドラインにしたがうだけです。共有 AMI の作成および使用方法の詳細については、[共有 AMI \(p. 102\)](#) を参照してください。

Red Hat のような組織のサービス契約に付属する AMI など、サードパーティーから AMI を購入できます。また、AMI を作成し、他の Amazon EC2 ユーザーに販売することもできます。AMI の購入と販売に関する詳細については、[有料 AMI \(p. 112\)](#) を参照してください。

## AMI の登録解除

AMI の利用が終わったら、その登録を解除できます。AMI の登録を解除すると、その AMI を使用して新しいインスタンスを起動できなくなります。その AMI から起動された既存のインスタンスは影響を受けません。詳細については、「[Linux AMI の登録解除 \(p. 163\)](#)」を参照してください。

## Amazon Linux 2 および Amazon Linux AMI

Amazon Linux 2 および Amazon Linux AMI は、AWS がサポートおよび保守管理している Linux イメージの 1 つです。次に示すのは Amazon Linux 2 および Amazon Linux AMI の特徴の一部です。

- Amazon EC2 で実行されるアプリケーションのため、安定性があり、安全で高性能な実行環境。
- Amazon EC2 ユーザーには追加料金なしで提供。
- MySQL、PostgreSQL、Python、Ruby、Tomcat など多くの一般的なパッケージの複数バージョンへのリポジトリアクセスが可能。
- 定期的な更新で最新のコンポーネントが追加される。更新は、インスタンスを実行するインストールの yum リポジトリでも利用可能。
- AWS CLI Amazon EC2 API および AMI ツール、Python 用の Boto ライブラリ、Elastic Load Balancing ツールなど、AWS サービスの統合を簡素化するパッケージが含まれる。

詳細については、「[Amazon Linux \(p. 165\)](#)」を参照してください。

## AMI タイプ

次の特性に基づき、使用する AMI を選択できます。

- リージョン（「[リージョン、アベイラビリティゾーン、および ローカルゾーン \(p. 7\)](#)」を参照）

- オペレーティングシステム
- アーキテクチャ (32 ビットまたは 64 ビット)
- 起動許可 (p. 96)
- ルートデバイスのストレージ (p. 96)

## 起動許可

AMI の所有者は、起動許可を指定することで可用性を決定します。起動許可は次のように分類されます。

起動許可	Description
パブリック	所有者はすべての AWS アカウントに起動許可を与えます。
明示的	所有者は特定の AWS アカウントに起動許可を与えます。
暗示的	所有者には AMI の暗示的起動許可があります。

Amazon や Amazon EC2 コミュニティではさまざまなパブリック AMI を提供しています。詳細については、「[共有 AMI \(p. 102\)](#)」を参照してください。開発者は自分の AMI に料金を請求できます。詳細については、「[有料 AMI \(p. 112\)](#)」を参照してください。

## ルートデバイスのストレージ

すべての AMI が Amazon EBS-Backed と Instance Store-Backed のいずれかに分類されます。前者は、AMI から起動されるインスタンスのルートデバイスが、Amazon EBS スナップショットから作成される Amazon EBS ボリュームであるということです。後者は、AMI から起動されるインスタンスのルートデバイスが、Amazon S3 に格納されたテンプレートから作成されるインスタンスストアボリュームであるということです。詳細については、「[Amazon EC2 ルートデバイスボリューム \(p. 16\)](#)」を参照してください。

次の表では、2 種類の AMI を使用した場合の重要な相違点をまとめています。

特徴	Amazon EBS-Backed AMI	Amazon Instance Store-Backed AMI
インスタンスの起動時間	通常 1 分以内	通常 5 分以内
ルートデバイスのサイズ制限	16 TiB	10 GiB
ルートデバイスボリューム	Amazon EBS ボリューム	インスタンスストアボリューム
データの永続性	デフォルトでは、インスタンスを終了するとルートボリュームは削除されます。 <sup>*</sup> Amazon EBS ボリュームにある他のデータはすべて、インスタンスの終了後もデフォルトで保持されます。	インスタンスストアボリューム上のデータは、インスタンスの存続中のみ使用できます。
変更	インスタンスの停止中に、インスタンスタイプ、カーネル、RAM ディスク、およびユーザーデータが変更可能	インスタンスの属性は、インスタンスを削除するまで固定。

特徴	Amazon EBS-Backed AMI	Amazon Instance Store-Backed AMI
料金	料金は、インスタンスの使用量、Amazon EBS ボリューム、Amazon EBS スナップショットとして保存した AMI に対して発生します。	インスタンスの使用量や Amazon S3 に保存した AMI に対して料金が発生します。
AMI の作成/バンドル	単一のコマンドまたは呼び出しを使用	AMI ツールをインストールして使用する必要があります
停止状態	インスタンスが実行されていない停止状態にできます。ただし、ルートボリュームは Amazon EBS に保持されます。	インスタンスが実行中も終了後も、停止状態にすることができない

\* デフォルトでは、Amazon EBS-Backed インスタンスのルートボリュームで、DeleteOnTermination フラグが `true` に設定されます。このフラグを変更し、終了後もボリュームを保持する方法については、「[永続的ルートデバイスボリュームへの変更 \(p. 19\)](#)」を参照してください。

## AMI のルートデバイスタイプの判別

コンソールを使用して AMI のルートデバイスタイプを判別するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[AMI] をクリックし、AMI を選択します。
3. 次のように、[詳細] タブで [ルートデバイスタイプ] の値を確認します。
  - 値が `ebs` の場合は Amazon EBS-Backed AMI です。
  - 値が `instance store` の場合、これは Instance store-Backed インスタンスです。

コマンドラインを使用して AMI のルートデバイスタイプを判別するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- `describe-images` (AWS CLI)
- `Get-EC2Image` (AWS Tools for Windows PowerShell)

## 停止状態

Amazon EBS-backed インスタンスは停止できますが、Amazon EC2 instance store-backed インスタンスは停止できません。停止すると、インスタンスの実行が停止します (ステータスが `running` から `stopping` を経て `stopped` に進む)。停止したインスタンスは Amazon EBS で保持されるため、再起動できます。`stopping` (停止) は `terminating` (終了) と異なります。`terminated` インスタンスは再起動できません。Amazon EC2 instance store-backed インスタンスは停止できないため、実行中が終了のいずれかになります。インスタンスが停止している場合に何が行われ、何を実行できるかの詳細については、「[インスタンスの停止と起動 \(p. 529\)](#)」を参照してください。

## デフォルトのデータストレージと永続性

ルートデバイスにインスタンスストアボリュームを使用するインスタンスでは、自動的にインスタンスストアが利用できます (ルートボリュームにルートパーティションが含まれ、追加のデータを保存できます)。1 つまたは複数の Amazon EBS ボリュームをアタッチすることで、永続的ストレージをインスタン

スに追加できます。インスタンスストアボリューム上のデータは、インスタンスが失敗または終了すると、削除されます。詳細については、「[インスタンスストアの存続期間 \(p. 1077\)](#)」を参照してください。

Amazon EBS をルートデバイスに使用するインスタンスには自動的に、Amazon EBS ボリュームがアタッチされます。ボリュームは、他のボリュームと同様に、ボリュームのリストに表示されます。ほとんどのインスタンスタイプでは、Amazon EBS-backed インスタンスは、デフォルトでインスタンスストアボリュームを保持しません。ロックデバイスマッピングを使用して、インスタンスストアボリュームまたは追加の Amazon EBS ボリュームを追加できます。詳細については、「[ロックデバイスマッピング \(p. 1100\)](#)」を参照してください。

## 作成時刻

Amazon EBS-backed AMI から起動するインスタンスは、instance store-backed AMI から起動するインスタンスよりも速く起動します。instance store-backed AMI からインスタンスを起動するときは、Amazon S3 からすべてのパートを取得しないとインスタンスを利用できません。Amazon EBS-backed AMI の場合、インスタンスの起動に必要な部分だけをスナップショットから取得するとインスタンスを利用できます。ただし、ルートデバイスに Amazon EBS ボリュームを使用するインスタンスのパフォーマンスは、残りの部分がスナップショットから取得され、ボリュームにロードされる少しの時間、遅くなります。インスタンスを停止し、再起動する場合は、状態が Amazon EBS ボリュームに保存されているため早く起動します。

## AMI の作成

Instance Store-Backed Linux AMI を作成するには、Amazon EC2 AMI ツールを使用して、当該のインスタンス上でインスタンスから AMI を作成する必要があります。

AMI の作成は、Amazon EBS Backed の AMI の方がはるかに簡単です。CreateImage API アクションは、Amazon EBS-Backed AMI を作成して登録します。AWS マネジメントコンソールにも、実行中のインスタンスから AMI を作成できるボタンがあります。詳細については、「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」を参照してください。

## 課金方法

Instance Store-Backed の AMI の場合、インスタンスの使用量と Amazon S3 への AMI の保存に対して課金されます。Amazon EBS-Backed の AMI の場合、インスタンスの使用料、Amazon EBS ボリュームストレージおよび使用量、AMI の Amazon EBS スナップショットとしての保存に対して課金されます。

Amazon EC2 Instance Store-Backed の AMI の場合、AMI をカスタマイズしたり、新しい AMI を作成したりするたびに、各 AMI のすべての部分が Amazon S3 に保存されます。そのため、カスタマイズした各 AMI のストレージフットプリントは、AMI の完全なサイズになります。Amazon EBS-Backed の AMI の場合、AMI をカスタマイズしたり、新しい AMI を作成したりするたびに、変更のみが保存されます。そのため、最初の AMI の後にカスタマイズする後続の AMI のストレージフットプリントははるかに小さくなり、AMI ストレージ料金が少なくなります。

Amazon EBS-backed instance が停止した場合、インスタンスの使用については課金されませんが、ボリュームストレージについては引き続き課金されます。インスタンスを起動した時点で、最低 1 分間分の使用料が課金されます。1 分経過した後は、使用した秒数のみ課金されます。たとえば、インスタンスを 20 秒間実行して停止した場合は、1 分間分課金されます。インスタンスを 3 分 40 秒実行した場合は、ちょうど 3 分 40 秒間分課金されます。インスタンスがアイドル状態で残っていて、そのインスタンスに接続していない場合でも、実行中のインスタンスに対して、1 秒ごとに最低 1 分間分の使用料が課金されます。

## Linux AMI 仮想化タイプ<sup>®</sup>

Linux Amazon マシンイメージでは、2 つの仮想化タイプ (準仮想化 (PV) およびハードウェア仮想マシン (HVM)) のどちらかを使用します。PV AMI と HVM AMI の主な違いは、起動の方法と、パフォーマンス向

上のための特別なハードウェア拡張機能 (CPU、ネットワーク、ストレージ) を利用できるかどうかという点です。

最適なパフォーマンスを得るために、インスタンスを起動するときには、現行世代のインスタンスタイプと HVM AMI を使用することをお勧めします。現行世代のインスタンスタイプの詳細については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。旧世代のインスタンスタイプを使用中で、アップグレードする場合は、「[アップグレードパス](#)」を参照してください。

#### HVM AMI

HVM AMI は、完全に仮想化された一連のハードウェアを備えており、イメージのルートブロックデバイスのマスター ブート レコードを実行することによって起動します。この仮想化タイプでは、ペアメタル ハードウェア上でオペレーティングシステムが動作するのと同様に、修正を行わなくても仮想マシン上でオペレーティングシステムを直接実行することができます。Amazon EC2 ホストシステムでは、ゲストに提供されている基盤となるハードウェアの一部またはすべてがエミュレートされます。

PV のゲストとは異なり、HVM のゲストは、ホストシステム上の基盤となるハードウェアへの高速なアクセスを可能にするハードウェア拡張を利用できます。Amazon EC2 で使用できる CPU 仮想化拡張機能の詳細については、Intel のウェブサイトの「[Intel Virtualization Technology](#)」を参照してください。HVM AMI は、拡張ネットワーキングと GPU 処理を利用する場合に必要です。専用のネットワークや GPU デバイスに命令を伝達するには、OS がネイティブ ハードウェア プラットフォームにアクセスできる必要があります。HVM 仮想化ではこのアクセスが可能です。詳細については、「[Linux の拡張ネットワーキング \(p. 737\)](#)」および「[Linux 高速コンピューティングインスタンス \(p. 253\)](#)」を参照してください。

すべてのインスタンスタイプが HVM AMI をサポートしています。

HVM AMI を見つけるには、コンソールまたは [describe-images](#) コマンドを使用して、AMI の仮想化タイプが `hvm` に設定されていることを確認します。

#### PV AMI

PV AMI は、PV-GRUB と呼ばれる特別なブートローダーを使用して起動します。このブートローダーによって起動サイクルが開始され、イメージの `menu.lst` ファイルで指定されているカーネルがチェーンコードされます。準仮想化のゲストは仮想化を明示的にサポートしていないホストのハードウェア上で実行されますが、これらのゲストは特別なハードウェア拡張（拡張ネットワーキングや GPU 処理など）を利用できません。従来、PV のゲストは HVM のゲストよりも多くの場合にパフォーマンスが向上しました。ただし、HVM 仮想化の機能強化や HVM AMI で PV ドライバが利用可能になったことにより、このようなパフォーマンスの向上はなくなりました。PV-GRUB の詳細や Amazon EC2 での使用方法については、「[独自の Linux カーネルを有効にする \(p. 176\)](#)」を参照してください。

次の旧世代のインスタンスタイプは、PV AMI をサポートします：C1、C3、HS1、M1、M3、M2、および T1。現行世代のインスタンスタイプは PV AMI をサポートしません。

次の AWS リージョンでは、PV インスタンスをサポートしています：アジアパシフィック（東京）、アジアパシフィック（シンガポール）、アジアパシフィック（シドニー）、欧州（フランクフルト）、欧州（アイルランド）、南米（サンパウロ）、米国東部（バージニア北部）、米国西部（北カリフォルニア）、米国西部（オレゴン）。

PV AMI を見つけるには、コンソールまたは [describe-images](#) コマンドを使用して、AMI の仮想化タイプが `paravirtual` に設定されていることを確認します。

#### PV on HVM

従来、準仮想化のゲストはストレージやネットワークの操作については、HVM のゲストよりも高いパフォーマンスを実現していました。これは、準仮想化のゲストでは I/O 用の特別なドライバー（ネットワークとディスクのハードウェアをエミュレートする際のオーバーヘッドが回避されます）を活用することができたためです。これに対して、HVM のゲストでは、エミュレートされたハードウェアに対する命令を変換する必要がありました。現在では、PV ドライバーを HVM のゲストで利用できるようになりました。このため、準仮想化された環境で実行するためのができないオペレーティングシステムでも、これらのドライバ

イバーを使用することで、ストレージやネットワークの I/O でパフォーマンスの向上を確認することができます。このような PV on HVM ドライバーを使用すると、HVM のゲストで、準仮想化のゲストと同じまたはより優れたパフォーマンスを実現できます。

## Linux AMI の検索

インスタンスを起動する前に、使用する AMI を選択する必要があります。AMI を選択するときに、起動するインスタンスに関して、次の要件を検討します。

- リージョン
- オペレーティングシステム
- アーキテクチャ: 32 ビット (i386)、64 ビット (x86\_64)、または 64 ビット ARM (arm64)
- ルートデバイスタイプ: Amazon EBS またはインスタンスストア
- プロバイダー (Amazon Web Services など)
- 追加のソフトウェア (SQL Server など)

Windows AMI の検索方法については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows AMI を見つける](#)」を参照してください。

### コンテンツ

- [Amazon EC2 コンソールを使用した Linux AMI の検索 \(p. 100\)](#)
- [AWS CLI を使用した AMI の検索 \(p. 101\)](#)
- [Systems Manager を使用して最新の Amazon Linux AMI を検索する \(p. 101\)](#)
- [クイックスタート AMI の検索 \(p. 101\)](#)

## Amazon EC2 コンソールを使用した Linux AMI の検索

Amazon EC2 コンソールを使用して Linux AMI を検索できます。[イメージ] ページを使用してすべての利用可能な AMI を検索できます。また、コンソールを使用してインスタンスを起動する場合は、[クイックスタート] タブでよく使用されている AMI を選択できます。AMI ID は各リージョンに固有です。

[AMI の選択] ページを使用して Linux AMI を検索するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、インスタンスを起動するリージョンを選択します。お客様は場所に関係なく、使用できるリージョンをどれでも選択できます。
3. コンソールダッシュボードで、[Launch Instance] を選択します。
4. [クイックスタート] タブで、よく使用されている AMI のいずれかをリストから選択します。必要な AMI が表示されていない場合は、[AWS Marketplace] または [Community AMIs] タブを選択して他の AMI を検索します。

[Images] ページを使用して Linux AMI を検索するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、インスタンスを起動するリージョンを選択します。お客様は場所に関係なく、使用できるリージョンをどれでも選択できます。
3. ナビゲーションペインで [AMIs] を選択します。
4. (オプション) [Filter] オプションを使用して、一覧表示された AMI を興味のある AMI に限定します。たとえば、AWS で指定されたすべての Linux AMI を表示するには、[Public images] を選択します。検

索バーを選択し、メニューから [Owner]、[Amazon images] の順に選択します。検索バーをもう一度選択し、[Platform] を選択します。次に、表示されたリストからオペレーティングシステムを選択します。

5. (オプション) [Show/Hide Columns] アイコンを選択して、ルートデバイスタイプなど、表示するイメージ属性を選択します。あるいは、一覧から AMI を選択し、[Details] タブにそのプロパティを表示できます。
6. AMI を選択する前に、その AMI が Instance Store-Backed と Amazon EBS-Backed のどちらであるかを確認し、その違いを認識しておくことが重要です。詳細については、「[ルートデバイスのストレージ \(p. 96\)](#)」を参照してください。
7. この AMI からインスタンスを起動するには、インスタンスを選択し、[Launch] を選択します。コンソールを使用したインスタンスの起動の詳細については、「[AMI からのインスタンスの起動 \(p. 450\)](#)」を参照してください。まだインスタンスを起動する準備ができていない場合は、後で使用するために AMI ID を記録します。

## AWS CLI を使用した AMI の検索

Amazon EC2 の AWS CLI コマンドを使用して、ニーズに合った Linux AMI のみが表示されるようにできます。ニーズに合った AMI が見つかったら、インスタンスの起動に使用するためにその ID を記録します。詳細については、『AWS CLI』の「[AWS Command Line Interface ユーザーガイド を使用したインスタンスの起動](#)」を参照してください。

`describe-images` コマンドは、フィルタリングパラメータをサポートしています。たとえば、Amazon が所有するパブリック AMI を表示するのに `--owners` パラメーターを使用します。

```
aws ec2 describe-images --owners self amazon
```

Amazon EBS-backed AMI のみを表示するには、前のコマンドに以下のフィルタを追加できます。

```
--filters "Name=root-device-type,Values=ebs"
```

### Important

`describe-images` コマンドから `--owners` フラグを省略すると、所有権に関係なく、起動許可を持つすべてのイメージが返されます。

## Systems Manager を使用して最新の Amazon Linux AMI を検索する

最新の Amazon Linux AMI の ID を AWS Systems Manager パラメータストアにクエリできます。詳細については、「[AWS Systems Manager パラメータストアを使用して最新の Amazon Linux AMI ID をクエリする](#)」を参照してください。

## クイックスタート AMI の検索

Amazon EC2 コンソールを使用してインスタンスを起動する場合、[Choose an Amazon Machine Image (AMI) (Amazon マシンイメージ (AMI) の選択)] ページの [Quick Start (クイックスタート)] タブには、一般的な AMI の一覧が含まれています。これらのクイックスタート AMI のいずれかを使用してインスタンスの起動を自動化するには、プログラムで現在の AMI バージョンの ID を探す必要があります。

クイックスタート AMI の現在のバージョンを探すには、すべての AMI を AMI 名で列挙し、作成日が最新の AMI を検索します。

Example 例: 現在の Amazon Linux 2 AMI を検索する

```
aws ec2 describe-images --owners amazon --filters 'Name=name,Values=amzn2-  
ami-hvm-2.0.?????????.?-x86_64-gp2' 'Name=state,Values=available' --query  
'reverse(sort_by(Images, &CreationDate))[:1].ImageId' --output text
```

Example 例: 現在の Amazon Linux AMI を検索する

```
aws ec2 describe-images --owners amazon --filters 'Name=name,Values=amzn-  
ami-hvm-????.???.??.????????-x86_64-gp2' 'Name=state,Values=available' --query  
'reverse(sort_by(Images, &CreationDate))[:1].ImageId' --output text
```

Example 例: 現在の Ubuntu Server 16.04 LTS AMI を検索する

```
aws ec2 describe-images --owners 099720109477 --filters 'Name=name,Values=ubuntu/images/  
hvm-ssd/ubuntu-xenial-16.04-amd64-server-????????' 'Name=state,Values=available' --query  
'reverse(sort_by(Images, &CreationDate))[:1].ImageId' --output text
```

Example 例: 現在の Red Hat Enterprise Linux 7.5 AMI を検索する

```
aws ec2 describe-images --owners 309956199498 --filters 'Name=name,Values=RHEL-7.5_HVM_GA*'  
'Name=state,Values=available' --query 'reverse(sort_by(Images, &CreationDate))  
[:1].ImageId' --output text
```

Example 例: 現在の SUSE Linux Enterprise Server 15 AMI を検索する

```
aws ec2 describe-images --owners amazon --filters 'Name=name,Values=suse-sles-15-  
v????????-hvm-ssd-x86_64' 'Name=state,Values=available' --query 'reverse(sort_by(Images,  
&CreationDate))[:1].ImageId' --output text
```

## 共有 AMI

共有 AMI は、開発者が作成し、他の開発者が利用できるようにした AMI です。Amazon EC2 を始める最も簡単な方法は、必要なコンポーネントが含まれている共有 AMI を使用して、カスタムコンテンツを追加することです。独自の AMI を作成し、他のユーザーと共有することもできます。

共有 AMI は、ご自分の判断で使用してください。Amazon は、他の Amazon EC2 ユーザーとの間で共有される AMI の統合性や安全性を保証できません。そのため、共有 AMI を取り扱う際は、ご自分のデータセンターに外部のコードをデプロイすることを検討するのと同じように、充分な注意を払う必要があります。信頼できる開発元の AMI を入手することをお勧めします。

Amazon のパブリックイメージにはエイリアスの所有者が設定されており、アカウントフィールドに `amazon` として表示されます。これを利用すれば、Amazon から AMI を簡単に見つけられます。他のユーザーは、AMI にエイリアスを設定できません。

AMI の作成の詳細については、「[Instance Store-Backed Linux AMI の作成](#)」または「[Amazon EBS-Backed Linux AMI の作成](#)」、「」を参照してください。AWS Marketplace でのアプリケーションの構築、配信、保守の詳細については、「[AWS Marketplace ユーザーガイド](#)」および「[AWS Marketplace Seller Guide](#)」を参照してください。

### コンテンツ

- [共有 AMI を見つける \(p. 103\)](#)
- [AMI を一般公開する \(p. 105\)](#)

- 特定の AWS アカウントと AMI を共有する (p. 106)
- ブックマークの使用 (p. 107)
- 共有 Linux AMI のガイドライン (p. 108)

## 共有 AMI を見つける

Amazon EC2 コンソールまたはコマンドラインを使用して、共有 AMI を検索できます。

### Note

AMI はリージョンのリソースです。したがって、共有 AMI (パブリックまたはプライベート) の検索は、その共有元のリージョン内から実行する必要があります。AMI を他のリージョンで利用できるようにするには、AMI をそのリージョンにコピーし、共有します。詳細については、「[AMI のコピー](#)」を参照してください。

## 共有 AMI を見つける (コンソール)

コンソールを使用して、共有しているプライベート AMI を見つけるには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで [AMIs] を選択します。
- 最初のフィルタで、[Private images] を選択します。お客様が共有しているすべての AMI が一覧表示されます。詳細な検索を行うには、検索バーを選択し、メニューに用意されたフィルタオプションを使用します。

コンソールを使用して、共有しているパブリック AMI を見つけるには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで [AMIs] を選択します。
- 最初のフィルタで、[Public images] を選択します。詳細な検索を行うには、検索バーを選択し、メニューに用意されたフィルタオプションを使用します。
- 興味のある種類の AMI のみを一覧表示するには、フィルタを使用します。たとえば、[Owner:] を選択して、[Amazon images] を選択すると、Amazon のパブリックイメージのみが表示されます。

## 共有 AMI を見つける (AWS CLI)

AMI を一覧表示するには、`-describe-images` コマンド (AWS CLI) を使用します。次の例のように、興味のある種類の AMI に絞って一覧表示できます。

例: すべてのパブリック AMI を一覧表示します。

次のコマンドは、所有しているパブリック AMI を含むすべてのパブリック AMI を一覧表示します。

```
aws ec2 describe-images --executable-users all
```

例: 明示的な起動許可を持つ AMI を一覧表示する

次のコマンドを使用すると、お客様が明示的な起動許可を持つ AMI が一覧表示されます。このリストには、お客様が所有する AMI は含まれていません。

```
aws ec2 describe-images --executable-users self
```

例: Amazon が所有する AMI を一覧表示する

次のコマンドを使用すると、Amazon が所有する AMI が一覧表示されます。Amazon のパブリックイメージにはエイリアスの所有者が設定されており、アカウントフィールドに `amazon` として表示されます。これを利用すれば、Amazon から AMI を簡単に見つけられます。他のユーザーは、AMI にエイリアスを設定できません。

```
aws ec2 describe-images --owners amazon
```

例: アカウントが所有する AMI を一覧表示する

次のコマンドを使用すると、指定した AWS アカウントが所有する AMI が一覧表示されます。

```
aws ec2 describe-images --owners 123456789012
```

例: フィルタを使用してスコープ AMI

表示される AMI の数を減らすには、フィルタを使用して、興味のある種類の AMI に限定して表示します。たとえば、次のフィルタを使用すると、EBS-backed AMI のみが表示されます。

```
--filters "Name=root-device-type,Values=ebs"
```

## 共有 AMI を使用する

共有 AMI を使用する前に、次の手順を実行して、インスタンスへの好ましくないアクセスを許可する認証情報が第三者により事前にインストールされていないことと、機密データを第三者に送信する可能性があるリモートログインが事前設定されていないことを確認します。システムセキュリティ改善についての詳細は、AMI で使用される Linux ディストリビューションの文書を確認してください。

インスタンスへのアクセスを誤って失わないように、SSH セッションを 2 つ開始して、見覚えのない認証情報を削除し、その後も SSH を使用してインスタンスにログインできることが確認されるまで、2 つ目のセッションを開いておくことをお勧めします。

- 未許可のパブリック SSH キーを特定し、無効にします。ファイル内の唯一のキーは、AMI の起動に使用したキーである必要があります。次のコマンドを使用すると、`authorized_keys` ファイルが見つかります。

```
[ec2-user ~]$ sudo find / -name "authorized_keys" -print -exec cat {} \;
```

- ルートユーザーにはパスワードベースの認証を無効にします。`sshd_config` ファイルを開き、次のように `PermitRootLogin` 行を編集します。

```
PermitRootLogin without-password
```

または、ルートユーザーとしてインスタンスにログインする機能を無効にできます。

```
PermitRootLogin No
```

`sshd` サービスを再起動します。

- インスタンスにログインできるユーザー アカウントが他にないか確認します。スーパーユーザー権限を持つアカウントが特に危険です。不明のアカウントがあれば、そのパスワードを削除するか、ロックします。
- 開いていても使用していないポートと、着信接続をリスニングしている実行中のネットワークサービスをチェックします。
- 事前設定されているリモートログインを防ぐには、既存の設定ファイルを削除し、`rsyslog` サービスを再起動してください。例:

```
[ec2-user ~]$ sudo rm /etc/rsyslog.config  
[ec2-user ~]$ sudo service rsyslog restart
```

- すべての cron ジョブが正当であることを確認します。

セキュリティ上のリスクとして考えられるパブリック AMI を発見した場合、AWS セキュリティチームにご連絡ください。詳細については、「[AWS セキュリティセンター](#)」を参照してください。

## AMI を一般公開する

Amazon EC2 では、自分の AMI を他の AWS アカウントと共有できます。すべての AWS アカウントに AMI の起動を許可する (AMI を一般公開する) ことも、特定の少数のアカウントだけに AMI の起動を許可することもできます (「[特定の AWS アカウントと AMI を共有する \(p. 106\)](#)」を参照してください)。お客様の AMI が他の AWS アカウントによって起動されても、お客様に料金は請求されません。AMI を起動するアカウントのみに料金が請求されます。

暗号化されたボリュームを持つ AMI を公開することはできません。

AMI はリージョンのリソースです。そのため、AMI を共有すると、そのリージョンで利用できるようになります。AMI を他のリージョンで利用できるようにするには、AMI をそのリージョンにコピーし、共有します。詳細については、「[AMI のコピー \(p. 155\)](#)」を参照してください。

AMI を共有するときに重要なデータが公開されないようにするには、「[共有 Linux AMI のガイドライン \(p. 108\)](#)」のセキュリティ考慮事項を読み、推奨アクションに従います。

### Note

AMI に製品コードがある場合、または暗号化されたボリュームのスナップショットが含まれている場合は、公開することはできません。特定の AWS アカウントと AMI を共有できます。

## すべての AWS アカウントで AMI を共有する (コンソール)

AMI を公開した後、コンソールを使用して同じリージョン内でインスタンスを起動すると、その AMI が [Community AMIs] に表示されます。AMI は、公開してから [Community AMIs] に表示されるまでに、しばらく時間がかかることがあります。また、AMI を非公開にした場合も、[Community AMIs] から削除されるまでにしばらく時間がかかることがあります。

コンソールを使用してパブリック AMI を共有するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで [AMIs] を選択します。
- リストから AMI を選択し、[Actions] から [Modify Image Permissions] を選択します。
- [Public] を選択し、[Save] を選択します。

## すべての AWS アカウントで AMI を共有する (AWS CLI)

各 AMI には、所有者以外でその AMI を使用してインスタンスを起動できる AWS アカウントを制御する launchPermission プロパティがあります。AMI の launchPermission プロパティを変更することで、AMI を一般公開したり (この場合、すべての AWS アカウントに起動許可が与えられます)、あるいは指定した AWS アカウントとのみ AMI を共有したりできます。

AMI の起動許可を持っているアカウントの一覧に対してアカウント ID の追加または削除ができます。AMI を公開するには、all グループを指定します。パブリック起動許可と明示的起動許可の両方を指定できます。

[To make an AMI public]

1. [modify-image-attribute](#) コマンドを次のように使用して、指定した AMI の launchPermission リストに all グループを追加します。

```
aws ec2 modify-image-attribute --image-id ami-0abcdef1234567890 --launch-permission "Add=[{Group=all}]"
```

2. AMI の起動許可を確認するには、次の [describe-image-attribute](#) コマンドを使用します。

```
aws ec2 describe-image-attribute --image-id ami-0abcdef1234567890 --attribute launchPermission
```

3. (オプション) AMI をプライベートに戻すには、その起動許可から all グループを削除します。AMI の所有者には常に起動許可が与えられるため、このコマンドの影響を受けないことにご注意ください。

```
aws ec2 modify-image-attribute --image-id ami-0abcdef1234567890 --launch-permission "Remove=[{Group=all}]"
```

## 特定の AWS アカウントと AMI を共有する

AMI を一般公開せず、特定の AWS アカウントとだけ共有することもできます。必要なものは AWS アカウント ID のみです。AMI を暗号化されたボリュームと共有する場合、ボリュームの暗号化に使用した CMK も共有する必要があります。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

AMI はリージョンのリソースです。そのため、AMI を共有すると、そのリージョンで利用できるようになります。AMI を他のリージョンで利用できるようにするには、AMI をそのリージョンにコピーし、共有します。詳細については、「[AMI のコピー \(p. 155\)](#)」を参照してください。

AMI を共有できる AWS アカウントの数に制限はありません。

## AMI を共有する (コンソール)

コンソールを使用して明示的な起動許可を与えるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [AMIs] を選択します。
3. リストで AMI を選択し、[Actions] から [Modify Image Permissions] を選択します。
4. [AWS Account Number] フィールドに AMI を共有するユーザーの AWS アカウント番号を指定し、[Add Permission] を選択します。

この AMI を複数のユーザーと共有するには、この手順を繰り返して、必要なすべてのユーザーを追加します。

5. スナップショットのボリューム作成権限を与えるには、[Add "create volume" permissions to the following associated snapshots when creating permissions] を選択します。

### Note

AMI を共有するために、AMI の参照先の Amazon EBS スナップショットを共有する必要はありません。共有する必要があるのは AMI 自体だけです。起動の際に、参照先の Amazon EBS スナップショットへのインスタンスアクセスが自動的に提供されます。ただし、AMI が参照するスナップショットを暗号化するために使用した CMK は共有する必要があります。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

6. 完了したら、[保存] を選択します。

7. (オプション) AMI を共有した AWS アカウント ID を表示するには、リストで AMI を選択し、[アクセス許可] タブを選択します。共有されている AMI を見つけるには、「[共有 AMI を見つける \(p. 103\)](#)」を参照してください。

## AMI を共有する (AWS CLI)

AMI を共有するには、次の例のように `modify-image-attribute` コマンド (AWS CLI) を使用します。

明示的な起動許可を与えるには

次のコマンドを使用すると、指定した AWS アカウントに指定した AMI の起動許可が与えられます。

```
aws ec2 modify-image-attribute --image-id ami-0abcdef1234567890 --launch-permission "Add=[{UserId=123456789012}]"
```

次のコマンドは、スナップショットのボリューム作成権限を付与します。

```
aws ec2 modify-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --attribute createVolumePermission --operation-type add --user-ids 123456789012
```

### Note

AMI を共有するために、AMI の参照先の Amazon EBS スナップショットを共有する必要はありません。共有する必要があるのは AMI 自体だけです。起動の際に、参照先の Amazon EBS スナップショットへのインスタンスアクセスが自動的に提供されます。ただし、AMI が参照するスナップショットを暗号化するために使用した CMK は共有する必要があります。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

アカウントに与えた起動許可を取り消すには

次のコマンドを使用すると、指定した AWS アカウントから指定した AMI の起動許可が削除されます。

```
aws ec2 modify-image-attribute --image-id ami-0abcdef1234567890 --launch-permission "Remove=[{UserId=123456789012}]"
```

次のコマンドは、スナップショットのボリューム作成権限を削除します。

```
aws ec2 modify-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --attribute createVolumePermission --operation-type remove --user-ids 123456789012
```

すべての起動許可を取り消すには

次のコマンドを使用すると、指定した AMI からパブリック起動許可と明示的起動許可がすべて削除されます。AMI の所有者には常に起動許可が与えられるため、このコマンドの影響を受けないようにご注意ください。

```
aws ec2 reset-image-attribute --image-id ami-0abcdef1234567890 --attribute launchPermission
```

## ブックマークの使用

パブリック AMI を作成した場合、あるいは AMI を別の AWS ユーザーと共有した場合は、ブックマークを作成できます。ブックマークを作成すると、ユーザーは自分のアカウントですばやく AMI にアクセスし、インスタンスを起動できます。これにより AMI リファレンスを簡単に共有できるため、時間をかけず、使用する AMI を見つけることができます。

AMI はパブリックであるか、ブックマークの送信先ユーザーと共有している必要があります。

## AMI のブックマークを作成するには

1. 次の情報が含まれる URL を入力します。region には AMI のリージョンを指定します。

```
https://console.aws.amazon.com/ec2/v2/home?  
region=region#LaunchInstanceWizard:ami=ami_id
```

たとえば、この URL は、ami-0abcdef1234567890 リージョンの us-east-1 AMI からインスタンスを起動します。

```
https://console.aws.amazon.com/ec2/v2/home?region=us-  
east-1#LaunchInstanceWizard:ami=ami-0abcdef1234567890
```

2. AMI を使用するユーザーにリンクを配信します。
3. ブックマークを使用するには、リンクを選択するか、そのリンクをコピーしてブラウザに貼り付けます。起動ウィザードが開きます。AMI が既に選択されています。

## 共有 Linux AMI のガイドライン

攻撃対象領域を縮小し、作成する AMI の信頼性を向上させるためには、次のガイドラインを使用します。

### Note

セキュリティのガイドラインのリストは、いずれも完全ではありません。共有 AMI を注意深く作成し、機密データが漏洩される可能性について十分考慮してください。

### トピック

- [使用する前に AMI ツールを更新する \(p. 108\)](#)
- [ルートのパスワードベースのリモートログインを無効にする \(p. 109\)](#)
- [ローカルルートアクセスを無効にする \(p. 109\)](#)
- [SSH ホストキーペアの削除 \(p. 109\)](#)
- [パブリックキー認証情報のインストール \(p. 110\)](#)
- [sshd DNS チェックの無効化 \(オプション\) \(p. 111\)](#)
- [公開元を明らかにする \(p. 111\)](#)
- [自身の保護 \(p. 112\)](#)

AWS Marketplace 向けの AMI を作成する場合、「[Building AMIs for AWS Marketplace](#)」のガイドライン、ポリシー、ベストプラクティスを参照してください。

AMI の安全な共有についての詳細は、次の記事を参照してください。

- [パブリック AMI を安全に共有し使用する方法](#)
- [パブリック AMI の公開: セキュリティ強化とクリーンアップの要件](#)

## 使用する前に AMI ツールを更新する

Instance Store-Backed の AMI の場合、使用する前に、AMI で Amazon EC2 AMI 作成ツールをダウンロードして、アップグレードすることをお勧めします。これにより、共有 AMI に基づく新しい AMI に最新の AMI ツールが与えられます。

Amazon Linux 2 では、aws-amitools-ec2 パッケージをインストールし、次のコマンドで PATH に AMI ツールを追加します。Amazon Linux AMI の場合、デフォルトで aws-amitools-ec2 パッケージが既にインストールされています。

```
[ec2-user ~]$ sudo yum install -y aws-amitools-ec2 && export PATH=$PATH:/opt/aws/bin > /etc/profile.d/aws-amitools-ec2.sh && . /etc/profile.d/aws-amitools-ec2.sh
```

次のコマンドを使用して AMI ツールをアップグレードします。

```
[ec2-user ~]$ sudo yum upgrade -y aws-amitools-ec2
```

他のディストリビューションの場合は、AMI ツールが最新版であることを確認してください。

## ルートのパスワードベースのリモートログインを無効にする

パブリック AMI に固定のルートパスワードを使用することは、セキュリティの面で危険であり、すぐに知られるおそれがあります。初回ログイン後にパスワードを変更するようにユーザーに依存していますが、変更されるまでの一瞬の間にパスワードが悪用される危険性があります。

この問題を解決するには、ルートユーザーのパスワードベースのリモートログインを無効にします。

ルートのパスワードベースのリモートログインを無効にするには

1. テキストエディタで `/etc/ssh/sshd_config` ファイルを開き、次の行を見つけ出します:

```
#PermitRootLogin yes
```

2. 行を次のように変更します:

```
PermitRootLogin without-password
```

この構成ファイルの場所は、ディストリビューションに応じて、または OpenSSH を実行していない場合は、異なることがあります。このような場合は、関連資料を参照してください。

## ローカルルートアクセスを無効にする

共有 AMI を使用する際のベストプラクティスは、直接ルートログインを無効にすることです。これを行うには、実行中のインスタンスにログインし、次のコマンドを発行します。

```
[ec2-user ~]$ sudo passwd -l root
```

### Note

このコマンドが `sudo` の使用に影響を及ぼすことはありません。

## SSH ホストキーペアの削除

パブリック AMI から派生した AMI を共有する場合は、`/etc/ssh` にある既存の SSH ホストキーペアを削除します。これにより、他のユーザーがお客様の AMI を使用してインスタンスを起動したときに、SSH は、新しい固有の SSH キーペアを生成するように強制されるため、セキュリティが強化され、「中間者」攻撃の可能性を減らします。

システムにある次のすべてのキーファイルを削除します。

- `ssh_host_dsa_key`
- `ssh_host_dsa_key.pub`
- `ssh_host_key`
- `ssh_host_key.pub`

- ssh\_host\_rsa\_key
- ssh\_host\_rsa\_key.pub
- ssh\_host\_ecdsa\_key
- ssh\_host\_ecdsa\_key.pub
- ssh\_host\_ed25519\_key
- ssh\_host\_ed25519\_key.pub

次のコマンドを使用して、これらのファイルをすべて確実に削除できます。

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_key.pub
```

#### Warning

**shred** などの安全な削除ユーティリティでは、ストレージメディアからファイルのすべてのコピーを削除できない可能性があります。ファイルの非表示のコピーは、ジャーナルファイルシステム (Amazon Linux のデフォルト ext4 を含む)、スナップショット、バックアップ、RAID、および一時キャッシュによって作成することができます。詳細については、[shred に関するドキュメント](#)を参照してください。

#### Important

パブリック AMI から既存の SSH ホストキーペアを削除することを忘れた場合、ルーチン監査プロセスから、AMI のインスタンスを実行するすべての顧客に向けて、セキュリティ上のリスクがある可能性について通知されます。短い猶予期間の後に、AMI にプライベートのマークが付けられます。

## パブリックキー認証情報のインストール

パスワードを使用したログインを防ぐように AMI を構成したら、ユーザーが別のメカニズムを使用してログインできるようにしておく必要があります。

ユーザーは、Amazon EC2 を使用すると、インスタンスの起動時にパブリックプライベートキーペア名を指定できます。RunInstances API 呼び出し ( またはコマンドライン API ツール ) で有効なキーペア名を指定すると、パブリックキー ( CreateKeyPair または ImportKeyPair の呼び出し後に Amazon EC2 がサーバー上に保持するキーペアの一部 ) を、インスタンスマタデータに対する HTTP Query を介してインスタンスで使用できるようになります。

SSH を使用してログインするには、AMI が起動時にキー値を取得し、それを /root/.ssh/authorized\_keys ( または AMI 上のその他のユーザー アカウントの同等項目 ) に付加する必要があります。ユーザーはキーペアを使用して AMI のインスタンスを起動し、ルートパスワードを入力せずにログインできます。

Amazon Linux や Ubuntu を始めとする多くのディストリビューションでは、cloud-init パッケージを使用して、設定されたユーザーのパブリックキー認証情報を挿入します。cloud-init をサポートしていないディストリビューションの場合は、システムスタートアップスクリプト ( 例: /etc/rc.local ) に次のコードを追加して、起動時にルートユーザーに対して指定したパブリックキーを取り込みます。

#### IMDSv2

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
```

```
if [ $? -eq 0 ] ; then
    cat /tmp/my-key >> /root/.ssh/authorized_keys
    chmod 700 /root/.ssh/authorized_keys
    rm /tmp/my-key
fi
```

#### IMDSv1

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
    cat /tmp/my-key >> /root/.ssh/authorized_keys
    chmod 700 /root/.ssh/authorized_keys
    rm /tmp/my-key
fi
```

この設定は、あらゆるユーザーアカウントに適用できます。root に限定する必要はありません。

#### Note

この AMI に基づいたインスタンスを再バンドルすると、起動時に使用されたキーが組み込まれます。キーへの組み込みを阻止するには、`authorized_keys` ファイルの を空にする (ファイルを削除する) か、このファイルを再バンドルから除外します。

## sshd DNS チェックの無効化 (オプション)

sshd DNS チェックを無効にすると、sshd セキュリティが若干低下します。ただし、DNS の解決策が失敗した場合は、SSH ログインが引き続き機能します。sshd チェックを無効にしなかった場合、DNS の解決策が失敗すると、すべてのログインが阻止されます。

sshd DNS チェックを無効にするには

- テキストエディタで `/etc/ssh/sshd_config` ファイルを開き、次の行を見つけ出します:

```
#UseDNS yes
```

- 行を次のように変更します:

```
UseDNS no
```

#### Note

この構成ファイルの場所は、ディストリビューションに応じて、または OpenSSH を実行していない場合は、異なることがあります。このような場合は、関連資料を参照してください。

## 公開元を明らかにする

現在のところ、AMI はそれぞれアカウント ID で表されるため、共有 AMI を提供したのが誰かを簡単に特定する方法はありません。

お客様の AMI の説明と AMI ID を [Amazon EC2 forum](#) に投稿することをお勧めします。これにより、新しい共有 AMI の使用に関心があるユーザーに便利な中心となる場所が提供されます。

## 自身の保護

共有する AMI に、機密性のあるデータやソフトウェアは保管しないことをお勧めします。共有 AMI を起動するユーザーは、それを再バンドルしたり、自分のものとして登録したりできる可能性があります。以下のガイドラインに従って、見落としやすいセキュリティ上のリスクを回避してください:

- `--exclude directory` で `ec2-bundle-vol` オプションを使用して、バンドル操作に含めたくない機密情報が入っているディレクトリおよびサブディレクトリをスキップすることをお勧めします。特に、イメージをバンドルするときに、すべてのユーザー所有の SSH パブリックキー/プライベートキーペアおよび SSH `authorized_keys` ファイルを除外します。Amazon パブリック AMI で、これらのファイルは、アカウントの場合は `/root/.ssh`、通常のユーザーアカウントの場合は `/home/user_name/.ssh/` に配置されています。詳細については、「[ec2-bundle-vol \(p. 137\)](#)」を参照してください。
- バンドルの前に必ずシェル履歴を削除してください。同じ AMI で複数のバンドルのアップロードを試行すると、シェル履歴にシークレットアクセスキーが含まれます。次の例は、インスタンス内からのバンドルの前に実行される最後のコマンドとなる必要があります。

```
[ec2-user ~]$ shred -u ~/.history
```

### Warning

上記の警告で示した `shred` の制限は、ここにも適用されます。

`bash` は、終了時に現在のセッション履歴をディスクに書き込むことに注意してください。`~/.bash_history` を削除後にインスタンスをログアウトし、再度ログインすると、`~/.bash_history` が再作成され、前のセッション中に実行されたすべてのコマンドが含まれています。

`bash` 以外の他のプログラムもディスクに履歴を書き込むため、注意して不要な dot ファイルと dot ディレクトリを削除または除外します。

- 実行中のインスタンスをバンドルするには、プライベートキーと X.509 証明書が必要です。これらの証明書およびその他の証明書を、バンドルされていない場所（インスタンスストアなど）に書き込みます。

## 有料 AMI

Amazon Kernel Image は、開発者から購入できる 有料 AMI です。

Amazon EC2 は AWS Marketplace と統合されており、開発者は自分が開発した AMI を他の Amazon EC2 ユーザーに有償で提供したり、インスタンスにサポートを提供したりできます。

AWS Marketplace は、EC2 インスタンスの起動に使用できる AMI など、AWS で実行されるソフトウェアを購入できるオンラインストアです。AWS Marketplace AMI は開発者ツールなどカテゴリ別に整理されており、ユーザーは要件に適合する製品を見つけることができます。AWS Marketplace の詳細については、[AWS Marketplace](#) のサイトを参照してください。

有料 AMI からのインスタンスの起動は、他の AMI からのインスタンスの起動と同じです。追加パラメータはありません。インスタンスは、AMI の所有者が設定した料金と、Amazon EC2 で m1.small インスタンスタイプを実行する場合の 1 時間あたりの料金など、関連ウェブサービスの標準使用料に基づいて課金されます。税金が加算されることもあります。有料 AMI の所有者は、特定のインスタンスがその有料 AMI から起動されたかどうかを確認できます。

### Important

Amazon DevPay は新しい販売者または製品の受付を停止しました。いまでは、AWS Marketplace が、ソフトウェアとサービスを AWS で販売する唯一の統一された e コマースプラットフォームです。AWS Marketplace でソフトウェアをデプロイし販売する方法については、「[Selling on AWS Marketplace](#)」を参照してください。AWS Marketplace は Amazon EBS-Backed AMI をサポートします。

## コンテンツ

- [ご自分の AMI を販売する \(p. 113\)](#)
- [有料 AMI を見つける \(p. 113\)](#)
- [有料 AMI の購入 \(p. 114\)](#)
- [インスタンスの製品コードを取得する \(p. 114\)](#)
- [有料サポートの利用 \(p. 115\)](#)
- [有料およびサポート対象の AMI の請求書 \(p. 115\)](#)
- [AWS Marketplace サブスクリプションの管理 \(p. 115\)](#)

## ご自分の AMI を販売する

AWS Marketplace を使用して AMI を販売できます。AWS Marketplace では体系的に買い物をすることができます。また、AWS Marketplace は、Amazon EBS-Backed AMI、リザーブドインスタンス、スポットインスタンスなどの AWS 機能もサポートします。

AWS Marketplace で AMI を販売する方法の詳細については、「[Selling on AWS Marketplace](#)」を参照してください。

## 有料 AMI を見つける

購入できる AMI を検索する方法はいくつかあります。たとえば、[AWS Marketplace](#)、Amazon EC2 コンソール、コマンドラインを使用できます。あるいは、開発者が有料 AMI に関する情報をお客様にお知らせすることができます。

### コンソールを使用して有料 AMI を見つける

コンソールを使用して有料 AMI を見つけるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [AMIs] を選択します。
3. 最初のフィルタで、[パブリックイメージ] を選択します。
4. 検索バーで、[所有者]、[AWS マーケットプレイス] の順に選択します。
5. 製品コードがわかっている場合は、[製品コード]を選択し、製品コードを入力します。

### AWS Marketplace を使用して有料 AMI を見つける

AWS Marketplace を使用して有料 AMI を見つけるには

1. [AWS Marketplace](#) を開く。
2. 検索ボックスにオペレーティングシステムの名前を入力して、[Go] をクリックします。
3. 検索結果をさらに絞るには、カテゴリまたはフィルタを利用します。
4. 各製品には、製品タイプ (AMI または Software as a Service) のラベルが付けられています。

### AWS CLI を使用して有料 AMI を見つける

次の `describe-images` コマンド (AWS CLI) を使用して、有料 AMI を見つけることができます。

```
aws ec2 describe-images --owners aws-marketplace
```

このコマンドは、有料 AMI の製品コードなど、各 AMI を説明するさまざまな詳細を返します。describe-images からの出力には、次のような製品コードのエントリがあります：

```
"ProductCodes": [
  {
    "ProductCodeId": "product_code",
    "ProductCodeType": "marketplace"
  }
],
```

製品コードがわかっている場合は、結果を製品コードでフィルタリングすることができます。次の例は、指定された製品コードを持つ最新の AMI を返します。

```
aws ec2 describe-images --owners aws-marketplace \
--filters "Name=product-code,Values=product_code" \
--query "sort_by(Images, &CreationDate)[-1].[ImageId]"
```

## 有料 AMI の購入

AMI を使用してインスタンスを起動するには、有料 AMI にサインアップする（購入する）必要があります。

通常、有料 AMI の販売者は、価格や購入サイトへのリンクなど、AMI に関する情報を提供します。リンクをクリックすると、最初に AWS へのログインが求められます。ログイン後、AMI を購入できます。

### コンソールを使用して有料 AMI を購入する

Amazon EC2 起動ウィザードを使用して有料 AMI を購入できます。詳細については、「[AWS Marketplace インスタンスの起動 \(p. 467\)](#)」を参照してください。

### AWS Marketplace を使用した製品のサブスクライブ

AWS Marketplace を使用するには、AWS アカウントが必要です。AWS Marketplace 製品からインスタンスを起動するには、Amazon EC2 サービスの利用にサインアップして、インスタンスの起動元から製品の受信登録をする必要があります。AWS Marketplace の製品を受信登録するには、2 つの方法があります。

- AWS Marketplace ウェブサイト: 1-Click デプロイメント機能で、事前に設定したソフトウェアをすばやく起動できます。
- Amazon EC2 起動ウィザード: AMI を検索し、ウィザードからインスタンスを直接起動できます。詳細については、「[AWS Marketplace インスタンスの起動 \(p. 467\)](#)」を参照してください。

## インスタンスの製品コードを取得する

インスタンスの AWS Marketplace 製品コードは、インスタンスマタデータを使用して取得できます。メタデータの取得については、[「インスタンスマタデータとユーザーデータ \(p. 593\)」](#) を参照してください。

製品コードを取得するには、次のコマンドを使用します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/product-codes
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/product-codes
```

インスタンスに製品コードが含まれる場合、Amazon EC2 はそれを返します。

## 有料サポートの利用

Amazon EC2 は、開発者がソフトウェア（またはそれに由来する AMI）のサポートを提供できるように手配します。開発者は、お客様がサインアップして使用できるサポート製品を提供することができます。サポート製品にサインアップすると、開発者はお客様に製品コードを渡します。お客様はそのコードをご自分の AMI に関連付ける必要があります。これにより、開発者は、ユーザーのインスタンスがサポート対象であることを確認できます。また、お客様が製品からインスタンスを実行すると、開発者が定めた製品の利用規約にしたがい、お客様に課金されます。

### Important

リザーブドインスタンスとともにサポート製品を使用することはできません。お客様は常に、サポート製品の販売者が指定した価格を支払います。

製品コードと自分の AMI を関連付けるには、次のコマンドの 1 つを使用します。ami\_id は AMI の ID で、product\_code は製品コードです。

- [modify-image-attribute](#) (AWS CLI)

```
aws ec2 modify-image-attribute --image-id ami_id --product-codes "product_code"
```

- [Edit-EC2ImageAttribute](#) (AWS Tools for Windows PowerShell)

```
PS C:\> Edit-EC2ImageAttribute -ImageId ami_id -ProductCode product_code
```

一度設定した製品コード属性を変更したり削除したりすることはできません。

## 有料およびサポート対象の AMI の請求書

有料またはサポートされた AMI の使用料金がお客様のクレジットカードに請求され、その金額を記載した E メールが毎月末に届きます。これは通常の Amazon EC2 使用料金とは別に請求されます。詳細については、「[Paying For AWS Marketplace Products](#)」を参照してください。

## AWS Marketplace サブスクリプションの管理

AWS Marketplace ウェブサイトでは、サブスクリプションの詳細の確認、使用に関するベンダー指示の表示、サブスクリプションの管理などを行うことができます。

サブスクリプションの詳細を確認するには

1. [AWS Marketplace](#) にログインします。
2. [Your Marketplace Account] を選択します。
3. [Manage your software subscriptions] を選択します。
4. 現在のすべてのサブスクリプションが表示されます。実行中のインスタンスに接続するためのユーザー名など、製品の使用に関する特定の取扱説明を表示するには、[Usage Instructions] を選択します。

### AWS Marketplace のサブスクリプションを取り消すには

1. サブスクリプションによって実行されていたすべてのインスタンスを終了したことを確認します。
  - a. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
  - b. ナビゲーションペインで、[インスタンス] を選択します。
  - c. 該当インスタンスを選択し、[Actions]、[Instance State]、[Terminate] の順に選択します。
  - d. 確認を求めるメッセージが表示されたら、[Yes, Terminate] を選択します。
2. 「[AWS Marketplace](#)」にログインし、[Your Marketplace Account]、[Manage your software subscriptions] の順にクリックします。
3. [Cancel subscription] を選択します。取り消しの確認を求めるプロンプトが表示されます。

#### Note

受信登録をキャンセルすると、その AMI からインスタンスを起動できなくなります。その AMI を再度使用するには、AWS Marketplace ウェブサイトまたは Amazon EC2 コンソールの起動ウィザードを使用して、その AMI を再度サブスクライブする必要があります。

## Amazon EBS-Backed Linux AMI の作成

Amazon EBS-Backed Linux AMI を作成するには、既存の Amazon EBS-Backed Linux AMI から起動したインスタンスから始めます。たとえば、AWS Marketplace から取得した AMI、[AWS Server Migration Service](#) が [VM Import/Export](#) を使用して作成した AMI、またはユーザーがアクセス可能なその他の任意の AMI です。ニーズに合わせてインスタンスをカスタマイズしたら、新しい AMI を作成し、登録します。新しい AMI を使用して、カスタマイズした新しいインスタンスを起動できます。

以下に説明された手順は、暗号化された Amazon EBS ボリューム (ルートボリュームを含む) でバックアップされた Amazon EC2 インスタンスにも、暗号化されていないボリューム同様に機能します。

AMI の作成プロセスは、Instance Store-Backed AMI の場合とは異なります。Amazon EBS-Backed インスタンスと Instance store-Backed インスタンスの違いの詳細と、インスタンスのルートデバイスタイプを判別する方法については、「[ルートデバイスのストレージ \(p. 96\)](#)」を参照してください。instance store-backed Linux AMI の作成に関する詳細については、「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」を参照してください。

Amazon EBS-backed Windows AMI の作成の詳細については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Amazon EBS-backed Windows AMI の作成](#)」を参照してください。

## Amazon EBS-Backed AMI の作成の概要

最初に、作成する AMI と同様の AMI からインスタンスを起動します。インスタンスに接続し、それをカスタマイズできます。インスタンスを正しく設定したら、AMI を作成する前にインスタンスを停止してデータ整合性を確認してから、次にイメージを作成します。作成した Amazon EBS-backed AMI は自動的に登録されます。

Amazon EC2 がインスタンスをシャットダウンしてから AMI を作成するのは、インスタンス上のすべての動作を停止し、作成プロセス中に一貫した状態が保たれるようにするためにです。インスタンスが一貫した状態にあり、適切に AMI を作成できる場合、インスタンスの電源を落として再起動しないよう、Amazon EC2 に指定できます。XFSなどの一部のファイルシステムでは、アクティビティのフリーズおよびフリーズ解除が可能なため、インスタンスを再起動しなくてもイメージを安全に作成できます。

AMI 作成プロセスの間、Amazon EC2 はインスタンスのルートボリュームとインスタンスにアタッチされている他の EBS ボリュームのスナップショットを作成します。AMI の登録を解除してスナップショットを削除するまで、スナップショットは課金の対象となります。詳細については、「[Linux AMI の登録解](#)

除 (p. 163)」を参照してください。インスタンスにアタッチされるいずれかのボリュームが暗号化されている場合、新しい AMI は、Amazon EBS 暗号化をサポートするインスタンスでのみ正常に起動します。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

ボリュームのサイズによっては、AMI 作成プロセスの完了に数分かかる場合があります(最長で 24 時間かかることがあります)。AMI を作成する前に、ボリュームのスナップショットを作成しておくと、効率が向上する可能性があります。この方法では、AMI を作成する際に作成する必要があるのは小さい差分スナップショットのみになるため、プロセスがよりすばやく完了します(スナップショット作成の合計時間は同じです)。詳細については、「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」を参照してください。

プロセスが完了すると、新しい AMI と、インスタンスのルートボリュームから作成されたスナップショットが与えられます。ユーザーが新しい AMI を使用してインスタンスを起動すると、Amazon はスナップショットを使用して、そのルートボリュームのために新しい EBS ボリュームを作成します。

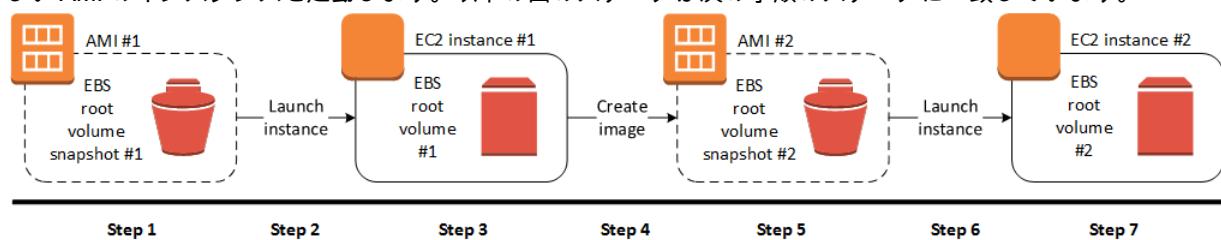
ルートデバイスボリュームに加えて、インスタンストアボリュームまたは EBS ボリュームをインスタンスに追加した場合、新しい AMI のブロックデバイスマッピングにこれらのボリュームの情報が含まれ、新しい AMI から起動するインスタンスのブロックデバイスマッピングに自動的にこれらのボリュームの情報が含まれます。新しいインスタンスのブロックデバイスマッピングに指定されているインスタンストアボリュームは新しく、AMI の作成に使用したインスタンスのインスタンストアボリュームからのデータは含まれていません。EBS ボリュームのデータは永続的です。詳細については、「[ブロックデバイスマッピング \(p. 1100\)](#)」を参照してください。

#### Note

EBS-backed AMI から新しいインスタンスを作成する場合、本稼働環境に移す前にそのルートボリュームと追加 EBS ストレージの両方を初期化する必要があります。詳細については、「[Amazon EBS ボリュームの初期化](#)」を参照してください。

## インスタンスからの Linux AMI の作成

AMI は、AWS マネジメントコンソール またはコマンドラインを使用して作成できます。次の図は、実行中の EC2 インスタンスから Amazon EBS-backed AMI を作成するプロセスをまとめたものです。既存の AMI から開始して、インスタンスを起動してカスタマイズし、そこから新しい AMI を作成して、最後に新しい AMI のインスタンスを起動します。以下の図のステップは次の手順のステップに一致しています。



### コンソールを使用してインスタンスから AMI を作成するには

- 新しい AMI の開始点として機能する適切な EBS-backed AMI を選択し、起動に先立って必要に応じて設定します。詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。
- [起動] を選択して、選択した EBS-backed AMI のインスタンスを起動します。デフォルト値をそのまま使ってウィザードを完了します。詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。
- インスタンスの実行中に、それに接続します。必要に応じてインスタンスで次のアクションを実行してインスタンスをカスタマイズできます。
  - ソフトウェアやアプリケーションをインストールする
  - データをコピーする

- 起動時間を短縮するために一時ファイルの消去、ハードディスクのデフラグ、占有領域の開放処理を行う。
  - 追加の Amazon EBS ボリュームをアタッチする。
4. (オプション) インスタンスにアタッチされているすべてのボリュームのスナップショットを作成する。スナップショット作成についての詳細は、[Amazon EBS スナップショットの作成 \(p. 972\)](#) を参照してください。
5. ナビゲーションペインで、[Instances] を選択し、インスタンスを選択して、[Actions]、[Image]、[Create Image] の順に選択します。

Tip

このオプションが無効になっている場合、そのインスタンスは Amazon EBS-Backed インスタンスではありません。

6. [Create Image] ダイアログボックスで、以下の情報を指定し、[Create Image] を選択します。
- イメージ名 – イメージの一意の名前。
  - イメージの説明 – イメージの説明 (オプション) (最大 255 文字)。
  - 再起動しない – このオプションはデフォルトでは選択されていません。Amazon EC2 はインスタンスをシャットダウンし、アタッチされていたすべてのボリュームのスナップショットを作成し、AMI を作成および登録して、インスタンスを再起動します。インスタンスをシャットダウンしない場合は、[再起動しない] を選択します。

Warning

[再起動しない] を選択した場合は、作成されたイメージのファイルシステムの整合性は保証されません。

- インスタンスピリューム – このセクションのフィールドでは、ルートボリュームを変更し、Amazon EBS およびインスタンスストアボリュームを追加できます。各フィールドについては、各フィールドの横の i アイコンで一時停止するとフィールドのヒントが表示されます。いくつかの重要な点を以下に示します。
    - ルートボリュームのサイズを変更するには、[Volume Type] 列で [Root] を見つけ、[Size (GiB)] に目的的の値を入力します。
    - [終了時に削除] を選択した場合、この AMI から作成されたインスタンスを終了すると、EBS ボリュームが削除されます。[終了時に削除] をオフにした場合は、インスタンスを終了しても、EBS ボリュームは削除されません。詳細については、「[インスタンスの削除で Amazon EBS ボリュームを保持する \(p. 549\)](#)」を参照してください。
    - Amazon EBS ボリュームを追加するには、[新しいボリュームの追加] を選択します (新しい行が追加されます)。[Volume Type] で [EBS] を選択し、行のフィールドに入力します。作成した AMI からインスタンスを起動すると、追加したボリュームは自動的にそのインスタンスにアタッチされます。空のボリュームはフォーマットしてマウントする必要があります。スナップショットベースのボリュームはマウントする必要があります。
    - インスタンスストアボリュームを追加するには、「[AMI にインスタンスストアボリュームを追加する \(p. 1084\)](#)」を参照してください。その後新しい AMI からインスタンスを起動すると、追加されたボリュームは自動的に初期化されてマウントされます。これらのボリュームには、AMI の作成に使用された実行中のインスタンスのインスタンスストアボリュームのデータは含まれません。
7. 作成中に AMI のステータスを表示するには、ナビゲーションペインで [AMI] を選択します。最初は、ステータスは pending ですが、数分後 available に変わります。

(オプション) 新しい AMI に作成されたスナップショットを表示するには、[Snapshots (スナップショット)] を選択します。ユーザーがこの AMI からインスタンスを起動すると、Amazon はこのスナップショットを使用して、ルートデバイスピリュームを作成します。

8. 新しい AMI からインスタンスを起動します。詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。
9. 新しい実行中インスタンスには、前のステップで適用したカスタム設定がすべて含まれます。

## コマンドラインを使用してインスタンスから AMI を作成するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [create-image \(AWS CLI\)](#)
- [New-EC2Image \(AWS Tools for Windows PowerShell\)](#)

## スナップショットからの Linux AMI の作成

インスタンスのルートデバイスピリュームのスナップショットがある場合、AWS マネジメントコンソールまたはコマンドラインを使用して、そのスナップショットから AMI を作成できます。

コンソールを使用してスナップショットから AMI を作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic Block Store] の [Snapshots] を選択します。
3. スナップショットを選択し、[Actions]、[Create Image] を選択します。
4. [Create Image from EBS Snapshot] ダイアログボックスで、AMI を作成するためのフィールドに入力し、[Create] を選択します。親インスタンスを再作成する場合は、親インスタンスと同じオプションを選択します。
  - Architecture: 32 ビットの場合は [i386] を、64 ビットの場合は [x86\_64] を選択します。
  - Root device name: ルートボリュームの適切な名前を入力します。詳細については、「[Linux インスタンスでのデバイスの名前付け \(p. 1098\)](#)」を参照してください。
  - 仮想化タイプ: この AMI から起動されるインスタンスで準仮想化 (PV) またはハードウェア仮想マシン (HVM) のいずれの仮想化を使用するかを選択します。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。
  - (PV 仮想化タイプのみ) Kernel ID および RAM disk ID: リストから AKI と ARI を選択します。デフォルトの AKI を選択するか、AKI を選択しない場合、この AMI を使用してインスタンスを起動するたびに AKI を指定する必要があります。また、デフォルトの AKI にインスタンスとの互換性がない場合、インスタンスのヘルスチェックが失敗する可能性があります。
  - (オプション) Block Device Mappings: ボリュームを追加するか、AMI のルートボリュームのデフォルト容量を増やします。ボリュームの容量を増やした場合のインスタンスのファイルシステムのサイズ変更の詳細については、「[ボリュームサイズ変更後の Linux ファイルシステムの拡張 \(p. 1011\)](#)」を参照してください。

コマンドラインを使用してスナップショットから AMI を作成するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [register-image \(AWS CLI\)](#)
- [Register-EC2Image \(AWS Tools for Windows PowerShell\)](#)

## Instance Store-Backed Linux AMI の作成

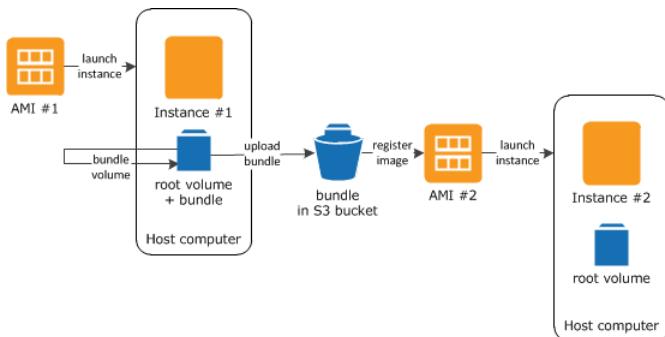
Instance Store-Backed Linux AMI を作成するには、既存の Instance Store-Backed Linux AMI から起動したインスタンスから始めます。ニーズに合わせてインスタンスをカスタマイズしたら、ボリュームをバンド

ルし、新しい AMI を登録します。新しい AMI を使用して、カスタマイズした新しいインスタンスを起動できます。

AMI の作成プロセスは、Amazon EBS-backed AMI の場合とは異なります。Amazon EBS-Backed インスタンスと Instance store-Backed インスタンスの違いの詳細と、インスタンスのルートデバイスタイプを判別する方法については、「[ルートデバイスのストレージ \(p. 96\)](#)」を参照してください。Amazon EBS-backed Linux AMI を作成する必要がある場合は、「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」を参照してください。

## Instance Store-Backed AMI の作成プロセスの概要

次の図は、Instance Store-Backed インスタンスから AMI を作成するプロセスをまとめたものです。



最初に、作成する AMI に似ている AMI からインスタンスを起動します。インスタンスに接続し、それをカスタマイズできます。インスタンスのカスタマイズが終わったら、それをバンドルできます。バンドルプロセスが完了するには数分間かかります。プロセスが完了すると、イメージマニフェスト (`image.manifest.xml`) とルートボリューム用のテンプレートを含むファイル (`image.part.xx`) で構成されるバンドルが作成されます。次に、バンドルを Amazon S3 バケットにアップロードし、AMI を登録します。

お客様が新しい AMI を使用してインスタンスを起動すると、Amazon はユーザーが Amazon S3 にアップロードしたバンドルを使用してインスタンスのルートボリュームを作成します。Amazon S3 のバンドルで使用されるストレージ領域については、お客様がその領域を削除するまでアカウントに料金が発生します。詳細については、「[Linux AMI の登録解除 \(p. 163\)](#)」を参照してください。

ルートデバイスピリュームに加えて、インスタンストアボリュームをインスタンスに追加した場合、新しい AMI のブロックデバイスマッピングにこれらのボリュームの情報が含まれ、新しい AMI から起動するインスタンスのブロックデバイスマッピングに自動的にこれらのボリュームの情報が含まれます。詳細については、「[ブロックデバイスマッピング \(p. 1100\)](#)」を参照してください。

## 前提条件

AMI を作成するには、最初に次のタスクを完了する必要があります。

- AMI ツールをインストールします。詳細については、「[AMI ツールを設定する \(p. 121\)](#)」を参照してください。
- AWS CLI をインストールします。詳細については、「[AWS Command Line Interface のセットアップ](#)」を参照してください。
- バンドルに Amazon S3 バケットがあることを確認します。Amazon S3 バケットを作成するには、Amazon S3 コンソールを開き、[Create Bucket (バケットの作成)] をクリックします。または、AWS CLI の `mb` コマンドを使用できます。
- AWS アカウント ID があることを確認します。詳細については、『AWS General Reference』の「[AWS アカウント ID](#)」を参照してください。

- アクセスキー ID とシークレットアクセスキーがあることを確認します。詳細については、『AWS General Reference』の「[アクセスキー](#)」を参照してください。
- X.509 証明書および対応するプライベートキーがあることを確認します。
  - X.509 証明書を作成する必要がある場合は、「[デジタル署名用証明書の管理 \(p. 123\)](#)」を参照してください。X.509 証明書とプライベートキーは、AMI の暗号化/復号に使用されます。
  - [中国 (北京)] \$EC2\_AMITOOL\_HOME/etc/ec2/amitools/cert-ec2-cn-north-1.pem 証明書を使用します。
  - [AWS GovCloud (US-West)] \$EC2\_AMITOOL\_HOME/etc/ec2/amitools/cert-ec2-gov.pem 証明書を使用します。
- インスタンスに接続し、カスタマイズします。たとえば、ソフトウェアとアプリケーションをインストールしたり、データをコピーしたり、一時ファイルを削除したり、Linux 設定を変更したりできます。

## タスク

- [AMI ツールを設定する \(p. 121\)](#)
- [Instance Store-Backed Amazon Linux インスタンスから AMI を作成する \(p. 124\)](#)
- [Instance Store-Backed Ubuntu インスタンスから AMI を作成する \(p. 127\)](#)
- [Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換する \(p. 131\)](#)

# AMI ツールを設定する

AMI ツールを使用して、Instance Store-Backed Linux AMI を作成および管理できます。ツールを使用するには、Linux インスタンスにインストールする必要があります。AMI ツールは RPM として使用できるとともに、RPM をサポートしていない Linux ディストリビューションでは .zip ファイルとして使用できます。

## RPM を使用して AMI ツールを設定するには

1. yum などの Linux ディストリビューション用のパッケージマネージャを使用して Ruby をインストールします。例:

```
[ec2-user ~]$ sudo yum install -y ruby
```

2. wget や curl などのツールを使用して RPM ファイルをダウンロードします。例:

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.noarch.rpm
```

3. 次のコマンドを使用して RPM ファイルの署名を確認する:

```
[ec2-user ~]$ rpm -K ec2-ami-tools.noarch.rpm
```

上のコマンドは、ファイルの SHA1 および MD5 ハッシュが OK. であることを示しています。ハッシュが NOT OK であることをコマンドが示している場合、次のコマンドを使用してファイルのヘッダー SHA1 および MD5 ハッシュを表示します。

```
[ec2-user ~]$ rpm -Kv ec2-ami-tools.noarch.rpm
```

次に、ファイルのヘッダー SHA1 および MD5 ハッシュを、以下の検証済み AMI ツールハッシュと比較し、ファイルの正統性を確認します。

- ヘッダー SHA1: a1f662d6f25f69871104e6a62187fa4df508f880

- MD5: 9faff05258064e2f7909b66142de6782

ファイルのヘッダー SHA1 および MD5 ハッシュが検証済み AMI ツールハッシュと一致する場合、次のステップに進みます。

4. 次のコマンドを使用して RPM をインストールします。

```
[ec2-user ~]$ sudo yum install ec2-ami-tools.noarch.rpm
```

5. [ec2-ami-tools-version \(p. 135\)](#) コマンドを使用してインストールした AMI ツールを検証します。

```
[ec2-user ~]$ ec2-ami-tools-version
```

#### Note

[cannot load such file -- ec2/amitools/version (LoadError)] などのロードエラーを受信した場合は、次のステップを実行し、AMI ツールをインストールした場所を RUBYLIB パスに追加します。

6. (オプション) 前のステップでエラーが発生した場合、AMI ツールをインストールした場所を RUBYLIB パスに追加します。

- a. 追加するパスを調べるには、次のコマンドを実行します。

```
[ec2-user ~]$ rpm -qil ec2-ami-tools | grep ec2/amitools/version
/usr/lib/ruby/site_ruby/ec2/amitools/version.rb
/usr/lib64/ruby/site_ruby/ec2/amitools/version.rb
```

上記の例では、以前のロードエラーから失われたファイルは /usr/lib/ruby/site\_ruby および /usr/lib64/ruby/site\_ruby にあります。

- b. 前のステップの場所を RUBYLIB パスに追加します。

```
[ec2-user ~]$ export RUBYLIB=$RUBYLIB:/usr/lib/ruby/site_ruby:/usr/lib64/ruby/
site_ruby
```

- c. [ec2-ami-tools-version \(p. 135\)](#) コマンドを使用してインストールした AMI ツールを検証します。

```
[ec2-user ~]$ ec2-ami-tools-version
```

#### zip ファイルを使用して AMI ツールを設定するには

1. Ruby をインストールし、apt-get など、Linux ディストリビューション用のパッケージマネージャを使用して解凍します。例:

```
[ec2-user ~]$ sudo apt-get update -y && sudo apt-get install -y ruby unzip
```

2. wget や curl などのツールを使用して .zip ファイルをダウンロードします。例:

```
[ec2-user ~]$ wget https://s3.amazonaws.com/ec2-downloads/ec2-ami-tools.zip
```

3. /usr/local/ec2 など、適切なインストールディレクトリにファイルを解凍します。

```
[ec2-user ~]$ sudo mkdir -p /usr/local/ec2
$ sudo unzip ec2-ami-tools.zip -d /usr/local/ec2
```

.zip ファイルには、フォルダ (ec2-ami-tools-**x.x.x**) が含まれます。ここで、**x.x.x** はツールのバージョン番号 (例: ec2-ami-tools-1.5.7) です。

- EC2\_AMITOOL\_HOME 環境変数を、ツールのインストールディレクトリに設定します。以下に例を示します。

```
[ec2-user ~]$ export EC2_AMITOOL_HOME=/usr/local/ec2/ec2-ami-tools-x.x.x
```

- ツールを PATH 環境変数に追加します。以下に例を示します。

```
[ec2-user ~]$ export PATH=$EC2_AMITOOL_HOME/bin:$PATH
```

- ec2-ami-tools-version (p. 135) コマンドを使用してインストールした AMI ツールを検証できます。

```
[ec2-user ~]$ ec2-ami-tools-version
```

## デジタル署名用証明書の管理

AMI ツールの特定のコマンドでは、デジタル署名用証明書 (X.509 証明書とも呼ばれる) が必要です。証明書を作成し、AWS にアップロードする必要があります。たとえば、証明書の作成に OpenSSL などのサードパーティ製のツールを使用できます。

デジタル署名用証明書を作成するには

- OpenSSL をインストールおよび設定します。
- プライベートキーを openssl genrsa コマンドを使用して作成し、出力を .pem ファイルで保存します。2048 ビットまたは 4096 ビット RSA キーの作成を推奨しています。

```
openssl genrsa 2048 > private-key.pem
```

- openssl req コマンドを使用して、証明書を作成します。

```
openssl req -new -x509 -nodes -sha256 -days 365 -key private-key.pem -outform PEM -out certificate.pem
```

証明書を AWS にアップロードするには、upload-signing-certificate コマンドを使用します。

```
aws iam upload-signing-certificate --user-name user-name --certificate-body file://path/to/certificate.pem
```

ユーザーの証明書を一覧表示するには、list-signing-certificates コマンドを使用します。

```
aws iam list-signing-certificates --user-name user-name
```

ユーザーのデジタル署名用証明書を無効化または再有効化するには、update-signing-certificate コマンドを使用します。次のコマンドは証明書を無効にします。

```
aws iam update-signing-certificate --certificate-id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE --status Inactive --user-name user-name
```

証明書を削除するには、delete-signing-certificate コマンドを使用します。

```
aws iam delete-signing-certificate --user-name user-name --certificate-id OFHPLP4ZULTHYPMSYEX7O4BEXAMPLE
```

## Instance Store-Backed インスタンスから AMI を作成する

次の手順では、instance store-backed インスタンスから instance store-backed AMI を作成します。開始する前に、必ず「[前提条件 \(p. 120\)](#)」を参照してください。

### トピック

- [Instance Store-Backed Amazon Linux インスタンスから AMI を作成する \(p. 124\)](#)
- [Instance Store-Backed Ubuntu インスタンスから AMI を作成する \(p. 127\)](#)

## Instance Store-Backed Amazon Linux インスタンスから AMI を作成する

このセクションでは、Amazon Linux インスタンスからの AMI の作成について説明します。以下の手順は、他の Linux ディストリビューションを実行するインスタンスでは機能しない可能性があります。Ubuntu 固有の手順については、「[Instance Store-Backed Ubuntu インスタンスから AMI を作成する \(p. 127\)](#)」を参照してください。

AMI ツールの使用準備を整えるには (HVM インスタンスのみ)

1. AMI ツールでは、GRUB のレガシーが正常に起動する必要があります。次のコマンドを使用して GRUB をインストールします。

```
[ec2-user ~]$ sudo yum install -y grub
```

2. 次のコマンドを使用して、パーティション管理パッケージをインストールします。

```
[ec2-user ~]$ sudo yum install -y gdisk kpartx parted
```

### Instance Store-Backed Amazon Linux インスタンスから AMI を作成するには

この手順では、「[前提条件 \(p. 120\)](#)」に記載された前提条件が満たされていることを前提としています。

1. インスタンスに認証情報をアップロードします。Amazon ではこれらの認証情報を使用して、お客様と Amazon EC2 だけがお客様の AMI にアクセスできるようにします。
  - a. 次のように、認証情報のための一時ディレクトリをインスタンスに作成します。

```
[ec2-user ~]$ mkdir /tmp/cert
```

それにより、作成したイメージから認証情報を除外できます。

- b. [scp \(p. 509\)](#) などの安全なコピーツールを使用して、コンピュータからインスタンスの /tmp/cert ディレクトリに X.509 証明書と対応するプライベートキーをコピーします。次の scp コマンドの -i **my-private-key.pem** オプションは、X.509 プライベートキーではなく、SSH でインスタンスに接続するために使用するプライベートキーです。例:

```
you@your_computer:~ $ scp -i my-private-key.pem /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /
```

```
path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717      0.7KB/s  00:00
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 685      0.7KB/s  00:00
```

または、これらがプレーンテキストファイルの場合、証明書とキーをテキストエディタで開き、コンテンツを /tmp/cert の新しいファイルにコピーできます。

2. インスタンス内から [ec2-bundle-vol \(p. 137\)](#) コマンドを実行して、Amazon S3 にアップロードするバンドルを準備します。-e オプションを指定して、認証情報を保存するディレクトリを除外します。デフォルトでは、バンドルプロセスで機密情報を含んでいる可能性があるファイルを除外します。ファイルには、\*.sw、\*.swo、\*.swp、\*.pem、\*.priv、\*id\_rsa\*、\*id\_dsa\*、\*.gpg、\*.jks、\*./ssh/authorized\_keys、\*./bash\_history などがあります。これらのファイルをすべて含めるには、--no-filter オプションを使用します。これらのファイルの一部を含めるには、--include オプションを使用します。

#### Important

AMI バンドルプロセスは、デフォルトで、ルートボリュームを表す /tmp ディレクトリに、圧縮され暗号化された一連のファイルを作成します。バンドルを格納するのに十分な空きディスク領域が /tmp にない場合、-d [/path/to/bundle/storage](#) オプションを使用して、バンドルを格納する別の場所を指定する必要があります。インスタンスによっては、工夫メモリストレージが /mnt または /media/ephemeral0 にマウントされて使用可能になっている場合があります。または、バンドルを格納する新しい Amazon EBS ボリュームを [作成 \(p. 949\)](#)、[アタッチ \(p. 952\)](#)、および [マウント \(p. 956\)](#)することもできます。

- a. ec2-bundle-vol コマンドは、root として実行する必要があります。ほとんどのコマンドで、sudo を使用することでアクセス許可を昇格させることができますが、この場合は、環境変数を維持するために sudo -E su を実行する必要があります。

```
[ec2-user ~]$ sudo -E su
```

これで、bash プロンプトにより root ユーザーとして識別されるようになったことと、root シェルにいることを示すハッシュタグにドル記号が置き換えられたことに注意してください。

```
[root ec2-user]#
```

- b. AMI のバンドルを作成するには、次のように [ec2-bundle-vol \(p. 137\)](#) コマンドを実行します。

```
[root ec2-user]# ec2-bundle-vol -k /tmp/cert/pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 123456789012 -r x86_64 -e /tmp/cert --
partition gpt
```

#### Note

中国 (北京) および AWS GovCloud (US-West) リージョンについては、--ec2cert パラメータを使用し、[前提条件 \(p. 120\)](#) に従って証明書を指定します。

イメージの作成には数分かかります。このコマンドが完了したら、/tmp(またはデフォルト以外の) ディレクトリにバンドルが含まれます (image.manifest.xml、および複数の image.part.**xx** ファイル)。

- c. root シェルを終了します。

```
[root ec2-user]# exit
```

3. (オプション) インスタンスストアをさらに追加するには、AMI 用の `image.manifest.xml` ファイルで、ブロックデバイスマッピングを編集します。詳細については、「[ブロックデバイスマッピング \(p. 1100\)](#)」を参照してください。

- a. `image.manifest.xml` ファイルのバックアップを作成します。

```
[ec2-user ~]$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

- b. `image.manifest.xml` ファイルの形式を変更し、読み取りと編集が簡単になるようにします。

```
[ec2-user ~]$ sudo xmllint --format /tmp/image.manifest.xml.bak > sudo /tmp/image.manifest.xml
```

- c. テキストエディタで `image.manifest.xml` のブロックデバイスマッピングを編集します。次の例は、`ephemeral1` インスタンスストアボリュームの新しいエントリを示しています。

Note

無効な種類のファイルの一覧については、「[ec2-bundle-vol \(p. 137\)](#)」を参照してください。

```
<block_device_mapping>
  <mapping>
    <virtual>ami</virtual>
    <device>sda</device>
  </mapping>
  <mapping>
    <virtual>ephemeral0</virtual>
    <device>sdb</device>
  </mapping>
  <mapping>
    <virtual>ephemeral1</virtual>
    <device>sdc</device>
  </mapping>
  <mapping>
    <virtual>root</virtual>
    <device>/dev/sda1</device>
  </mapping>
</block_device_mapping>
```

- d. `image.manifest.xml` ファイルを保存し、テキストエディタを終了します。

4. バンドルを Amazon S3 にアップロードするには、次のように [ec2-upload-bundle \(p. 148\)](#) コマンドを実行します。

```
[ec2-user ~]$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name -m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

Important

米国東部 (バージニア北部) 以外のリージョンで AMI を登録するには、`--region` オプションと、すでにターゲットリージョンに存在するバケットパス、またはターゲットリージョンで作成できる一意のバケットパスの両方でターゲットリージョンを指定する必要があります。

5. (オプション) バンドルを Amazon S3 にアップロードしたら、次の `rm` コマンドを使用して、インスタンスの `/tmp` ディレクトリからバンドルを削除できます。

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

Important

`-d /path/to/bundle/storage` で Step 2 (p. 125) オプションを使用してパスを指定した場合は、`/tmp` ではなくそのパスを使用します。

- AMI を登録するには、次のように `register-image` コマンドを実行します。

```
[ec2-user ~]$ aws ec2 register-image --image-location my-s3-
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --virtualization-
type hvm
```

Important

`ec2-upload-bundle` (p. 148) コマンドでリージョンを以前に指定した場合は、このコマンドでもう一度そのリージョンを指定します。

## Instance Store-Backed Ubuntu インスタンスから AMI を作成する

このセクションでは、インスタンストアボリュームをルートボリュームとして使用する Ubuntu Linux インスタンスからの AMI の作成について説明します。以下の手順は、他の Linux ディストリビューションを実行するインスタンスでは機能しない可能性があります。Amazon Linux に固有の手順については、「Instance Store-Backed Amazon Linux インスタンスから AMI を作成する (p. 124)」を参照してください。

AMI ツールの使用準備を整えるには (HVM インスタンスのみ)

AMI ツールでは、GRUB のレガシーが正常に起動する必要があります。ただし、Ubuntu は GRUB 2 を使用するように設定されています。インスタンスで GRUB のレガシーを使用しているかどうか確認し、使用していない場合はインストールして設定する必要があります。

AMI ツールが正常に機能するためには、HVM インスタンスにパーティションツールがインストールされている必要があります。

- GRUB Legacy (バージョン 0.9x 未満) をインスタンスにインストールする必要があります。GRUB Legacy が存在していることを確認し、必要な場合はインストールしてください。
  - GRUB インストールのバージョンを確認します。

```
ubuntu:~$ grub-install --version
grub-install (GRUB) 1.99-21ubuntu3.10
```

この例では、GRUB バージョンが 0.9x 以上のため、GRUB Legacy をインストールする必要があります。Step 1.b (p. 127) に進みます。GRUB Legacy が既にある場合、「Step 2 (p. 127)」までスキップできます。

- 次のコマンドを使用して grub パッケージをインストールします。

```
ubuntu:~$ sudo apt-get install -y grub
```

- お使いのディストリビューションのパッケージマネージャを使用して、次のパーティション管理パッケージをインストールします。
  - `gdisk` (ディストリビューションによっては代わりにパッケージ `gptfdisk` が呼び出される場合があります)。
  - `kpartx`

- parted

次のコマンドを使用します。

```
ubuntu:-$ sudo apt-get install -y gdisk kpartx parted
```

- インスタンスのカーネルパラメータを確認します。

```
ubuntu:-$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.2.0-54-virtual root=UUID=4f392932-ed93-4f8f-
aee7-72bc5bb6ca9d ro console=ttyS0 xen_emul_unplug=unnecessary
```

カーネルおよびルートデバイスのパラメータ `ro`、`console=ttyS0`、および `xen_emul_unplug=unnecessary` を書き留めます。オプションは異なる場合があります。

- /boot/grub/menu.lst でカーネルエントリを確認してください。

```
ubuntu:-$ grep ^kernel /boot/grub/menu.lst
kernel  /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro console=hvc0
kernel  /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
kernel  /boot/memtest86+.bin
```

`console` パラメータが `hvc0` をポイントしている (`ttyS0` ではない) こと、および `xen_emul_unplug=unnecessary` パラメータが未指定であることに注意してください。ここでも、オプションは異なる場合があります。

- /boot/grub/menu.lst ファイルを任意のテキストエディタで (vim や nano など) で編集して、コンソールを変更し、先ほど確認したパラメータをブートエントリに追加します。

```
title      Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual
root      (hd0)
kernel    /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs
          ro console=ttyS0 xen_emul_unplug=unnecessary
initrd    /boot/initrd.img-3.2.0-54-virtual

title      Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual (recovery mode)
root      (hd0)
kernel    /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro
          single console=ttyS0 xen_emul_unplug=unnecessary
initrd    /boot/initrd.img-3.2.0-54-virtual

title      Ubuntu 12.04.3 LTS, memtest86+
root      (hd0)
kernel    /boot/memtest86+.bin
```

- カーネルエントリに適切なパラメータが含まれていることを確認します。

```
ubuntu:-$ grep ^kernel /boot/grub/menu.lst
kernel  /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro console=ttyS0
          xen_emul_unplug=unnecessary
kernel  /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single
          console=ttyS0 xen_emul_unplug=unnecessary
kernel  /boot/memtest86+.bin
```

- (Ubuntu 14.04 以降のみ) Ubuntu 14.04 で起動する Instance Store-Backed Ubuntu AMI は GPT のパーティションテーブルおよび /boot/efi にマウントされた別の EFI のパーティションを使用します。ec2-bundle-vol コマンドはこの起動パーティションをバンドルしません。そのため、次の例に示すように EFI のパーティションの /etc/fstab エントリをコメントアウトする必要があります。

```
LABEL=cloudimg-rootfs / ext4 defaults 0 0
#LABEL=UEFI /boot/efi vfat defaults 0 0
/dev/xvdb /mnt auto defaults,nobootwait,comment=cloudconfig 0 2
```

## Instance Store-Backed Ubuntu インスタンスから AMI を作成するには

この手順では、「[前提条件 \(p. 120\)](#)」に記載された前提条件が満たされていることを前提としています。

1. インスタンスに認証情報をアップロードします。Amazon ではこれらの認証情報を使用して、お客様と Amazon EC2 だけがお客様の AMI にアクセスできるようにします。
  - a. 次のように、認証情報のための一時ディレクトリをインスタンスに作成します。

```
ubuntu:~$ mkdir /tmp/cert
```

それにより、作成したイメージから認証情報を除外できます。

- b. [scp \(p. 509\)](#) などの安全なコピーツールを使用して、コンピュータからインスタンスの /tmp/cert ディレクトリに X.509 証明書とプライベートキーをコピーします。次の scp コマンドの -i *my-private-key.pem* オプションは、X.509 プライベートキーではなく、SSH でインスタンスに接続するために使用するプライベートキーです。例:

```
you@your_computer:~ $ scp -i my-private-key.pem /
path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /
path/to/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-
user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 685 0.7KB/s 00:00
```

または、これらがプレーンテキストファイルの場合、証明書とキーをテキストエディタで開き、コンテンツを /tmp/cert の新しいファイルにコピーできます。

2. インスタンスから [ec2-bundle-vol \(p. 137\)](#) コマンドを実行して、Amazon S3 にアップロードするバンドルを準備します。-e オプションを指定して、認証情報を保存するディレクトリを除外します。デフォルトでは、バンドルプロセスで機密情報を含んでいる可能性があるファイルを除外します。ファイルには、\*.sw, \*.swo, \*.swp, \*.pem, \*.priv, \*id\_rsa\*, \*id\_dsa\*, \*.gpg, \*.jks, \*/.ssh/authorized\_keys、\*/.bash\_history などがあります。これらのファイルをすべて含めるには、--no-filter オプションを使用します。これらのファイルの一部を含めるには、--include オプションを使用します。

### Important

AMI バンドルプロセスは、デフォルトで、ルートボリュームを表す /tmp ディレクトリに、圧縮され暗号化された一連のファイルを作成します。バンドルを格納するのに十分な空きディスク領域が /tmp にない場合、-d */path/to/bundle/storage* オプションを使用して、バンドルを格納する別の場所を指定する必要があります。インスタンスによっては、エフェメラルストレージが /mnt または /media/ephemeral0 にマウントされて使用可能になっている場合があります。または、バンドルを格納する新しい Amazon EBS ボリュームを作成 (p. 949)、アタッチ (p. 952)、およびマウント (p. 956) することもできます。

- a. ec2-bundle-vol コマンドは、root として実行する必要があります。ほとんどのコマンドで、sudo を使用することでアクセス許可を昇格させることができますが、この場合は、環境変数を維持するため sudo -E su を実行する必要があります。

```
ubuntu:~$ sudo -E su
```

これで、bash プロンプトにより root ユーザーとして識別されるようになったことと、root シェルにいることを示すハッシュタグにドル記号が置き換えられたことに注意してください。

```
root@ubuntu:#
```

- b. AMI のバンドルを作成するには、次のように [ec2-bundle-vol \(p. 137\)](#) コマンドを実行します。

```
root@ubuntu:# ec2-bundle-vol -k /tmp/cert/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
-c /tmp/cert/cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r
x86_64 -e /tmp/cert --partition gpt
```

#### Important

Ubuntu 14.04 以降の HVM インスタンスの場合、`--partition mbr` フラグを追加して起動手順を正しくバンドルします。それ以外の場合は、新しく作成された AMI は起動しません。

イメージの作成には数分かかります。このコマンドが完了したら、tmp ディレクトリにバンドルが含まれます (`image.manifest.xml`、および複数の `image.part.xx` ファイル)。

- c. root シェルを終了します。

```
root@ubuntu:# exit
```

3. (オプション) インスタンスストアをさらに追加するには、AMI 用の `image.manifest.xml` ファイルで、ブロックデバイスマッピングを編集します。詳細については、「[ブロックデバイスマッピング \(p. 1100\)](#)」を参照してください。

- a. `image.manifest.xml` ファイルのバックアップを作成します。

```
ubuntu:~$ sudo cp /tmp/image.manifest.xml /tmp/image.manifest.xml.bak
```

- b. `image.manifest.xml` ファイルの形式を変更し、読み取りと編集が簡単になるようにします。

```
ubuntu:~$ sudo xmllint --format /tmp/image.manifest.xml.bak > /tmp/
image.manifest.xml
```

- c. テキストエディタで `image.manifest.xml` のブロックデバイスマッピングを編集します。次の例は、`ephemeral1` インスタンスストアボリュームの新しいエントリを示しています。

```
<block_device_mapping>
  <mapping>
    <virtual>ami</virtual>
    <device>sda</device>
  </mapping>
  <mapping>
    <virtual>ephemeral0</virtual>
    <device>sdb</device>
  </mapping>
  <mapping>
    <virtual>ephemeral1</virtual>
    <device>sdc</device>
  </mapping>
  <mapping>
    <virtual>root</virtual>
    <device>/dev/sda1</device>
```

```
</mapping>  
</block_device_mapping>
```

- d. `image.manifest.xml` ファイルを保存し、テキストエディタを終了します。
4. バンドルを Amazon S3 にアップロードするには、次のように [ec2-upload-bundle \(p. 148\)](#) コマンドを実行します。

```
ubuntu:~$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name -m /tmp/  
image.manifest.xml -a your_access_key_id -s your_secret_access_key
```

**Important**

米国東部（バージニア北部）以外のリージョンで AMI を登録する予定の場合、`--region` オプションと、すでにターゲットリージョンに存在するバケットパス、またはターゲットリージョンで作成できる一意のバケットパスの両方でターゲットリージョンを指定する必要があります。

5. (オプション) バンドルを Amazon S3 にアップロードしたら、次の `rm` コマンドを使用して、インスタンスの `/tmp` ディレクトリからバンドルを削除できます。

```
ubuntu:~$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

**Important**

`-d /path/to/bundle/storage` で [Step 2 \(p. 129\)](#) オプションを使用してパスを指定した場合、`/tmp` ではなく以下と同じパスを使用します。

6. AMI を登録するには、次のように AWS CLI の `register-image` コマンドを実行します。

```
ubuntu:~$ aws ec2 register-image --image-location my-s3-  
bucket/bundle_folder/bundle_name/image.manifest.xml --name AMI_name --virtualization-  
type hvm
```

**Important**

`ec2-upload-bundle (p. 148)` コマンドでリージョンを以前に指定した場合は、このコマンドでもう一度そのリージョンを指定します。

7. (Ubuntu 14.04 以降) `/etc/fstab` の EFI エントリをコメント解除します。それ以外の場合、実行中のインスタンスは再起動できません。

## Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換する

Instance Store-Backed Linux AMI は、Amazon EBS-Backed Linux AMI に変換できます。

**Important**

Instance Store-Backed Windows AMI から Amazon EBS-Backed Windows AMI への変換、および所有していない AMI の変換はできません。

Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換するには

1. Amazon EBS-Backed AMI から Amazon Linux インスタンスを起動します。詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。Amazon Linux インスタンスには、AWS CLI と AMI ツールがプリインストールされています。

2. Instance Store-Backed AMI をバンドルするのに使用した X.509 プライベートキーをインスタンスにアップロードします。Amazon はこのキーを使用して、お客様と Amazon EC2 だけがお客様の AMI にアクセスできるようにします。

- a. 次のように、X.509 プライベートキーのインスタンスに一時ディレクトリを作成します。

```
[ec2-user ~]$ mkdir /tmp/cert
```

- b. [scp \(p. 509\)](#) などの安全なコピーツールを使用して、コンピュータから /tmp/cert ディレクトリに X.509 プライベートキーをコピーします。次のコマンドの **my-private-key** パラメータは、SSH でインスタンスに接続するために使用するプライベートキーです。例:

```
you@your_computer:~ $ scp -i my-private-key.pem /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem ec2-user@ec2-203-0-113-25.compute-1.amazonaws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00
```

3. AWS アクセスキーおよび秘密キーの環境変数を設定します。

```
[ec2-user ~]$ export AWS_ACCESS_KEY_ID=your_access_key_id
[ec2-user ~]$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

4. 新しい AMI の Amazon EBS ボリュームを準備します。

- a. [create-volume](#) コマンドを使用して、インスタンスと同じアベイラビリティゾーンに空の Amazon EBS ボリュームを作成します。コマンド出力のボリューム ID を書き留めてください。

**Important**

この Amazon EBS ボリュームは、元のインスタンストアのルートボリュームと同じサイズ以上である必要があります。

```
[ec2-user ~]$ aws ec2 create-volume --size 10 --region us-west-2 --availability-zone us-west-2b
```

- b. [attach-volume](#) コマンドを使用して、Amazon EBS-Backed インスタンスにボリュームをアタッチします。

```
[ec2-user ~]$ aws ec2 attach-volume --volume-id volume_id --instance-id instance_id --device /dev/sdb --region us-west-2
```

5. バンドルのフォルダを作成します。

```
[ec2-user ~]$ mkdir /tmp/bundle
```

6. /tmp/bundle コマンドを使用して、Instance Store-Backed AMI のバンドルを [ec2-download-bundle \(p. 143\)](#) にダウンロードします。

```
[ec2-user ~]$ ec2-download-bundle -b my-s3-bucket/bundle_folder/bundle_name -m image.manifest.xml -a $AWS_ACCESS_KEY_ID -s $AWS_SECRET_ACCESS_KEY --privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. [ec2-unbundle \(p. 147\)](#) コマンドを使用して、バンドルからイメージファイルを再作成します。

- a. バンドルフォルダにディレクトリを変更します。

```
[ec2-user ~]$ cd /tmp/bundle/
```

- b. [ec2-unbundle \(p. 147\)](#) コマンドを実行します。

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey /path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. バンドルを解除したイメージから新しい Amazon EBS ボリュームにファイルをコピーします。

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

9. バンドルを解除した新しいパーティションのボリュームを調査します。

```
[ec2-user bundle]$ sudo partprobe /dev/sdb1
```

10. ブロックデバイスの一覧を表示してマウントするデバイス名を選択します。

```
[ec2-user bundle]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
/dev/sda    202:0    0   8G  0 disk 
##/dev/sda1 202:1    0   8G  0 part /
/dev/sdb    202:80   0  10G  0 disk 
##/dev/sdb1 202:81   0  10G  0 part
```

この例では、マウントするパーティションは /dev/sdb1 ですが、デバイス名はおそらく異なります。ボリュームが仕切られていない場合は、マウントするデバイスは /dev/sdb に似ています（デバイスパーティションの末尾に数値なし）。

11. 新しい Amazon EBS ボリュームのマウントポイントを作成し、ボリュームをマウントします。

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs
[ec2-user bundle]$ sudo mount /dev/sdb1 /mnt/ebs
```

12. EBS ボリュームの /etc/fstab ファイルを任意のテキストエディタ (vim や nano など) で開き、インスタンスストア (エフェメラル) ボリュームのエントリがあれば削除します。Amazon EBS ボリュームが /mnt/ebs に取付けられるため、fstab ファイルは /mnt/ebs/etc/fstab にあります。

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/      /          ext4      defaults,noatime  1   1
tmpfs        /dev/shm   tmpfs     defaults          0   0
devpts       /dev/pts   devpts    gid=5,mode=620  0   0
sysfs        /sys       sysfs    defaults          0   0
proc         /proc      proc     defaults          0   0
/dev/sdb     /media/ephemeral0 auto      defaults,comment=cloudconfig  0
2
```

この例では、最後の行を削除する必要があります。

13. ボリュームをアンマウントし、インスタンスからデタッチします。

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ aws ec2 detach-volume --volume-id volume_id --region us-west-2
```

14. 次のように、新しい Amazon EBS ボリュームから AMI を作成します。

- 新しい Amazon EBS ボリュームのスナップショットを作成します。

```
[ec2-user bundle]$ aws ec2 create-snapshot --region us-west-2 --description "your_snapshot_description" --volume-id volume_id
```

- スナップショットが完了していることを確認します。

```
[ec2-user bundle]$ aws ec2 describe-snapshots --region us-west-2 --snapshot-id snapshot_id
```

- c. 元の AMI で使用されたプロセッサーアーキテクチャ、仮想化タイプ、カーネルイメージ (aki) を、describe-images コマンドを使用して特定します。このステップでは、元の Instance Store-Backed AMI の AMI ID が必要です。

```
[ec2-user bundle]$ aws ec2 describe-images --region us-west-2 --image-id ami_id --output text
IMAGES x86_64 amazon/amzn-ami-pv-2013.09.2.x86_64-s3 ami-8ef297be amazon available
public machine aki-fc8f11cc instance-store paravirtual xen
```

この例では、アーキテクチャは x86\_64 で、カーネルイメージ ID は aki-fc8f11cc です。次のステップでこれらの値を使用します。前述のコマンドの出力では ari ID もリストされますので、これも書き留めます。

- d. 新しい Amazon EBS ボリュームのスナップショット ID と前のステップで書き留めた値を使用して、新しい AMI を登録します。前述のコマンド出力に ari ID がリストされていた場合は、その ID を次のコマンドで --ramdisk-id **ari\_id** を使用して指定します。

```
[ec2-user bundle]$ aws ec2 register-image --region us-west-2 --name your_new_ami_name --block-device-mappings DeviceName=device-name,Ebs={SnapshotId=snapshot_id} --virtualization-type paravirtual --architecture x86_64 --kernel-id aki-fc8f11cc --root-device-name device-name
```

15. (オプション) 新しい AMI からインスタンスを起動できることをテストした後で、この手順で作成した Amazon EBS ボリュームを削除できます。

```
aws ec2 delete-volume --volume-id volume_id
```

## AMI ツールリファレンス

AMI ツールコマンドを使用して、Instance Store-Backed Linux AMI を作成および管理できます。ツールをセットアップする方法は、「[AMI ツールを設定する \(p. 121\)](#)」を参照してください。

アクセスキーの詳細については、「[AWS アクセスキーを管理するためのベストプラクティス](#)」を参照してください。

### コマンド

- [ec2-ami-tools-version \(p. 135\)](#)
- [ec2-bundle-image \(p. 135\)](#)
- [ec2-bundle-vol \(p. 137\)](#)
- [ec2-delete-bundle \(p. 141\)](#)
- [ec2-download-bundle \(p. 143\)](#)
- [ec2-migrate-manifest \(p. 145\)](#)
- [ec2-unbundle \(p. 147\)](#)
- [ec2-upload-bundle \(p. 148\)](#)
- [AMI ツール用の一般的なオプション \(p. 150\)](#)

## ec2-ami-tools-version

### 説明

AMI ツールのバージョンについて説明します。

### 構文

**ec2-ami-tools-version**

### 出力

バージョン情報。

### 例

このコマンド例では、使用中の AMI ツールのバージョン情報を表示します。

```
[ec2-user ~]$ ec2-ami-tools-version
1.5.2 20071010
```

## ec2-bundle-image

### 説明

ループバックファイル内に作成されるオペレーティングシステムイメージから instance store-backed Linux AMI を作成します。

### 構文

```
ec2-bundle-image -c path -k path -u account -i path [-d path] [--ec2cert path]
[-r architecture] [--productcodes code1,code2,...] [-B mapping] [-p prefix]
```

### オプション

**-c, --cert** パス

ユーザーの PEM エンコード RSA パブリックキー証明書ファイル。

必須: はい

**-k, --privatekey** パス

PEM エンコードされる RSA キーファイルへのパス。このバンドルをバンドル解除するには、このキーを指定する必要があるため、安全な場所に保管してください。このキーは AWS アカウントに登録されている必要があります。

必須: はい

**-u, --user** アカウント

ダッシュのない、ユーザーの AWS アカウント ID。

必須: はい

**-i, --image** パス

バンドルするイメージへのパス。

必須: はい

**-d, --destination** パス

バンドルを作成するディレクトリ。

デフォルト: /tmp

必須: いいえ

--ec2cert パス

イメージマニフェストの暗号化に使用される Amazon EC2 X.509 パブリックキー証明書へのパス。

us-gov-west-1 および cn-north-1 リージョンではデフォルト以外のパブリックキー証明書を使用し、その証明書へのパスは、このオプションで指定する必要があります。証明書へのパスは、AMI ツールのインストール方法によって異なります。Amazon Linux の場合、証明書の場所は /opt/aws/amitools/ec2/etc/ec2/amitools/ です。「[AMI ツールを設定する \(p. 121\)](#)」の RPM または ZIP ファイルから AMI ツールをインストールした場合、証明書の場所は \$EC2\_AMITOOL\_HOME/etc/ec2/amitools/ です。

必須: us-gov-west-1 および cn-north-1 リージョンのみ。

-r, --arch アーキテクチャ

イメージアーキテクチャ。コマンドラインでアーキテクチャを指定しない場合、バンドルの開始時に入力を求められます。

有効な値: i386 | x86\_64

必須: いいえ

--productcodes code1、code2、...

登録時にイメージにアタッチする、カンマ区切りの製品コード。

必須: いいえ

-B, --block-device-mapping マッピング

インスタンスタイプが指定されたデバイスをサポートする場合に、この AMI のインスタンスにプロックデバイスを公開する方法を定義します。

キーと値のペアのカンマ区切りのペアを指定します。名キーは仮想名であり、各値は対応するデバイス名です。仮想名には以下が含まれています。

- ami — インスタンスによって判断されるルートファイルシステムデバイス
- root — カーネルによって判断されるルートファイルシステムデバイス
- swap — インスタンスによって判断されるスワップデバイス
- ephemeralN — N 番目のインスタンスストアボリューム

必須: いいえ

-p, --prefix プレフィックス

バンドル済み AMI ファイルのファイル名プレフィックス。

デフォルト: イメージファイルの名前。たとえば、イメージパスが /var/spool/my-image/version-2/debian.img である場合、デフォルトのプレフィックスは debian.img です。

必須: いいえ

--kernel kernel\_id

廃止.カーネルを設定するには、[register-image](#) を使用します。

必須: いいえ

--ramdisk ramdisk\_id

廃止.必要に応じて、[register-image](#) を使用して RAM ディスクを設定します。

必須: いいえ

## 出力

バンドルプロセスのステージとステータスを記述するステータスマッセージ。

### 例

この例は、ループバックファイルで作成されたオペレーティングシステムイメージから、バンドルされた AMI を作成します。

```
[ec2-user ~]$ ec2-bundle-image -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -i image.img -d bundled/ -r x86_64
Please specify a value for arch [i386]:
Bundling image file...
Splitting bundled/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
Created image.part.04
Created image.part.05
Created image.part.06
Created image.part.07
Created image.part.08
Created image.part.09
Created image.part.10
Created image.part.11
Created image.part.12
Created image.part.13
Created image.part.14
Generating digests for each part...
Digests generated.
Creating bundle manifest...
ec2-bundle-image complete.
```

## ec2-bundle-vol

### Description

インスタンスのルートデバイスピリュームを圧縮、暗号化、署名することで、instance store-backed Linux AMI を作成します。

Amazon EC2 はインスタンスから製品コード、カーネル設定、RAM ディスク設定、およびブロックデバイスマッピングを継承しようとします。

デフォルトでは、バンドルプロセスで機密情報を含んでいる可能性があるファイルを除外します。ファイルに

は、\*.sw、\*.swo、\*.swp、\*.pem、\*.priv、\*id\_rsa\*、\*id\_dsa\*、\*.gpg、\*.jks、\*./ssh/authorized\_keys、\*./bash\_history などがあります。これらのファイルをすべて含めるには、--no-filter オプションを使用します。これらのファイルの一部を含めるには、--include オプションを使用します。

詳細については、「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」を参照してください。

### 構文

```
ec2-bundle-vol -c path -k path -u account [-d path] [--ec2cert path] [-r architecture] [--productcodes code1,code2,...] [-B mapping] [--all] [-e directory1,directory2,...] [-i file1,file2,...] [--no-filter] [-p prefix] [-s size] [--[no-]inherit] [-v volume] [-P type] [-S script] [--fstab path] [--generate-fstab] [--grub-config path]
```

## オプション

**-c, --cert** パス

ユーザーの PEM エンコード RSA パブリックキー証明書ファイル。

必須: はい

**-k, --privatekey** パス

ユーザーの PEM エンコード RSA キーファイルへのパス。

必須: はい

**-u, --user** アカウント

ダッシュのない、ユーザーの AWS アカウント ID。

必須: はい

**-d, --destination** 送信先

バンドルを作成するディレクトリ。

デフォルト: /tmp

必須: いいえ

**--ec2cert** パス

イメージマニフェストの暗号化に使用される Amazon EC2 X.509 パブリックキー証明書へのパス。

us-gov-west-1 および cn-north-1 リージョンではデフォルト以外のパブリックキー証明書を使用し、その証明書へのパスは、このオプションで指定する必要があります。証明書へのパスは、AMI ツールのインストール方法によって異なります。Amazon Linux の場合、証明書の場所は /opt/aws/amitools/ec2/etc/ec2/amitools/ です。「[AMI ツールを設定する \(p. 121\)](#)」の RPM または ZIP ファイルから AMI ツールをインストールした場合、証明書の場所は \$EC2\_AMITOOL\_HOME/etc/ec2/amitools/ です。

必須: us-gov-west-1 および cn-north-1 リージョンのみ。

**-r, --arch** アーキテクチャ

イメージアーキテクチャ。コマンドラインでこれを指定しない場合、バンドルの開始時に入力を求められます。

有効な値: i386 | x86\_64

必須: いいえ

**--productcodes** code1、code2、...

登録時にイメージにアタッチする、カンマ区切りの製品コード。

必須: いいえ

**-B, --block-device-mapping** マッピング

インスタンスタイプが指定されたデバイスをサポートする場合に、この AMI のインスタンスにブロックデバイスを公開する方法を定義します。

キーと値のペアのカンマ区切りのペアを指定します。名キーは仮想名であり、各値は対応するデバイス名です。仮想名には以下が含まれています。

- **ami** — インスタンスによって判断されるルートファイルシステムデバイス
- **root** — カーネルによって判断されるルートファイルシステムデバイス
- **swap** — インスタンスによって判断されるスワップデバイス
- **ephemeralN** — N 番目のインスタンストアボリューム

必須: いいえ

-a, --all

リモートでマウントされたファイルシステムのディレクトリを含めて、すべてのディレクトリをバンドルします。

必須: いいえ

-e, --exclude directory1, directory2, ...

バンドルオペレーションから除外する絶対ディレクトリパスとファイルのリスト。このパラメータは --all オプションを上書きします。除外を指定すると、パラメータとともにリストされたディレクトリとサブディレクトリは、ボリュームにバンドルされません。

必須: いいえ

-i, --include file1, file2, ...

バンドルオペレーションに含めるファイルのリスト。指定されたファイルは、それ以外の場合は AMI から除外されます。これは、機密情報が含まれる可能性があるためです。

必須: いいえ

--no-filter

指定した場合、AMI からファイルは除外されません。これは、機密情報が含まれる可能性があるためです。

必須: いいえ

-p, --prefix プレフィックス

バンドル済み AMI ファイルのファイル名プレフィックス。

デフォルト: image

必須: いいえ

-s, --size サイズ

作成するイメージファイルの MB (1024 \* 1024 バイト) 単位のサイズ。最大サイズは 10240 MB です。

デフォルト: 10240

必須: いいえ

--[no-]inherit

イメージがインスタンスのメタデータを継承するかどうかを示します (デフォルトでは継承します)。--inherit を有効にし、インスタンスマタデータにアクセスできない場合、バンドルは失敗します。

必須: いいえ

-v, --volume ボリューム

バンドルを作成する、マウントされたボリュームへの絶対パス。

デフォルト: ルートディレクトリ (/)。

必須: いいえ

-P, --partition タイプ

ディスクイメージでパーティションテーブルを使用するかどうかを示します。パーティションテーブルタイプを指定しない場合、デフォルトでは、該当する場合はボリュームの親ブロックデバイスで使用されるタイプになります。それ以外の場合、デフォルトは gpt です。

有効な値: mbr | gpt | none

必須: いいえ

-S, --script スクリプト

バンドルの直前に実行するカスタマイズスクリプト。スクリプトでは単一の引数である、ボリュームのマウントポイントが予期されます。

必須: いいえ

--fstab パス

イメージにバンドルする fstab へのパス。これを指定しない場合、Amazon EC2 は /etc/fstab をバンドルします。

必須: いいえ

--generate-fstab

Amazon EC2 で提供される fstab を使用してボリュームをバンドルします。

必須: いいえ

--grub-config

イメージにバンドルする別の grub 設定ファイルへのパス。デフォルトでは、ec2-bundle-vol は /boot/grub/menu.lst または /boot/grub/grub.conf が、クローンされたイメージ上に存在することを想定します。このオプションにより、別の grub 設定ファイルへのパスを指定することができ、このファイルはデフォルトに上書きしてコピーされます (存在する場合)。

必須: いいえ

--kernel kernel\_id

廃止. カーネルを設定するには、[register-image](#) を使用します。

必須: いいえ

--ramdiskramdisk\_id

廃止. 必要に応じて、[register-image](#) を使用して RAM ディスクを設定します。

必須: いいえ

## 出力

バンドルのステージとステータスを説明するステータスマッセージ。

## 例

この例では、ローカルマシンのルートファイルシステムのスナップショットを圧縮、暗号化、署名することで、バンドルされた AMI を作成します。

```
[ec2-user ~]$ ec2-bundle-vol -d /mnt -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -u 111122223333 -r x86_64
Copying / into the image file /mnt/image...
Excluding:
  sys
  dev/shm
  proc
  dev/pts
  proc/sys/fs/binfmt_misc
  dev
  media
  mnt
```

```
proc
sys
tmp/image
mnt/img-mnt
1+0 records in
1+0 records out
mke2fs 1.38 (30-Jun-2005)
warning: 256 blocks unused.

Splitting /mnt/image.gz.crypt...
Created image.part.00
Created image.part.01
Created image.part.02
Created image.part.03
...
Created image.part.22
Created image.part.23
Generating digests for each part...
Digests generated.
Creating bundle manifest...
Bundle Volume complete.
```

## ec2-delete-bundle

### Description

Amazon S3 ストレージから、指定されたバンドルを削除します。バンドルを削除した後で、対応する AMI からインスタンスを起動することはできません。

### 構文

```
ec2-delete-bundle -b bucket -a access_key_id -s secret_access_key [-t token]
[--url url] [--region region] [--sigv version] [-m path] [-p prefix] [--clear]
[--retry] [-y]
```

### オプション

-b, --bucket バケット

バンドルされた AMI に続いてオプションの '/' 区切りパスプレフィックスを含む Amazon S3 バケットの名前

必須: はい

-a, --access-key *access\_key\_id*

AWS アクセスキー ID。

必須: はい

-s, --secret-key *secret\_access\_key*

AWS シークレットアクセスキー。

必須: はい

-t, --delegation-token トークン

AWS リクエストに渡す委任トークン。詳細については、「[一時的なセキュリティ認証情報の使用](#)」を参照してください。

必須: 一時的なセキュリティ認証情報を使用している場合のみ。

デフォルト: AWS\_DELEGATION\_TOKEN 環境変数の値 (設定されている場合)。

--region リージョン

リクエスト署名で使用するリージョン。

デフォルト: us-east-1

必須: 署名バージョン 4 を使用する場合は必須

--sigv バージョン

リクエストに署名するときに使用する署名バージョン。

有効な値: 2 | 4

デフォルト: 4

必須: いいえ

-m, --manifest パス

マニフェストファイルへのパス。

必須: --prefix または --manifest のどちらかを指定する必要があります。

-p, --prefix プレフィックス

バンドルされた AMI ファイル名プレフィックス。プレフィックス全体を指定します。たとえば、プレフィックスが image.img である場合は、-p image.img ではなく -p image を使用します。

必須: --prefix または --manifest のどちらかを指定する必要があります。

--clear

指定されたバンドルを削除した後で空の場合は、Amazon S3 バケットを削除します。

必須: いいえ

--retry

すべての Amazon S3 エラーで、オペレーションあたり最大 5 回まで自動的に再試行します。

必須: いいえ

-y, --yes

すべてのプロンプトへの答えが [yes] であると自動的に想定します。

必須: いいえ

## 出力

Amazon EC2 は、削除プロセスのステージとステータスを示すステータスマッセージを表示します。

## 例

この例では、Amazon S3 からバンドルを削除します。

```
[ec2-user ~]$ ec2-delete-bundle -b aws-s3-bucket1 -a your_access_key_id -s your_secret_access_key
Deleting files:
aws-s3-bucket1/
image.manifest.xml
aws-s3-bucket1/
image.part.00
aws-s3-bucket1/
image.part.01
```

```
aws-s3-bucket1/
image.part.02
aws-s3-bucket1/
image.part.03
aws-s3-bucket1/
image.part.04
aws-s3-bucket1/
image.part.05
aws-s3-bucket1/image.part.06
Continue? [y/n]
y
Deleted aws-s3-bucket1/image.manifest.xml
Deleted aws-s3-bucket1/image.part.00
Deleted aws-s3-bucket1/image.part.01
Deleted aws-s3-bucket1/image.part.02
Deleted aws-s3-bucket1/image.part.03
Deleted aws-s3-bucket1/image.part.04
Deleted aws-s3-bucket1/image.part.05
Deleted aws-s3-bucket1/image.part.06
ec2-delete-bundle complete.
```

## ec2-download-bundle

### 説明

指定された instance store-backed Linux AMI を Amazon S3 ストレージからダウンロードします。

### 構文

```
ec2-download-bundle -b bucket -a access_key_id -s secret_access_key -k path
[--url url] [--region region] [--sigv version] [-m file] [-p prefix] [-d
directory] [--retry]
```

### オプション

-b, --bucket /バケット

バンドルが存在する Amazon S3 バケットの名前。この後に、オプションで '/' 区切りのパスプレフィックスが続きます。

必須: はい

-a, --access-key *access\_key\_id*

AWS アクセスキー ID。

必須: はい

-s, --secret-key *secret\_access\_key*

AWS シークレットアクセスキー。

必須: はい

-k, --privatekey *パス*

マニフェストの復号に使用されるプライベートキー。

必須: はい

--url *url*

Amazon S3 サービスの URL。

デフォルト: <https://s3.amazonaws.com/>

必須: いいえ  
--region リージョン  
リクエスト署名で使用するリージョン。  
デフォルト: us-east-1  
必須: 署名バージョン 4 を使用する場合は必須  
--sigv バージョン  
リクエストに署名するときに使用する署名バージョン。  
有効な値: 2 | 4  
デフォルト: 4  
必須: いいえ  
-m, --manifest ファイル  
マニフェストファイル名(パスなし)。マニフェスト (-m) またはプレフィックス (-p) を指定することをお勧めします。  
必須: いいえ  
-p, --prefix プレフィックス  
バンドル済み AMI ファイルのファイル名プレフィックス。  
デフォルト: image  
必須: いいえ  
-d, --directory ディレクトリ  
ダウンロードしたバンドルが保存されているディレクトリ。ディレクトリが存在している必要があります。  
デフォルト: 現在の作業ディレクトリ。  
必須: いいえ  
--retry  
すべての Amazon S3 エラーで、オペレーションあたり最大 5 回まで自動的に再試行します。  
必須: いいえ

## 出力

ダウンロードプロセスの多様な段階ステータスを示すメッセージが表示されます。

## 例

この例では、bundled ディレクトリを作成(Linux mkdir コマンドを使用)し、Amazon S3 `aws-s3-bucket1` バケットからバンドルをダウンロードします。

```
[ec2-user ~]$ mkdir bundled
[ec2-user ~]$ ec2-download-bundle -b aws-s3-bucket1/bundles/bundle_name -m
image.manifest.xml -a your_access_key_id -s your_secret_access_key -k pk-
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d mybundle
Downloading manifest image.manifest.xml from aws-s3-bucket1 to mybundle/
image.manifest.xml ...
Downloading part image.part.00 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.00 ...
```

```
Downloaded image.part.00 from aws-s3-bucket1
Downloading part image.part.01 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.01 ...
Downloaded image.part.01 from aws-s3-bucket1
Downloading part image.part.02 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.02 ...
Downloaded image.part.02 from aws-s3-bucket1
Downloading part image.part.03 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.03 ...
Downloaded image.part.03 from aws-s3-bucket1
Downloading part image.part.04 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.04 ...
Downloaded image.part.04 from aws-s3-bucket1
Downloading part image.part.05 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.05 ...
Downloaded image.part.05 from aws-s3-bucket1
Downloading part image.part.06 from aws-s3-bucket1/bundles/bundle_name to mybundle/
image.part.06 ...
Downloaded image.part.06 from aws-s3-bucket1
```

## ec2-migrate-manifest

### 説明

別のリージョンをサポートするように instance store-backed Linux AMI (証明書、カーネル、RAM ディスクなど) を変更します。

### 構文

```
ec2-migrate-manifest -c path -k path -m path {(-a access_key_id -s
secret_access_key --region region) | (--no-mapping)} [--ec2cert ec2_cert_path]
[--kernel kernel_id] [--ramdisk ramdisk_id]
```

### オプション

-c, --cert パス

ユーザーの PEM エンコード RSA パブリックキー証明書ファイル。

必須: はい

-k, --privatekey パス

ユーザーの PEM エンコード RSA キーファイルへのパス。

必須: はい

--manifest パス

マニフェストファイルへのパス。

必須: はい

-a, --access-key access\_key\_id

AWS アクセスキー ID。

必須: 自動マッピングを使用する場合は必須です。

-s, --secret-key secret\_access\_key

AWS シークレットアクセキー。

必須: 自動マッピングを使用する場合は必須です。

--region リージョン

マッピングファイル内で検索するリージョン。

必須: 自動マッピングを使用する場合は必須です。

--no-mapping

カーネルと RAM ディスクの自動マッピングを無効にします。

移行中、Amazon EC2 は、コピー先リージョン用に設計されたカーネルと RAM ディスクで、マニフェストファイルのカーネルと RAM ディスクを置き換えます。--no-mapping パラメータを指定しない場合、`ec2-migrate-bundle` は `DescribeRegions` および `DescribeImages` オペレーションを使用して、自動化されたマッピングを実行します。

必須: 自動マッピングに使用される `-a`、`-s`、および `--region` オプションを指定しない場合は必須です。

--ec2cert パス

イメージマニフェストの暗号化に使用される Amazon EC2 X.509 パブリックキー証明書へのパス。

`us-gov-west-1` および `cn-north-1` リージョンではデフォルト以外のパブリックキー証明書を使用し、その証明書へのパスは、このオプションで指定する必要があります。証明書へのパスは、AMI ツールのインストール方法によって異なります。Amazon Linux の場合、証明書の場所は `/opt/aws/amitools/ec2/etc/ec2/amitools/` です。「[AMI ツールを設定する \(p. 121\)](#)」の ZIP ファイルから AMI ツールをインストールした場合、証明書の場所は `$EC2_AMITOOL_HOME/etc/ec2/amitools/` です。

必須: `us-gov-west-1` および `cn-north-1` リージョンのみ。

--kernel kernel\_id

選択するカーネルの ID。

Important

カーネルと RAM ディスクではなく PV-GRUB を使用することをお勧めします。詳細については、「[独自の Linux カーネルを有効にする \(p. 176\)](#)」を参照してください。

必須: いいえ

--ramdisk ramdisk\_id

選択する RAM ディスクの ID。

Important

カーネルと RAM ディスクではなく PV-GRUB を使用することをお勧めします。詳細については、「[独自の Linux カーネルを有効にする \(p. 176\)](#)」を参照してください。

必須: いいえ

## 出力

バンドルプロセスのステージとステータスを記述するステータスマッセージ。

## 例

この例では、`my-ami.manifest.xml` マニフェストで指定された AMI を米国から EU にコピーします。

```
[ec2-user ~]$ ec2-migrate-manifest --manifest my-ami.manifest.xml --cert cert-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --privatekey pk-HKZYKTAIG2ECMXYIBH3HXV4ZBZQ55CLO.pem --region eu-west-1
```

```
Backing up manifest...
Successfully migrated my-ami.manifest.xml It is now suitable for use in eu-west-1.
```

## ec2-unbundle

### 説明

instance store-backed Linux AMI からバンドルを再作成します。

### 構文

```
ec2-unbundle -k path -m path [-s source_directory] [-d destination_directory]
```

### オプション

-k, --privatekey パス

PEM エンコードされる RSA キーファイルへのパス。

必須: はい

-m, --manifest パス

マニフェストファイルへのパス。

必須: はい

-s, --source *source\_directory*

バンドル含むディレクトリ。

デフォルト: 現在のディレクトリ。

必須: いいえ

-d, --destination *destination\_directory*

AMI をバンドル解除するディレクトリ。宛先ディレクトリが存在している必要があります。

デフォルト: 現在のディレクトリ。

必須: いいえ

### 例

この Linux および UNIX の例では、image.manifest.xml ファイルに指定された AMI をバンドル解除します。

```
[ec2-user ~]$ mkdir unbundled
$ ec2-unbundle -m mybundle/image.manifest.xml -k pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -s
mybundle -d unbundled
$ ls -l unbundled
total 1025008
-rw-r--r-- 1 root root 1048578048 Aug 25 23:46 image.img
```

### 出力

バンドル解除プロセスの多様な段階ステータスを示すメッセージが表示されます。

## ec2-upload-bundle

### 説明

instance store-backed Linux AMI のバンドルを Amazon S3 にアップロードし、アップロードされたオブジェクトで適切な ACL を設定します。詳細については、「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」を参照してください。

### 構文

```
ec2-upload-bundle -b bucket -a access_key_id -s secret_access_key [-t token] -m path [--url url] [--region region] [--sigv version] [--acl acl] [-d directory] [--part part] [--retry] [--skipmanifest]
```

### オプション

-b, --bucket バケット

バンドルを保存する Amazon S3 バケットの名前。その後にオプションで '/' 区切りのパスプレフィックスが続きます。バケットが存在しない場合、バケット名を使用できる場合はバケットが作成されます。

必須: はい

-a, --access-key access\_key\_id

AWS アクセスキー ID。

必須: はい

-s, --secret-key secret\_access\_key

お客様の AWS シークレットアクセスキー。

必須: はい

-t, --delegation-token トークン

AWS リクエストに渡す委任トークン。詳細については、「[一時的なセキュリティ認証情報の使用](#)」を参照してください。

必須: 一時的なセキュリティ認証情報を使用している場合のみ。

デフォルト: AWS\_DELEGATION\_TOKEN 環境変数の値 (設定されている場合)。

-m, --manifest パス

マニフェストファイルへのパス。マニフェストファイルはバンドルプロセス中に作成され、バンドルを含むディレクトリにあります。

必須: はい

--url url

廃止. バケットの場所が (--region ではなく) EU に制約されない限り、代わりに eu-west-1 オプションを使用します。--location フラグは、その特定の場所の制限を対象にする唯一の方法です。

Amazon S3 エンドポイントサービスの URL。

デフォルト: https://s3.amazonaws.com/

必須: いいえ

--region リージョン

宛先の S3 バケットに対してリクエスト署名で使用するリージョン。

- バケットが存在せず、リージョンを指定しない場合、ツールは (us-east-1 で) 場所の制約のないバケットを作成します。
- バケットが存在せず、リージョンを指定した場合、ツールは指定したリージョンでバケットを作成します。
- バケットが存在し、リージョンを指定しない場合、ツールはバケットの場所を使用します。
- バケットが存在し、リージョンとして us-east-1 を指定した場合、ツールはエラーメッセージなしでバケットの実際の場所を使用し、一致する既存のファイルは上書きされます。
- バケットが存在し、バケットの実際の場所に一致しない (us-east-1 以外の) リージョンを指定した場合、ツールはエラーで終了します。

バケットが (EU ではなく) eu-west-1 の場所に制約されている場合は、代わりに --location フラグを使用します。--location フラグは、その特定の場所の制限を対象にする唯一の方法です。

デフォルト: us-east-1

必須: 署名バージョン 4 を使用する場合は必須

--sigv バージョン

リクエストに署名するときに使用する署名バージョン。

有効な値: 2 | 4

デフォルト: 4

必須: いいえ

--acl acl

バンドルされたイメージのアクセスコントロールリストのポリシー。

有効な値: public-read | aws-exec-read

デフォルト: aws-exec-read

必須: いいえ

-d, --directory ディレクトリ

バンドルされた AMI 部分を含むディレクトリ。

デフォルト: マニフェストファイルを含むディレクトリ (-m オプションを参照)。

必須: いいえ

--part パート

指定された部分とそれ以降のすべての部分のアップロードを開始します。たとえば、--part 04 と指定します。

必須: いいえ

--retry

すべての Amazon S3 エラーで、オペレーションあたり最大 5 回まで自動的に再試行します。

必須: いいえ

--skipmanifest

マニフェストをアップロードしません。

必須: いいえ

--location 場所

廃止. バケットの場所が (--region ではなく) EU に制約されない限り、代わりに eu-west-1 オプションを使用します。--location フラグは、その特定の場所の制限を対象にする唯一の方法です。

宛先 Amazon S3 バケットの場所の制約。バケットが存在し、バケットの実際の場所に一致しない場所を指定する場合、ツールはエラーで終了します。バケットが存在し、場所を指定しない場合、ツールはバケットの場所を使用します。バケットが存在しない場合に場所を指定すると、ツールは、指定した場所でバケットを作成します。バケットが存在せず、場所を指定しない場合、ツールは (us-east-1 で) 場所の制約のないバケットを作成します。

デフォルト: --region を指定した場合、場所はその指定したリージョンに設定されます。--region を指定しない場合、場所はデフォルトで us-east-1 になります。

必須: いいえ

## 出力

Amazon EC2 は、アップロードプロセスのステージとステータスを示すステータスマッセージを表示します。

## 例

この例では、image.manifest.xml マニフェストで指定されたバンドルをアップロードします。

```
[ec2-user ~]$ ec2-upload-bundle -b aws-s3-bucket1/bundles/bundle_name -m image.manifest.xml
-a your_access_key_id -s your_secret_access_key
Creating bucket...
Uploading bundled image parts to the S3 bucket aws-s3-bucket1 ...
Uploaded image.part.00
Uploaded image.part.01
Uploaded image.part.02
Uploaded image.part.03
Uploaded image.part.04
Uploaded image.part.05
Uploaded image.part.06
Uploaded image.part.07
Uploaded image.part.08
Uploaded image.part.09
Uploaded image.part.10
Uploaded image.part.11
Uploaded image.part.12
Uploaded image.part.13
Uploaded image.part.14
Uploading manifest ...
Uploaded manifest.
Bundle upload completed.
```

## AMI ツール用の一般的なオプション

AMI ツールのほとんどで、以下の任意のパラメータを使用できます。

--help, -h

ヘルプメッセージを表示します。

--version

バージョンと著作権表記を表示します。

--manual

手動のエントリを表示します。

--batch

インタラクティブなプロンプトを制約するバッチモードで実行します。

--debug

問題のトラブルシューティング時に役立つ可能性がある情報を表示します。

## EBS-Backed AMI での暗号化の利用

Amazon EBS スナップショットを使用した AMI は Amazon EBS 暗号化の利点を活かすことができます。データおよびルートボリュームの両方のスナップショットを暗号化して AMI にアタッチできます。インスタンスを起動し、完全な EBS 暗号化サポートも含めてイメージをコピーできます。これらのオペレーションの暗号化パラメータは、AWS KMS が利用できるすべてのリージョンでサポートされています。

暗号化された EBS ボリュームを持つ EC2 インスタンスは、他のインスタンスと同様に AMI から起動します。また、暗号化されていない EBS スナップショットでバックアップされている AMI からインスタンスを起動するとき、起動中に一部またはすべてのボリュームを暗号化できます。

EBS ボリュームと同じように、AMI のスナップショットは、デフォルトの AWS Key Management Service カスタマーマスターキー(CMK)、または指定したカスタマー管理キーにより暗号化できます。いずれの場合も、選択したキーを使用するアクセス許可が必要です。

暗号化されたスナップショットを持つ AMI は、AWS アカウント間で共有できます。詳細については、「[共有 AMI](#)」を参照してください。

## インスタンスの起動シナリオ

Amazon EC2 インスタンスは、ブロックデバイスマッピングを通じて提供されるパラメータを備えた `RunInstances` アクションを使用して、AMI から起動されます。このパラメータは、AWS マネジメント コンソールにより、または直接 Amazon EC2 API または CLI を使用して提供されます。ブロックデバイスマッピングに関する詳細については、「[ブロックデバイスマッピング](#)」を参照してください。AWS CLI からブロックデバイスマッピングを制御する例については、「[EC2 インスタンスを起動、リスト、および終了する](#)」を参照してください。

デフォルトでは、明示的な暗号化パラメータがない場合、AMI のソーススナップショットから EBS ボリュームを復元しているときに、`RunInstances` アクションは AMI のソーススナップショットの既存の暗号化状態を維持します。デフォルトでの暗号化が有効な場合、AMI から作成されたすべてのボリューム(暗号化されたスナップショットから作成されたか暗号化されていないスナップショットから作成されたかに関係なく)が暗号化されます。デフォルトでの暗号化有効にされていない場合、インスタンスは AMI の暗号化状態を維持します。

インスタンスを起動し、同時に、暗号化パラメータを指定して、新しい暗号化状態を生成されるボリュームに適用することもできます。そのため、以下の動作が観察されます。

### 暗号化パラメータなしでの起動

- デフォルトでの暗号化が有効にされている場合を除き、暗号化されていないスナップショットは、暗号化されていないボリュームに復元されます。デフォルトでの暗号化が有効にされている場合は、新しく作成されるすべてのボリュームが暗号化されます。
- 所有する暗号化されたスナップショットは、同じ CMK に暗号化されるボリュームに復元されます。
- 所有していない暗号化されたスナップショット(たとえば、AMI が共有されている)は、ユーザーの AWS アカウントのデフォルト CMK に暗号化されているボリュームに復元されます。

デフォルトの動作は、暗号化パラメータを指定してオーバーライドできます。利用できるパラメータは、`Encrypted` と `KmsKeyId` です。`Encrypted` パラメータのみを設定すると、次のような結果になります。

`Encrypted` を設定し、`KmsKeyId` を指定しない場合のインスタンス起動動作

- 暗号化されていないスナップショットは、ユーザーの AWS アカウントのデフォルト CMK により暗号化されている EBS ボリュームに復元されます。
- 所有する暗号化されたスナップショットは、同じ CMK により暗号化された EBS ボリュームに復元されます。(つまり、`Encrypted` パラメータには効果がありません。)
- 所有していない暗号化されたスナップショット(つまり、AMI が共有されている)は、ユーザーの AWS アカウントのデフォルト CMK により暗号化されているボリュームに復元されます。(つまり、`Encrypted` パラメータには効果がありません。)

`Encrypted` と `KmsKeyId` 両方のパラメータを設定すると、暗号化オペレーションにデフォルトではない CMK を指定できます。結果として次のように動作します。

#### `Encrypted` と `KmsKeyId` が両方設定されたインスタンス

- 暗号化されていないスナップショットは、指定された CMK により暗号化された EBS ボリュームに復元されます。
- 暗号化されたスナップショットは、元の CMK ではなく、指定された CMK に暗号化された EBS ボリュームに復元されます。

`Encrypted` パラメータも設定せずに `KmsKeyId` を送信するとエラーが発生します。

以下のセクションでは、デフォルトではない暗号化パラメータを使用して AMI からインスタンスを起動する例を示します。これらの各シナリオでは、`RunInstances` アクションに指定するパラメータにより、スナップショットからボリュームを復元中に暗号化の状態が変化します。

#### Note

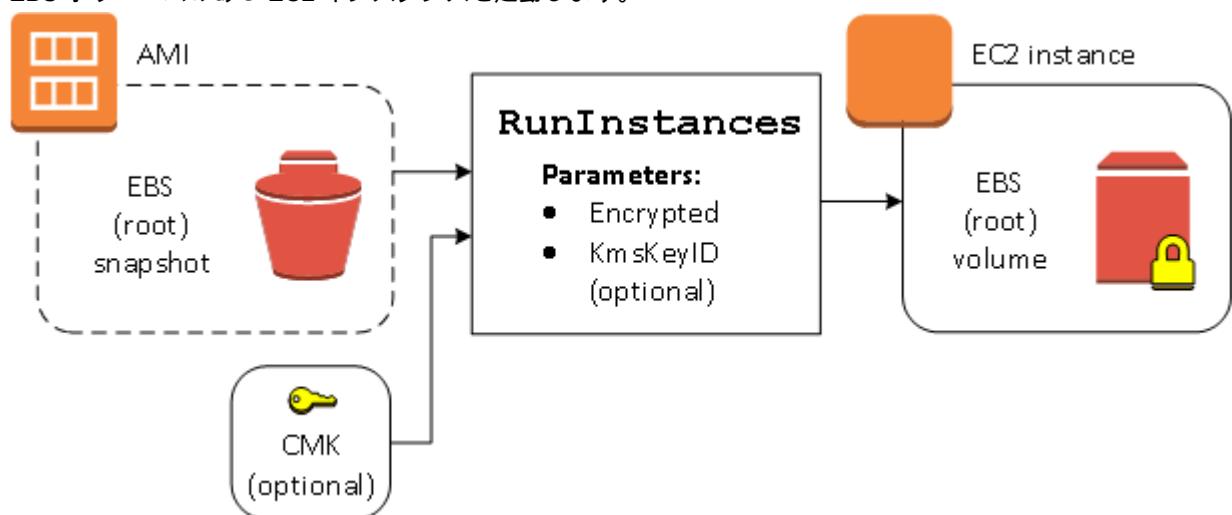
コンソールで AMI からインスタンスを起動する詳細な手順については、「[インスタンスの起動](#)」を参照してください。

`RunInstances` API のドキュメントについては、「[RunInstances](#)」を参照してください。

AWS Command Line Interface の `run-instances` コマンドのドキュメントについては、「[run-instances](#)」を参照してください。

## 起動時にボリュームを暗号化する

この例では、暗号化されていないスナップショットでバックアップされた AMI を使用して、暗号化された EBS ボリュームのある EC2 インスタンスを起動します。

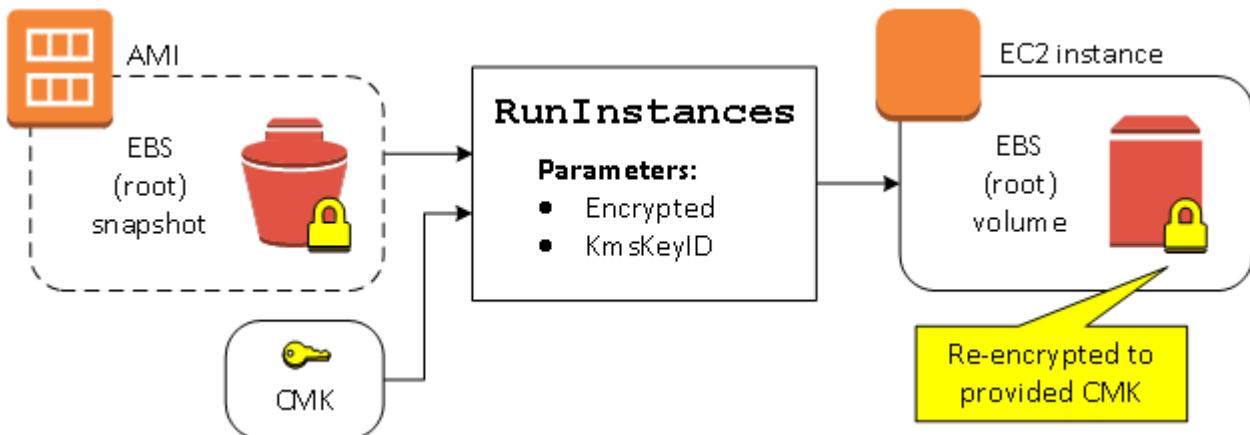


`Encrypted` パラメータのみを使用すると、このインスタンスのボリュームが暗号化されます。`KmsKeyId` パラメータの指定はオプションです。キー ID を指定しない場合、AWS アカウントのデフォルト CMK を

使用して、ボリュームを暗号化します。所有する別の CMK にボリュームを暗号化するには、`KmsKeyId` パラメータを指定します。

## 起動時にボリュームを再暗号化する

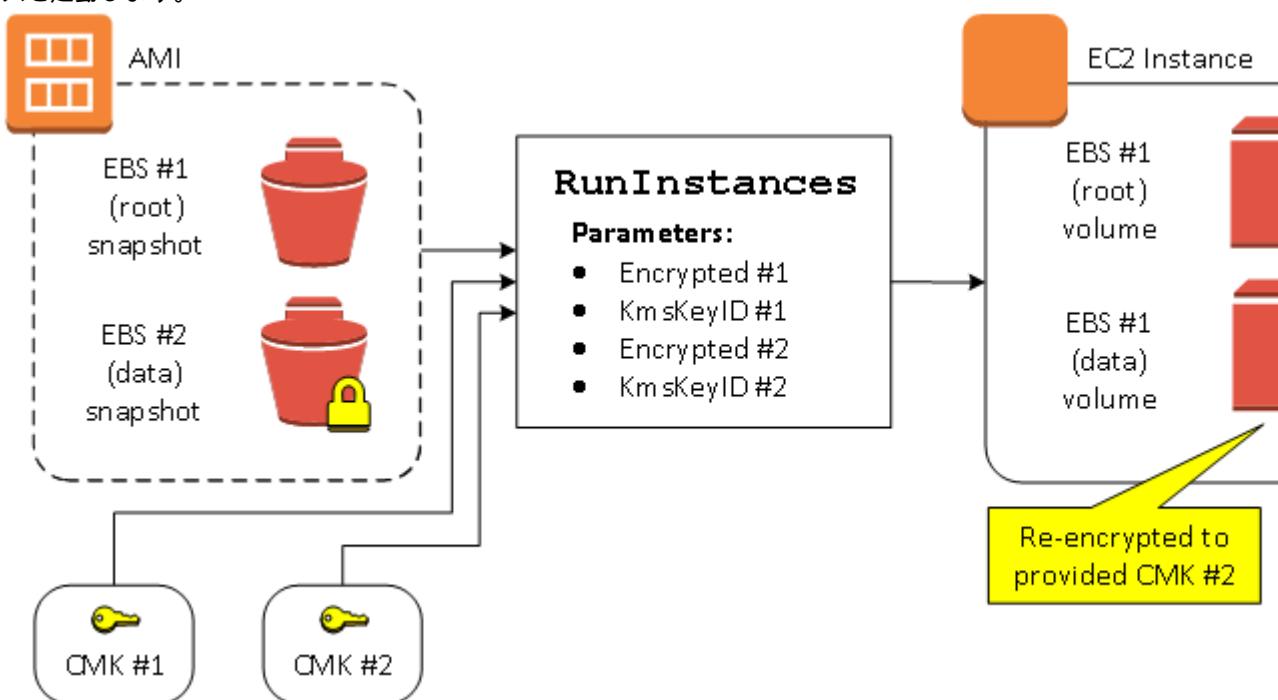
この例では、暗号化されたスナップショットでバックアップされた AMI を使用して、新しい CMK により暗号化された EBS ボリュームのある EC2 インスタンスを起動します。



AMI を所有していて、暗号化パラメータを指定しない場合、作成されるインスタンスにはスナップショットと同じキーにより暗号化されたボリュームがあります。AMI が所有ではなく共有され。暗号化パラメータを指定しない場合、ボリュームはデフォルト CMK により暗号化されます。暗号化パラメータがここに示すように指定されている場合、ボリュームは指定された CMK により暗号化されます。

## 起動時に複数のボリュームの暗号化状態を変更する

このより複雑な例では、複数のスナップショット（暗号化状態はそれぞれ異なります）でバックアップされた AMI を使用して、新しく暗号化されたボリュームと再暗号化されたボリュームがある EC2 インスタンスを起動します。



このシナリオでは、RunInstances アクションにソーススナップショットそれぞれに対する暗号化パラメータが指定されます。可能な暗号化パラメータがすべて指定されると、AMI を所有しているかどうかに関係なく、作成されるインスタンスは同じです。

## イメージコピーのシナリオ

Amazon EC2 AMI は、AWS マネジメントコンソール を通じてまたは直接 Amazon EC2 API または CLI を使用し、CopyImage アクションを使用してコピーされます。

デフォルトでは、明示的な暗号化パラメータがない場合、コピー中 CopyImage アクションは AMI のソーススナップショットの既存の暗号化状態を維持します。AMI をコピーし、同時に、暗号化パラメータを指定して、新しい暗号化状態を関連付けられている EBS スナップショットに適用することができます。そのため、以下の動作が観察されます。

### 暗号化パラメータなしでのコピー

- デフォルトでの暗号化が有効にされている場合を除き、暗号化されていないスナップショットは、別の暗号化されていないスナップショットにコピーされます。デフォルトでの暗号化が有効にされている場合は、新しく作成されるすべてのスナップショットが暗号化されます。
- 所有する暗号化されたスナップショットは、同じキーで暗号化されたスナップショットにコピーされます。
- 所有していない暗号化されたスナップショット (つまり、AMI が共有されている) は、ユーザーの AWS アカウントのデフォルト CMK により暗号化されているスナップショットにコピーされます。

これらすべてのデフォルトの動作は、暗号化パラメータを指定してオーバーライドできます。利用できるパラメータは、Encrypted と KmsKeyId です。Encrypted パラメータのみを設定すると、次のような結果になります。

### Encrypted を設定し、KmsKeyId を指定しない場合のコピーイメージ動作

- 暗号化されていないスナップショットは、AWS アカウントのデフォルト CMK により暗号化されたスナップショットにコピーされます。
- 暗号化されたスナップショットは、同じ CMK により暗号化されたスナップショットにコピーされます。(つまり、Encrypted パラメータには効果がありません。)
- 所有していない暗号化されたスナップショット (つまり、AMI が共有されている) は、ユーザーの AWS アカウントのデフォルト CMK により暗号化されているボリュームにコピーされます。(つまり、Encrypted パラメータには効果がありません。)

Encrypted と KmsKeyId 両方のパラメータを設定すると、暗号化オペレーションにカスタマーマネージド CMK を指定できます。結果として次のように動作します。

### Encrypted と KmsKeyId の両方を設定した場合のコピーイメージ動作

- 暗号化されていないスナップショットは、指定された CMK により暗号化されたスナップショットにコピーされます。
- 暗号化されたスナップショットは、元の CMK ではなく、指定された CMK に暗号化されたスナップショットにコピーされます。

Encrypted パラメータも設定せずに KmsKeyId を送信するとエラーが発生します。

以下のセクションでは、デフォルトではない暗号化パラメータを使用して AMI をコピーし、結果として暗号化状態が変化する例を示します。

#### Note

AMI をコピーするための詳細な手順については、「[AMI のコピー](#)」を参照してください。  
CopyImage API のドキュメントについては、「[CopyImage](#)」を参照してください。

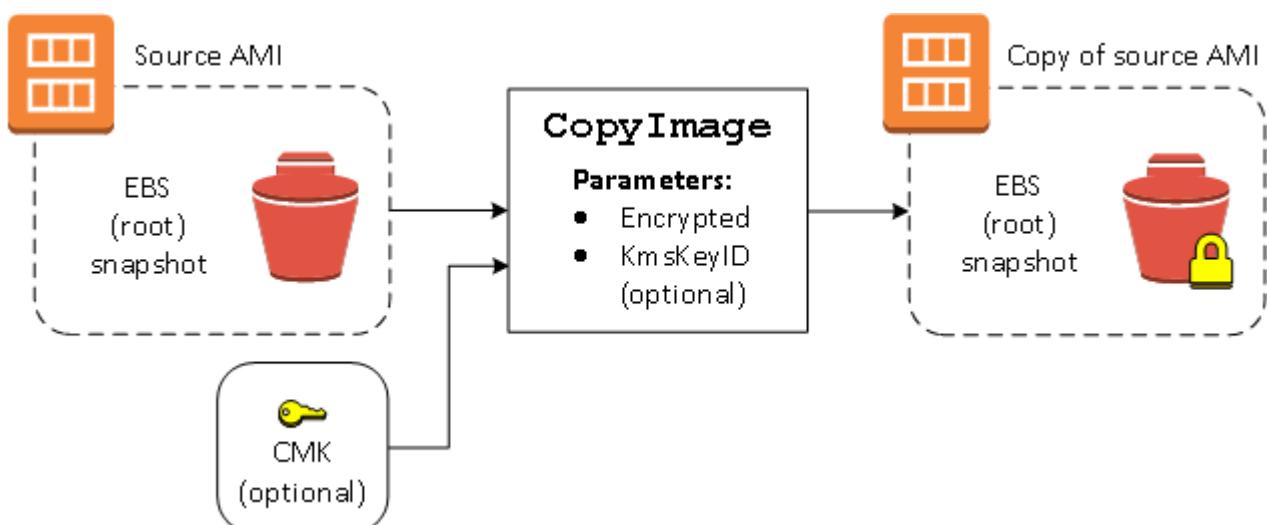
AWS Command Line Interface のコマンド `copy-image` のドキュメントについては、「[copy-image](#)」を参照してください。

## コピー時に暗号化されていないイメージを暗号化する

このシナリオでは、暗号化されていないルートのスナップショットでバックアップされた AMI は、暗号化されたルートのスナップショットと合わせて AMI にコピーされます。CMK の選択を含む 2 つの暗号化パラメータを指定した `CopyImage` アクションが呼び出されます。その結果、ルートスナップショットの暗号化ステータスが変更され、ターゲット AMI は、元のスナップショットと同じデータを含むが指定されたキーを使用して暗号化されたルートスナップショットにバックアップされます。いずれかの AMI で起動するインスタンスに対する料金と同様に、両方の AMI でスナップショットのストレージコストが発生します。

### Note

[デフォルトで暗号化 \(p. 1017\)](#)を有効にすると、AMI のすべてのスナップショットで `Encrypted` パラメータを `true` に設定するのと同じ効果があります。



`Encrypted` パラメータを設定すると、このインスタンスの単一のスナップショットが暗号化されます。`KmsKeyId` パラメータを指定しない場合は、デフォルトの CMK がスナップショットコピーの暗号化に使用されます。

### Note

複数のスナップショットがあるイメージをコピーして、それぞれの暗号化状態を個々に設定することもできます。

## AMI のコピー

Amazon マシンイメージ (AMI) は、AWS マネジメントコンソール、AWS Command Line Interface または SDK、または Amazon EC2 API (すべて `CopyImage` アクションをサポート) を使用して、AWS リージョン内または AWS リージョン間でコピーできます。Amazon EBS-backed AMI と instance store-backed AMI のいずれもコピーできます。暗号化されたスナップショットで AMI をコピーし、コピープロセス中に暗号化ステータスを変更することもできます。

ソース AMI をコピーすると、同一ではあるが独自の識別子を使用するターゲット AMI となります。Amazon EBS-backed AMI の場合は、それぞれのバックアップするスナップショットは、デフォルトでは、同一だが区別されるターゲットスナップショットにコピーされます。(唯一の例外は、スナップショットを暗号化または再暗号化するときです。) ソース AMI は、ターゲット AMI に影響を及ぼさずに変更または登録解除できます。逆の場合も同様です。

AMI のコピーには課金されません。ただし、標準のストレージ料金とデータ転送料金が適用されます。EBS-backed AMI をコピーする場合は、追加の EBS スナップショットのストレージに対して料金が発生します。

AWS は元の AMI から新しい AMI に起動許可、ユーザー定義のタグ、Amazon S3 バケット許可をコピーしません。コピー操作が完了すると、起動許可、ユーザー定義のタグ、Amazon S3 バケット許可を新しい AMI に適用できます。

直接取得したか、共有されたかに関わらず、AWS Marketplace から取得した AMI をコピーすることはできません。代わりに AWS Marketplace AMI を使用して EC2 インスタンスを起動し、インスタンスから AMI を作成します。詳細については、「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」を参照してください。

## Instance Store-Backed AMI をコピーするアクセス許可

Instance Store-Backed AMI をコピーするために IAM を使用する場合、ユーザーは次の Amazon S3 へのアクセス許可が必要です。`s3:CreateBucket`、`s3:GetBucketAcl`、`s3>ListAllMyBuckets`、`s3:GetObject`、`s3:PutObject` および `s3:PutObjectAcl`。

次のポリシー例では、指定されたバケットの AMI ソースを、指定されたリージョンにコピーできます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3>ListAllMyBuckets",
            "Resource": [
                "arn:aws:s3:::*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::ami-source-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3>CreateBucket",
                "s3:GetBucketAcl",
                "s3:PutObjectAcl",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amis-for-123456789012-in-us-east-1*"
            ]
        }
    ]
}
```

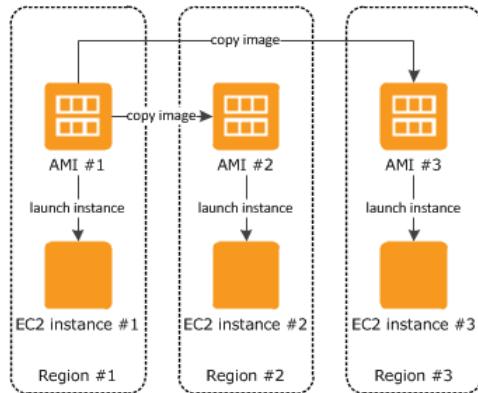
AMI ソースバケットの Amazon リソースネーム (ARN) を調べるには、<https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開き、ナビゲーションペインで [AMI] を選択し、[Source] 列でバケット名を見つけます。

## リージョン間のコピー

地理的に分散したリージョンに AMI をコピーすると、次のような利点があります。

- 一貫性のあるグローバルなデプロイメント: 1つのリージョンから別のリージョンに AMI をコピーすることで、一貫性のあるインスタンスを同じ AMI から別のリージョンに起動できます。
- スケーラビリティ: ユーザーの場所にかかわらず、ユーザーのニーズに合ったグローバルアプリケーションをより簡単に設計できます。
- パフォーマンス: アプリケーションを配布したり、アプリケーションの重要なコンポーネントをユーザーの近くに配置したりすることでパフォーマンスを向上できます。また、インスタンスの種類やその他の AWS サービスなど、リージョン固有の機能を活用することもできます。
- 高可用性: アプリケーションを設計し、AWS リージョン全体にわたってデプロイして可用性を高めることができます。

次の図は、ソース AMI と異なるリージョンにある 2 つのコピーされた AMI、およびそこから起動される EC2 インスタンスの関係を示します。AMI からインスタンスを起動すると、AMI が存在する同じリージョンに存在します。ソース AMI を変更し、それらの変更をターゲットリージョンの AMI に反映させる場合、ソース AMI をターゲットリージョンに再度コピーする必要があります。



instance store-backed AMI を最初にリージョンにコピーするときに、そのリージョンにコピーされた AMI に Amazon S3 バケットを作成します。そのリージョンにコピーするすべての Instance Store-Backed AMI が、このバケットに保存されます。バケット名の形式は次のとおりです: `amis-for-#####-in-#####-##`  
**##例:** `amis-for-123456789012-in-us-east-2-yhjmxvp6`。

### 前提条件

AMI をコピーする前に、ソース AMI のすべてのコンテンツが、異なるリージョンでの実行をサポートするように更新されていることを確認する必要があります。たとえば、データベース接続文字列や同様のアプリケーション設定データが、適切なリソースを指すように更新する必要があります。それ以外の場合、対象のリージョンの新しい AMI から起動したインスタンスは元のリージョンのリソースをまだ使用している可能性があり、それによりパフォーマンスとコストに影響が及ぶことがあります。

### 制限

- コピー先のリージョンには、AMI の同時コピーが 50 個までという制限があります。
- 準仮想化 (PV) AMI がサポートされていないリージョンに、PV AMI をコピーすることはできません。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。

## アカウント間のコピー

AWS アカウント間で AMI を共有できます。AMI の共有は AMI の所有権には影響しません。所有しているアカウントには、リージョンのストレージ料金が適用されます。詳細については、「[特定の AWS アカウントと AMI を共有する \(p. 106\)](#)」を参照してください。

自分のアカウントと共有された AMI をコピーした場合、アカウントのコピー先の AMI の所有者は自分になります。コピー元の AMI の所有者には、Amazon EBS または Amazon S3 の標準転送料金が課金され、コピー先の AMI の所有者には、コピー先リージョンのストレージ料金が課金されます。

### リソースのアクセス許可

AMI を別のアカウントから共有した場合、この AMI をコピーするには、関連 EBS スナップショット (Amazon EBS-backed AMI; の場合) であっても関連 S3 バケット (instance-store-backed AMI の場合) であっても、コピー元の AMI の所有者から AMI をバックアップするストレージの読み取り権限を付与してもらう必要があります。共有 AMI に暗号化されたスナップショットがある場合、所有者はキーも共有する必要があります。

## 暗号化とコピー

次の表は、各種 AMI コピーのシナリオにおける暗号化サポートを示します。暗号化されたスナップショットを生成するために暗号化されていないスナップショットをコピーすることはできますが、暗号化されていないスナップショットを生成するために暗号化されたスナップショットをコピーすることはできません。

シナリオ	説明	サポート対象
1	非暗号化から非暗号化	はい
2	暗号化から暗号化	はい
3	非暗号化から暗号化	はい
4	暗号化から非暗号化	いいえ

### Note

CopyImage アクション中の暗号化は Amazon EBS-backed AMI にのみ適用されます。Instance Store-Backed AMI はスナップショットに依存しないため、コピーを使用して暗号化ステータスを変更することはできません。

デフォルト (暗号化パラメータを指定しない) では、AMI をバックアップするスナップショットは元の暗号化ステータスとともにコピーされます。暗号化されていないスナップショットにバックアップされた AMI をコピーすると、やはり暗号化されていない同一のターゲットスナップショットになります。ソース AMI が暗号化されたスナップショットにバックアップされている場合は、コピーすると、同じカスタマーマスターキー (CMK) により暗号化された同一のターゲットスナップショットになります。複数のスナップショットにバックアップされた AMI をコピーした場合、デフォルトでは、元の暗号化ステータスが各ターゲットスナップショットで維持されます。

AMI をコピー中に暗号化パラメータを指定した場合、バックアップスナップショットを暗号化または再暗号化できます。以下の例は、ターゲット AMI の暗号化状態を変更するために CopyImage アクションに暗号化パラメータを提供する、デフォルトではないケースを示しています。

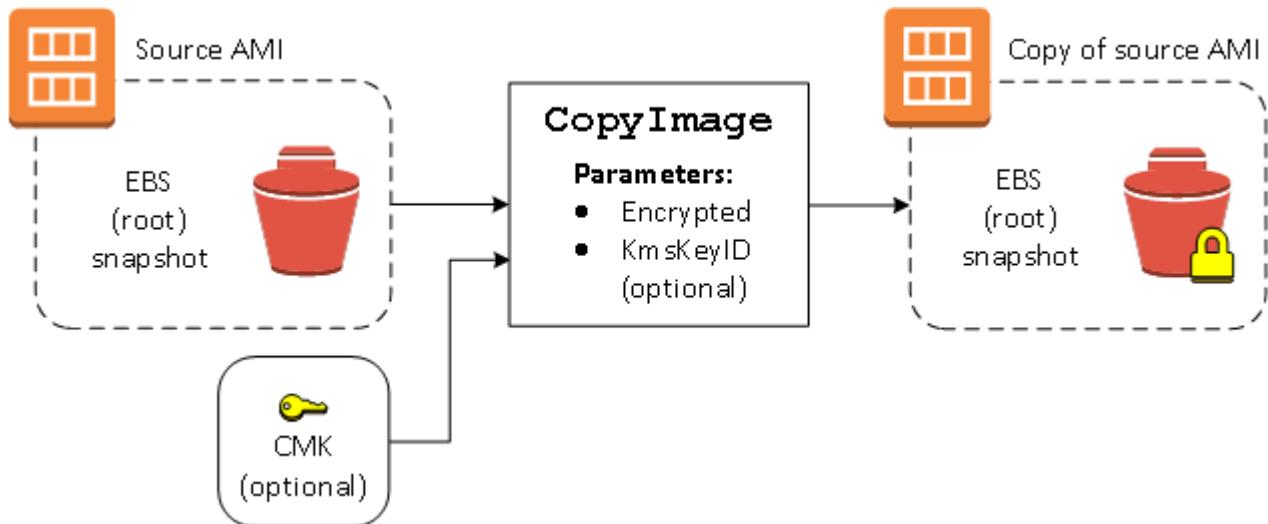
### 暗号化されていないソース AMI の暗号化されたターゲット AMI へのコピー

このシナリオでは、暗号化されていないルートのスナップショットでバックアップされた AMI は、暗号化されたルートのスナップショットと合わせて AMI にコピーされます。CMK の選択を含む 2 つの暗号化

パラメータを指定した CopyImage アクションが呼び出されます。その結果、ルートスナップショットの暗号化ステータスが変更され、ターゲット AMI は、元のスナップショットと同じデータを含むが指定されたキーを使用して暗号化されたルートスナップショットにバックアップされます。いずれかの AMI で起動するインスタンスに対する料金と同様に、両方の AMI でスナップショットのストレージコストが発生します。

Note

[デフォルトで暗号化 \(p. 1017\)](#)を有効にすると、AMI のすべてのスナップショットで Encrypted パラメータを true に設定するのと同じ効果があります。



Encrypted パラメータを設定すると、このインスタンスの単一のスナップショットが暗号化されます。KmsKeyId パラメータを指定しない場合は、デフォルトの CMK がスナップショットコピーの暗号化に使用されます。

暗号化されたスナップショットを持つ AMI のコピーの詳細については、「[EBS-Backed AMI での暗号化の利用 \(p. 151\)](#)」を参照してください。

## AMI のコピー

AMI は次の手順でコピーできます。

### 前提条件

Amazon EBS スナップショットによってバックアップされた AMI を作成、または取得します。Amazon EC2 コンソールを使用して AWS が提供するさまざまな AMI を検索できます。詳細については、「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」と「[Linux AMI の検索 \(p. 100\)](#)」を参照してください。

コンソールを使用して AMI をコピーするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. コンソールのナビゲーションバーから、AMI を含むリージョンを選択します。ナビゲーションペインで、[Images]、[AMIs] の順に選択し、リージョンで利用できる AMI のリストを表示します。
3. コピーする AMI を選択して、[Actions]、[Copy AMI] の順に選択します。
4. [AMI のコピー] ダイアログボックスで、下記の情報を指定して [AMI のコピー] を選択します。
  - 送信先リージョン: AMI をコピーするリージョン。
  - 名前: 新しい AMI の名前。この名前にはオペレーティングシステム情報を含めることができます（この情報は AMI の詳細としては表示されません）。

- 説明: デフォルトでは、オリジナルからコピーを見分けられるように、ソース AMI に関する情報が説明に含まれています。この説明は必要に応じて変更できます。
  - 暗号化: ターゲットスナップショットを暗号化するか、別のキーを使用して再暗号化する場合は、このフィールドを選択します。デフォルトで暗号化をデフォルトで有効にしている場合は、[Encryption (暗号化)] オプションが設定され、AMI コンソールから設定解除することはできません。
  - マスターキー: ターゲットスナップショットを暗号化するための KMS キー。
5. コピーオペレーションが開始したことを知らせる確認ページが表示され、新しい AMI の ID が提供されます。

コピー操作の進行状況をすぐに確認するには、提供されたリンクをクリックしてください。後で進行状況を確認するには [Done] を選択し、自分の準備ができたときに、ナビゲーションバーを使用してターゲットリージョン (ある場合) を切り替え、AMI のリストから AMI を見つけます。

ターゲット AMI の初期ステータスは pending です。ステータスが available になると、オペレーションは完了します。

AWS CLI を使用して AMI をコピーするには

AMI は、[copy-image](#) コマンドを使用してコピーできます。コピー元リージョンおよび送信先リージョンの両方を指定する必要があります。コピー元のリージョンは、--source-region パラメータを使用して指定します。--region パラメータまたは環境変数を使用して送信先リージョンを指定できます。詳細については、「[AWS コマンドラインインターフェイスの設定](#)」を参照してください。

コピー時にターゲットスナップショットを暗号化する場合は、--encrypted および --kms-key-id の追加のパラメータを指定する必要があります。

Tools for Windows PowerShell を使用して AMI をコピーするには

AMI は、[Copy-EC2Image](#) コマンドを使用してコピーできます。コピー元リージョンおよび送信先リージョンの両方を指定する必要があります。コピー元のリージョンは、-SourceRegion パラメータを使用して指定します。-Region パラメータまたは Set-AWSDefaultRegion コマンドを使用して送信先リージョンを指定できます。リージョンの詳細については、「[AWS リージョンの指定](#)」を参照してください。

コピー時にターゲットスナップショットを暗号化する場合は、-Encrypted および -KmsKeyId の追加のパラメータを指定する必要があります。

## 保留中の AMI コピー操作を中止する

保留中の AMI のコピーは、次の手順で停止できます。

コンソールを使用して AMI のコピー操作を中止するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションバーのリージョンセレクターから対象のリージョンを選択します。
- ナビゲーションペインで [AMIs] を選択します。
- コピーを中止する AMI を選択し、[Actions]、[Deregister] を選択します。
- 確認を求められたら、[Continue] を選択します。

コマンドラインを使用して AMI コピー操作を中止するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [deregister-image](#) (AWS CLI)
- [Unregister-EC2Image](#) (AWS Tools for Windows PowerShell)

## 請求情報の取得

オンデマンドインスタンス または スポットインスタンス を起動する前に、または リザーブドインスタンス を購入する前に、Amazon マシンイメージ (AMI) に関連付けられたプラットフォームの詳細と請求情報を確認できます。スポットインスタンス では、プラットフォームの詳細を使用して、AMI が スpot インスタンス でサポートされていることを確認できます。リザーブドインスタンス を購入するときに、[Platform (プラットフォーム)] で AMI の PlatformDetails にマッピングする正しい値が選択されていることを確認できます。インスタンスを起動する前に、または リザーブドインスタンス を購入する前に請求情報を知ることで、間違った AMI からインスタンスを誤って起動し、予定外のコストが発生する可能性を減らすことができます。

インスタンスの料金の詳細については、「[Amazon EC2 料金表](#)」を参照してください。

### 目次

- [AMI 請求情報フィールド \(p. 161\)](#)
- [プラットフォーム詳細および使用状況オペレーション請求コード \(p. 161\)](#)
- [プラットフォーム詳細および請求情報の表示 \(p. 162\)](#)

## AMI 請求情報フィールド

`describe-images` コマンドは、AMI に関する次の情報を返します。

### PlatformDetails

AMI の請求コードに関連付けられたプラットフォームの詳細。たとえば、Red Hat Enterprise Linux と指定します。

### UsageOperation

Amazon EC2 インスタンスのオペレーション、および AMI に関連付けられている請求コード。UsageOperation は、AWS のコストと使用状況レポート (CUR) および [AWS 價格表 API](#) の `lineitem/Operation` 列に対応します。UsageOperation コードのリストについては、次のセクションの「[プラットフォーム詳細および使用状況オペレーション請求コード \(p. 161\)](#)」を参照してください。

## プラットフォーム詳細および使用状況オペレーション 請求コード

次の表に、AMI で `describe-images` コマンドを実行したときに返される PlatformDetails および UsageOperation の値を示します。

PlatformDetails	UsageOperation **
Linux/UNIX	RunInstances
Red Hat BYOL Linux	RunInstances:00g0
Red Hat Enterprise Linux	RunInstances:0010
SQL Server Enterprise	RunInstances:0100
SQLServer Standard	RunInstances:0004

PlatformDetails	UsageOperation **
SQL Server Web	RunInstances:0200
SUSE Linux	RunInstances:000g
Windows	RunInstances:0002
Windows BYOL	RunInstances:0800
Windows with SQL Server Enterprise *	RunInstances:0102
Windows with SQL Server Standard *	RunInstances:0006
Windows with SQL Server Web *	RunInstances:0202

\* 2 つのソフトウェアライセンスが 1 つの AMI に関連付けられている場合、PlatformDetails フィールドには両方が表示されます。

\*\* スポットインスタンスを実行している場合、AWS のコストと使用状況レポートの [lineitem/operation](#) は、ここに記載されている UsageOperation 値と異なる場合があります。たとえば、[lineitem/operation](#) に RunInstances:0010:SV006 が表示されている場合は、Amazon EC2 が VPC ゾーン #6 で米国東部（バージニア）で Red Hat Enterprise Linux スポットインスタンス時間を行っていることを示します。

## プラットフォーム詳細および請求情報の表示

AMI に関連付けられたプラットフォームの詳細と請求情報を表示するには

[describe-images](#) コマンドを使用します。

```
$ aws ec2 describe-images --image-ids ami-0123456789EXAMPLE
```

次の出力例は、PlatformDetails フィールドと UsageOperation フィールドを示しています。この例では、ami-0123456789EXAMPLE プラットフォームは Red Hat Enterprise Linux であり、使用オペレーションは RunInstances:0010 です。

```
{
    "Images": [
        {
            "VirtualizationType": "hvm",
            "Description": "Provided by Red Hat, Inc.",
            "Hypervisor": "xen",
            "EnaSupport": true,
            "SriovNetSupport": "simple",
            "ImageId": "ami-0123456789EXAMPLE",
            "State": "available",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "Ebs": {
                        "SnapshotId": "snap-111222333444aaabb",
                        "DeleteOnTermination": true,
                        "VolumeType": "gp2",
                        "VolumeSize": 10,
                        "Encrypted": false
                    }
                }
            ],
        }
    ]
},
```

```
        "Architecture": "x86_64",
        "ImageLocation": "123456789012/RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2",
        "RootDeviceType": "ebs",
        "OwnerId": "123456789012",
        "PlatformDetails": "Red Hat Enterprise Linux",
        "UsageOperation": "RunInstances:0010",
        "RootDeviceName": "/dev/sda1",
        "CreationDate": "2019-05-10T13:17:12.000Z",
        "Public": true,
        "ImageType": "machine",
        "Name": "RHEL-8.0.0_HVM-20190618-x86_64-1-Hourly2-GP2"
    }
}
]
```

予定外のコストを発生させないように、AWS のコストと使用状況レポートの請求情報が、[describe-images](#) から返された請求情報と一致していることを確認できます。ami-0123456789EXAMPLE を使用してインスタンスを起動した場合は、AWS のコストと使用状況レポートでインスタンスの請求情報を確認できます。インスタンス ID を検索し、[lineitem/Operation](#) 列で対応する値を確認します。この例では、値は RunInstances:0010 である必要があります。

インスタンスに関連付けられたプラットフォームの詳細と請求情報を表示するには

インスタンスを起動した後、インスタンスマタデータの `billingProducts` フィールドを調べることによって、請求情報を見つけることができます。詳細については、「[インスタンスアイデンティティドキュメントと署名の取得 \(p. 619\)](#)」を参照してください。または、[describe-instances](#) コマンドを使用してインスタンスの AMI ID を取得し、前述の手順で説明した [describe-images](#) コマンドを使用して、レスポンスの `PlatformDetails` フィールドおよび `UsageOperation` フィールドから請求情報を取得することもできます。

## Linux AMI の登録解除

AMI の利用が終わったら、その登録を解除できます。AMI の登録を解除すると、それを使用して新しいインスタンスを起動できなくなります。

AMI の登録を解除しても、AMI から既に起動したインスタンスに影響を与えることはありません。そのようなインスタンスの使用に対しては引き続き課金されます。そのため、使用が終わったら、インスタンスを終了することをお勧めします。

AMI のクリーンアップに使用する手順は、Amazon EBS-Backed と Instance Store-Backed で異なります。詳細については、「[AMI のルートデバイスタイプの判別 \(p. 97\)](#)」を参照してください。

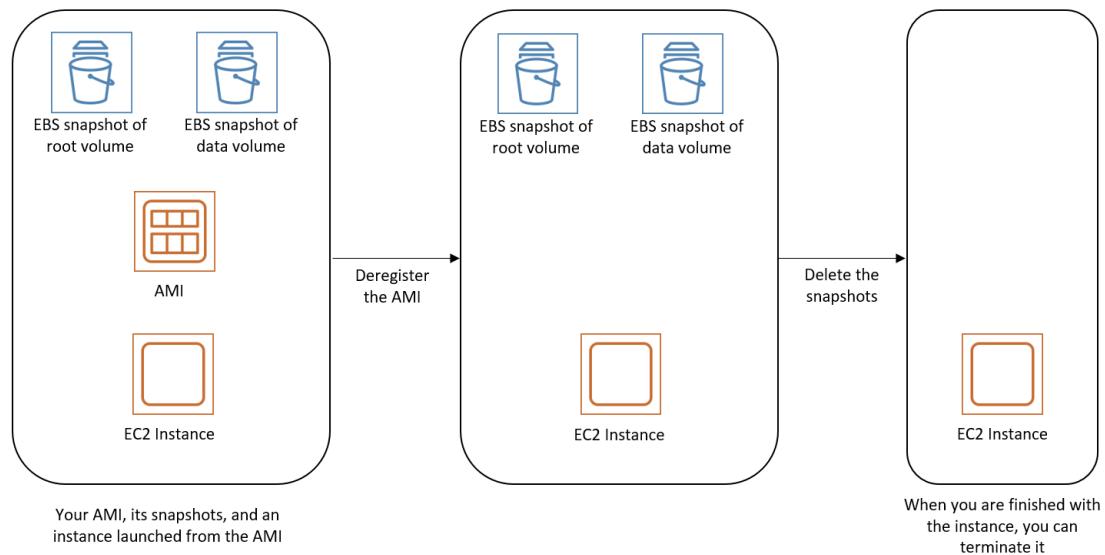
### コンテンツ

- [Amazon EBS-Backed AMI のクリーンアップ \(p. 163\)](#)
- [Instance Store-Backed AMI をクリーンアップする \(p. 164\)](#)

## Amazon EBS-Backed AMI のクリーンアップ<sup>®</sup>

Amazon EBS-Backed AMI の登録を解除しても、AMI 作成プロセス中にインスタンスのボリューム用に作成したスナップショットには影響しません。スナップショットのストレージは引き続き課金されます。そのため、使用が終わったスナップショットは削除することをお勧めします。

次の図は、Amazon EBS-Backed AMI のクリーンアッププロセスについてまとめたものです。



### Amazon EBS-Backed AMI をクリーンアップするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [AMIs] を選択します。AMI を選択し、その ID — を書き留めます。これは、次のステップで正しいスナップショットを見つけるのに役立ちます。[Actions]、[Deregister] の順に選択します。確認を求められたら、[Continue] を選択します。

#### Note

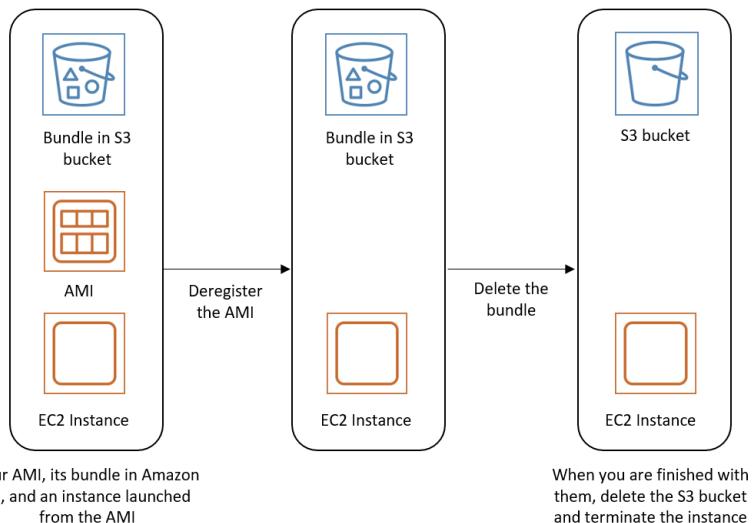
コンソールで AMI がリストから削除されるまで、数分ほどかかります。ステータスを更新するには、[Refresh] を選択します。

3. ナビゲーションペインで、[Snapshots] を選択し、スナップショットを選択します ([Description] 列で AMI ID を探します)。[Actions] を選択してから、[Delete Snapshot] を選択します。確認を求めるメッセージが表示されたら、[Yes, Delete] を選択します。
4. (オプション) AMI から起動したインスタンスの使用が終わったら、それを終了します。ナビゲーションペインで、[インスタンス] を選択します。インスタンスを選択後、[Actions]、[Instance State]、[Terminate (削除)] の順に選択します。確認を求めるメッセージが表示されたら、[Yes, Terminate] を選択します。

## Instance Store-Backed AMI をクリーンアップする

Instance Store-Backed AMI の登録を解除しても、AMI の作成時に Amazon S3 にアップロードしたファイルには影響しません。Amazon S3 のそのファイルの使用に対しては引き続き課金されます。そのため、使用が終わったら、それらのファイルは削除することをお勧めします。

次の図は、Instance Store-Backed AMI のクリーンアッププロセスをまとめたものです。



### Instance Store-Backed AMI をクリーンアップするには

1. 次のように、[deregister-image](#) コマンドを使用して AMI の登録を解除します。

```
aws ec2 deregister-image --image-id ami_id
```

2. 次のように [ec2-delete-bundle \(p. 141\)](#) (AMI ツール) コマンドを使用して、Amazon S3 のバンドルを削除します。

```
ec2-delete-bundle -b myawsbucket/myami -a your_access_key_id -s your_secret_access_key  
-p image
```

3. (オプション) AMI から起動したインスタンスの使用が終わったら、次のように、[terminate-instances](#) コマンドを使用してそれを終了します。

```
aws ec2 terminate-instances --instance-ids instance_id
```

4. (オプション) バンドルをアップロードした Amazon S3 バケットの使用が終わったら、バケットを削除できます。Amazon S3 バケットを削除するには、Amazon S3 コンソールを開き、バケットを選択してから、[Actions]、[Delete] の順に選択します。

## Amazon Linux

Amazon Linux は Amazon ウェブ サービス ( AWS ) が提供します。これは Amazon EC2 上で実行するアプリケーションのために安定した安全で高性能な実行環境を提供できるよう設計されています。またこれには、起動設定ツールおよび多くの AWS 人気ライブラリやツールなど、AWS の統合を容易にするいくつかのパッケージも含まれています。AWS は Amazon Linux を実行しているすべてのインスタンスの現行のセキュリティとメンテナンスの更新を提供します。CentOS ( および同様のディストリビューション ) で開発された多くのアプリケーションは、Amazon Linux で実行されます。

### 目次

- [Amazon Linux の入手可能性 \(p. 166\)](#)
- [Amazon Linux インスタンスへの接続 \(p. 166\)](#)

- [Amazon Linux イメージの特定 \(p. 166\)](#)
- [AWS コマンドラインツール \(p. 167\)](#)
- [パッケージリポジトリ \(p. 168\)](#)
- [Extras Library \(Amazon Linux 2\) \(p. 170\)](#)
- [参照のためのソースパッケージへのアクセス \(p. 171\)](#)
- [cloud-init \(p. 171\)](#)
- [Amazon Linux 通知にサブスクライブする \(p. 173\)](#)
- [Amazon Linux 2 を仮想マシンとしてオンプレミスで実行する \(p. 174\)](#)

## Amazon Linux の入手可能性

AWS は Amazon Linux 2 と Amazon Linux AMI を提供しています。別の Linux ディストリビューションから Amazon Linux に移行する場合、Amazon Linux 2 に移行することをお勧めします。

Amazon Linux AMI の最後のバージョンである 2018.03 は、2020 年 12 月 31 日に標準サポートが終了します。詳細については、ブログ投稿「[Amazon Linux AMI のサポート終了](#)」を参照してください。現在 Amazon Linux AMI を使用している場合は、Amazon Linux 2 に移行することをお勧めします。Amazon Linux 2 に移行するには、インスタンスを起動するか、現在の Amazon Linux 2 イメージから仮想マシンを作成します。アプリケーションと必要なパッケージをインストールします。アプリケーションをテストし、Amazon Linux 2 で実行するために必要な変更を加えます。

詳細については、「[Amazon Linux 2](#)」および「[Amazon Linux AMI](#)」を参照してください。Amazon Linux Docker コンテナイメージの場合、Docker Hub の「[amazonlinux](#)」を参照してください。

## Amazon Linux インスタンスへの接続

デフォルトでは、Amazon Linux はリモートルート SSH を許可しません。また、パスワード認証は、パスワードのブルートフォース攻撃を防ぐために無効になっています。Amazon Linux インスタンスへの SSH ログインを有効にするには、起動時にキーペアをインスタンスに提供する必要があります。インスタンスを起動するときに使用するセキュリティグループで、SSH アクセスを許可するよう設定する必要があります。デフォルトでは、SSH を使用してリモートログインできる唯一のアカウントは ec2-user です。このアカウントには sudo 特権もあります。リモートルートログインを有効にする場合は、このログインが、キーペアおよびセカンダリユーザーを使用する場合よりも安全性が低いことに注意してください。

## Amazon Linux イメージの特定

各イメージには、そのイメージを特定する一意の /etc/image-id ファイルが含まれています。このファイルには、イメージに関する次の情報が含まれています。

- `image_name`、`image_version`、`image_arch` — イメージを作成するときに Amazon が使用したビルドレシピの値。
- `image_stamp` — イメージの作成中に生成されたランダムな一意の 16 進値。
- `image_date` — YYYYMMDDhhmmss 形式の、イメージを作成した UTC 時間
- `recipe_name`、`recipe_id` — イメージを作成するときに Amazon が使用したビルドレシピの名前と ID。

Amazon Linux には、インストールされている現在のリリースを示す /etc/system-release ファイルが含まれています。このファイルは、yum を使用して更新され、system-release RPM の一部です。

Amazon Linux には、CPE 仕様に従う、機械による読み取りが可能なバージョンの /etc/system-release も含まれています。「[/etc/system-release-cpe](#)」を参照してください。

## Amazon Linux 2

現在のバージョンの Amazon Linux 2 用の /etc/image-id の例を以下に示します。

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn2-ami-hvm"
image_version="2"
image_arch="x86_64"
image_file="amzn2-ami-hvm-2.0.20180810-x86_64.xfs.gpt"
image_stamp="8008-2abd"
image_date="20180811020321"
recipe_name="amzn2 ami"
recipe_id="c652686a-2415-9819-65fb-4dee-9792-289d-1e2846bd"
```

現在のバージョンの Amazon Linux 2 用の /etc/system-release の例を以下に示します。

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux 2
```

次は、Amazon Linux 2 の /etc/os-release の例です。

```
[ec2-user ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"
HOME_URL="https://amazonlinux.com/"
```

## Amazon Linux AMI

現在の Amazon Linux AMI 用の /etc/image-id の例を以下に示します。

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn-ami-hvm"
image_version="2018.03"
image_arch="x86_64"
image_file="amzn-ami-hvm-2018.03.0.20180811-x86_64.ext4.gpt"
image_stamp="cc81-f2f3"
image_date="20180811012746"
recipe_name="amzn ami"
recipe_id="5b283820-dc60-a7ea-d436-39fa-439f-02ea-5c802dbd"
```

現在の Amazon Linux AMI 用の /etc/system-release の例を以下に示します。

```
[ec2-user ~]$ cat /etc/system-release
Amazon Linux AMI release 2018.03
```

## AWS コマンドラインツール

Amazon Linux AMI または Amazon Linux 2 のデフォルトリポジトリには、AWS の統合および使用のための次のコマンドラインツールが含まれています。Amazon Linux AMI のパッケージの詳細なリストについては、「[Amazon Linux AMI 2017.09 パッケージ](#)」を参照してください。

- aws-amitools-ec2
- aws-apitools-as
- aws-apitools-cfn
- aws-apitools-ec2
- aws-apitools-elb
- aws-apitools-mon
- aws-cfn-bootstrap
- aws-cli

Amazon Linux 2 および最小バージョンの Amazon Linux (amzn-ami-minimal-\* および amzn2-ami-minimal-\*) には、これらのすべてのパッケージが含まれているとは限りません。ただし、これらのパッケージは、次のコマンドを使用してデフォルトリポジトリからインストールできます。

```
[ec2-user ~]$ sudo yum install -y package_name
```

IAM ロールを使用して起動したインスタンスの場合、認証情報ファイルのインストール後、AWS\_CREDENTIAL\_FILE、JAVA\_HOME、AWS\_PATH、PATH および製品固有の環境変数を準備するためのシンプルなスクリプトが組み込まれます。これにより、上記のツールの設定が簡素化されます。

また、次に説明するように、複数のバージョンの API と AMI ツールをインストールできるように、これらのツールの希望するバージョンにアクセスするシンボリックリンクを /opt/aws に配置しました。

/opt/aws/bin

インストールされている各ツールディレクトリにある、/bin ディレクトリへのシンボリックリンク。

/opt/aws/{apitools|amitools}

インストールされた最新バージョンにアタッチされている *name-version* フォームと *name* シンボリックリンクのディレクトリに製品がインストールされます。

/opt/aws/{apitools|amitools}/{*name*}/environment.sh

/etc/profile.d/aws-apitools-common.sh により使用され、EC2\_HOME などの製品固有の環境変数を設定します。

## パッケージリポジトリ

Amazon Linux 2 および Amazon Linux AMI は、各 Amazon EC2 の AWS リージョンでホストされているオンラインパッケージリポジトリと一緒に使用するように設計されています。これらのリポジトリは、Amazon Linux 2 および Amazon Linux AMI パッケージの継続的な更新だけでなく、何百もの一般的なオープンソースのサーバーアプリケーションへのアクセスを提供します。リポジトリはすべてのリージョンに存在し、yum 更新ツールを使用してアクセスできます。各リージョンでリポジトリをホストしているため、データ転送料金なしで、更新を迅速にデプロイできます。

Amazon Linux 2 および Amazon Linux AMI のセキュリティと機能強化は定期的に更新されます。インスタンスでデータまたはカスタム設定を保存する必要がない場合は、最新の AMI を使用して新しいインスタンスを開始できます。インスタンスでデータまたはカスタム設定を保存する必要がある場合は、Amazon Linux パッケージリポジトリを介してこれらのインスタンスを維持できます。これらのリポジトリには、更新されたすべてのパッケージが含まれます。実行中のインスタンスにこれらの更新を適用するよう選択できます。新しいバージョンの AMI がリリースされても、古いバージョンの AMI と更新パッケージは引き続き利用できます。

### Important

インスタンスがリポジトリにアクセスするには、インターネットへのアクセスが必要です。

パッケージをインストールするには、次のコマンドを使用します。

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

Amazon Linux AMI の場合、Enterprise Linux (EPEL) リポジトリの追加パッケージへのアクセスが設定されていますが、デフォルトでは有効になっていません。Amazon Linux 2 は EPEL リポジトリを使用するように設定されています。EPEL は、リポジトリのパッケージのほか、サードパーティ製のパッケージを提供します。AWS はサードパーティ製のパッケージをサポートしていません。EPEL レポジトリは、次のコマンドを使って有効にできます。

- 複数 Amazon Linux 2:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- Amazon Linux AMI の場合:

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

Amazon Linux に必要なアプリケーションが含まれていない場合は、アプリケーションを Amazon Linux インスタンスに直接インストールするだけです。Amazon Linux はパッケージ管理に RPM と yum を使用しています。これは、新しいアプリケーションをインストールする最も簡単な方法です。Amazon Linux 中央リポジトリで利用可能なアプリケーションは多数あるので、アプリケーションがそのリポジトリで利用できるかどうかを最初に必ず確認する必要があります。これらのアプリケーションは、Amazon Linux インスタンスに簡単に追加できます。

実行中の Amazon Linux インスタンスにアプリケーションをアップロードするには、scp または sftp を使用し、インスタンスにログオンしてアプリケーションを設定します。組み込みの cloud-init パッケージから PACKAGE\_SETUP アクションを使用して、インスタンスの起動時にアプリケーションをアップロードすることができます。詳細については、「[cloud-init \(p. 171\)](#)」を参照してください。

## セキュリティの更新

セキュリティの更新は、パッケージリポジトリと更新された AMI を使用して提供されます。セキュリティアラートは、[Amazon Linux セキュリティセンター](#)で公開されます。AWS セキュリティポリシーの詳細については、またはセキュリティの問題を報告するには、[AWS セキュリティセンター](#)にアクセスしてください。

Amazon Linux は、起動時にクリティカルまたは重要なセキュリティ更新をダウンロードおよびインストールするよう設定されています。起動後にユースケースに必要な更新を行うことをお勧めします。たとえば、起動時にすべての更新（セキュリティ更新だけでなく）を適用したり、各更新を評価してシステムに適用可能なものののみを適用することができます。これは、cloud-init 設定 repo\_upgrade を使用して制御されます。次の cloud-init 設定のスニペットは、インスタンス初期化に渡すユーザーデータテキストで設定を変更する方法を示しています。

```
#cloud-config
repo_upgrade: security
```

repo\_upgrade の有効な値は次のとおりです。

security

Amazon によってセキュリティ更新としてマークされた保留中のクリティカルまたは重要な更新を適用します。

bugfix

Amazon によってバグフィックスとしてマークされた更新を適用します。バグフィックスは大きなサイズの更新セットで、セキュリティ更新および他のさまざまな小さなバグに対する修正が含まれます。

all

分類に関係なく、使用できる適切な更新すべてを適用します。

none

起動時に更新をインスタンスに適用しません。

`repo_upgrade` のデフォルトの設定は `security` です。つまり、ユーザーデータに異なる値を指定しない場合、Amazon Linux はその時点でのインストールされたパッケージの起動時にセキュリティアップグレードを実行します。Amazon Linux はまた、`/etc/motd` ファイルを使用してログインする際に利用可能なアップデートの数をリストすることによって、インストールされたパッケージのアップデートを通知します。これらの更新をインストールするには、インスタンスで `sudo yum upgrade` を実行する必要があります。

## リポジトリの設定

Amazon Linux では、AMI はスナップショットとして扱われ、`yum update -y` の実行時には、最新のパッケージを常に提供するリポジトリおよび更新構造を備えています。

このリポジトリ構造は、Amazon Linux のあるバージョンから次のバージョンへのローリング更新を可能にするように、連続的な更新が配信されるように設定されています。たとえば、前のバージョンの Amazon Linux AMI ( 2017.09 やそれより前のバージョンなど ) からインスタンスを起動し、`yum update -y` を実行する場合でも、最新のパッケージが利用できます。

ローリング更新を無効にするには、`lock-on-launch` 機能を有効にします。Lock-on-launch 機能は、インスタンスが、指定したリリースの AMI からのみ更新を受け取るようにロックします。たとえば、2017.09 AMI に移行する準備ができるまでは、2018.03 AMI を起動したときに、2018.03 AMI より前にリリースされた更新のみを受け取るよう指定できます。

### Important

最新ではないリポジトリのバージョンに固定すると、それ以上の更新を受け取ることができません。更新を継続的に受け取るには、最新の AMI を使用するか、必ず最新とされているリポジトリで AMI を更新する必要があります。

新しいインスタンスで `lock-on-launch` を有効にするには、次のユーザーデータを `cloud-init` に渡してインスタンスを起動します。

```
#cloud-config
repo_releasever: 2017.09
```

既存のインスタンスを現在の AMI バージョンにロックするには

1. 編集 `/etc/yum.conf`.
2. `releasever=latest` をコメントアウトします。
3. キャッシュをクリアするには、`yum clean all` を実行します。

## Extras Library (Amazon Linux 2)

Amazon Linux 2 では、Extras Library を使用してアプリケーションおよびソフトウェア更新をインスタンスにインストールできます。このようなソフトウェア更新は、トピックと呼ばれます。特定のバージョン

のトピックをインストールしたり、最新バージョンを使用するためにバージョン情報を省略したりすることができます。

使用可能なトピックのリストを表示するには、次のコマンドを使用します。

```
[ec2-user ~]$ amazon-linux-extras list
```

トピックを有効にし、パッケージの最新バージョンをインストールして最新の状態を維持するには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo amazon-linux-extras install topic
```

トピックを有効にし、パッケージの特定のバージョンをインストールして安定性を確保するには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo amazon-linux-extras install topic=version topic=version
```

## 参照のためのソースパッケージへのアクセス

Amazon Linux で提供されているツールを使用して、インスタンスにインストールしたパッケージソースを参照するために表示できます。ソースパッケージは、Amazon Linux およびオンラインパッケージリポジトリに含まれるすべてのパッケージで利用できます。インストールするソースパッケージのパッケージ名を確認し、`yumdownloader --source` コマンドを使用して実行中のインスタンス内にソースを表示します。例:

```
[ec2-user ~]$ yumdownloader --source bash
```

ソース RPM は解凍できます。そして、標準の RPM ツールを使用して、参照するためにソースツリーを表示できます。デバッグが完了したら、パッケージを利用できます。

## cloud-init

cloud-init パッケージは、Canonical によって構築されたオープンソースアプリケーションであり、Amazon EC2 などのクラウドコンピューティング環境で Linux イメージをブートストラップするときに使用されます。Amazon Linux にはカスタマイズされたバージョンの cloud-init が含まれています。これにより、起動時のインスタンスに対するアクションを指定することができます。インスタンスの起動時に、ユーザーデータフィールドを使用して必要なアクションを cloud-init に渡すことができます。つまり、さまざまなユースケースに対して共通の AMI を使用し、起動時にその AMI を動的に設定できます。Amazon Linux はまた、ec2 ユーザーアカウントの初期設定を実行するために cloud-init を使用します。

詳細については、「[cloud-init ドキュメント](#)」を参照してください。

Amazon Linux は、`/etc/cloud/cloud.cfg.d` と `/etc/cloud/cloud.cfg` にある cloud-init アクションを使用します。独自の cloud-init アクションファイルを `/etc/cloud/cloud.cfg.d` に作成することができます。このディレクトリ内のすべてのファイルは、cloud-init で読み取られます。それらは辞書と同じ順序に読み取られ、後のファイルは以前のファイルの値を上書きします。

cloud-init パッケージは、起動時にインスタンスのこれらの（およびその他の）共通の設定タスクを実行します。

- ・デフォルトのロケールを設定。
- ・ホスト名を設定。
- ・ユーザーデータの解析と処理。

- ホストプライベート SSH キーの生成。
- 容易にログインおよび管理できるように、ユーザーのパブリック SSH キーを `.ssh/authorized_keys` に追加する。
- パッケージ管理のためにリポジトリを準備する
- ユーザーデータで定義されたパッケージアクションの処理。
- ユーザーデータにあるユーザースクリプトの実行。
- インスタンスストアボリュームをマウントする（該当する場合）
  - デフォルトでは、`ephemeral0` インスタンスストアボリュームがある場合は `/media/ephemeral0` にマウントされ、有効なファイルシステムが含まれます。それ以外の場合は、マウントされません。
  - デフォルトでは、インスタンスに関連付けられたスワップボリュームがマウントされます (`m1.small` および `c1.medium` インスタンスタイプの場合のみ)。
  - 次の cloud-init ディレクティブを使用して、デフォルトのインスタンスストアボリュームマウントを上書きすることができます。

```
#cloud-config
mounts:
- [ ephemeral0 ]
```

マウントをより詳細にコントロールするには、cloud-init ドキュメントの「[マウント](#)」を参照してください。

- TRIM をサポートするインスタンスストアボリュームは、インスタンスの起動時にはフォーマットされないため、マウントする前にパーティション化してフォーマットする必要があります。詳細については、「[インスタンスストアボリュームの TRIM のサポート \(p. 1087\)](#)」を参照してください。`disk_setup` モジュールを使用して、起動時にインスタンスストアボリュームをパーティションおよびフォーマットすることができます。詳細については、cloud-init ドキュメントの「[Disk Setup](#)」を参照してください。

## サポートされているユーザーデータ形式

cloud-init パッケージは、さまざまな形式のユーザーデータを処理できます。

- Gzip
  - ユーザーデータが `gzip` で圧縮されている場合、cloud-init はデータを解凍し、適切に処理します。
- MIME マルチパート
  - MIME マルチパートファイルを使用して、複数のデータタイプを指定できます。たとえば、ユーザーデータスクリプトとクラウド設定タイプの両方を指定できます。マルチパートファイルのパートの形式が、サポートされている形式のいずれかの場合、そのパートは cloud-init で処理できます。
- Base64 デコード
  - ユーザーデータが `base64` でエンコードされている場合、cloud-init は、デコードされたデータをサポートされているタイプのいずれかとして認識できるか確認します。デコードされたデータを認識できる場合、データをデコードし、適切に処理します。認識できない場合、`base64` データは変更されません。
- ユーザーデータスクリプト
  - 「`#!`」または「`Content-Type: text/x-shellscript`」で始まります。
  - このスクリプトは、初回の起動サイクル時に `/etc/init.d/cloud-init-user-scripts` によって実行されます。これは起動プロセスの後半（初期設定アクションが実行された後）に実行されます。
- インクルードファイル
  - 「`#include`」または「`Content-Type: text/x-include-url`」で始まります。
  - このコンテンツはインクルードファイルです。ファイルには URL の一覧（1行に1つの URL）が含まれます。各 URL が読み取られ、そのコンテンツが同じルールセットを使用して渡されます。URL から読み取られたコンテンツは `gzip`、MIME マルチパート、またはプレーンテキスト形式になります。

- クラウド設定データ
  - 「#cloud-config」または「Content-Type: text/cloud-config」で始まります。
  - このコンテンツはクラウド設定データです。サポートされている設定形式のコメント付きサンプルについて、例を参照してください。
- Upstart ディレクトリ
  - 「#upstart-job」または「Content-Type: text/upstart-job」で始まります。
  - このコンテンツは /etc/init のファイルに格納され、upstart は他の upstart ディレクトリごとにコンテンツを消費します。
- クラウドブートフック
  - 「#cloud-boothook」または「Content-Type: text/cloud-boothook」で始まります。
  - このコンテンツはブートフックデータです。このデータは /var/lib/cloud にあるファイルに保存され、すぐに実行されます。
  - これは最初に使用可能な「フック」です。1回だけ実行するためのメカニズムはありません。ブートフックは自身でこの点に対処する必要があります。環境変数 INSTANCE\_ID でインスタンス ID が指定されています。この変数を使用して、インスタンスあたり1つのブートフックデータのセットを提供します。

## Amazon Linux 通知にサブスクライブする

新しい AMI が使用可能になったときに通知を受けるには、Amazon SNS を使用してサブスクライブします。

Amazon Linux の通知をサブスクライブするには

1. Amazon SNS コンソール (<https://console.aws.amazon.com/sns/v3/home>) を開きます。
2. ナビゲーションバーで、必要に応じてリージョンを [米国東部 (バージニア北部)] に変更します。購読する SNS 通知が作成されたリージョンを選択する必要があります。
3. ナビゲーションペインで、[Subscriptions]、[Create subscription] の順に選択します。
4. [サブスクリプションの作成] ダイアログボックスで、次の操作を行います。
  - a. [Amazon Linux 2] [トピックの ARN] には、以下の Amazon リソースネーム (ARN) をコピーして貼り付けます。**arn:aws:sns:us-east-1:137112412989:amazon-linux-2-ami-updates**
  - b. [Amazon Linux] [トピックの ARN] には、以下の Amazon リソースネーム (ARN) をコピーして貼り付けます。**arn:aws:sns:us-east-1:137112412989:amazon-linux-ami-updates**
  - c. [Protocol] で [Email] を選択します。
  - d. [Endpoint] に、通知を受信するために使用できる E メールアドレスを入力します。
  - e. [Create subscription] を選択します。
5. 「AWS Notification - Subscription Confirmation」という件名の確認メールを受け取ります。メールを開いて [Confirm subscription] を選択して受信登録を完了します。

AMI がリリースされるごとに、対応するトピックの受信者に通知が送信されます。このような通知を停止するには、以下の手順を使用してサブスクリプション解除します。

Amazon Linux の通知の受信登録を解除するには

1. Amazon SNS コンソール (<https://console.aws.amazon.com/sns/v3/home>) を開きます。
2. ナビゲーションバーで、必要に応じてリージョンを [米国東部 (バージニア北部)] に変更します。SNS 通知が作成されたリージョンを使用する必要があります。
3. ナビゲーションペインで、[サブスクリプション] を選択し、サブスクリプションを選択したら、[アクション]、[サブスクリプションの削除] の順に選択します。

- 
4. 確認を求めるメッセージが表示されたら、[削除] を選択します。

## Amazon Linux 2 を仮想マシンとしてオンプレミスで実行する

オンプレミスの開発とテストには、Amazon Linux 2 仮想マシン (VM) イメージを使用します。これらのイメージは、以下の仮想化プラットフォームで使用できます。

- VMWare
- KVM
- VirtualBox (Oracle VM)
- Microsoft Hyper-V

サポートされているいずれかの仮想プラットフォームで Amazon Linux 2 仮想マシンイメージを使用するには、次の操作を行います。

- ステップ 1: `seed.iso` 起動イメージを準備する (p. 174)
- ステップ 2: Amazon Linux 2 VM イメージのダウンロード (p. 176)
- ステップ 3: 新しい VM を起動して接続する (p. 176)

### ステップ 1: `seed.iso` 起動イメージを準備する

`seed.iso` 起動イメージには、新しい VM の起動に必要な初期設定情報 (例: ネットワーク設定、ホスト名、ユーザーデータ) が含まれます。

Note

`seed.iso` 起動イメージには、VM の起動に必要な設定情報のみ含まれています。Amazon Linux 2 オペレーティングシステムファイルは含まれていません。

`seed.iso` 起動イメージを生成するには、2 つの設定ファイルが必要です。

- `meta-data` — このファイルには、VM のホスト名と静的ネットワーク設定が含まれます。
- `user-data` — このファイルを使用して、ユーザー アカウントを設定し、パスワード、キーペア、およびアクセスメカニズムを指定します。デフォルトでは、Amazon Linux 2 VM イメージでは、`ec2-user` のユーザー アカウントを作成します。デフォルトのユーザー アカウントのパスワードを設定するには、`user-data` 設定ファイルを使用します。

#### `seed.iso` 起動ディスクを作成するには

1. `seedconfig` という名前の新しいフォルダを作成し、そのフォルダに移動します。
2. `meta-data` 設定ファイルを作成します。
  - a. `meta-data` という名前の新しいファイルを作成します。
  - b. 任意のテキストエディタを使用して `meta-data` ファイルを開き、以下を追加します。

```
local-hostname: vm_hostname
# eth0 is the default network interface enabled in the image. You can configure
static network settings with an entry like the following.
network-interfaces: |
    auto eth0
    iface eth0 inet static
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
Amazon Linux 2 を仮想マシン  
としてオンプレミスで実行する

```
address 192.168.1.10
network 192.168.1.0
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.1.254
```

**vm\_hostname** を任意の VM ホスト名に置き換え、必要に応じてネットワーク設定を行います。

- c. meta-data 設定ファイルを保存して閉じます。

VM ホスト名 (amazonlinux.onprem) を指定し、デフォルトのネットワークインターフェイス (eth0) を構成し、必要なネットワークデバイスの静的 IP アドレスを指定する meta-data 構成ファイルの例については、[サンプルの Seed.iso ファイル](#)を参照してください。

3. user-data 設定ファイルを作成します。

- a. user-data という名前の新しいファイルを作成します。
- b. 任意のテキストエディタを使用して user-data ファイルを開き、以下を追加します。

```
#cloud-config
#vim:syntax=yaml
users:
# A user by the name `ec2-user` is created in the image by default.
- default
chpasswd:
  list: |
    ec2-user:plain_text_password
# In the above line, do not add any spaces after 'ec2-user:'.
```

**plain\_text\_password** を、デフォルトの ec2-user ユーザーアカウントの任意のパスワードに置き換えます。

- c. (オプション) デフォルトでは、cloud-init は VM が起動される度にネットワーク設定に適用されます。ブート起動時の cloud-init によるネットワーク設定の適用を無効にし、最初の起動時のネットワーク設定を保持するには、以下を追加します。

```
# NOTE: Cloud-init applies network settings on every boot by default. To retain
network settings from first
boot, add following 'write_files' section:
write_files:
- path: /etc/cloud/cloud.cfg.d/80_disable_network_after_firstboot.cfg
  content: |
    # Disable network configuration after first boot
    network:
      config: disabled
```

- d. user-data 設定ファイルを保存して閉じます。

また、他のユーザーアカウントを作成して、アクセスメカニズム、パスワード、およびキーペアを指定することもできます。サポートされるディレクティブについては、「[モジュール](#)」を参照してください。3人のユーザーを追加で作成し、デフォルトの ec2-user ユーザーアカウントのカスタムパスワードを指定する user-data ファイルの例については、[サンプル Seed.iso ファイル](#)を参照してください。

4. seed.iso および meta-data 設定ファイルを使用して、user-data 起動イメージを作成します。

Linux の場合は、genisoimage などのツールを使用します。seedconfig フォルダに移動し、次のコマンドを実行します。

```
$ genisoimage -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

macOS の場合は、hdutil などのツールを使用します。seedconfig フォルダの 1 つ上に移動し、次のコマンドを実行します。

```
$ hdutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata  
seedconfig/
```

## ステップ 2: Amazon Linux 2 VM イメージのダウンロード

サポートされている仮想化プラットフォームごとに異なる Amazon Linux 2 VM イメージをお使いいただけます。選択したプラットフォームに合う VM イメージをダウンロードします。

- [VMWare](#)
- [KVM](#)
- [Oracle VirtualBox](#)
- [Microsoft Hyper-V](#)

## ステップ 3: 新しい VM を起動して接続する

新しい VM を起動して接続するには、seed.iso 起動イメージ (ステップ 1 で作成済み) と Amazon Linux 2 VM イメージ (ステップ 2 でダウンロード済み) が必要です。ステップは、選択した VM プラットフォームによって異なります。

最初の起動時に seed.iso 起動イメージを VM に接続する必要があります。seed.iso は最初の起動時のみ評価されます。

VM を起動したら、user-data 設定ファイルで定義されているいざれかのユーザーアカウントを使用してログインします。VMWare 以外の仮想化プラットフォームでは、seed.iso 起動イメージは、VM の初回ログイン後に切断することができます。

# ユーザー提供カーネル

Amazon EC2 インスタンスでカスタムカーネルが必要な場合は、必要としているものに近い AMI でインスタンスを起動し、カスタムカーネルをインスタンス上でコンパイルして、新しいカーネルを参照するように menu.lst ファイルを変更します。このプロセスは AMI が使用する仮想化タイプによって異なります。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。

## コンテンツ

- [HVM AMI \(GRUB\) \(p. 176\)](#)
- [AMI の準仮想化 \(PV-GRUB\) \(p. 177\)](#)

## HVM AMI (GRUB)

HVM インスタンスボリュームは実際の物理ディスクのように扱われます。起動プロセスは、パーティション分割ディスクとブートローダーを備えるベアメタルオペレーティングシステムの起動プロセスに似ています。ブートローダーは現在サポートされているすべての Linux ディストリビューションで使用できます。もっとも一般的なブートローダーは GRUB です。以下のセクションでは、カスタムカーネルを使用するための GRUB の設定について説明します。

## HVM AMI 向けの GRUB の設定

次の例は、HVM AMI 向けの menu.lst 設定ファイルです。この例では、Amazon Linux 2018.03 (この AMI の元々のカーネル) と Vanilla Linux 4.16.4 (<https://www.kernel.org/>) の新しいバージョンの Vanilla Linux カーネル) の 2 つのカーネルエントリを選択できます。Vanilla エントリは、この AMI の元々のエントリからコピーされました。kernel と initrd パスは新しい場所に更新されました。[default 0] パラメータは、ブートローダーをそれが検出した最初のエントリ (この場合、Vanilla エントリ) にポイントします。fallback 1 パラメータは、最初のエントリの起動に問題が発生した場合、次のエントリにブートローダーをポイントします。

起動が遅くなるため、デフォルトでは GRUB はインスタンスのコンソールに出力を送信しません。詳細については、「[インスタンスコンソール出力 \(p. 1168\)](#)」を参照してください。カスタムカーネルをインストールする場合は、以下の例に示すように、hiddenmenu 行を削除して serial 行および terminal 行を /boot/grub/menu.lst に追加し、GRUB 出力を有効にすることを検討してください。

### Important

起動処理中に大量のデバッグ情報を表示することは避けてください。シリアルコンソールは高速データ転送をサポートしていません。

```
default=0
fallback=1
timeout=5
serial --unit=0 --speed=9600
terminal --dumb --timeout=5 serial console

title Vanilla Linux 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
initrd /boot/initramfs-4.14.26-46.32.amzn1.x86_64.img
```

menu.lst ファイルにフォールバックカーネルを指定する必要はありません。ただし、新しいカーネルをテストするときは、フォールバックを設定することをお勧めします。GRUB では、新しいカーネルにエラーがあった場合に別のカーネルにフォールバックできます。フォールバックカーネルを設定すると、新しいカーネルが見つからない場合でもインスタンスを起動できます。

新しい Vanilla Linux カーネルが失敗した場合、出力は次の例のようになります。

```
^M Entry 0 will be booted automatically in 3 seconds. ^M Entry 0 will be booted
automatically in 2 seconds. ^M Entry 0 will be booted automatically in 1 seconds.

Error 13: Invalid or unsupported executable format
[ 0.000000] Initializing cgroup subsys cpuset
```

## AMI の準仮想化 (PV-GRUB)

準仮想化 (PV) を使用する Amazon マシンイメージでは、起動プロセスで PV-GRUB と呼ばれるシステムが利用されます。PV-GRUB は、パッチが適用されたバージョンの GNU GRUB 0.97 を実行する準仮想化ブートローダーです。インスタンスを起動すると、PV-GRUB では起動プロセスが開始され、お客様のイメージの menu.lst ファイルが指定するカーネルがチェーンロードされます。

PV-GRUB は標準の grub.conf または menu.lst コマンドを認識しますこれにより、現在サポートされているすべての Linux ディストリビューションとともに利用できます。Ubuntu 10.04 LTS、Oracle

Enterprise Linux、CentOS 5.x など、古いディストリビューションでは特別な「ec2」や「xen」カーネルパッケージが必要です。新しいディストリビューションでは、デフォルトのカーネルパッケージに必要なドライバーが含まれています。

最新の準仮想 AMI では、デフォルトで PV-GRUB AKI を使用します (Amazon EC2 Launch Wizard Quick Start メニューで利用できるすべての準仮想 Linux AMI が含まれています)。そのため、使用するカーネルにディストリビューションとの互換性がある場合、インスタンスで別のカーネルを使用するために必要な追加の手順はありません。インスタンスでカスタムカーネルを実行するには、必要としているものに近い AMI でインスタンスを起動する方法が最適です。この方法では、カスタムカーネルをインスタンス上でコンパイルして、「[GRUB の設定 \(p. 178\)](#)」で説明されているように、そのカーネルで起動するように menu.lst ファイルを変更します。

AMI のカーネルイメージが PV-GRUB AKI であることを Amazon EC2 コマンドラインツール (チェックするカーネルイメージ ID を代用します) で次の `describe-images` コマンドを実行して検証できます。

```
aws ec2 describe-images --filters Name=image-id,Values=aki-880531cd
```

Name フィールドが `pv-grub` で始まるかどうかを確認します。

#### トピック

- [PV-GRUB の制約事項 \(p. 178\)](#)
- [準仮想化 AMI 向けの GRUB の設定 \(p. 178\)](#)
- [Amazon PV-GRUB カーネルイメージ ID \(p. 179\)](#)
- [PV-GRUB の更新 \(p. 181\)](#)

## PV-GRUB の制約事項

PV-GRUB には次の制約事項があります。

- PV-GRUB の 64 ビットバージョンを使用して 32 ビットカーネルを起動したり、PV-GRUB の 32 ビットバージョンを使用して 64 ビットカーネルを起動したりすることはできません。
- PV-GRUB AKI の使用時には、Amazon ラムディスクイメージ (ARI) を指定できません。
- AWS は、PV-GRUB が EXT2、EXT3、EXT4、JFS、XFS、ReiserFS のファイルシステム形式で動作することをテストし、確認しています。その他のファイルシステム形式では動作しない場合があります。
- PV-GRUB は、gzip、bzip2、lzo、xz 圧縮形式を利用して圧縮されたカーネルを起動できます。
- Cluster AMI は PV-GRUB をサポートせず、また、必要としません。完全ハードウェア仮想化 (HVM) が使用されるためです。準仮想インスタンスは PV-GRUB を使用して起動します。一方、HVM インスタンスピリュームは実際のディスクのように扱われ、その起動プロセスはパーティション分割ディスクとブートローダーを備えるペアメタルオペレーティングシステムの起動プロセスに似ています。
- PV-GRUB バージョン 1.03 以前では、GPT パーティショニングをサポートしません。MBR パーティショニングがサポートされています。
- Amazon EBS で Logical Volume Manager (LVM) を使用する場合、LVM の外側に別の起動パーティションが必要です。その場合、LVM で論理ボリュームを作成できます。

## 準仮想化 AMI 向けの GRUB の設定

PV-GRUB を起動するには、GRUB menu.lst ファイルがイメージに含まれている必要があります。このファイルの最も一般的な場所は `/boot/grub/menu.lst` です。

次の例は、PV-GRUB AKI を使用して AMI を起動する menu.lst 設定ファイルです。この例では、Amazon Linux 2018.03 (この AMI の元々のカーネル) と Vanilla Linux 4.16.4 (<https://www.kernel.org/>) の新しいバージョンの Vanilla Linux カーネル) の 2 つのカーネルエントリを選択できます。Vanilla エン

トリは、この AMI の元々のエントリからコピーされました。kernel と initrd パスは新しい場所に更新されました。default 0 パラメータは、ブートローダーをそれが検出した最初のエントリ (この場合、Vanilla エントリ) にポイントします。fallback 1 パラメータは、最初のエントリの起動に問題が発生した場合、次のエントリにブートローダーをポイントします。

```
default 0
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-4.14.26-46.32.amzn1.x86_64.img
```

menu.lst ファイルにフォールバックカーネルを指定する必要はありません。ただし、新しいカーネルをテストするときは、フォールバックを設定することをお勧めします。PV-GRUB では、新しいカーネルにエラーがあった場合に別のカーネルにフォールバックできます。フォールバックカーネルを設定すると、新しいカーネルが見つからない場合でもインスタンスを起動できます。

PV-GRUB は、次の場所で menu.lst をチェックします。その際、それが検出した最初の場所が利用されます。

- (hd0)/boot/grub
- (hd0,0)/boot/grub
- (hd0,0)/grub
- (hd0,1)/boot/grub
- (hd0,1)/grub
- (hd0,2)/boot/grub
- (hd0,2)/grub
- (hd0,3)/boot/grub
- (hd0,3)/grub

PV-GRUB 1.03 以前では、このリストの最初の 2 つの場所うちの 1 つのみがチェックされることに注意してください。

## Amazon PV-GRUB カーネルイメージ ID

PV-GRUB AKI はすべての Amazon EC2 リージョンで利用できます。32 ビットと 64 ビットの両方のアーキテクチャタイプに AKI があります。最新の AMI では、デフォルトで PV-GRUB AKI が使用されます。

すべてのバージョンの PV-GRUB AKI がすべてのインスタンスタイプと互換性があるとは限らないため、常に最新バージョンの PV-GRUB AKI を使用することをお勧めします。次の [describe-images](#) コマンドを使用し、現在のリージョンの PV-GRUB AKI のリストを取得します。

```
aws ec2 describe-images --owners amazon --filters Name=name,Values=pv-grub-* .gz
```

PV-GRUB は、ap-southeast-2 リージョンで利用できる唯一の AKI であることにご注意ください。このリージョンにコピーする AMI が、このリージョンで利用できる PV-GRUB のバージョンを使用していることを確認してください。

各リージョンの現在の AKI ID は次のとおりです。新しい AMI は、hd0 AKI を使用して登録します。

Note

以前 hd00 AKI が利用可能であった地域では、後方互換性のために引き続き提供されます。

ap-northeast-1、アジアパシフィック (東京)

イメージ ID	イメージ名
aki-f975a998	pv-grub-hd0_1.05-i386.gz
aki-7077ab11	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-1、アジアパシフィック (シンガポール) リージョン

イメージ ID	イメージ名
aki-17a40074	pv-grub-hd0_1.05-i386.gz
aki-73a50110	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-2、アジアパシフィック (シドニー)

イメージ ID	イメージ名
aki-ba5665d9	pv-grub-hd0_1.05-i386.gz
aki-66506305	pv-grub-hd0_1.05-x86_64.gz

eu-central-1、欧州 (フランクフルト)

イメージ ID	イメージ名
aki-1419e57b	pv-grub-hd0_1.05-i386.gz
aki-931fe3fc	pv-grub-hd0_1.05-x86_64.gz

eu-west-1、欧州 (アイルランド)

イメージ ID	イメージ名
aki-1c9fd86f	pv-grub-hd0_1.05-i386.gz
aki-dc9ed9af	pv-grub-hd0_1.05-x86_64.gz

sa-east-1、南米 (サンパウロ)

イメージ ID	イメージ名
aki-7cd34110	pv-grub-hd0_1.05-i386.gz
aki-912fbcf9	pv-grub-hd0_1.05-x86_64.gz

us-east-1、米国東部 (バージニア北部)

イメージ ID	イメージ名
aki-04206613	pv-grub-hd0_1.05-i386.gz
aki-5c21674b	pv-grub-hd0_1.05-x86_64.gz

us-gov-west-1, AWS GovCloud (US-West)

イメージ ID	イメージ名
aki-5ee9573f	pv-grub-hd0_1.05-i386.gz
aki-9ee55bff	pv-grub-hd0_1.05-x86_64.gz

us-west-1、米国西部 (北カリフォルニア)

イメージ ID	イメージ名
aki-43cf8123	pv-grub-hd0_1.05-i386.gz
aki-59cc8239	pv-grub-hd0_1.05-x86_64.gz

us-west-2、米国西部 (オレゴン)

イメージ ID	イメージ名
aki-7a69931a	pv-grub-hd0_1.05-i386.gz
aki-70cb0e10	pv-grub-hd0_1.05-x86_64.gz

## PV-GRUB の更新

すべてのバージョンの PV-GRUB AKI がすべてのインスタンスタイプと互換性があるとは限らないため、常に最新バージョンの PV-GRUB AKI を使用することをお勧めします。また、古いバージョンの PV-GRUB はすべてのリージョンで使用できるわけではないため、旧バージョンを使用する AMI を、そのバージョンをサポートしないリージョンにコピーした場合、カーネルのイメージを更新するまで、その AMI から起動されたインスタンスを起動できなくなります。次の手順を使用してインスタンスの PV-GRUB のバージョンを確認し、必要に応じて更新します。

### PV-GRUB のバージョンを確認するには

1. インスタンスのカーネル ID を見つけます。

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute kernel --region region

{
    "InstanceId": "instance_id",
    "KernelId": "aki-70cb0e10"
}
```

このインスタンスのカーネル ID は、aki-70cb0e10 です。

2. このカーネル ID のバージョン情報を表示します。

```
aws ec2 describe-images --image-ids aki-70cb0e10 --region region

{
    "Images": [
        {
            "VirtualizationType": "paravirtual",
            "Name": "pv-grub-hd0_1.05-x86_64.gz",
            ...
            "Description": "PV-GRUB release 1.05, 64-bit"
        }
    ]
}
```

このカーネルイメージは PV-GRUB 1.05 です。PV-GRUB のバージョンが最新バージョン ([Amazon PV-GRUB カーネルイメージ ID \(p. 179\)](#) を参照) でない場合、次の手順を使用して更新する必要があります。

#### PV-GRUB のバージョンを更新するには

インスタンスが古いバージョンの PV-GRUB を使用している場合は、最新バージョンに更新する必要があります。

1. [Amazon PV-GRUB カーネルイメージ ID \(p. 179\)](#) で、使用するリージョンとプロセッサーアーキテクチャーの最新の PV-GRUB AKI を特定します。
2. インスタンスを停止します。使用されるカーネルイメージを変更するには、インスタンスを停止する必要があります。

```
aws ec2 stop-instances --instance-ids instance_id --region region
```

3. インスタンスに使用するカーネルイメージを変更します。

```
aws ec2 modify-instance-attribute --instance-id instance_id --kernel kernel_id --region region
```

4. インスタンスを再起動します。

```
aws ec2 start-instances --instance-ids instance_id --region region
```

# Amazon EC2 インスタンス

Amazon EC2 を初めて使用する場合は、次のトピックを参照して使用を開始してください。

- [Amazon EC2 とは \(p. 1\)](#)
- [Amazon EC2 でのセットアップ \(p. 22\)](#)
- [Amazon EC2 Linux インスタンスの開始方法 \(p. 26\)](#)
- [インスタンスのライフサイクル \(p. 443\)](#)

実稼働環境を起動する前に、以下の質問に答える必要があります。

Q.ニーズに最も合っているインスタンスタイプはどれですか。

Amazon EC2 には、アプリケーションを実行するために必要な CPU、メモリ、ストレージ、ネットワークキャパシティーを選択できるようにするため、さまざまなインスタンスタイプが用意されています。詳細については、「[インスタンスタイプ \(p. 183\)](#)」を参照してください。

Q.ニーズに最も合っている購入オプションはどれですか。

Amazon EC2 では、オンデマンドインスタンス (デフォルト)、スポットインスタンス、および リザーブドインスタンスをサポートします。詳細については、「[インスタンス購入オプション \(p. 274\)](#)」を参照してください。

Q.ニーズに合っているルートボリュームのタイプはどれですか。

各インスタンスは Amazon EBS またはインスタンスストアによってサポートされています。必要なルートボリュームのタイプに基づいて AMI を選択します。詳細については、「[ルートデバイスのストレージ \(p. 96\)](#)」を参照してください。

Q.EC2 インスタンス群およびハイブリッド環境のマシンを管理できますか。

AWS Systems Manager では、Amazon EC2 インスタンス、オンプレミスのインスタンス、および他のクラウドプロバイダーの VM を含むハイブリッド環境の仮想マシン (VM) の設定をセキュアにリモートで管理できます。詳細については、「[AWS Systems Manager ユーザーガイド](#)」を参照してください。

## インスタンスタイプ

インスタンスを起動するときは、指定したインスタンスタイプによって、インスタンスに使用するホストコンピュータのハードウェアが決まります。インスタンスタイプごとに、コンピューティング、メモリ、およびストレージの機能は異なり、これらの機能に基づいてインスタンスマリナーにグループ分けされます。インスタンスタイプは、インスタンス上で実行するアプリケーションやソフトウェアの要件に基づいて選択します。

Amazon EC2 は、基になっているハードウェアに関係なく、各インスタンスに、一貫した予測可能な CPU 能力を提供します。

Amazon EC2 は、CPU、メモリ、インスタンストレージなどのホストコンピュータの一部のリソースを、特定のインスタンス専用に割り当てます。Amazon EC2 は、ネットワークやディスクサブシステムなどの他のホストコンピュータリソースをインスタンス間で共有します。ホストコンピュータの各インスタンスが、これらの共有リソースの 1 つを可能な限り利用しようとする場合、それぞれのインスタンスは、そのリソースの共有分を等しく受け取ります。ただし、リソースの使用率が低い場合は、1 つのインスタンスがそのリソースのより多くの部分を利用できます。

各インスタンスタイプは、共有リソースからより高い、またはより低い最小性能を提供します。たとえば、高速の I/O パフォーマンスを実行するインスタンスタイプは、共有リソースに対してより大きな割り当てを取得します。共有リソースをより大きく配分することによって、I/O 性能のばらつきを抑えるこ

ともできます。ほとんどのアプリケーションでは、中程度の I/O 性能があれば十分です。ただし、より高い、またはより一貫した I/O パフォーマンスを必要とするアプリケーションの場合は、より I/O パフォーマンスの高いインスタンスタイプを使用することを検討してください。

## コンテンツ

- 利用可能なインスタンスタイプ (p. 184)
- ハードウェア仕様 (p. 186)
- AMI 仮想化タイプ (p. 186)
- Nitro ベースのインスタンス (p. 187)
- ネットワーキング機能とストレージ機能 (p. 187)
- インスタンス制限 (p. 190)
- 汎用インスタンス (p. 190)
- コンピュート最適化インスタンス (p. 231)
- メモリ最適化インスタンス (p. 236)
- ストレージ最適化インスタンス (p. 245)
- Linux 高速コンピューティングインスタンス (p. 253)
- インスタンスタイプの検索 (p. 266)
- インスタンスタイプを変更する (p. 267)
- インスタンスタイプに関する推奨事項の取得 (p. 271)

## 利用可能なインスタンスタイプ<sup>†</sup>

Amazon EC2 では、次の表で示すインスタンスタイプが提供されます。

### 現行世代のインスタンス

最適なパフォーマンスを得るために、新しいインスタンスを起動するときには、現行世代のインスタンスタイプを使用することをお勧めします。どのリージョンまたはアベイラビリティゾーンでどのインスタンスタイプが利用可能かを判断するには、[describe-instance-type-offerings](#) コマンドを使用します。インスタンスタイプの特性を取得するには、[describe-instance-types](#) コマンドを使用します。

現行世代のインスタンスタイプの詳細については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

インスタンスファミリー	現行世代のインスタンスタイプ <sup>†</sup>
汎用	a1.medium   a1.large   a1.xlarge   a1.2xlarge   a1.4xlarge     m4.large   m4.xlarge   m4.2xlarge   m4.4xlarge     m4.10xlarge   m4.16xlarge   m5.large   m5.xlarge     m5.2xlarge   m5.4xlarge   m5.8xlarge   m5.12xlarge     m5.16xlarge   m5.24xlarge   m5.metal   m5a.large     m5a.xlarge   m5a.2xlarge   m5a.4xlarge   m5a.8xlarge     m5a.12xlarge   m5a.16xlarge   m5a.24xlarge   m5ad.large     m5ad.xlarge   m5ad.2xlarge   m5ad.4xlarge   m5ad.8xlarge     m5ad.12xlarge   m5ad.16xlarge   m5ad.24xlarge     m5d.large   m5d.xlarge   m5d.2xlarge   m5d.4xlarge     m5d.8xlarge   m5d.12xlarge   m5d.16xlarge   m5d.24xlarge     m5d.metal   m5dn.large   m5dn.xlarge   m5dn.2xlarge     m5dn.4xlarge   m5dn.8xlarge   m5dn.12xlarge     m5dn.16xlarge   m5dn.24xlarge   m5n.large   m5n.xlarge     m5n.2xlarge   m5n.4xlarge   m5n.8xlarge   m5n.12xlarge     m5n.16xlarge   m5n.24xlarge   t2.nano   t2.micro

インスタンスファミリー	現行世代のインスタンスタイプ
	t2.small   t2.medium   t2.large   t2.xlarge   t2.2xlarge   t3.nano   t3.micro   t3.small   t3.medium   t3.large   t3.xlarge   t3.2xlarge   t3a.nano   t3a.micro   t3a.small   t3a.medium   t3a.large   t3a.xlarge   t3a.2xlarge
コンピューティングの最適化	c4.large   c4.xlarge   c4.2xlarge   c4.4xlarge   c4.8xlarge   c5.large   c5.xlarge   c5.2xlarge   c5.4xlarge   c5.9xlarge   c5.12xlarge   c5.18xlarge   c5.24xlarge   c5.metal   c5d.large   c5d.xlarge   c5d.2xlarge   c5d.4xlarge   c5d.9xlarge   c5d.12xlarge   c5d.18xlarge   c5d.24xlarge   c5d.metal   c5n.large   c5n.xlarge   c5n.2xlarge   c5n.4xlarge   c5n.9xlarge   c5n.18xlarge   c5n.metal
メモリ最適化	r4.large   r4.xlarge   r4.2xlarge   r4.4xlarge   r4.8xlarge   r4.16xlarge   r5.large   r5.xlarge   r5.2xlarge   r5.4xlarge   r5.8xlarge   r5.12xlarge   r5.16xlarge   r5.24xlarge   r5.metal   r5a.large   r5a.xlarge   r5a.2xlarge   r5a.4xlarge   r5a.8xlarge   r5a.12xlarge   r5a.16xlarge   r5a.24xlarge   r5ad.large   r5ad.xlarge   r5ad.2xlarge   r5ad.4xlarge   r5ad.12xlarge   r5ad.24xlarge   r5d.large   r5d.xlarge   r5d.2xlarge   r5d.4xlarge   r5d.8xlarge   r5d.12xlarge   r5d.16xlarge   r5d.24xlarge   r5d.metal   r5dn.large   r5dn.xlarge   r5dn.2xlarge   r5dn.4xlarge   r5dn.8xlarge   r5dn.12xlarge   r5dn.16xlarge   r5dn.24xlarge   r5n.large   r5n.xlarge   r5n.2xlarge   r5n.4xlarge   r5n.8xlarge   r5n.12xlarge   r5n.16xlarge   r5n.24xlarge   u-6tb1.metal   u-9tb1.metal   u-12tb1.metal   u-18tb1.metal   u-24tb1.metal   x1.16xlarge   x1.32xlarge   x1e.xlarge   x1e.2xlarge   x1e.4xlarge   x1e.8xlarge   x1e.16xlarge   x1e.32xlarge   z1d.large   z1d.xlarge   z1d.2xlarge   z1d.3xlarge   z1d.6xlarge   z1d.12xlarge   z1d.metal
ストレージの最適化	d2.xlarge   d2.2xlarge   d2.4xlarge   d2.8xlarge   h1.2xlarge   h1.4xlarge   h1.8xlarge   h1.16xlarge   i3.large   i3.xlarge   i3.2xlarge   i3.4xlarge   i3.8xlarge   i3.16xlarge   i3.metal   i3en.large   i3en.xlarge   i3en.2xlarge   i3en.3xlarge   i3en.6xlarge   i3en.12xlarge   i3en.24xlarge   i3en.metal
高速コンピューティング	f1.2xlarge   f1.4xlarge   f1.16xlarge   g3s.xlarge   g3.4xlarge   g3.8xlarge   g3.16xlarge   g4dn.xlarge   g4dn.2xlarge   g4dn.4xlarge   g4dn.8xlarge   g4dn.12xlarge   g4dn.16xlarge   p2.xlarge   p2.8xlarge   p2.16xlarge   p3.2xlarge   p3.8xlarge   p3.16xlarge   p3dn.24xlarge   inf1.xlarge   inf1.2xlarge   inf1.6xlarge   inf1.24xlarge

## 旧世代のインスタンス

お客様のアプリケーションが旧世代のインスタンス用に最適化されており、アップグレードはまだこれからという場合でも、Amazon Web Services では、それらの旧世代のインスタンスを提供しています。最高のパフォーマンスを得るために最新世代のインスタンスのご利用をお勧めしていますが、これらの旧世

代のインスタンスのサポートも継続します。現在、旧世代のインスタンスを使用している場合は、適切なアップグレードとなる最新世代のインスタンスを確認できます。詳細については、「[旧世代のインスタンス](#)」を参照してください。

インスタンスファミリー	旧世代のインスタンスタイプ
汎用	m1.small   m1.medium   m1.large   m1.xlarge   m3.medium   m3.large   m3.xlarge   m3.2xlarge   t1.micro
コンピューティングの最適化	c1.medium   c1.xlarge   cc2.8xlarge   c3.large   c3.xlarge   c3.2xlarge   c3.4xlarge   c3.8xlarge
メモリ最適化	m2.xlarge   m2.2xlarge   m2.4xlarge   cr1.8xlarge   r3.large   r3.xlarge   r3.2xlarge   r3.4xlarge   r3.8xlarge
ストレージの最適化	hs1.8xlarge   i2.xlarge   i2.2xlarge   i2.4xlarge   i2.8xlarge
高速コンピューティング	g2.2xlarge   g2.8xlarge

## ハードウェア仕様

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

お客様のニーズに最適なインスタンスタイプを決定するには、インスタンスを起動し、独自のベンチマークアプリケーションを使用することをお勧めします。支払いはインスタンス秒単位であるため、決定する前に複数のインスタンスタイプをテストすると、便利なうえ、コストを抑えることができます。

決定を行った後でも、ニーズが変化したときは、インスタンスのサイズを変更できます。詳細については、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。

### Note

Amazon EC2 インスタンスは、インスタンスタイプ製品ページで指定されているとおり、通常、64 ビット仮想 Intel プロセッサで実行されます。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。ただし、64 ビット CPU に関する業界の命名規則が原因でわかりにくくなっています。チップ製造元の Advanced Micro Devices (AMD) は、Intel x86 命令セットをベースとして商業的に初めて成功した 64 ビットアーキテクチャを導入しました。その結果、このアーキテクチャーはチップ製造元にかかわらず AMD64 と幅広く呼ばれています。Windows および複数の Linux ディストリビューションがこの慣習に従っています。インスタンスが Intel ハードウェアで実行されているにもかかわらず、Ubuntu または Windows EC2 インスタンスの内部システム情報に CPU アーキテクチャが AMD64 と表示されるのはこのためです。

## AMI 仮想化タイプ

インスタンスの仮想化タイプは、インスタンスの起動に使用する AMI によって決まります。現行世代のインスタンスタイプは、ハードウェア仮想マシン (HVM) のみをサポートしています。以前の世代のインスタンスタイプの中には、準仮想化 (PV) をサポートするものがあり、一部の AWS リージョンは PV インスタンスをサポートしています。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。

最適なパフォーマンスを得るために、HVM AMI を使用することをお勧めします。さらに、拡張ネットワーキングのメリットを活用するには、HVM AMI が必要です。HVM 仮想化は、AWS プラットフォームによって提供されるハードウェアアシストテクノロジーを使用します。HVM 仮想化を使用すると、ゲスト VM はネイティブハードウェアプラットフォーム上で動作しているかのように動作します。ただし、パフォーマンスの向上のために PV ネットワークとストレージドライバは使用します。

## Nitro ベースのインスタンス

Nitro システムは、高パフォーマンス、高可用性、高セキュリティを実現する AWS で構築されたハードウェアとソフトウェアのコンポーネントの集合です。また、ペアメタル機能を備えることで、仮想化オーバーヘッドを排除するとともに、ホストハードウェアへのフルアクセスを要求するワークロードをサポートします。

### Nitro コンポーネント

Nitro システムには、以下のコンポーネントが含まれます。

- Nitro ハイパー/バイザー - メモリと CPU の割り当てを管理し、ほとんどのワークロードのペアメタルと見分けがつかないようなパフォーマンスを提供する軽量ハイパー/バイザー。
- Nitro カード
  - ローカル NVMe ストレージボリューム
  - ネットワーキングハードウェアのサポート
  - 管理
  - モニタリング
  - セキュリティ
- Nitro セキュリティチップ (マザーボードに統合)

### インスタンスタイプ

以下のインスタンスが Nitro システムに基づいています。

- A1、C5、C5d、C5n、G4、I3en、Inf1、M5、M5a、M5ad、M5d、M5dn、M5n、p3dn.24xlarge、R5、R5a、R5ad および z1d
- ペアメタル: c5.metal, c5d.metal, c5n.metal, i3.metal, i3en.metal, m5.metal, m5d.metal, r5.metal, r5d.metal, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, u-24tb1.metal, and z1d.metal

### リソース

詳細については、以下の動画を参照してください。

- [AWS re:Invent 2017: The Amazon EC2 Nitro System Architecture](#)
- [AWS re:Invent 2017: Amazon EC2 Bare Metal Instances](#)
- [The Nitro Project: Next-Generation EC2 Infrastructure](#)

## ネットワーキング機能とストレージ機能

インスタンスタイプを選択すると、使用できるネットワーキング機能とストレージ機能が決まります。インスタンスタイプの情報を取得するには、[describe-instance-types](#) コマンドを使用します。

### ネットワーキング機能

- IPv6 は、現行世代のすべてのインスタンスタイプと、旧世代の C3、R3、I2 のインスタンスタイプでサポートされています。
- インスタンスタイプのネットワーキングと帯域幅のパフォーマンスを最大化するには、次のことを実行できます。
  - サポートされるインスタンスタイプをクラスター・プレイスメント・グループで起動し、ハイパフォーマンスコンピューティング (HPC) アプリケーション用にインスタンスを最適化します。共通のクラス

ターブレイスマントグループのインスタンスは、高帯域幅、低レイテンシーのネットワーキングから利点を得られます。詳細については、「[プレイスメントグループ \(p. 791\)](#)」を参照してください。

- サポートされる現行世代のインスタンスタイプ用の拡張ネットワーキングを有効にして、パケット毎秒 (PPS) のパフォーマンスを大幅に高め、ネットワークのストレスとレイテンシーを低減することができます。詳細については、「[Linux の拡張ネットワーキング \(p. 737\)](#)」を参照してください。
- 拡張ネットワーキングに対して有効になっている現行世代のインスタンスタイプには、次のネットワーキングパフォーマンス属性があります。
  - 同じリージョン内でのプライベート IPv4 または IPv6 を介したトラフィックでは、シングルフロートラフィックで 5 Gbps、マルチフロートラフィックで最大 25 Gbps をサポートしています（インスタンスタイプによって異なります）。
  - 同じリージョン内でのインスタンスと Amazon S3 バケットとの間では、パブリック IP アドレス空間または VPC エンドポイントを介したトラフィックに、使用可能なすべてのインスタンスの集計帯域幅を使用できます。
- サポートされている最大 MTU は、インスタンスタイプによって異なります。すべての Amazon EC2 インスタンスタイプは、標準イーサネット V2 1500 MTU フレームをサポートします。すべての現行世代のインスタンスは 9001 MTU、またはジャンボフレームをサポートし、一部の旧世代のインスタンスも同様にそれらをサポートします。詳細については、「[EC2 インスタンスの最大ネットワーク送信単位 \(MTU\) \(p. 801\)](#)」を参照してください。

## ストレージ機能

- インスタンスタイプの中には、EBS ボリュームとインスタンスストアボリュームをサポートするものや、EBS ボリュームのみをサポートするものがあります。インスタンスストアボリュームをサポートする一部のインスタンスタイプは、ソリッドステートドライブ (SSD) を使用して非常に高いランダム I/O パフォーマンスを提供します。インスタンスタイプによっては、NVMe インスタンスストアボリュームをサポートしていないものがあります。インスタンスタイプによっては、NVMe EBS ボリュームをサポートしていないものがあります。詳細については、「[Linux インスタンスの Amazon EBS および NVMe \(p. 1027\)](#)」および「[NVMe SSD ボリューム \(p. 1086\)](#)」を参照してください。
- Amazon EBS I/O 専用の容量をさらに多く取得するには、一部のインスタンスタイプを EBS 最適化インスタンスとして起動してください。インスタンスタイプの中には、デフォルトで EBS 最適化されるものがあります。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

## ネットワーキング機能とストレージ機能の概要

次の表に、現行世代のインスタンスタイプでサポートされるネットワーキング機能とストレージ機能をまとめています。

	EBS のみ	NVMe EBS	インスタンスストア	配置グループ	拡張ネットワーキング
A1	はい	はい	いいえ	はい	ENA
C4	はい	いいえ	いいえ	はい	Intel 82599 VF
C5	はい	はい	いいえ	はい	ENA
C5d	いいえ	はい	NVMe *	はい	ENA
C5n	はい	はい	いいえ	はい	ENA
D2	いいえ	いいえ	HDD	はい	Intel 82599 VF
F1	いいえ	いいえ	NVMe *	はい	ENA
G3	はい	いいえ	いいえ	はい	ENA

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
ネットワーキング機能とストレージ機能

	EBS のみ	NVMe EBS	インスタンスストア	配置グループ	拡張ネットワーキング
G4	いいえ	はい	NVMe *	はい	ENA
HS1	いいえ	いいえ	HDD *	はい	ENA
I3	いいえ	いいえ	NVMe *	はい	ENA
I3en	いいえ	はい	NVMe *	はい	ENA
M4	はい	いいえ	いいえ	はい	m4.16xlarge: ENA  他のすべてのサ イズ: インテル 82599 VF
M5	はい	はい	いいえ	はい	ENA
M5a	はい	はい	いいえ	はい	ENA
M5ad	いいえ	はい	NVMe *	はい	ENA
M5d	いいえ	はい	NVMe *	はい	ENA
M5dn	いいえ	はい	NVMe *	はい	ENA
M5n	はい	はい	いいえ	はい	ENA
P2	はい	いいえ	いいえ	はい	ENA
P3	はい	いいえ	いいえ	はい	ENA
P3dn	いいえ	はい	NVMe *	はい	ENA
R4	はい	いいえ	いいえ	はい	ENA
R5	はい	はい	いいえ	はい	ENA
R5a	はい	はい	いいえ	はい	ENA
R5ad	いいえ	はい	NVMe *	はい	ENA
R5d	いいえ	はい	NVMe *	はい	ENA
R5dn	いいえ	はい	NVMe *	はい	ENA
R5n	はい	はい	いいえ	はい	ENA
T2	はい	いいえ	いいえ	いいえ	いいえ
T3	はい	はい	いいえ	いいえ	ENA
T3a	はい	はい	いいえ	いいえ	ENA
u-xtb1.metal	はい	はい	いいえ	いいえ	ENA
X1	いいえ	いいえ	SSD	はい	ENA
X1e	いいえ	いいえ	SSD *	はい	ENA
z1d	いいえ	はい	NVMe *	はい	ENA

\* ルートデバイスピリュームは、Amazon EBS ボリュームにする必要があります。

次の表に、以前の世代のインスタンスタイプでサポートされるネットワーキング機能とストレージ機能をまとめています。

	インスタンスストア	配置グループ	拡張ネットワーキング
C3	SSD	はい	Intel 82599 VF
G2	SSD	はい	いいえ
I2	SSD	はい	Intel 82599 VF
M3	SSD	いいえ	いいえ
R3	SSD	はい	Intel 82599 VF

## インスタンス制限

リージョンで起動できるインスタンスの合計数には制限があります。また、一部のインスタンスタイプにはその他の制限もあります。

デフォルトの制限の詳細については、「[Amazon EC2 で実行できるインスタンスの数はいくつですか？](#)」を参照してください。

現在の制限の表示、または現在の制限の引き上げリクエストについての詳細については、「[Amazon EC2 サービスの制限 \(p. 1130\)](#)」を参照してください。

## 汎用インスタンス

汎用インスタンスは、コンピューティング、メモリ、およびネットワーキングリソースをバランスよく備えており、さまざまなワークロードに使用できます。

### A1 インスタンス

A1 インスタンスは、Arm ワークシステムがサポートするスケールアウト型ワークロードに最適です。これらのインスタンスは、以下の用途に適しています。

- ・ウェブサーバー
- ・コンテナ化されたマイクロサービス
- ・キャッシュフリート
- ・分散型データストア
- ・Arm の命令セットが必要なアプリケーション

詳細については、「[Amazon EC2 A1 インスタンス](#)」を参照してください。

### M5、M5a、M5ad、M5d、M5dn、および M5n インスタンス

これらのインスタンスは、クラウドインフラストラクチャの実現に最適で、コンピューティング、メモリ、およびネットワーキングリソースをバランスよく備えており、クラウドにデプロイされる広範なアプリケーションに使用されます。M5 インスタンスは、以下の用途に適しています。

- ・ウェブサーバーとアプリケーションサーバー
- ・中小規模のデータベース
- ・ゲームサーバー

- キャッシュフリート
- SAP、Microsoft SharePoint、クラスター・コンピューティング、その他のエンタープライズ・アプリケーションを実行するバックエンド・サーバー

m5.metaL および m5d.metaL インスタンスでは、プロセッサやメモリなどのホスト・サーバーの物理リソースにアプリケーションから直接アクセスすることができます。これらのインスタンスは、次の用途に適しています。

- 仮想環境で利用できない、または完全にサポートされていない低レベルのハードウェア機能（例：Intel VT）へのアクセスを必要とするワークロード
- ライセンスやサポートを目的として非仮想化環境で実行する必要があるアプリケーション

詳細については、「[Amazon EC2 M5 インスタンス](#)」を参照してください。

### T2、T3、および T3a インスタンス

これらのインスタンスは、ベースライン・レベルの CPU パフォーマンスを維持し、ワークロードの必要に応じてより高いレベルまでバーストすることもできます。無制限インスタンスは、必要なときに、任意の期間にわたって高い CPU パフォーマンスを保持できます。詳細については、「[バースト可能パフォーマンスインスタンス \(p. 199\)](#)」を参照してください。これらのインスタンスは、以下の用途に適しています。

- ウェブサイトとウェブ・アプリケーション
- コードリポジトリ
- 開発、ビルト、テスト、およびステージング環境
- マイクロサービス

詳細については、「[Amazon EC2 T2 インスタンス](#)」および「[Amazon EC2 T3 インスタンス](#)」を参照してください。

### コンテンツ

- [ハードウェア仕様 \(p. 191\)](#)
- [インスタンスのパフォーマンス \(p. 194\)](#)
- [ネットワークパフォーマンス \(p. 194\)](#)
- [SSD I/O パフォーマンス \(p. 196\)](#)
- [インスタンスの機能 \(p. 197\)](#)
- [リリースノート \(p. 198\)](#)
- [バースト可能パフォーマンスインスタンス \(p. 199\)](#)

## ハードウェア仕様

以下に示しているのは、汎用インスタンスのハードウェア仕様の要約です。

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
a1.medium	1	2
a1.large	2	4
a1.xlarge	4	8
a1.2xlarge	8	16

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
汎用インスタンス

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
a1.4xlarge	16	32
m4.large	2	8
m4.xlarge	4	16
m4.2xlarge	8	32
m4.4xlarge	16	64
m4.10xlarge	40	160
m4.16xlarge	64	256
m5.large	2	8
m5.xlarge	4	16
m5.2xlarge	8	32
m5.4xlarge	16	64
m5.8xlarge	32	128
m5.12xlarge	48	192
m5.16xlarge	64	256
m5.24xlarge	96	384
m5.metal	96	384
m5a.large	2	8
m5a.xlarge	4	16
m5a.2xlarge	8	32
m5a.4xlarge	16	64
m5a.8xlarge	32	128
m5a.12xlarge	48	192
m5a.16xlarge	64	256
m5a.24xlarge	96	384
m5ad.large	2	8
m5ad.xlarge	4	16
m5ad.2xlarge	8	32
m5ad.4xlarge	16	64
m5ad.8xlarge	32	128
m5ad.12xlarge	48	192
m5ad.16xlarge	64	256

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
汎用インスタンス

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
m5ad.24xlarge	96	384
m5d.large	2	8
m5d.xlarge	4	16
m5d.2xlarge	8	32
m5d.4xlarge	16	64
m5d.8xlarge	32	128
m5d.12xlarge	48	192
m5d.16xlarge	64	256
m5d.24xlarge	96	384
m5d.metal	96	384
m5dn.large	2	8
m5dn.xlarge	4	16
m5dn.2xlarge	8	32
m5dn.4xlarge	16	64
m5dn.8xlarge	32	128
m5dn.12xlarge	48	192
m5dn.16xlarge	64	256
m5dn.24xlarge	96	384
m5n.large	2	8
m5n.xlarge	4	16
m5n.2xlarge	8	32
m5n.4xlarge	16	64
m5n.8xlarge	32	128
m5n.12xlarge	48	192
m5n.16xlarge	64	256
m5n.24xlarge	96	384
t2.nano	1	0.5
t2.micro	1	1
t2.small	1	2
t2.medium	2	4
t2.large	2	8

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
t2.xlarge	4	16
t2.2xlarge	8	32
t3.nano	2	0.5
t3.micro	2	1
t3.small	2	2
t3.medium	2	4
t3.large	2	8
t3.xlarge	4	16
t3.2xlarge	8	32
t3a.nano	2	0.5
t3a.micro	2	1
t3a.small	2	2
t3a.medium	2	4
t3a.large	2	8
t3a.xlarge	4	16
t3a.2xlarge	8	32

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

CPU オプションの指定についての詳細は、「[CPU オプションの最適化 \(p. 571\)](#)」を参照してください。

## インスタンスのパフォーマンス

EBS 最適化インスタンスは、インスタンスからの Amazon EBS I/O とその他のネットワークトラフィックとの競合を排除することによって、EBS ボリュームの安定した高パフォーマンスを実現できます。一部の汎用インスタンスは、追加料金なしでデフォルトで EBS 最適化されています。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

一部の汎用インスタンスタイプでは、Linux でプロセッサの C ステートと P ステートを制御できます。C ステートは非アクティブ時のコアのスリープレベルを制御し、P ステートは希望するコアからのパフォーマンス (CPU 周波数) を制御します。詳細については、「[EC2 インスタンスタイプのプロセッサのステート制御 \(p. 561\)](#)」を参照してください。

## ネットワークパフォーマンス

サポート対象のインスタンスタイプで、拡張されたネットワーキング機能を有効にすることができます。拡張ネットワーキングでは、パケット毎秒 (PPS) が非常に大きく、ネットワークのストレスが少なく、レイテンシーが低くなります。詳細については、「[Linux の拡張ネットワーキング \(p. 737\)](#)」を参照してください。

拡張されたネットワーキングのための Elastic Network Adapter (ENA) を使用するインスタンスタイプは、高いパケット/秒パフォーマンスと一貫して低いレイテンシーを同時に実現します。ほとんどのアプリケーション

ションでは、高いレベルのネットワークパフォーマンスが一貫して必要なわけではありませんが、データの送受信時にアクセスする帯域幅を増やすことでメリットを得られます。ENA を使用し、「最大 10 Gbps」または「最大 25 Gbps」のネットワークパフォーマンスをサポートするインスタンスサイズでは、ネットワーク I/O クレジットメカニズムを使用して、平均帯域幅使用率に基づいてインスタンスにネットワーク帯域幅を割り当てます。これらのインスタンスでは、ネットワーク帯域幅がベースライン制限を下回るとクレジットを獲得し、これらのクレジットをネットワークデータ転送を実行するときに使用できます。

以下に示しているのは、拡張ネットワークをサポートする汎用インスタンスのネットワークパフォーマンスの要約です。

インスタンスタイプ	ネットワークパフォーマンス	拡張ネットワーキング
t2.nano   t2.micro   t2.small   t2.medium   t2.large   t2.xlarge   t2.2xlarge	最大 1 Gbps	
t3.nano   t3.micro   t3.small   t3.medium   t3.large   t3.xlarge   t3.2xlarge   t3a.nano   t3a.micro   t3a.small   t3a.medium   t3a.large   t3a.xlarge   t3a.2xlarge	最大 5 Gbps	ENA (p. 738)
m4.large	中	Intel 82599 VF (p. 751)
m4.xlarge   m4.2xlarge   m4.4xlarge	高	Intel 82599 VF (p. 751)
a1.medium   a1.large   a1.xlarge   a1.2xlarge     a1.4xlarge   m5.large   m5.xlarge   m5.2xlarge   m5.4xlarge   m5a.large   m5a.xlarge   m5a.2xlarge   m5a.4xlarge   m5a.8xlarge     m5ad.large   m5ad.xlarge     m5ad.2xlarge     m5ad.4xlarge   m5ad.8xlarge   m5d.large   m5d.xlarge   m5d.2xlarge   m5d.4xlarge	最大 10 Gbps	ENA (p. 738)
m4.10xlarge	10 Gbps	Intel 82599 VF (p. 751)
m5.8xlarge   m5.12xlarge     m5a.12xlarge   m5ad.12xlarge   m5d.8xlarge   m5d.12xlarge	10 Gbps	ENA (p. 738)
m5a.16xlarge   m5ad.16xlarge	12 Gbps	ENA (p. 738)
m5.16xlarge   m5a.24xlarge     m5ad.24xlarge   m5d.16xlarge	20 Gbps	ENA (p. 738)

インスタンスタイプ	ネットワークパフォーマンス	拡張ネットワーキング
m5dn.4xlarge 以下   m5n.4xlarge 以下	最大 25 Gbps	<a href="#">ENI (p. 738)</a>
m4.16xlarge   m5.24xlarge   m5.metal   m5d.24xlarge   m5d.metal   m5dn.8xlarge   m5n.8xlarge	25 Gbps	<a href="#">ENI (p. 738)</a>
m5dn.12xlarge   m5n.12xlarge	50 Gbps	<a href="#">ENI (p. 738)</a>
m5dn.16xlarge   m5n.16xlarge	75 Gbps	<a href="#">ENI (p. 738)</a>
m5dn.24xlarge   m5n.24xlarge	100 Gbps	<a href="#">ENI (p. 738)</a>

## SSD I/O パフォーマンス

カーネルバージョン 4.4 以降の Linux AMI を使用し、インスタンスで利用可能なすべての SSD ベースのインスタンスマップリュームを使用する場合は、以下の表に示されている IOPS (4,096 バイトブロックサイズ) のパフォーマンスを得ることができます (キューの深さの飽和度において)。それ以外の場合、IOPS パフォーマンスは低下します。

インスタンスサイズ	100% のランダム読み取り時 IOPS	書き込み IOPS
m5ad.large *	30,000	15,000
m5ad.xlarge *	59,000	29,000
m5ad.2xlarge *	117,000	57,000
m5ad.4xlarge *	234,000	114,000
m5ad.8xlarge	466,666	233,333
m5ad.12xlarge	700,000	340,000
m5ad.16xlarge	933,333	466,666
m5ad.24xlarge	1,400,000	680,000
m5d.large *	30,000	15,000
m5d.xlarge *	59,000	29,000
m5d.2xlarge *	117,000	57,000
m5d.4xlarge *	234,000	114,000
m5d.8xlarge	466,666	233,333
m5d.12xlarge	700,000	340,000
m5d.16xlarge	933,333	466,666

インスタンスサイズ	100% のランダム読み取り時 IOPS	書き込み IOPS
m5d.24xlarge	1,400,000	680,000
m5d.metal	1,400,000	680,000
m5dn.large *	30,000	15,000
m5dn.xlarge *	59,000	29,000
m5dn.2xlarge *	117,000	57,000
m5dn.4xlarge *	234,000	114,000
m5dn.8xlarge	466,666	233,333
m5dn.12xlarge	700,000	340,000
m5dn.16xlarge	933,333	466,666
m5dn.24xlarge	1,400,000	680,000

\* これらのインスタンスの場合、最大で指定されたパフォーマンスを得ることができます。

インスタンスに SSD ベースのインスタンスストアボリュームを使用するほど、アーカイブできる書き込み IOPS の数は減少します。これは、SSD コントローラーが実行する必要がある追加の作業が原因です。SSD コントローラーは、利用可能な領域を見つけ、既存のデータを再書き込みし、未使用的領域を消去して、再書き込みができるようになります。このガベージコレクションというプロセスにより、SSD への内部的な書き込み増幅が発生し、ユーザーの書き込み操作に対する SSD 書き込み操作の割合として表示されます。書き込み操作が 4,096 バイトの倍数でないか、4,096 バイトの境界に整合していない場合、パフォーマンスの低下はさらに大きくなります。少量のバイト数または整合していないバイト数で書き込む場合、SSD コントローラーは周辺のデータを読み取り、その結果を新しい場所に保存する必要があります。このパターンにより、書き込み増幅が大幅に増え、レイテンシーが増加し、I/O パフォーマンスが大きく低下します。

SSD コントローラーは、複数の方法を利用すると、書き込み増幅の影響を減らすことができます。このような方法の 1 つには、SSD インスタンスストレージに領域を予約し、コントローラーが書き込み操作に利用できる領域をより効率的に管理できるようにすることです。これをオーバープロビジョニングと呼びます。インスタンスに提供された SSD ベースのインスタンスストアボリュームには、オーバープロビジョニングに対して予約された領域がありません。書き込み増幅を減らすには、ボリュームの 10% を未使用的の状態のままにし、SSD コントローラーがこれをオーバープロビジョニングに使用できるようにすることをお勧めしますこれにより、使用できるストレージは減りますが、ディスクが総容量に近づいた場合でもパフォーマンスを向上させることができます。

TRIM をサポートするインスタンスストアボリュームの場合、TRIM コマンドを使用して、書き込んだデータが不要になったときはいつでも SSD コントローラーに通知することができます。これにより、より多くの空き領域がコントローラーに与えられ、その結果書き込み増幅が減り、パフォーマンスが向上します。詳細については、「[インスタンスストアボリュームの TRIM のサポート \(p. 1087\)](#)」を参照してください。

## インスタンスの機能

以下に示しているのは、汎用インスタンスの機能の要約です。

	EBS のみ	NVMe EBS	インスタンスストア	配置グループ
A1	はい	はい	いいえ	あり

	EBS のみ	NVMe EBS	インスタンスストア	配置グループ
M4	はい	いいえ	いいえ	あり
M5	はい	はい	いいえ	あり
M5a	はい	はい	いいえ	あり
M5ad	いいえ	はい	NVMe *	あり
M5d	いいえ	はい	NVMe *	はい
M5dn	いいえ	はい	NVMe *	はい
M5n	はい	はい	いいえ	はい
T2	はい	いいえ	いいえ	いいえ
T3	はい	はい	いいえ	なし
T3a	はい	はい	いいえ	なし

\* ルートデバイスボリュームは、Amazon EBS ボリュームにする必要があります。

詳細については、以下を参照してください。

- [Linux インスタンスの Amazon EBS および NVMe \(p. 1027\)](#)
- [Amazon EC2 インスタンスストア \(p. 1076\)](#)
- [プレイスメントグループ \(p. 791\)](#)

## リリースノート

- M5、M5d、および T3 インスタンスは、第 1 世代 (Skylake-SP) または第 2 世代 (Cascade Lake) の 3.1 GHz Intel Xeon Platinum 8000 シリーズプロセッサーを搭載しています。
- M5a、M5ad、および T3a インスタンスは、2.5 GHz AMD EPYC 7000 シリーズプロセッサを搭載しています。
- A1 インスタンスは、64 ビット Arm アーキテクチャベースの 2.3 GHz AWS Graviton プロセッサを搭載しています。
- インスタンスタイプが M4、M5、M5a、M5ad、M5d、t2.large 以上の場合、および t3.large 以上、および t3a.large 以上のは場合は、64 ビットの HVM AMI が必要です。これらのインスタンスはハイメモリであるため、そのキャパシティーを活用するには 64 ビットのオペレーティングシステムが必要です。HVM AMI は、ハイメモリインスタンスタイプの準仮想化 (PV) AMI よりも優れたパフォーマンスを提供します。さらに、拡張ネットワーキングを利用するには、HVM AMI を使用する必要があります。
- A1 インスタンスには、次の要件があります。
  - 64 ビット Arm アーキテクチャ用の AMI を使用する必要があります。
  - [NVMe ドライバー \(p. 1027\)](#)がインストールされている必要があります。
  - [Elastic Network Adapter \(ENA\) ドライバー \(p. 738\)](#)がインストールされている必要があります。
  - ACPI テーブルを含む UEFI による起動と、PCI デバイスの ACPI ホットプラグをサポートしている必要があります。

以下の AMI はこれらの要件を満たしています。

- Amazon Linux 2 (64 ビット Arm)
- Ubuntu 16.04 以降 (64 ビット Arm)

- Red Hat Enterprise Linux 7.6 以降 (64 ビット Arm)
- SUSE Linux Enterprise Server 15 以降 (64 ビット Arm)
- M5、M5a、M5ad、M5d、M5dn、M5n、T3、および T3a インスタンスには、次の要件があります。
  - NVMe ドライバー (p. 1027)がインストールされている必要があります。
  - Elastic Network Adapter (ENA) ドライバー (p. 738)がインストールされている必要があります。

以下の Linux AMI はこれらの要件を満たしています。

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (linux-aws カーネル) 以降
- Red Hat Enterprise Linux 7.4 以降
- SUSE Linux Enterprise Server 12 SP2 以降
- CentOS 7.4.1708 以降
- FreeBSD 11.1 以降
- Debian GNU/Linux 9 以降
- A1、M5、M5a、M5ad、M5d、M5dn、M5n、T3、および T3a インスタンスは、ネットワークインターフェイス、EBS ボリューム、および NVMe インスタンスストアボリュームを含め、最大 28 のアタッチをサポートしています。インスタンスごとに 1 つ以上のネットワークインターフェイスのアタッチメントがあります。たとえば、EBS のみのインスタンスに追加のネットワークインターフェイスのアタッチがない場合は、そのインスタンスに 27 個の EBS ボリュームをアタッチできます。
- ベアメタルインスタンスを起動すると、基盤となるサーバーが起動します。これには、すべてのハードウェアやファームウェアコンポーネントの確認が含まれます。つまり、インスタンスが実行状態になつてからネットワーク経由で使用できるようになるまでに 20 分かかることがあります。
- ベアメタルインスタンスから EBS ボリュームまたはセカンダリネットワークインターフェイスをアタッチまたはデタッチするには、PCIe のネイティブホットプラグサポートが必要です。Amazon Linux 2 および最新バージョンの Amazon Linux AMI は PCIe ネイティブホットプラグをサポートしていますが、以前のバージョンではサポートされていません。次の Linux カーネル設定オプションを有効にする必要があります。

```
CONFIG_HOTPLUG_PCI_PCIE=y
CONFIG_PCIEASPM=y
```

- ベアメタルインスタンスでは、I/O ポートベースのシリアルデバイスではなく、PCI ベースのシリアルデバイスを使用しています。アップストリームの Linux カーネルと最新の Amazon Linux AMI は、このデバイスをサポートしています。また、ベアメタルインスタンスでは、システムが PCI ベースのシリアルデバイスを自動的に使用できるようにする ACPI SPCR テーブルも使用できます。最新の Windows AMI では、自動的に PCI ベースのシリアルデバイスが使用されます。
- A1、M5、M5a、M5ad、M5d、M5dn、M5n、T3、および T3a インスタンスには、API リクエストによるクリーンシャットダウンをサポートするための system-logind または acpid がインストールされている必要があります。
- リージョンで起動できるインスタンスの合計数には制限があります。また、一部のインスタンスタイプにはその他の制限もあります。詳細については、「[Amazon EC2 で実行できるインスタンスの数はいくつですか？](#)」を参照してください。制限の引き上げをリクエストするには、「[Amazon EC2 インスタンス申請フォーム](#)」を使用します。

## バースト可能パフォーマンスインスタンス

T3、T3a、および T2 インスタンスを含むバーストパフォーマンスインスタンスは、ベースラインレベルの CPU パフォーマンスを実現するとともに、ワークロードの必要に応じて高いレベルまでバーストする機能を実現できるように設計されています。バーストパフォーマンスインスタンスは、汎用アプリケーションの幅広い用途に適しています。たとえば、マイクロサービス、低レイテンシーのインタラクティブなアプ

リケーション、小中規模のデータベース、仮想デスクトップ、開発、ビルト & ステージング環境、コードリポジトリ、製品プロトタイプなどがあります。

バーストパフォーマンスインスタンスは、CPU 使用率にクレジットを使用する唯一のインスタンスタイプです。T2 インスタンスの料金体系と追加ハードウェアの詳細については、「[Amazon EC2 料金表](#)」および「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

アカウントを作成してから 12 か月未満の場合は、特定の使用制限内で t2.micro インスタンスを使用できます。詳細については、「[AWS 無料利用枠](#)」を参照してください。

## コンテンツ

- [バーストパフォーマンスインスタンスの要件 \(p. 200\)](#)
- [ベストプラクティス \(p. 200\)](#)
- [バースト可能パフォーマンスインスタンスの CPU クレジットおよびベースラインパフォーマンス \(p. 200\)](#)
- [バーストパフォーマンスインスタンスの無制限モード \(p. 203\)](#)
- [バーストパフォーマンスインスタンスのスタンダードモード \(p. 211\)](#)
- [バーストパフォーマンスインスタンスの使用 \(p. 222\)](#)
- [CPU クレジットの監視 \(p. 227\)](#)

## バーストパフォーマンスインスタンスの要件

これらの インスタンスを作成するための要件を以下に示します。

- これらのインスタンスは、オンデマンドインスタンス、リザーブドインスタンス、ハードウェア専有インスタンス、およびスポットインスタンス としては使用できますが、スケジュールされたインスタンス としては使用できません。また、T2 インスタンスは Dedicated Host ではサポートされていません。詳細については、「[インスタンス購入オプション \(p. 274\)](#)」を参照してください。
- 選択するインスタンスのサイズが、オペレーティングシステムおよびアプリケーションの最小メモリ要件を満たしていることを確認します。多量のメモリおよび CPU リソースを消費するグラフィカルユーザーインターフェースを使用するオペレーティングシステム (Windows など) は、多くのユースケースで t2.micro 以上のインスタンスサイズを必要とする場合があります。時間が経つにつれてメモリおよび CPU の要件が増大したときは、同じインスタンスタイプ、または別のインスタンスタイプのより大きなインスタンスサイズにスケールすることができます。
- 追加の要件については、「[汎用インスタンスのリリースノート \(p. 198\)](#)」を参照してください。

## ベストプラクティス

これらのベストプラクティスに従って、バーストパフォーマンスインスタンスの利点を最大限に活用してください。

- 推奨される AMI の使用 – 必要なドライバーを提供する AMI を使用します。詳細については、「[リリースノート \(p. 198\)](#)」を参照してください。
- インスタンスの復旧をオンにする – EC2 インスタンスをモニタリングし、何らかの原因でインスタンスに障害が発生した場合に、インスタンスを自動的に復旧するための CloudWatch アラームを作成します。詳細については、「[Amazon CloudWatch アラームへの復旧アクションの追加 \(p. 667\)](#)」を参照してください。

## バースト可能パフォーマンスインスタンスの CPU クレジットおよびベースラインパフォーマンス

従来の Amazon EC2 インスタンスタイプはパフォーマンスが一定ですが、バーストパフォーマンスインスタンスはベースラインレベルの CPU パフォーマンスを提供しながら、そのベースラインレベルを超えて

バーストする機能を備えています。ベースラインパフォーマンスとバースト機能は、CPU クレジットにより管理されます。

CPU クレジットは、100% の使用率で実行されるフル CPU コアのパフォーマンスを 1 分間実現します。その他の vCPU、使用率、時間数の組み合わせを CPU クレジットと同じにすることができます。たとえば、1 個の CPU クレジットは 1 台の vCPU を使用率 50% で 2 分間実行するか、または 2 台の vCPU を使用率 25% で 2 分間実行するとの等しくなります。

## 目次

- [CPU クレジットの獲得 \(p. 201\)](#)
- [CPU クレジットの獲得率 \(p. 202\)](#)
- [CPU クレジット蓄積制限 \(p. 202\)](#)
- [CPU 存続期間の蓄積 \(p. 203\)](#)
- [ベースラインパフォーマンス \(p. 203\)](#)

## CPU クレジットの獲得

各バーストパフォーマンスインスタンスは、インスタンスサイズに応じて、1 時間当たりの CPU クレジットを絶えず一定の割合で(ミリ秒レベルの細かさで)獲得します。クレジットを蓄積または消費する会計処理もミリ秒レベルの細かさで実施されるため、CPU クレジットの浪費について心配する必要はありません。CPU の短期バーストでは CPU クレジットのごく一部しか使用されません。

バーストパフォーマンスインスタンスが使用する CPU リソースが、ベースラインパフォーマンスに必要な CPU リソースよりも少ない場合(アイドル時など)、未使用的 CPU クレジットが CPU クレジット残高に蓄積されます。バーストパフォーマンスインスタンスがベースラインパフォーマンスレベルの上にバーストする必要が生じた場合、蓄積されたクレジットを消費します。バーストパフォーマンスインスタンスに蓄積されるクレジットが多いほど、より高いパフォーマンスが必要な場合にベースラインパフォーマンスレベルを超えてバーストできる時間が増えます。

以下の表は、バーストパフォーマンスインスタンスのタイプ、1 時間あたりに CPU クレジットを獲得するレート、インスタンスが累積できる最大獲得 CPU クレジット数、インスタンスごとの vCPU の数、およびフルコアパフォーマンス(单一の vCPU を使用した場合)に対するパーセントで表したベースラインパフォーマンスレベルの一覧です。

インスタンスタイプ	1 時間あたりに受け取る CPU クレジット	蓄積可能な最大獲得クレジット*	vCPU	vCPU 別のベースラインパフォーマンス
T2				
t2.nano	3	72	1	5%
t2.micro	6	144	1	10%
t2.small	12	288	1	20%
t2.medium	24	576	2	20%**
t2.large	36	864	2	30%**
t2.xlarge	54	1296	4	22.5%**
t2.2xlarge	81.6	1958.4	8	17%**
T3				
t3.nano	6	144	2	5%**

インスタンスタイプ	1 時間あたりに受け取る CPU クレジット	蓄積可能な最大獲得クレジット*	vCPU	vCPU 別のベースラインパフォーマンス
t3.micro	12	288	2	10%**
t3.small	24	576	2	20%**
t3.medium	24	576	2	20%**
t3.large	36	864	2	30%**
t3.xlarge	96	2304	4	40%**
t3.2xlarge	192	4608	8	40%**
T3a				
t3a.nano	6	144	2	5%**
t3a.micro	12	288	2	10%**
t3a.small	24	576	2	20%**
t3a.medium	24	576	2	20%**
t3a.large	36	864	2	30%**
t3a.xlarge	96	2304	4	40%**
t3a.2xlarge	192	4608	8	40%**

\* 蓄積できるクレジットの数は、24 時間で獲得できるクレジットの数と同じです。

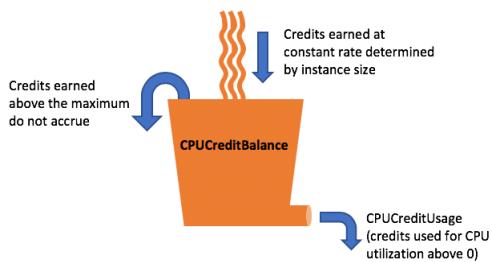
\*\* このテーブルのベースラインのパフォーマンスは vCPU 別です。複数の vCPU があるインスタンスサイズの場合、インスタンスのベースライン CPU 使用率を計算するには、vCPU のパーセントを vCPU の数で乗算します。たとえば、t3.large インスタンスに 2 つの vCPU があると、このインスタンスのベースラインの CPU 使用率は 60% です (2 つの vCPU x 1 つの vCPU のベースラインパフォーマンスの 30%)。CloudWatch では、CPU 使用率は vCPU 別に表示されます。ベースラインパフォーマンスで動作する t3.large インスタンスの CPU 使用率は、CloudWatch の CPU メトリクスに 30% として表示されます。

## CPU クレジットの獲得率

1 時間あたりに獲得する CPU クレジット数は、インスタンスのサイズによって決まります。たとえば、t3.nano は 1 時間あたり 6 クレジットを獲得しますが、t3.small は 1 時間あたり 24 クレジットを獲得します。前記の表は、すべてのインスタンスのクレジット獲得率を示しています。

## CPU クレジット蓄積制限

実行中のインスタンスで獲得されたクレジットが失効することはありませんが、インスタンスが蓄積できる獲得クレジットの数には制限があります。制限は、CPU クレジット残高により決まります。下記の図に示されているとおり、制限に到達すると、獲得された新しいクレジットはすべて破棄されます。フルバケットは CPU クレジット残高制限を示し、スピルオーバーは制限を超えた新しく獲得されたクレジットを示します。



CPU クレジット残高制限は、各 インスタンスのサイズによって異なります。たとえば、`t3.micro` インスタンスは CPU クレジット残高で最大 288 の獲得 CPU クレジットを蓄積できます。前記の表は、各 インスタンスに累積できる獲得クレジットの最大数を示しています。

#### Note

T2 スタンダードインスタンスは、起動クレジットも獲得します。起動クレジットは、CPU クレジット残高制限に対してカウントされません。T2 インスタンスがその起動クレジットを消費しておらず、獲得クレジットを蓄積しながら 24 時間以上アイドル状態が続いた場合、CPU クレジット残高は制限を上回って表示されます。詳細については、「[起動クレジット \(p. 212\)](#)」を参照してください。

T3 および T3a インスタンスは起動クレジットを獲得しません。これらのインスタンスはデフォルトで `unlimited` として起動するため、起動クレジットなしでも起動後すぐにバーストできます。

#### CPU 存続期間の蓄積

実行中のインスタンスの CPU クレジットは失効しません。

T3 および T3a では、CPU クレジットバランスは、インスタンスが停止して起動すると、7 日間保持された後失われます。7 日以内にインスタンスを起動する場合、クレジットは失われません。

T2 では、CPU クレジットバランスは、インスタンスが停止して起動すると引き継がれません。T2 インスタンスを停止した場合、蓄積されたすべてのクレジットが失われます。

詳細については、「[CloudWatch メトリクスの表 \(p. 228\)](#)」の `CPUCreditBalance` を参照してください。

#### ベースラインパフォーマンス

インスタンスが 1 時間あたりに獲得したクレジット数は、CPU 使用率で表されます。これは、ベースラインパフォーマンス、またはベースラインと呼ばれます。たとえば、`t3.nano` インスタンスは 2 つの vCPU を使用して 1 時間あたり 6 クレジットを獲得するので、ベースラインパフォーマンスは vCPU あたり 5% (3/60 分) となります。`t3.xlarge` インスタンスは 4 つの vCPU を使用して 1 時間あたり 96 クレジットを獲得するので、ベースラインパフォーマンスは vCPU あたり 40% (24/60 分) となります。

#### バーストパフォーマンスインスタンスの無制限モード

`unlimited` として設定されたバーストパフォーマンスインスタンスは、必要なときに任意の期間にわたって高い CPU パフォーマンスを保持できます。24 時間ごとのインスタンスの平均 CPU 使用率またはインスタンスの存続期間のいずれか短い方の時間で、インスタンスの平均 CPU 使用率がベースライン以下になった場合、1 時間ごとのインスタンス価格は自動的にすべての CPU 使用率スパイクをカバーします。

汎用のワークロードではほとんどの場合、`unlimited` として設定されたインスタンスは追加料金なしで十分なパフォーマンスを提供します。長時間にわたって高い CPU 使用率でインスタンスを実行する場合には、vCPU 時間ごとに均一追加料金が発生します。インスタンスの価格設定については、「[Amazon EC2 料金表](#)」と、「[Amazon EC2 オンデマンド料金](#)」の無制限の料金表のセクションを参照してください。

## Important

`t2.micro` インスタンスを AWS 無料利用枠オファーで使用し、`unlimited` として設定すると、ローリング期間の 24 時間あたりの平均使用率がインスタンスのベースライン (p. 203) を越えた場合に料金が発生します。

T3 インスタンスは、デフォルトで `unlimited` として起動します。T3スポットインスタンスを `unlimited` として起動し、CPU クレジットを計上するためのアイドル時間なしに、すぐに短期間使用する場合は、余剰クレジットの料金が発生します。24 時間の平均 CPU 使用率がベースラインを超えた場合は、余剰クレジットに対しても課金されます。T3スポットインスタンスは標準 (p. 211) モードで起動して、コストが高くならないようにすることをお勧めします。詳細については、「余剰クレジットにより料金が発生することがある (p. 207)」および「T3スポットインスタンス (p. 395)」を参照してください。

## 目次

- [無制限モードの概念 \(p. 204\)](#)
  - [無制限のバーストパフォーマンスインスタンスの仕組み \(p. 204\)](#)
  - [Unlimited モードと固定 CPU を使用する場合 \(p. 205\)](#)
  - [余剰クレジットにより料金が発生することがある \(p. 207\)](#)
  - [T2 無制限の起動クレジットはありません \(p. 207\)](#)
  - [無制限モードの有効化 \(p. 207\)](#)
  - [無制限とスタンダードを切り替えるとクレジットはどうなるか \(p. 208\)](#)
  - [クレジット使用状況のモニタリング \(p. 208\)](#)
- [例: 無制限モード \(p. 208\)](#)
  - [例 1: T3 無制限でのクレジット使用についての説明 \(p. 208\)](#)
  - [例 2: T2 無制限でのクレジット使用についての説明 \(p. 209\)](#)

## 無制限モードの概念

`unlimited` モードはバーストパフォーマンスインスタンスのクレジットの設定オプションです。これにより、実行中または停止中のインスタンスをいつでも有効または無効にできます。AWS リージョンごとのアカウントレベルで、バーストパフォーマンスインスタンスファミリーごとに、デフォルトのクレジットオプションとして `unlimited` を設定できます。これにより、アカウント内のすべての新しいバーストパフォーマンスインスタンスが、デフォルトのクレジットオプションを使用して起動されます。

### Note

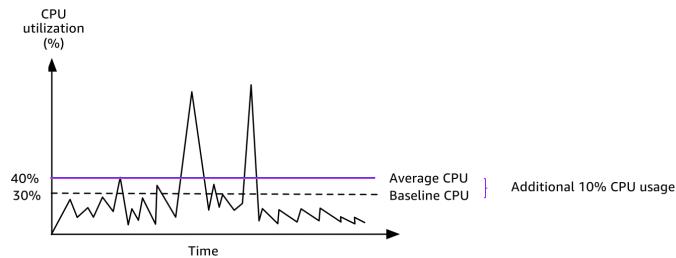
T3 および T3a インスタンスは、デフォルトで `unlimited` として起動します。T2 インスタンスは、デフォルトで `standard` として起動します。デフォルトは、AWS リージョンごとにアカウントレベルで変更できます。詳細については、「アカウントのデフォルトのクレジット指定の設定 (p. 226)」を参照してください。

## 無制限のバーストパフォーマンスインスタンスの仕組み

`unlimited` として設定されたバーストパフォーマンスインスタンスが CPU クレジット残高を使い切った場合、ベースラインを越えるバーストには余剰クレジットを消費します。その CPU 利用率がベースラインを下回った場合、獲得した CPU クレジットを使用して、先に消費された余剰クレジットの支払いが行われます。CPU クレジットを獲得して余剰クレジットを支払う機能により、Amazon EC2 は 24 時間にわたるインスタンスの CPU 使用率を平均化できるようになります。24 時間の平均 CPU 使用率がベースラインを超えると、インスタンスは追加使用量に対して vCPU 時間あたりの一一定の追加料金で請求されます。

以下のグラフは、`t3.large` の CPU 使用率を示します。`t3.large` のベースラインの CPU 使用率は 30% です。インスタンスが 24 時間にわたって平均 30% 以下の CPU 使用率で実行されている場合、コストはインスタンスの 1 時間あたりの料金ですでにカバーされているため、追加料金はかかりません。ただ

し、グラフに示されているように、インスタンスが 24 時間にわたって平均 40% の CPU 使用率で実行されている場合、vCPU 時間あたりの一定の追加料金で CPU 使用率の 10% がインスタンスに対して追加請求されます。



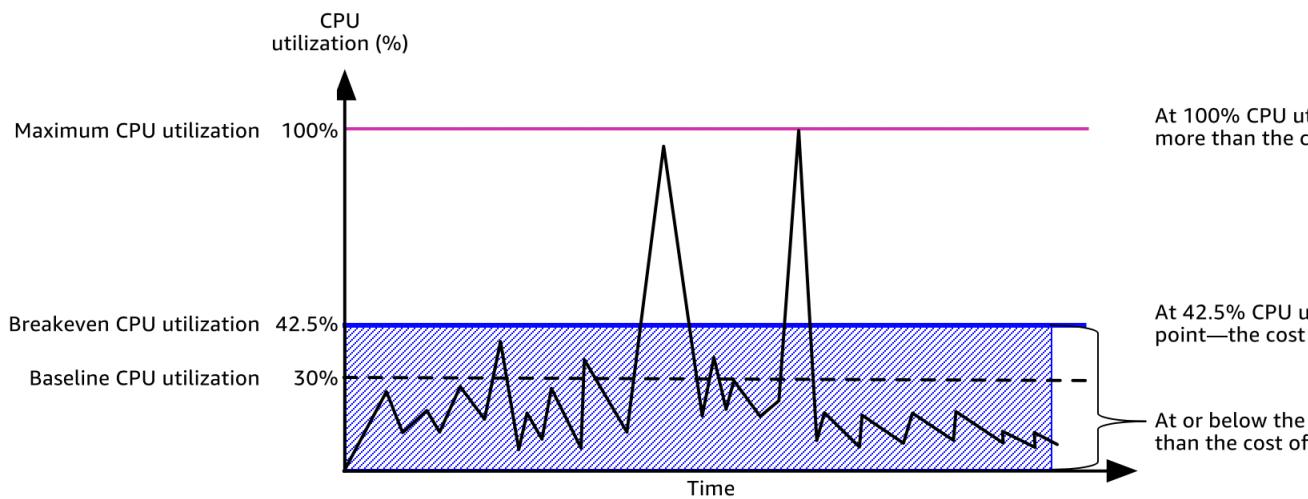
各インスタンスタイプの vCPU あたりのベースラインパフォーマンス、および各インスタンスタイプで獲得するクレジットの詳細については、「[クレジットの表 \(p. 201\)](#)」を参照してください。

### Unlimited モードと固定 CPU を使用する場合

`unlimited` モード (例: T3) でバーストパフォーマンスインスタンスと、固定パフォーマンスインスタンス (例: M5) のどちらを使用するかを決める場合は、損益分岐点 CPU 使用率を判断する必要があります。バーストパフォーマンスインスタンスの損益分岐点 CPU 使用率は、バーストパフォーマンスインスタンスが固定パフォーマンスインスタンスと同じコストになるポイントです。損益分岐点 CPU 使用率は、次のことを判断するのに役立ちます。

- 24 時間の平均 CPU 使用率が損益分岐点の CPU 使用率以下である場合は、バーストパフォーマンスインスタンスを `unlimited` モードで使用すると、固定パフォーマンスインスタンスと同じパフォーマンスを維持しながら、バーストパフォーマンスインスタンスを低価格で使用できます。
- 24 時間の平均 CPU 使用率が損益分岐点 CPU 使用率を上回る場合、バーストパフォーマンスインスタンスは、同等サイズの固定パフォーマンスインスタンスよりもコストが高くなります。T3 インスタンスが 100% CPU で継続的にバーストする場合、同等サイズの M5 インスタンスの約 1.5 倍の価格を支払うことになります。

次のグラフは、`t3.large` のコストが `m5.large` と同じ場合の損益分岐点の CPU 使用率を示しています。`t3.large` の損益分岐点の CPU 使用率は 42.5% です。平均 CPU 使用率が 42.5% の場合、`t3.large` の実行コストは `m5.large` と同じです。平均 CPU 使用率が 42.5% を超える場合は、より高価になります。ワークロードの平均 CPU 使用率が 42.5% 未満であれば、`m5.large` と同じパフォーマンスを得ながら、`t3.large` を低価格で使用することができます。



次の表は、unlimited モードまたは固定パフォーマンスインスタンスでバーストパフォーマンスインスタンスを使用する方が安価な場合を判断できるように、損益分岐点 CPU 使用率のしきい値を計算する方法を示しています。表の列には A から K のラベルが付けられています。

インスタンスタイプ	vCPU	T3 の料金 */ 時間	M5 の料金 */ 時間	料金の違い	vCPU 別の T3 ベースラインパフォーマンス (%)	余剰クレジットに対する vCPU 時間あたりの料金 (分)	vCPU 時間あたりの料金 (分)	vCPU ごとに利用可能な追加のバースト (分)	利用可能な追加 CPU (%)	損益分岐点 CPU %
A	B	C	D	E = D - C	F	G	H = G / 60	I = E / H	J = (I / 60) / B	K = F + J
t3.large	2	0.0835 USD	0.096 USD	0.0125 USD	30%	0.05 USD	0.000833 USD	15	12.5%	42.5%

\* 料金は us-east-1 および Linux OS に基づいています。

テーブルは以下の情報を提供します。

- 列 A は、インスタンスタイプ t3.large を示します。
- 列 B は、t3.large の vCPU の数を示します。
- 列 C は、1 時間あたりの t3.large の料金を示します。
- 列 D は、1 時間あたりの m5.large の料金を示します。
- 列 E は、t3.large と m5.large の差額を示しています。
- 列 F は、t3.large の vCPU あたりのベースラインパフォーマンスを示しており、この場合は 30% です。ベースラインでは、インスタンスの 1 時間あたりのコストが CPU 使用率のコストになります。
- G 列は、獲得したクレジットを使い切った後にインスタンスが 100% CPU でバーストした場合に請求される vCPU 時間あたりの一定の追加料金を示しています。
- H 列は、獲得したクレジットを使い切った後にインスタンスが 100% CPU でバーストした場合に請求される vCPU 時間 (分)あたりの一定の追加料金を示しています。
- I 列は、m5.large と同じ 1 時間あたりの料金が発生している間に、100% CPU で t3.large が 1 時間あたりにバーストできる追加の時間 (分) を示しています。
- J 列は、インスタンスが m5.large と同じ 1 時間あたりの料金が発生している間にバーストする可能性がある、ベースラインを超えた追加の CPU 使用率 (%) を示しています。
- 列 K は、m5.large 以上支払わなくても t3.large がバーストする可能性がある損益分岐点 CPU 使用率 (%) を示しています。この使用率を超えた t3.large のコストは m5.large よりも高くなります。

次の表は、同様のサイズの M5 インスタンスタイプと比較した、T3 インスタンスタイプの損益分岐点 CPU 使用率 (%) を示しています。

T3 インスタンスタイプ	M5 と比較した T3 の損益分岐点 CPU 使用率 (%)
t3.large	42.5%
t3.xlarge	52.5%
t3.2xlarge	52.5%

## 余剰クレジットにより料金が発生することがある

インスタンスの平均 CPU 使用率がベースライン以下の場合、インスタンスに追加料金は発生しません。インスタンスは 24 時間で [クレジット最大数 \(p. 201\)](#) を獲得 (たとえば、t3.micro インスタンスは 24 時間で最大 288 クレジット獲得可能) するため、課金されることなく余剰クレジットを最大まで消費できます。

ただし、CPU 使用率がベースラインを上回ったままの場合、インスタンスは消費した余剰クレジットを支払うのに十分なクレジットを獲得できません。支払われない余剰クレジットに対して、vCPU 時間ごとに均一追加料金が発生します。

先に消費された余剰クレジットは、以下のいずれかの状況に当てはまるとき料金が発生します。

- 消費された余剰クレジットが、インスタンスが 24 時間に獲得できる [最大クレジット数 \(p. 201\)](#) を超えている。最大数を越えて消費された余剰クレジットは、時間の最後に課金されます。
- インスタンスが停止または終了した。
- インスタンスは `unlimited` から `standard` に切り替わります。

消費された余剰クレジットは、CloudWatch メトリクス `CPUSurplusCreditBalance` により追跡されます。課金された余剰クレジットは、CloudWatch メトリクス `CPUSurplusCreditsCharged` で追跡できます。詳細については、「[バーストパフォーマンスインスタンスの追加 CloudWatch メトリクス \(p. 227\)](#)」を参照してください。

## T2 無制限の起動クレジットはありません

T2 スタンダードインスタンスが [起動クレジット \(p. 212\)](#) を受け取っても、T2 無制限インスタンスは起動クレジットを受け取りません。T2 無制限インスタンスは、24 時間のローリング枠内または存続期間のどちらか短いほうで平均 CPU 使用率がベースラインを越えない限り、追加料金なしでいつでもベースラインを超えるバーストができます。したがって、T2 無制限インスタンスは、起動直後の高パフォーマンスを実現するために起動クレジットを必要としません。

T2 インスタンスが `standard` から `unlimited` に切り替えられた場合、残りの `CPUCreditBalance` が引き継がれる前に、蓄積された起動クレジットが `CPUCreditBalance` から削除されます。

### Note

T3 および T3a インスタンスが起動クレジットを獲得することはできません。

## 無制限モードの有効化

T3 および T3a インスタンスは、デフォルトで `unlimited` として起動します。T2 インスタンスはデフォルトで `standard` として起動しますが、起動時に `unlimited` を有効にできます。

実行中または停止中のインスタンスで、`unlimited` から `standard`、`standard` から `unlimited` へいつでも切り替えることができます。詳細については、「[バーストパフォーマンスインスタンスを無制限またはスタンダードとして起動する \(p. 223\)](#)」および「[バーストパフォーマンスインスタンスのクレジット指定の変更 \(p. 226\)](#)」を参照してください。

AWS リージョンごとのアカウントレベルで、バーストパフォーマンスインスタンスマリーゴーに、デフォルトのクレジットオプションとして `unlimited` を設定できます。これにより、アカウント内のすべての新しいバーストパフォーマンスインスタンスが、デフォルトのクレジットオプションを使用して起動されます。詳細については、「[アカウントのデフォルトのクレジット指定の設定 \(p. 226\)](#)」を参照してください。

Amazon EC2 コンソールまたは AWS CLI を使用して、バーストパフォーマンスインスタンスが `unlimited` として設定されているか `standard` として設定されているかを確認できます。詳細については、「[バーストパフォーマンスインスタンスのクレジット指定の表示 \(p. 225\)](#)」および「[デフォルトのクレジット指定の表示 \(p. 227\)](#)」を参照してください。

## 無制限とスタンダードを切り替えるとクレジットはどうなるか

`CPUCreditBalance` は、インスタンスが蓄積したクレジットの数を追跡する CloudWatch メトリクスです。`CPUSurplusCreditBalance` は、インスタンスが消費した余剰クレジットの数を追跡する CloudWatch メトリクスです。

`unlimited` として設定されたインスタンスを `standard` に変更すると、以下の状況が発生します

- `CPUCreditBalance` 値は変更されずに引き継がれます。
- `CPUSurplusCreditBalance` 値にはすぐに課金されます。

`standard` インスタンスが `unlimited` に切り替えられると、以下の状況が発生します。

- `CPUCreditBalance` 値に含まれる、蓄積された獲得クレジットが引き継がれます。
- `T2` スタンダードインスタンスでは、起動クレジットがすべて `CPUCreditBalance` 値から削除され、蓄積された獲得クレジットを含む残りの `CPUCreditBalance` 値が引き継がれます。

## クレジット使用状況のモニタリング

インスタンスが、ベースラインが提供するよりも多くクレジットを消費していないか確認するには、CloudWatch メトリクスを使用して使用率を追跡し、クレジット使用量を通知する時間ごとのアラームを設定できます。詳細については、「[CPU クレジットの監視 \(p. 227\)](#)」を参照してください。

### 例: 無制限モード

次の例では、`unlimited` として設定されているインスタンスのクレジットの使用について説明します。

#### 例

- [例 1: T3 無制限でのクレジット使用についての説明 \(p. 208\)](#)
- [例 2: T2 無制限でのクレジット使用についての説明 \(p. 209\)](#)

#### 例 1: T3 無制限でのクレジット使用についての説明

この例では、`unlimited` として起動する `t3.nano` インスタンスの CPU 使用率と、どのように獲得および余剰クレジットを消費して CPU パフォーマンスを維持するかを見ます。

`t3.nano` インスタンスは、24 時間のローリング期間に渡って最大で 144 CPU クレジットを獲得し、それを 144 分の vCPU 使用と引き換えることができます。CPU クレジット残高 (CloudWatch メトリクス `CPUCreditBalance` で示される) が消耗すると、余剰 CPU クレジット — まだ獲得していない — を消費して必要なだけバーストします。`t3.nano` インスタンスは 24 時間あたり最大 144 クレジットを獲得するため、すぐに課金されることなく余剰クレジットを最大まで消費できます。CPU クレジットを 144 以上消費した場合、差分については時間の最後に課金されます。

以下のグラフにある例の目的は、`CPUCreditBalance` を使いきった後でも余剰クレジットを使用してインスタンスをバーストさせる方法を示すことです。以下のワークフローは、グラフの番号付きの点を参照します。

P1 – グラフの 0 時において、インスタンスは `unlimited` として起動され、すぐにクレジットを獲得します。このインスタンスは起動時からアイドル状態になり (CPU 使用率は 0%)、クレジットは消費されません。すべての未消費のクレジットはクレジット残高に蓄積されます。最初の 24 時間は、`CPUCreditUsage` は 0 で、`CPUCreditBalance` 値は、最大の 144 に達します。

P2 – 次の 12 時間では、CPU 使用率はベースラインの 5% を下回る 2.5% です。インスタンスは消費するよりも多くのクレジットを獲得しますが、`CPUCreditBalance` 値は、最大 144 クレジットを超えることはできません。

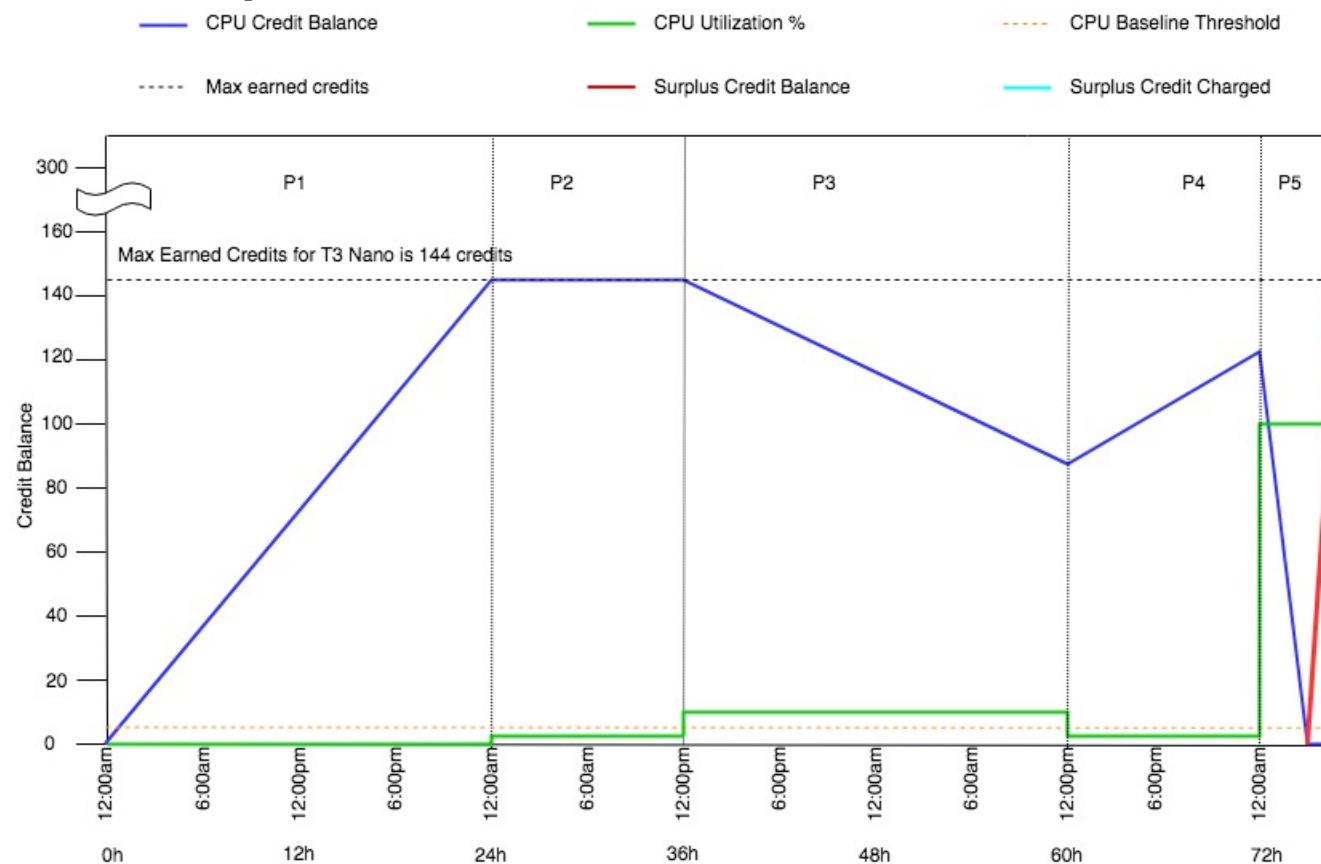
P3 – 次の 24 時間では、CPU 使用率は 7% (ベースラインを上回る) で 57.6 クレジットの消費を必要とします。インスタンスは獲得するよりも多くのクレジットを消費し、CPUCreditBalance 値は、86.4 クレジットに低減します。

P4 – 次の 12 時間では、CPU 使用率は 2.5% に減少し (ベースラインを下回る) で 36 クレジットの消費を必要とします。同時に、インスタンスは 72 クレジットを獲得します。インスタンスは消費するよりも多くのクレジットを獲得し、CPUCreditBalance 値は、122 クレジットに増加します。

P5 – 次の 5 時間で、インスタンスは 100% の CPU 使用率でバーストし、バーストを保持するために 570 クレジットを消費します。この期間の約 1 時間に、インスタンスは CPUCreditBalance 全体の 122 クレジットを使い切り、高い CPU パフォーマンスを維持するために、この期間に合計 448 (570-122=448) の余剰クレジットを使用し始めます。CPUSurplusCreditBalance 値が 144 CPU クレジット (`t3.nano` インスタンスが 24 時間に獲得できるクレジットの最大数) に達すると、その後に消費される余剰クレジットは獲得クレジットで相殺することはできません。その後に消費される余剰クレジットの量は 304 (448-144=304) クレジットで、時間の終了後に 304 クレジットに対して少額の追加料金が発生します。

P6 – 次の 13 時間では、CPU 使用率は 5% (ベースライン) です。インスタンスは消費したのと同量のクレジットを獲得するため、CPUSurplusCreditBalance の支払いを超過しません。CPUSurplusCreditBalance 値は、144 クレジットのままでです。

P7 – この例の過去 24 時間では、インスタンスはアイドル状態で、CPU 使用率は 0% です。この間、インスタンスは、CPUSurplusCreditBalance の支払いに使用する 144 クレジットを獲得します。



#### 例 2: T2 無制限でのクレジット使用についての説明

この例では、`unlimited` として起動する `t2.nano` インスタンスの CPU 使用率と、どのように獲得および余剰クレジットを消費して CPU パフォーマンスを維持するかを見ます。

**t2.nano** インスタンスは、24 時間のローリング期間に渡って最大で 72 CPU クレジットを獲得し、それを 72 分の vCPU 使用と引き換えることができます。CPU クレジット残高 (CloudWatch メトリクス `CPUCreditBalance` で示される) が消耗すると、余剰 CPU クレジット — まだ獲得していない — を消費して必要なだけバーストします。**t2.nano** インスタンスは 24 時間あたり最大 72 クレジットを獲得するため、すぐに課金されることなく余剰クレジットを最大まで消費できます。CPU クレジットを 72 以上消費した場合、差分については時間の最後に課金されます。

以下のグラフにある例の目的は、`CPUCreditBalance` を使い切った後でも余剰クレジットを使用してインスタンスをバーストさせる方法を示すことです。グラフ中のタイムライン開始時点で、インスタンスは 24 時間に獲得可能なクレジットの最大数と同じクレジット残高を蓄積しているものとします。以下のワークフローは、グラフの番号付きの点を参照します。

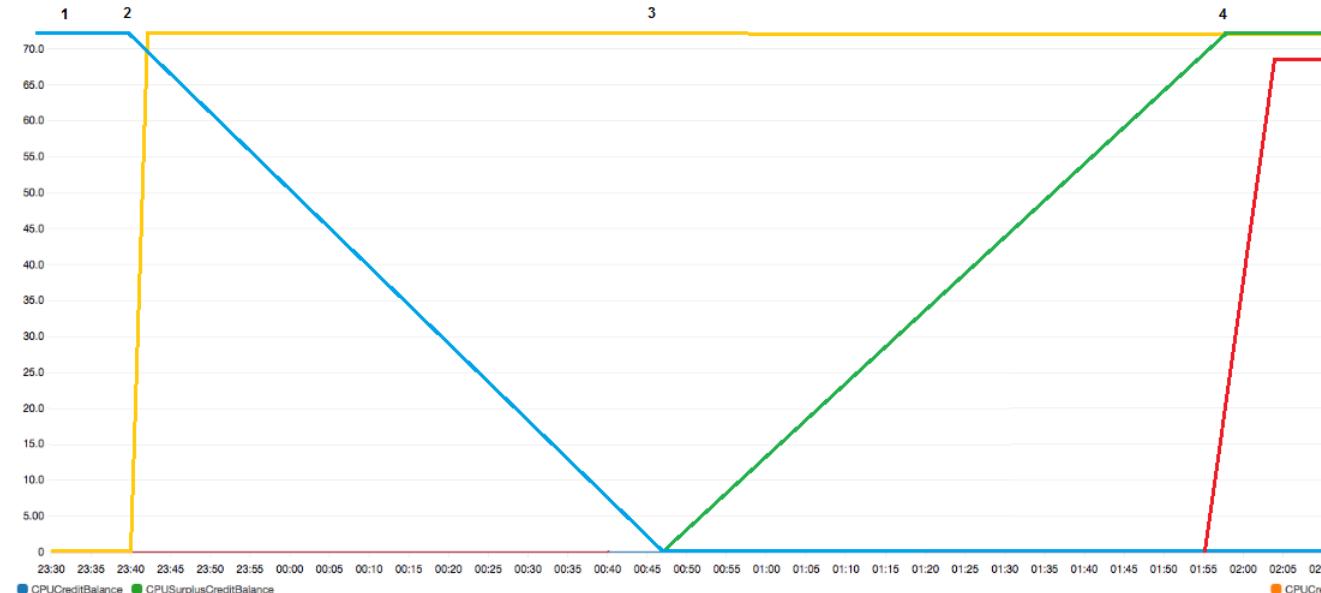
1 – 最初の 10 分間、`CPUCreditUsage` は 0 で、`CPUCreditBalance` 値は最大の 72 のままで。

2 – 23:40 に CPU 使用率が増加すると、インスタンスは CPU クレジットを消費し `CPUCreditBalance` 値が減少します。

3 – 00:47 頃、インスタンスは `CPUCreditBalance` をすべて消耗し、高い CPU パフォーマンスを維持するために余剰クレジットを消費し始めます。

4 – `CPUSurplusCreditBalance` 値が 72 CPU クレジットに達する 1:55 まで余剰クレジットが消費されます。これは、**t2.nano** インスタンスが 24 時間で獲得できる最大値と同じです。その後に消費される余剰クレジットは、24 時間以内の獲得クレジットで相殺することはできません。そのため、時間終了時に少額の追加料金が発生します。

5 – インスタンスは 2:20 頃まで余剰クレジットを消費し続けます。この時点で、CPU 使用率がベースラインを下回ると、インスタンスは 1 時間あたり 3 クレジット (または 5 分ごとに 0.25 クレジット) を獲得し始めます。これは、`CPUSurplusCreditBalance` の支払いに使用されます。`CPUSurplusCreditBalance` 値が 0 まで減った後、インスタンスは 5 分ごとに 0.25 クレジットの割合で `CPUCreditBalance` に獲得クレジットを蓄積し始めます。



All metrics    Graphed metrics (4)    Graph options

	Label	Details	Statistic	Period
■	CPUCreditBalance	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUCreditBalance	Maximum	5 Minutes
■	CPUCreditUsage	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUCreditUsage	Maximum	5 Minutes
■	CPUSurplusCreditBalance	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUSurplusCreditBalance	Maximum	5 Minutes
■	CPUSurplusCreditsCharged	EC2 * InstanceId:i-0aa4b948d7eb37d6b * CPUSurplusCreditsCharged	Maximum	5 Minutes

## 請求書の計算

余剰クレジットは vCPU 時間あたり 0.05 USD かかります。インスタンスは、1:55 から 2:20 の間におよそ 25 余剰クレジットを消費し、これは 0.42 vCPU 時間に相当します。

このインスタンスの追加料金は、 $0.42 \text{ vCPU Hours} \times 0.05 \text{ USD/vCPU Hour} = 0.021 \text{ USD}$  で、四捨五入すると 0.02 USD です。

これが、この T2 無制限インスタンスの月末請求書です。

Amazon Elastic Compute Cloud running Linux/UNIX		
\$0.0058 per On Demand Linux t2.nano Instance Hour	720.000 Hrs	\$4.18
Amazon Elastic Compute Cloud T2 CPU Credits		
\$0.05 per vCPU-Hour of T2 CPU credits	0.420 vCPU-Hours	\$0.02

1 時間ごとの料金の発生を通知する請求アラートを設定して、必要に応じてアクションを実行できます。

## バーストパフォーマンスインスタンスのスタンダードモード

standard として設定されたバーストパフォーマンスインスタンスは、平均 CPU 使用率がインスタンスのベースラインパフォーマンスを一貫して下回るワークロードに適しています。ベースラインより上にバーストする場合、インスタンスは CPU クレジット残高に蓄積されたクレジットを消費します。インスタンスの累積クレジットが足りなくなりそうな場合、パフォーマンスは徐々にベースラインパフォーマンスレベルまで下がるため、累積 CPU クレジット残高を使い切った場合でも、パフォーマンスが急激に低下することはありません。詳細については、「[バースト可能パフォーマンスインスタンスの CPU クレジットおよびベースラインパフォーマンス \(p. 200\)](#)」を参照してください。

### コンテンツ

- [スタンダードモードの概念 \(p. 211\)](#)
  - [スタンダードのバーストパフォーマンスインスタンスの仕組み \(p. 212\)](#)
  - [起動クレジット \(p. 212\)](#)
  - [起動クレジット制限 \(p. 213\)](#)
  - [起動クレジットと獲得クレジットの違い \(p. 213\)](#)
- [例: スタンダードモード \(p. 214\)](#)
  - [例 1: T3 スタンダードでのクレジット使用についての説明 \(p. 214\)](#)
  - [例 2: T2 スタンダードでのクレジット使用についての説明 \(p. 215\)](#)
    - [期間 1: 1 ~ 24 時間 \(p. 215\)](#)
    - [期間 2: 25 ~ 36 時間 \(p. 216\)](#)
    - [期間 3: 37 ~ 61 時間 \(p. 217\)](#)
    - [期間 4: 62 ~ 72 時間 \(p. 218\)](#)
    - [期間 5: 73 ~ 75 時間 \(p. 219\)](#)
    - [期間 6: 76 ~ 90 時間 \(p. 220\)](#)
    - [期間 7: 91 ~ 96 時間 \(p. 221\)](#)

### スタンダードモードの概念

standard モードはバーストパフォーマンスインスタンスの設定オプションです。これにより、実行中または停止中のインスタンスをいつでも有効または無効にできます。AWS リージョンごとのアカウントレベルで、バーストパフォーマンスインスタンスファミリーごとに、デフォルトのクレジットオプションとして standard を設定できます。これにより、アカウント内のすべての新しいバーストパフォーマンスインスタンスが、デフォルトのクレジットオプションを使用して起動されます。

#### Note

T3 および T3a インスタンスは、デフォルトで `unlimited` として起動します。T2 インスタンスは、デフォルトで `standard` として起動します。デフォルトは、AWS リージョンごとにアカウ

ントレベルで変更できます。詳細については、「[アカウントのデフォルトのクレジット指定の設定 \(p. 226\)](#)」を参照してください。

## スタンダードのバーストパフォーマンスインスタンスの仕組み

standard に設定されているバーストパフォーマンスインスタンスが実行状態の場合、1 時間当たりの獲得クレジットを絶えず一定の割合で(ミリ秒レベルの細かさで)獲得します。T2 スタンダードインスタンスが停止すると、蓄積されたクレジットがすべて失われ、クレジット残高はゼロにリセットされます。再起動されると、新しい起動クレジットのセットを受け取り、獲得したクレジットの蓄積を始めます。T3 および T3a スタンダードでは、CPU クレジットバランスは、インスタンスが停止して起動すると、7 日間保持された後失われます。7 日以内にインスタンスを起動する場合、クレジットは失われません。

T2 スタンダードインスタンスは、獲得クレジットと起動クレジットの 2 種類の CPU クレジットを受け取ります。T2 スタンダードインスタンスが実行状態の場合、1 時間当たりの獲得クレジットを絶えず一定の割合で(ミリ秒レベルの細かさで)獲得します。スタート時のインスタンスは、良いスタートアップエクスペリエンスのためのクレジットをまだ獲得していません。したがって、スタートアップエクスペリエンスを積み重ねるために、スタート時にクレジットを獲得します。インスタンスは、獲得クレジットを蓄積しながら最初にそのクレジットを消費します。

T3 および T3a スタンダードインスタンスは起動クレジットを受け取れません。

### 起動クレジット

T2 スタンダードインスタンスは、起動時またはスタート時に vCPU あたり 30 起動クレジットを獲得します。たとえば、t2.micro インスタンスは vCPU が 1 つのため 30 起動クレジット、t2.xlarge インスタンスには vCPU が 4 つあるため 120 起動クレジットを取得します。起動クレジットは、インスタンスが獲得クレジットを蓄積できるようになる前に、起動してすぐにバーストできるよう、最適な起動エクスペリエンスを提供するために設計されています。

起動クレジットは、獲得クレジットよりも先に消費されます。未使用の起動クレジットは CPU クレジット残高に蓄積されますが、CPU クレジット残高制限に対してカウントされません。たとえば、t2.micro インスタンスの CPU クレジット残高制限は 144 獲得クレジットです。起動された後 24 時間アイドルのままであった場合、その CPU クレジット残高は 174 に到達し(30 起動クレジット + 144 獲得クレジット)、制限を上回ります。ただし、インスタンスが 30 起動クレジットを消費した後は、クレジット残高が 144 を超えることはありません。各インスタンスサイズの CPU クレジット残高制限の詳細については、「[クレジットの表 \(p. 201\)](#)」を参照してください。

次の表は、起動または開始の際に受け取る初期 CPU クレジットの割り当てと vCPU の数を示しています。

インスタンスタイプ	起動クレジット	vCPU
t1.micro	15	1
t2.nano	30	1
t2.micro	30	1
t2.small	30	1
t2.medium	60	2
t2.large	60	2
t2.xlarge	120	4
t2.2xlarge	240	8

## 起動クレジット制限

T2 スタンダードインスタンスが起動クレジットを受け取る回数には制限があります。デフォルトの制限は、リージョンごとにローリング期間の 24 時間あたり各アカウントで合計で 100 回の T2 スタンダードインスタンスの起動または開始と設定されています。たとえば、24 時間以内にインスタンスが 100 回停止および開始した場合、24 時間以内に 100 インスタンスが起動された場合、または他の組み合わせが 100 回の開始と同じである場合、制限に到達します。新しいアカウントでは、使用量に基づいて増える下限が設定される場合があります。

### Tip

ワークロードに必要なパフォーマンスを常に確実に得るには、[バーストパフォーマンスインスタンスの無制限モード \(p. 203\)](#) に切り替えるか、またはより大きいインスタンスサイズの使用を検討してください。

## 起動クレジットと獲得クレジットの違い

次の表に、起動クレジットと獲得クレジットの違いを示します。

	起動クレジット	獲得クレジット
クレジットの獲得率	<p>T2 スタンダードインスタンスは、起動時またはスタート時に vCPU あたり 30 起動クレジットを獲得します。</p> <p>T2 インスタンスが <code>unlimited</code> から <code>standard</code> に切り替えられた場合、切り替えの時点では起動クレジットを取得しません。</p>	各 T2 インスタンスは、インスタンスサイズに応じて、1 時間当たりの CPU クレジットを絶えず一定の割合で（ミリ秒レベルの細かさで）獲得します。インスタンスサイズごとに獲得される CPU クレジット数の詳細については、「 <a href="#">クレジットの表 (p. 201)</a> 」を参照してください。
クレジットの獲得制限	起動クレジット受け取り制限は、リージョンごとにローリング期間の 24 時間あたり各アカウントで合計で 100 回の T2 スタンダードインスタンスの起動または開始と設定されています。新しいアカウントでは、使用量に基づいて増える下限が設定される場合があります。	T2 インスタンスは、CPU クレジット残高制限より多くのクレジットを蓄積することはできません。CPU クレジット残高がその制限に到達した場合、制限に到達した後に獲得されたクレジットはすべて破棄されます。起動クレジットは制限に對してはカウントされません。各 T2 インスタンスサイズの CPU クレジット残高制限の詳細については、「 <a href="#">クレジットの表 (p. 201)</a> 」を参照してください。
クレジットの使用	起動クレジットは、獲得クレジットよりも先に消費されます。	獲得クレジットは、すべての起動クレジットを消費した後にのみ消費されます。
クレジットの有効期限	T2 スタンダードインスタンスが実行中の場合、起動クレジットは期限切れになります。T2 スタンダードインスタンスが停止し、T2 無制限に切り替えられた場合、すべての起動クレジットが失われます。	T2 インスタンスが実行中の場合、蓄積した獲得クレジットは期限切れになりません。T2 インスタンスが停止すると、蓄積された獲得クレジットはすべて失われます。

蓄積された起動クレジットと蓄積された獲得クレジットの数は、CloudWatch メトリクス `CPUCreditBalance` によって追跡されます。詳細については、「[CloudWatch メトリクスの表 \(p. 228\)](#)」の `CPUCreditBalance` を参照してください。

## 例: スタンダードモード

次の例では、インスタンスが `standard` として設定された場合の、クレジットの使用について説明します。

### 例

- [例 1: T3 スタンダードでのクレジット使用についての説明 \(p. 214\)](#)
- [例 2: T2 スタンダードでのクレジット使用についての説明 \(p. 215\)](#)

### 例 1: T3 スタンダードでのクレジット使用についての説明

この例では、`standard` として起動した `t3.nano` インスタンスが、獲得クレジットを、獲得、蓄積、消費する方法について示します。クレジットバランスが、蓄積された獲得クレジットを反映するかについて示します。

#### Note

`standard` として設定された T3 および T3a インスタンスは起動クレジットを受け取れません。

実行中の `t3.nano` インスタンスは、24 時間ごとに 144 クレジットを獲得します。このクレジットバランスの制限は、144 の獲得クレジットです。制限に到達すると、獲得された新しいクレジットはすべて破棄されます。獲得および蓄積できるクレジット数の詳細については、[クレジットの表 \(p. 201\)](#)を参照してください。

T3 スタンダードインスタンスを起動し、すぐに使用することができます。または、T3 スタンダードインスタンスを起動し、何日間かアイドル状態にしてから、そこでアプリケーションを実行する場合があります。インスタンスを使用中か、アイドル状態であるかによって、クレジットが消費されるか、あるいは蓄積されるかが決まります。インスタンスが起動してから 24 時間アイドル状態のままの場合、蓄積できる獲得クレジットの最大数となり、クレジットバランスが制限に達します。

この例では、起動後に 24 時間アイドル状態のままとなるインスタンスについて説明します。また、96 時間にわたる 7 つの期間で、クレジットが獲得、蓄積、消費、破棄される率と、各期間の終了時点でのクレジット残高の値を示します。

以下のワークフローは、グラフの番号付きの点を参照します。

P1 – グラフの 0 時において、インスタンスは `standard` として起動され、すぐにクレジットを獲得します。このインスタンスは起動時からアイドル状態になり (CPU 使用率は 0%)、クレジットは消費されません。すべての未消費のクレジットはクレジット残高に蓄積されます。最初の 24 時間は、`CPUCreditUsage` は 0 で、`CPUCreditBalance` 値は、最大の 144 に達します。

P2 – 次の 12 時間では、CPU 使用率はベースラインの 5% を下回る 2.5% です。インスタンスは消費するよりも多くのクレジットを獲得しますが、`CPUCreditBalance` 値は、最大 144 クレジットを超えることはできません。制限を超えて獲得されたクレジットはすべて破棄されます。

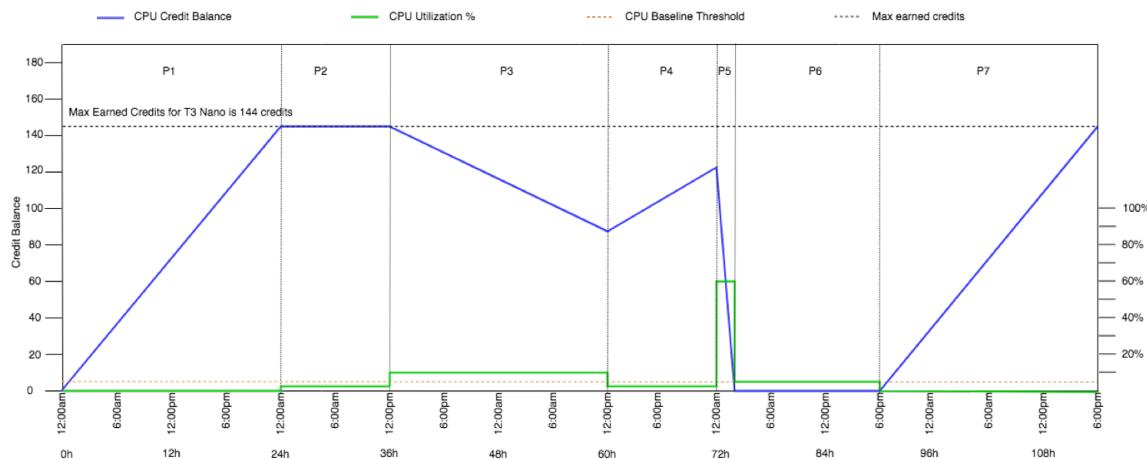
P3 – 次の 24 時間では、CPU 使用率は 7% (ベースラインを上回る) で 57.6 クレジットの消費を必要とします。インスタンスは獲得するよりも多くのクレジットを消費し、`CPUCreditBalance` 値は、86.4 クレジットに低減します。

P4 – 次の 12 時間では、CPU 使用率は 2.5% に減少し (ベースラインを下回る) で 36 クレジットの消費を必要とします。同時に、インスタンスは 72 クレジットを獲得します。インスタンスは消費するよりも多くのクレジットを獲得し、`CPUCreditBalance` 値は、122 クレジットに増加します。

P5 – 次の 2 時間で、インスタンスは 100% の CPU 使用率でバーストし、`CPUCreditBalance` 全体の 122 クレジットを使い切ります。この期間の終わりに、`CPUCreditBalance` は 0 に、CPU 使用率は、ベースラインのパフォーマンスレベル 5% に強制的に低下させられます。ベースラインで、インスタンスは消費した分のクレジットを獲得します。

P6 – 次の 14 時間では、CPU 使用率は 5% (ベースライン) です。インスタンスは消費した分のクレジットを獲得します。CPUCreditBalance 値は、0 のままです。

P7 – この例の過去 24 時間では、インスタンスはアイドル状態で、CPU 使用率は 0% です。この間、インスタンスは、CPUCreditBalance に蓄積する 144 クレジットを獲得します。



## 例 2: T2 スタンダードでのクレジット使用についての説明

この例では、standard が起動クレジットおよび獲得クレジットを獲得、蓄積、消費する際に、t2.nano インスタンスがどのように起動されるかについて示します。クレジット残高に、蓄積された獲得クレジットだけでなく、蓄積された起動クレジットがどのように反映されるかについて示します。

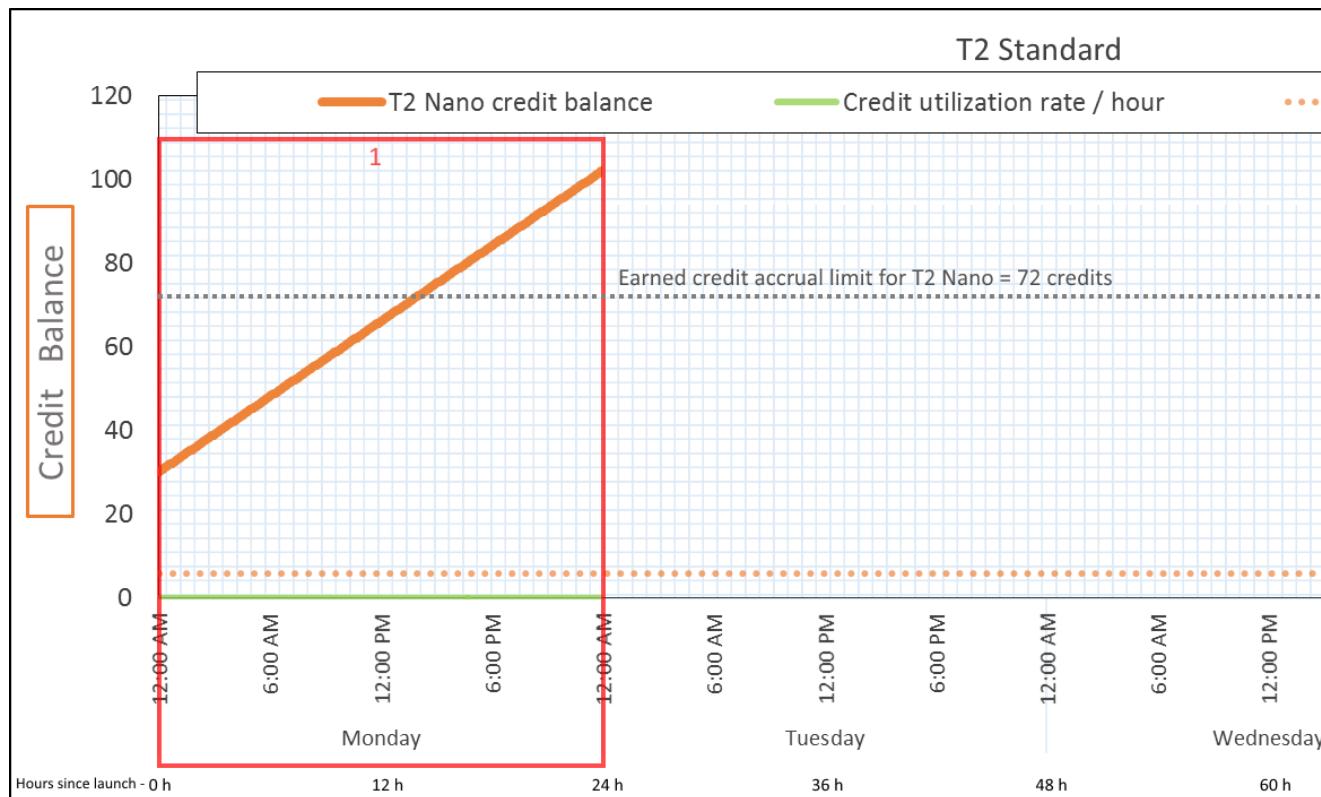
t2.nano インスタンスは、起動時に 30 起動クレジットを獲得し、24 時間ごとに 72 クレジットを獲得します。このクレジット残高の制限は 72 獲得クレジットです。起動クレジットはこの制限に対してカウントされません。制限に到達すると、獲得された新しいクレジットはすべて破棄されます。獲得および蓄積できるクレジット数の詳細については、[クレジットの表 \(p. 201\)](#)を参照してください。の制限事項の詳細については、「[起動クレジット制限 \(p. 213\)](#)」を参照してください。

T2 スタンダードインスタンスを起動し、すぐに使用することができます。または、T2 スタンダードインスタンスを起動し、何日間かアイドル状態にしてから、そこでアプリケーションを実行する場合があります。インスタンスを使用中か、アイドル状態であるかによって、クレジットが消費されるか、あるいは蓄積されるかが決まります。インスタンスが起動後に 24 時間アイドル状態のままである場合、クレジット残高は制限を超えて表示されます。これは、蓄積された獲得クレジットと蓄積された起動クレジットの両方が残高に反映されるためです。ただし、CPU を使用すると、起動クレジットが最初に使用されます。その後、この制限は、蓄積できる獲得クレジットの最大数を常に反映します。

この例では、起動後に 24 時間アイドル状態のまとなるインスタンスについて説明します。また、96 時間にわたる 7 つの期間で、クレジットが獲得、蓄積、消費、破棄される率と、各期間の終了時点でのクレジット残高の値を示します。

### 期間 1: 1 ~ 24 時間

グラフの 0 時において、T2 インスタンスは standard として起動され、すぐに 30 クレジットを獲得します。実行状態の間はクレジットを獲得します。このインスタンスは起動時からアイドル状態になり（—CPU 使用率は 0%—）、クレジットは消費されません。すべての未消費のクレジットはクレジット残高に蓄積されます。起動後約 14 時間で、クレジット残高は 72 (30 起動クレジット + 42 獲得クレジット) となり、これはインスタンスが 24 時間に獲得できる数と同等になります。起動後 24 時間で、クレジット残高は 72 クレジットを超えます。これは、未消費の起動クレジットがクレジット残高に蓄積されるためです（クレジット残高は—102 クレジット: 30 起動クレジット + 72 獲得クレジット）。



クレジットの消費率	24 時間あたり 0 クレジット (0% の CPU 使用率)
クレジットの獲得率	24 時間あたり 72 クレジット
クレジットの破棄率	24 時間あたり 0 クレジット
クレジット残高	102 クレジット (30 起動クレジット + 72 獲得クレジット)

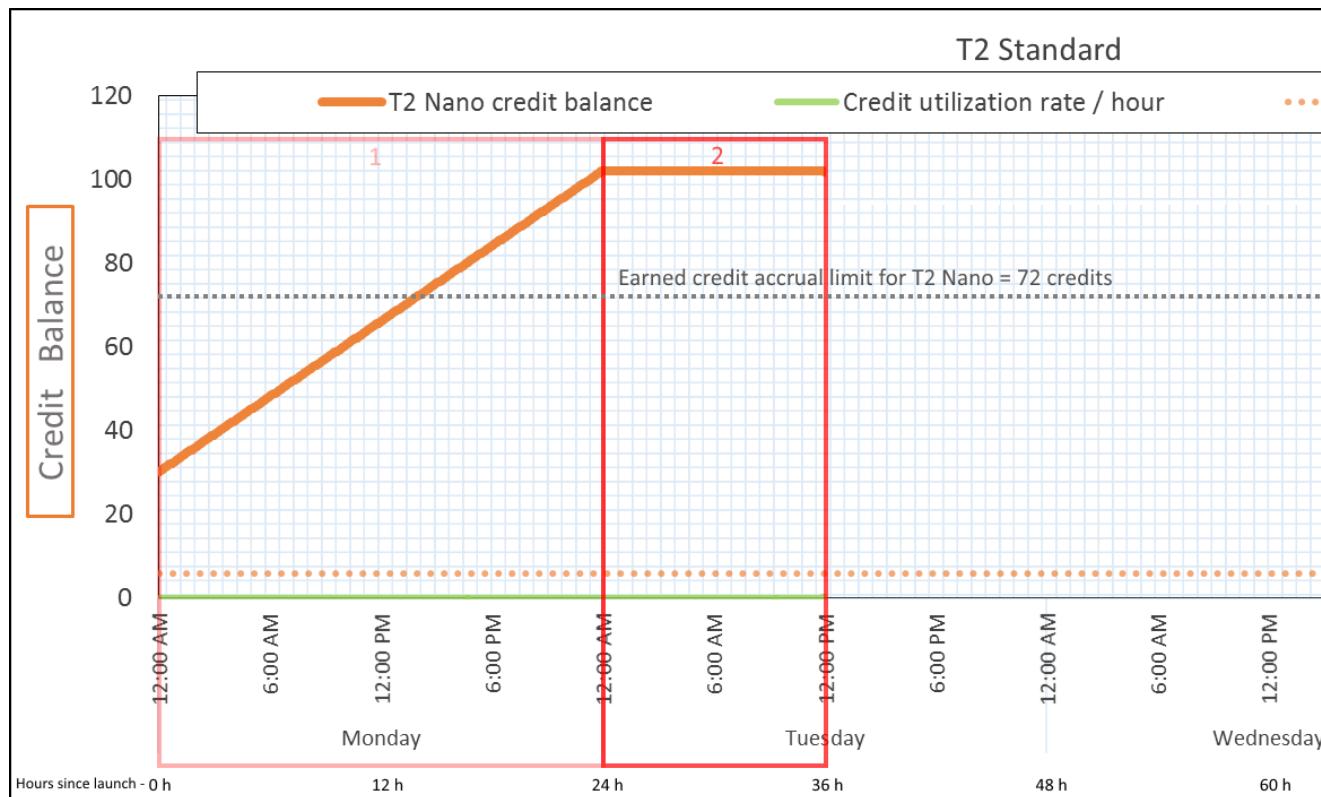
## 結論

起動後に CPU の使用がない場合、インスタンスは 24 時間に獲得できるよりも多くのクレジットを蓄積します (30 起動クレジット + 72 獲得クレジット = 102 クレジット)。

実際のシナリオでは、EC2 インスタンスは起動中および実行中にも少数のクレジットを消費します。それにより、残高がこの例の理論的な最大値に達することを防ぎます。

## 期間 2: 25 ~ 36- 時間

次の 12 時間に、インスタンスは引き続きアイドル状態のままとなり、クレジットを獲得しますが、クレジット残高は増えません。102 クレジット (30 起動クレジット + 72 獲得クレジット) で頭打ちとなります。クレジット残高は制限である 72 の蓄積された獲得クレジットに達したため、新しく獲得されたクレジットは破棄されます。



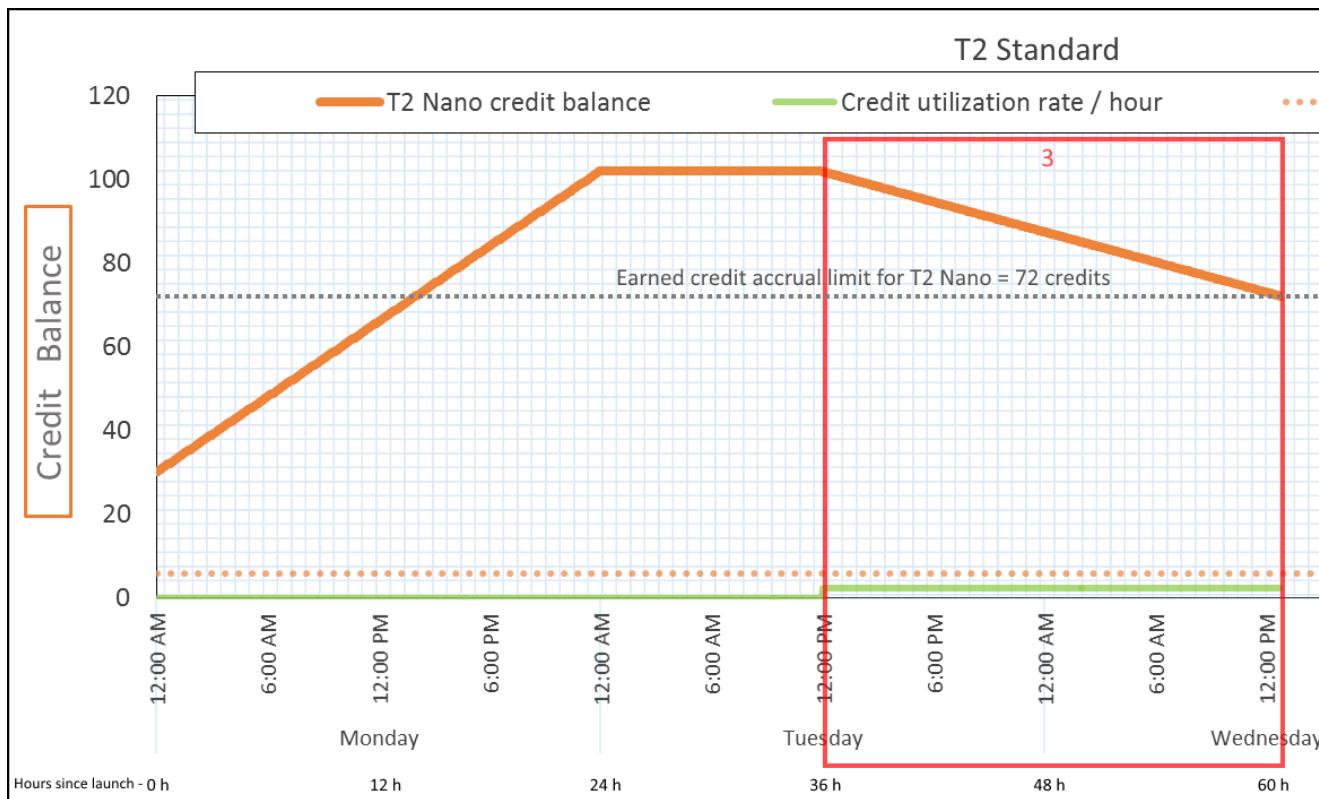
クレジットの消費率	24 時間あたり 0 クレジット (0% の CPU 使用率)
クレジットの獲得率	24 時間あたり 72 クレジット (1 時間で 3 クレジット)
クレジットの破棄率	24 時間あたり 72 クレジット (100% のクレジット獲得率)
クレジット残高	102 クレジット (30 起動クレジット + 72 獲得クレジット) — 残高は変更されません

## 結論

インスタンスはクレジットを継続して獲得しますが、クレジット残高が制限に達した場合、獲得クレジットはそれ以上蓄積されません。制限に到達すると、新しく獲得されたクレジットはすべて破棄されます。起動クレジットは、クレジット残高制限に対してカウントされません。残高に蓄積された起動クレジットが含まれている場合、残高は制限を超えて表示されます。

## 期間 3: 37 ~ 61 時間

次の 25 時間で、インスタンスは 2% の CPU を使用します。これには 30 クレジットが必要です。同じ期間に 75 クレジットを取得しますが、クレジット残高は減ります。残高が減るのは、蓄積された起動クレジットが最初に消費されますが、クレジット残高が既に 72 獲得クレジットという制限に達しているため、新しく獲得されたクレジットは破棄されるためです。



クレジットの消費率	24 時間あたり 28.8 クレジット (1 時間ごとに 1.2 クレジット、2% の CPU 使用率、40% のクレジット獲得率)— 25 時間以上で 30 クレジット
クレジットの獲得率	24 時間あたり 72 クレジット
クレジットの破棄率	24 時間あたり 72 クレジット (100% のクレジット獲得率)
クレジット残高	72 クレジット (30 起動クレジットが消費され、72 獲得クレジットが未使用のまま)

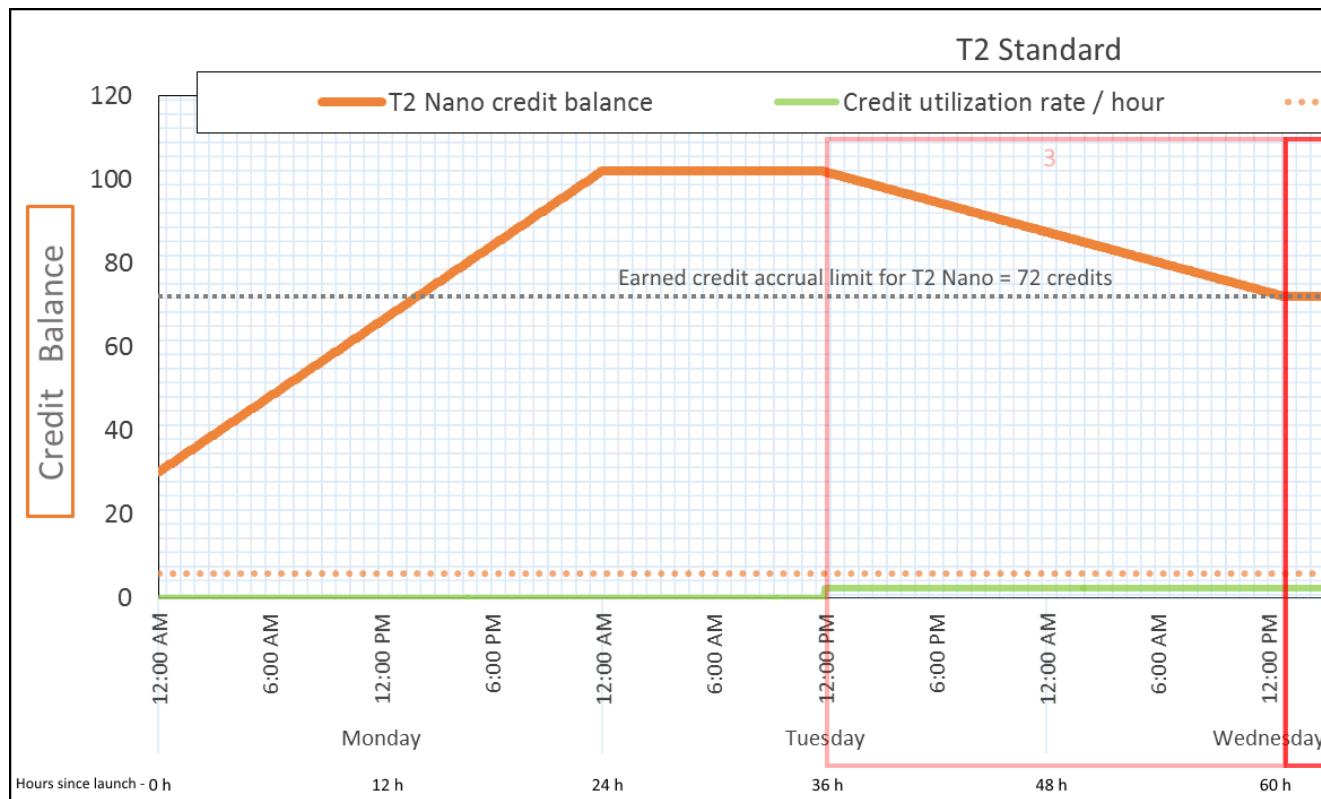
## 結論

インスタンスは、獲得クレジットを消費する前に、起動クレジットを最初に消費します。起動クレジットは、クレジット制限に対してカウントされません。起動クレジットが消費された後で、24 時間に獲得できる数よりも残高が高くなることはありません。さらに、インスタンスの実行中は、それ以上クレジットを獲得することはできません。

## 期間 4: 62 ~ 72 時間

次の 11 時間で、インスタンスは 2% の CPU を使用します。これには 13.2 クレジットが必要です。これは前の期間の CPU 使用率と同じですが、残高は減りません。72 クレジットのままでです。

残高が減らないのは、クレジットの獲得率がクレジットの消費率よりも高いためです。また、インスタンスは 13.2 クレジットを消費する時間に、33 クレジットを獲得します。ただし、残高の制限は 72 クレジットであるため、制限を超えて獲得されたクレジットは破棄されます。残高は 72 で頭打ちとなります。これは期間 2 の 102 クレジットという頭打ちとは異なりますが、蓄積された起動クレジットがないためです。



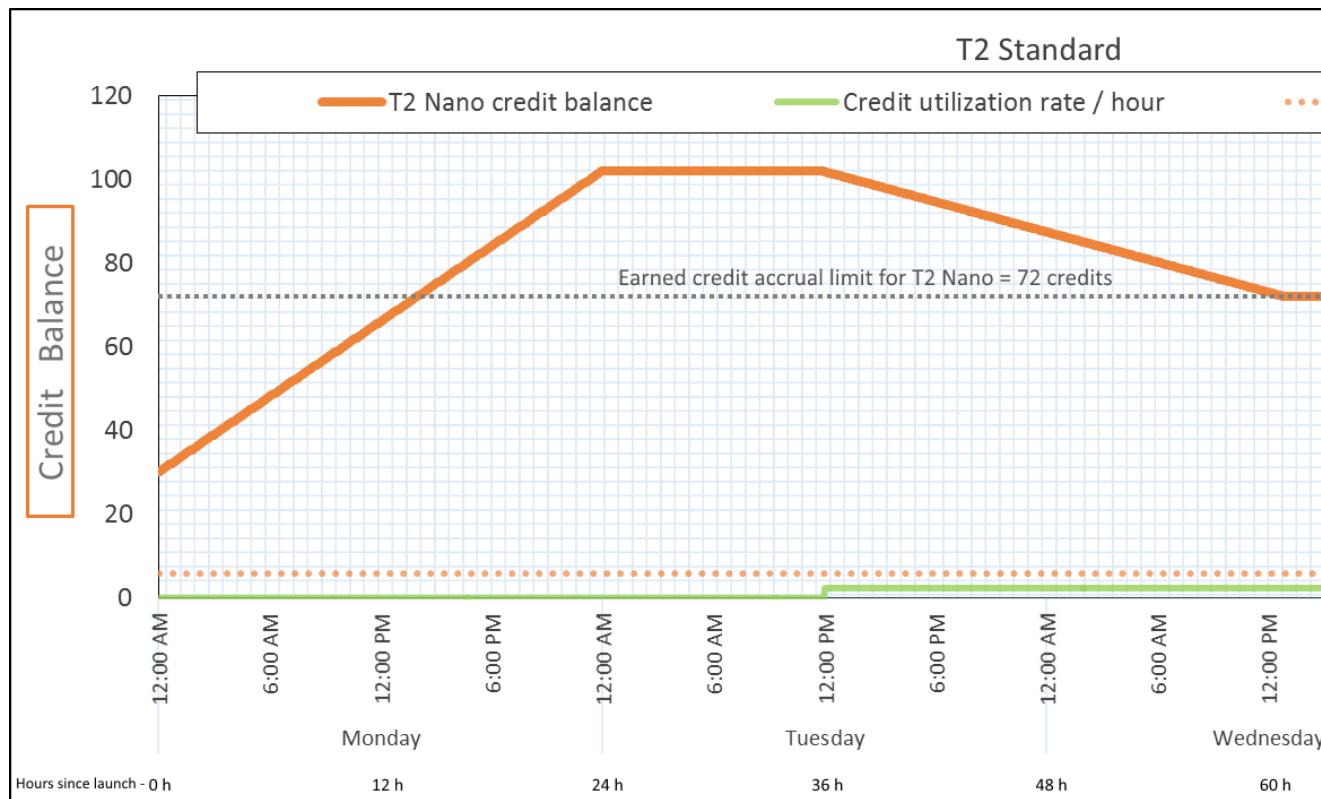
クレジットの消費率	24 時間あたり 28.8 クレジット (1 時間ごとに 1.2 クレジット、2% の CPU 使用率、40% のクレジット獲得率)— 11 時間以上で 13.2 クレジット
クレジットの獲得率	24 時間あたり 72 クレジット
クレジットの破棄率	24 時間あたり 43.2 クレジット (60% のクレジット獲得率)
クレジット残高	72 クレジット (0 起動クレジット、72 獲得クレジット)— 残高は上限

### まとめ

起動クレジットの消費後、クレジット残高の制限はインスタンスが 24 時間に獲得できるクレジット数によって決まります。インスタンスが、消費するよりも多くのクレジットを獲得した場合、制限を超えて新しく獲得されたクレジットは破棄されます。

### 期間 5: 73 ~ 75 時間

次の 3 時間で、インスタンスは 20% の CPU 使用率でバーストします。これには 36 クレジットが必要です。インスタンスは同じ 3 時間で 9 クレジットを獲得します。これにより、実際の残高は 27 クレジット減ります。3 時間の最後に、クレジット残高は 45 の蓄積された獲得クレジットとなります。



クレジットの消費率	24 時間あたり 288 クレジット (1 時間ごとに 12 クレジット、20% の CPU 使用率、400% のクレジット獲得率)—3 時間以上で 36 クレジット
クレジットの獲得率	24 時間あたり 72 クレジット (3 時間で 9 クレジット)
クレジットの破棄率	24 時間あたり 0 クレジット
クレジット残高	45 クレジット (前の残高 (72) - 消費したクレジット (36) + 獲得したクレジット (9))—残高は 24 時間あたり 216 クレジットの率で減少 (消費率 288/24 + 獲得率 72/24 = 残高減少率 216/24)

## 結論

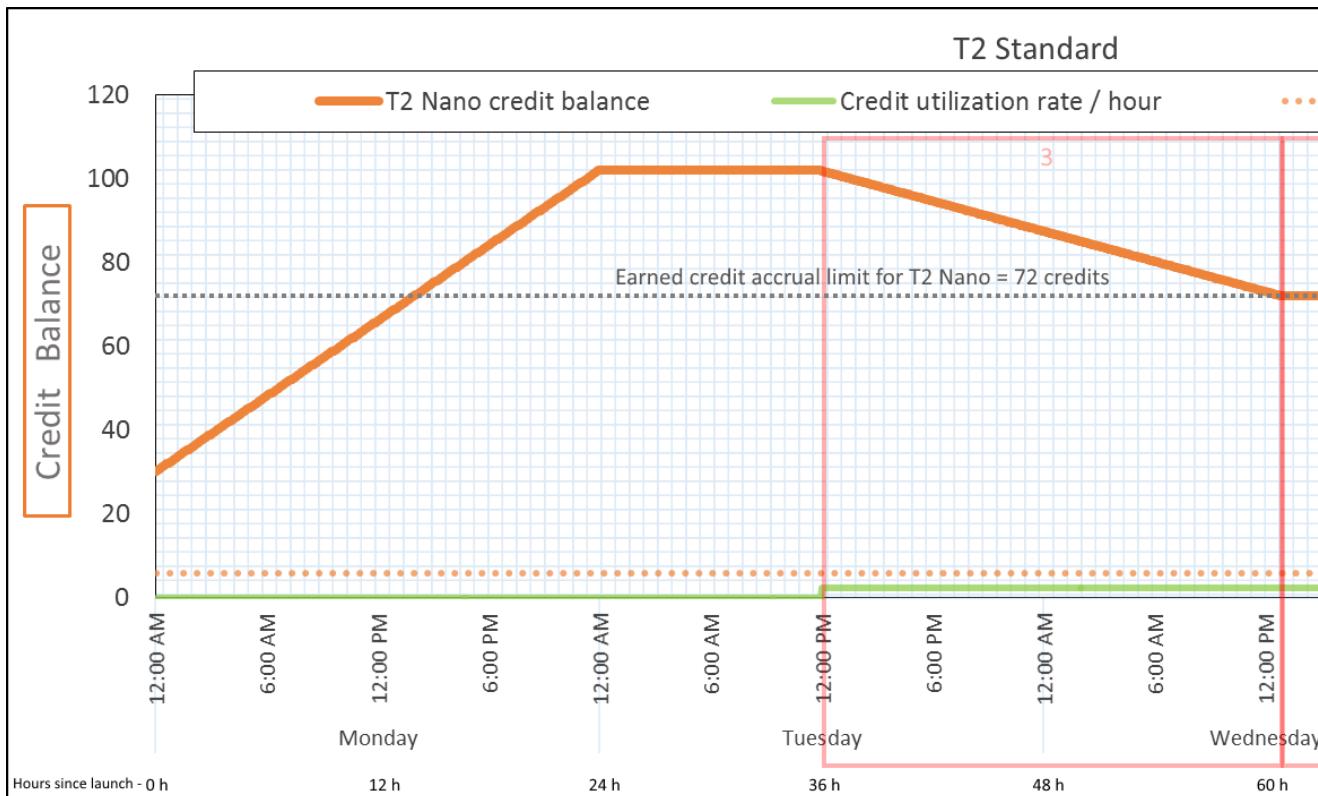
インスタンスが、獲得するよりも多くのクレジットを消費する場合、クレジット残高は減ります。

### 期間 6: 76 ~ 90 時間

次の 15 時間で、インスタンスは 2% の CPU を使用します。これには 18 クレジットが必要です。これは、期間 3 および 4 と同じ CPU 使用率です。ただし、期間 3 で残高が減り、期間 4 で頭打ちになりましたが、この期間の残高は増えます。

期間 3 で、蓄積された起動クレジットが消費されました。また、クレジットの制限を超えて獲得されたクレジットは破棄され、クレジット残高が減りました。期間 4 で、インスタンスが消費したクレジットは獲得したクレジットよりも少なくなりました。限度額を超えて獲得したクレジットはすべて破棄されたため、残高は最大 72 クレジットとなりました。

この期間に蓄積された起動クレジットではなく、残高の蓄積された獲得クレジットの数は制限を下回っています。獲得クレジットは破棄されません。さらに、インスタンスは消費するよりも多くのクレジットを獲得し、クレジット残高が増えます。



クレジットの消費率	24 時間あたり 28.8 クレジット (1 時間ごとに 1.2 クレジット、2% の CPU 使用率、40% のクレジット獲得率)— 15 時間以上で 18 クレジット
クレジットの獲得率	24 時間あたり 72 クレジット (15 時間で 45 クレジット)
クレジットの破棄率	24 時間あたり 0 クレジット
クレジット残高	72 クレジット (残高は 24 時間あたり 43.2 クレジットの率で増えます— 変更率 = 消費率 28.8/24 + 獲得率 72/24)

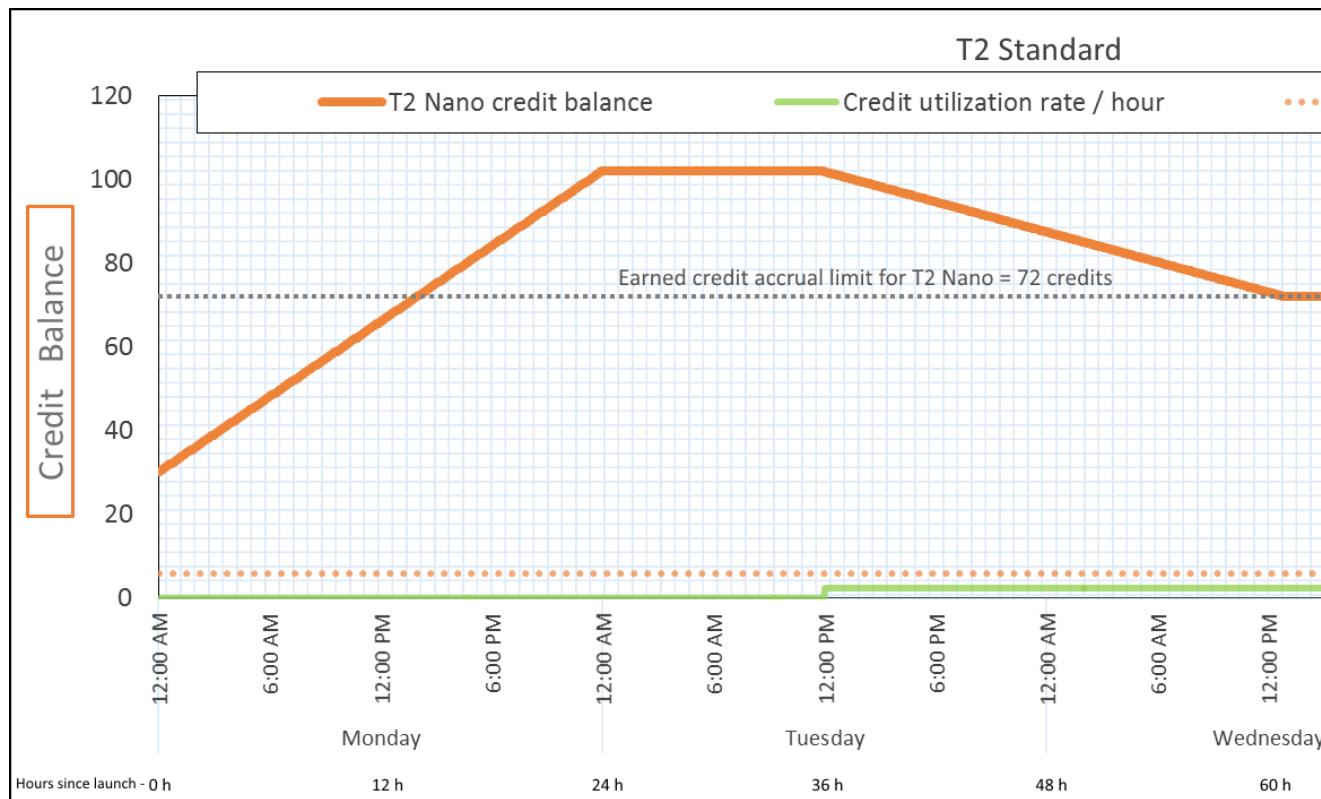
## 結論

インスタンスが、獲得するよりも少ないクレジットを消費する場合、クレジット残高は増えます。

### 期間 7: 91 ~ 96 時間

次の 6 時間は、インスタンスはアイドル状態になり (—CPU 使用率は 0%—)、クレジットは消費されません。これは、期間 2 の —CPU 使用率と同じですが、残高は 102 クレジットで頭打ちになりません。インスタンスのクレジット残高の制限である 72 クレジットで頭打ちになります。

期間 2 で、クレジット残高には蓄積された 30 起動クレジットが含まれます。起動クレジットは期間 3 で消費されました。実行中のインスタンスはそれ以上起動クレジットを取得できません。クレジット残高の制限に達すると、制限を超えて獲得されたクレジットは破棄されます。



クレジットの消費率	24 時間あたり 0 クレジット (0% の CPU 使用率)
クレジットの獲得率	24 時間あたり 72 クレジット
クレジットの破棄率	24 時間あたり 72 クレジット (100% のクレジット獲得率)
クレジット残高	72 クレジット (0 起動クレジット、72 獲得クレジット)

## 結論

インスタンスはクレジットを継続して獲得しますが、クレジット残高の制限に達した場合、獲得クレジットはそれ以上蓄積されません。制限に到達すると、新しく獲得されたクレジットはすべて破棄されます。クレジット残高の制限は、インスタンスが 24 時間に獲得できるクレジット数によって決まります。クレジット残高の制限の詳細については、[クレジットの表 \(p. 201\)](#)を参照してください。

## バーストパフォーマンスインスタンスの使用

これらのインスタンスの起動、モニタリング、および変更の手順は似ています。主な違いは、起動時のデフォルトのクレジット指定です。デフォルトのクレジット指定を変更しない場合、デフォルトは次のようにになります。

- T3 および T3a インスタンスは、デフォルトで `unlimited` として起動します。
- T2 インスタンスは、デフォルトで `standard` として起動します。

## コンテンツ

- バーストパフォーマンスインスタンスを無制限またはスタンダードとして起動する (p. 223)
- Auto Scaling グループを使用してバーストパフォーマンスインスタンスを無制限で起動する (p. 224)
- バーストパフォーマンスインスタンスのクレジット指定の表示 (p. 225)
- バーストパフォーマンスインスタンスのクレジット指定の変更 (p. 226)
- アカウントのデフォルトのクレジット指定の設定 (p. 226)
- デフォルトのクレジット指定の表示 (p. 227)

### バーストパフォーマンスインスタンスを無制限またはスタンダードとして起動する

T3 および T3a インスタンスは、デフォルトで `unlimited` として起動します。T2 インスタンスは、デフォルトで `standard` として起動します。

これらのインスタンスの AMI とドライバーの要件の詳細については、「[リリースノート \(p. 198\)](#)」を参照してください。

インスタンスの起動には、Amazon EBS ボリュームをルートデバイスとして使用する必要があります。詳細については、「[Amazon EC2 ルートデバイスボリューム \(p. 16\)](#)」を参照してください。

Amazon EC2 コンソール、AWS SDK、コマンドラインツール、または Auto Scaling グループを使用して、インスタンスを `unlimited` または `standard` として起動できます。詳細については、「[Auto Scaling グループを使用してバーストパフォーマンスインスタンスを無制限で起動する \(p. 224\)](#)」を参照してください。

バーストパフォーマンスインスタンスを無制限またはスタンダードとして起動するには (コンソール)

- 「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」の手順に従います。
- [インスタンスタイプの選択] ページで、インスタンスタイプを選択し、[次の手順: インスタンスの詳細設定] を選択します。
- クレジット指定を選択します。T3 および T3a のデフォルトは `unlimited` で、T2 のデフォルトは `standard` です。
  - T3 または T3a インスタンスを `standard` として起動するには、[インスタンスの詳細を設定] ページの [T2/T3 無制限] で、[有効化] をオフにします。
  - T2 インスタンスを `unlimited` として起動するには、[Configure Instance Details (インスタンスの詳細を設定)] ページの [T2/T3 Unlimited (T2/T3 無制限)] で、[Enable (有効化)] を選択します。
- ウィザードに従って続行します。[Review Instance Launch (インスタンス作成の確認)] ページでオプションの確認が終了したら、[Launch (起動)] を選択します。詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。

バーストパフォーマンスインスタンスを無制限またはスタンダードとして起動するには (AWS CLI)

`run-instances` コマンドを使用して、インスタンスを起動します。`--credit-specification CpuCredits=` パラメータを使用してクレジット指定を指定します。有効なクレジット指定は `unlimited` と `standard` です。

- T3 および T3a で、`--credit-specification` パラメータを含めない場合、インスタンスはデフォルトで `unlimited` として起動します。
- T2 で、`--credit-specification` パラメータを含めない場合、インスタンスはデフォルトで `standard` として起動します。

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t3.micro --key-name MyKeyPair --credit-specification "CpuCredits=unlimited"
```

## Auto Scaling グループを使用してバーストパフォーマンスインスタンスを無制限で起動する

バーストパフォーマンスインスタンスが起動または開始する際、優れたポートストラップエクスペリエンスには CPU クレジットが必要です。Auto Scaling グループを使用してインスタンスを起動する場合は、インスタンスを `unlimited` として設定することをお勧めします。そうする場合、インスタンスは Auto Scaling グループによって自動的に起動または再開されたときに余剰クレジットを使用します。余剰クレジットを使用することで、パフォーマンスの制限を防ぐことができます。

### 起動テンプレートの作成

インスタンスを Auto Scaling グループで `unlimited` として起動するには、起動に起動テンプレートを使用する必要があります。起動設定では、インスタンスを `unlimited` として起動することはサポートされません。

インスタンスを無制限として起動する起動テンプレートを作成するには (コンソール)

1. [Auto Scaling グループの起動テンプレートの作成](#) の手順に従います。
2. [テンプレートコンテンツの起動] の [インスタンスタイプ] で、T3、T3a、または T2 インスタンスサイズを選択します。
3. Auto Scaling グループでインスタンスを `unlimited` として起動するには、[高度な詳細] の [T2/T3 Unlimited (T2/T3 無制限)] で、[有効化] を選択します。
4. 起動テンプレートパラメータの定義が終了したら、[Create launch template (起動テンプレートの作成)] を選択します。詳細については、『Amazon EC2 Auto Scaling ユーザーガイド』の「[Auto Scaling グループの起動テンプレートの作成](#)」を参照してください。

インスタンスを無制限として起動する起動テンプレートを作成するには (AWS CLI)

`create-launch-template` コマンドを使用して、`unlimited` を CPU 使用率に関するクレジット指定として指定します。

- T3 および T3a で、`CreditSpecification={CpuCredits=unlimited}` 値を含めない場合、インスタンスはデフォルトで `unlimited` として起動します。
- T2 で、`CreditSpecification={CpuCredits=unlimited}` 値を含めない場合、インスタンスはデフォルトで `standard` として起動します。

```
aws ec2 create-launch-template --launch-template-name MyLaunchTemplate
--version-description FirstVersion --launch-template-data
ImageId=ami-8c1be5f6,InstanceType=t3.medium,CreditSpecification={CpuCredits=unlimited}
```

### 起動テンプレートを使用して Auto Scaling グループを関連付ける

起動テンプレートを Auto Scaling グループに関連付けるには、起動テンプレートを使用して Auto Scaling グループを作成するか、または既存の Auto Scaling グループに起動テンプレートを追加します。

起動テンプレートを使用して Auto Scaling グループを作成するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 画面の上部のナビゲーションバーで、起動テンプレートを作成したときに使用したのと同じリージョンを選択します。
3. ナビゲーションペインで [Auto Scaling グループ]、[Auto Scaling グループの作成] の順に選択します。
4. [Launch Template (起動テンプレート)] で、起動テンプレートを選択し、[次のステップ] を選択します。
5. Auto Scaling グループ用のフィールドに入力します。[Review page (確認ページ)] で設定の確認を終えたら、[Create Auto Scaling group (Auto Scaling グループの作成)] を選択します。詳細については、

『Amazon EC2 Auto Scaling ユーザーガイド』の「[起動テンプレートを使用した Auto Scaling グループの作成](#)」を参照してください。

起動テンプレートを使用して Auto Scaling グループを作成するには (AWS CLI)

`create-auto-scaling-group` AWS CLI コマンドを使用して、`--launch-template` パラメータを指定します。

既存の Auto Scaling グループに起動テンプレートを追加するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 画面の上部のナビゲーションバーで、起動テンプレートを作成したときに使用したのと同じリージョンを選択します。
3. ナビゲーションペインで、[Auto Scaling Groups] をクリックします。
4. Auto Scaling グループの一覧から Auto Scaling グループを選択し、[アクション]、[編集] の順に選択します。
5. [Details (詳細)] タブの [Launch Template (起動テンプレート)] で起動テンプレートを選択して、[Save (保存)] を選択します。

既存の Auto Scaling グループに起動テンプレートを追加するには (AWS CLI)

`update-auto-scaling-group` AWS CLI コマンドを使用して、`--launch-template` パラメータを指定します。

#### バーストパフォーマンスインスタンスのクレジット指定の表示

実行中または停止中のインスタンスのクレジット指定 (`unlimited` または `standard`) を表示できます。

バーストインスタンスのクレジット指定を表示するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 左ナビゲーションペインで [インスタンス] を選択し、インスタンスを選択します。
3. [Description (説明)] を選択し、[T2/T3 Unlimited (T2/T3 無制限)] フィールドを表示します。
  - 値が `Enabled` の場合、インスタンスは `unlimited` として設定されます。
  - 値が `Disabled` の場合、インスタンスは `standard` として設定されます。

バーストパフォーマンスインスタンスのクレジット指定を記述するには (AWS CLI)

`describe-instance-credit-specifications` コマンドを使用します。1つ以上のインスタンス ID を指定しない場合、以前に `unlimited` クレジット仕様で設定されていたインスタンスだけでなく、`unlimited` クレジット指定のすべてのインスタンスが返されます。たとえば、T3 インスタンスを M4 インスタンスにサイズ変更し、`unlimited` に設定している場合、Amazon EC2 は M4 インスタンスを返します。

#### Example

```
aws ec2 describe-instance-credit-specifications --instance-id i-1234567890abcdef0
```

出力例を次に示します。

```
{
  "InstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CpuCredits": "unlimited"
    }
  ]
}
```

```
}
```

## バーストパフォーマンスインスタンスのクレジット指定の変更

実行中または停止中のインスタンスのクレジット指定は、`unlimited` と `standard` の間でいつでも切り替えることができます。

バーストパフォーマンスインスタンスのクレジット指定を変更するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 左ナビゲーションペインで [インスタンス] を選択し、インスタンスを選択します。複数のインスタンスのクレジット指定を一度に変更するには、適用可能なインスタンスをすべて選択します。
3. [Actions (アクション)]、[Instance Settings (インスタンスの設定)]、[Change T2/T3 Unlimited (T2/T3 無制限の変更)] の順に選択します。

### Note

[Change T2/T3 Unlimited (T2/T3 無制限の変更)] オプションは、T3、T3a、または T2 インスタンスを選択した場合にのみ有効になります。

4. クレジット指定を `unlimited` に変更するには、[有効化] を選択します。クレジット指定を `standard` に変更するには、[無効化] を選択します。インスタンスの現在のクレジット指定は、インスタンス ID の後の括弧内に表示されます。

バーストパフォーマンスインスタンスのクレジット指定を変更するには (AWS CLI)

`modify-instance-credit-specification` コマンドを使用します。`--instance-credit-specification` パラメータを使用して、インスタンスとクレジット指定を指定します。有効なクレジット指定は `unlimited` と `standard` です。

### Example

```
aws ec2 modify-instance-credit-specification --region us-east-1 --instance-credit-specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

出力例を次に示します。

```
{
  "SuccessfulInstanceCreditSpecifications": [
    {
      "InstanceId": "i- 1234567890abcdef0"
    }
  ],
  "UnsuccessfulInstanceCreditSpecifications": []
}
```

## アカウントのデフォルトのクレジット指定の設定

デフォルトのクレジット指定は、AWS リージョンごとにアカウントレベルで設定できます。デフォルトのクレジット指定は、インスタンスファミリーごと (T2、T3、または T3a) に指定します。

AWS マネジメントコンソール でインスタンスの起動ウィザードを使用してインスタンスを起動する場合、[T2/T3 無制限] の値はアカウントレベルのデフォルトのクレジット指定よりも優先されます。AWS CLI を使用してインスタンスを起動する場合、アカウント内のすべての新しいバーストパフォーマンスインスタンスは、デフォルトのクレジットオプションを使用して起動されます。既存の実行中または停止中のインスタンスのクレジット指定には影響しません。

modify-default-credit-specification API は非同期オペレーションとして AWS リージョンレベルで機能し、各アベイラビリティーゾーンのクレジットオプションを変更します。リージョン内のすべてのゾーンは 5 分以内に更新されます。ただし、このオペレーション中に起動したインスタンスは、ゾーンが更新されるまで新しいクレジットオプションを利用できない場合があります。更新が発生したかどうかを確認するには、get-default-credit-specification を呼び出して、デフォルトのクレジット指定の更新を確認します。詳細については、「[デフォルトのクレジット指定の表示 \(p. 227\)](#)」を参照してください。

#### Note

インスタンスファミリーのデフォルトのクレジット指定は、継続した 5 分間に 1 回のみ変更でき、継続した 24 時間中に最大 4 回変更できます。

アカウントレベルでデフォルトのクレジット指定を設定するには (AWS CLI)

`modify-default-credit-specification` コマンドを使用します。--cpu-credits パラメータを使用して、AWS リージョン、インスタンスファミリー、およびデフォルトのクレジット指定を指定します。有効なデフォルトのクレジット指定は、unlimited および standard です。

```
aws ec2 modify-default-credit-specification --region us-east-1 --instance-family t2 --cpu-credits unlimited
```

#### デフォルトのクレジット指定の表示

AWS リージョンごとにアカウントレベルで、バーストパフォーマンスインスタンスファミリーのデフォルトのクレジット指定を表示できます。

アカウントレベルでデフォルトのクレジット指定を表示するには (AWS CLI)

`get-default-credit-specification` コマンドを使用します。AWS リージョンとインスタンスファミリーを指定します。

```
aws ec2 get-default-credit-specification --region us-east-1 --instance-family t2
```

#### CPU クレジットの監視

CloudWatch コンソールの Amazon EC2 インスタンス別メトリクスの各インスタンスのクレジットバランスを確認できます。

##### トピック

- [バーストパフォーマンスインスタンスの追加 CloudWatch メトリクス \(p. 227\)](#)
- [CPU クレジット使用状況の計算 \(p. 229\)](#)

#### バーストパフォーマンスインスタンスの追加 CloudWatch メトリクス

T3、T3a、および T2 インスタンスにはこれらの追加 CloudWatch メトリクスがあり、5 分ごとに更新されます。

- CPUCreditUsage – 測定期間に消費された CPU クレジットの数。
- CPUCreditBalance – インスタンスが蓄積する CPU クレジット数。このバランスは CPU がバーストする際に枯渇し、CPU クレジットは獲得するよりも速い速度で使用されます。
- CPUSurplusCreditBalance – CPUCreditBalanceがゼロになった時に CPU パフォーマンスを保持するために消費される、余剰 CPU クレジットの数。
- CPUSurplusCreditsCharged – 24 時間で獲得できる [CPU クレジットの最大数 \(p. 201\)](#) を越えた、追加料金が発生する分の余剰 CPU クレジットの数。

最後の 2 つのメトリクスは `unlimited` として設定されたインスタンスにのみ適用されます。

バーストパフォーマンスインスタンスの CloudWatch メトリクスの説明を次の表に示します。詳細については、「[インスタンスの利用可能な CloudWatch メトリクスのリスト表示 \(p. 644\)](#)」を参照してください。

メトリクス	説明
CPUCreditUsage	<p>CPU 使用率に関してインスタンスで消費される CPU クレジットの数。1 つの CPU クレジットは、1 個の vCPU が 100% の使用率で 1 分間実行されること、または、vCPU、使用率、時間の同等の組み合わせ（たとえば、1 個の vCPU が 50% の使用率で 2 分間実行されるか、2 個の vCPU が 25% の使用率で 2 分間実行される）に相当します。</p> <p>CPU クレジットメトリクスは、5 分間隔でのみ利用可能です。5 分を超える期間を指定する場合は、Sum 統計の代わりに Average 統計を使用します。</p> <p>単位: クレジット (vCPU 分)</p>
CPUCreditBalance	<p>インスタンスが起動または開始後に蓄積した獲得 CPU クレジットの数。T2 スタンダードの場合、CPUCreditBalance には蓄積された起動クレジットの数も含まれます。</p> <p>クレジットは、獲得後にクレジット残高に蓄積され、消費されるとクレジット残高から削除されます。クレジットバランスには、インスタンスサイズによって決まる上限があります。制限に到達すると、獲得された新しいクレジットはすべて破棄されます。T2 スタンダードの場合、起動クレジットは制限に対してカウントされません。</p> <p>CPUCreditBalance のクレジットは、インスタンスがそのベースライン CPU 使用率を超えてバーストするために消費できます。</p> <p>インスタンスが実行中の場合、CPUCreditBalance のクレジットは期限切れになりません。T3 または T3a インスタンスが停止すると、CPUCreditBalance 値は 7 日間保持されます。その後、蓄積されたすべてのクレジットが失われます。T2 インスタンスが停止すると、CPUCreditBalance 値は保持されず、蓄積されたすべてのクレジットが失われます。</p> <p>CPU クレジットメトリクスは、5 分間隔でのみ利用可能です。</p> <p>単位: クレジット (vCPU 分)</p>
CPUSurplusCreditBalance	<p>CPUCreditBalance 値がゼロの場合に unlimited インスタンスによって消費された余剰クレジットの数。</p> <p>CPUSurplusCreditBalance 値は獲得した CPU クレジットによって支払われます。余剰クレジットの数が、24 時間にインスタンスが獲得できるクレジットの最大数を超えている場合、最大数を超えて消費された余剰クレジットに対しては料金が発生します。</p> <p>単位: クレジット (vCPU 分)</p>
CPUSurplusCreditsCharged	<p>獲得 CPU クレジットにより支払われないために追加料金が発生した、消費された余剰クレジットの数。</p> <p>消費された余剰クレジットは、以下のいずれかの状況に当てはまるとき料金が発生します。</p>

メトリクス	説明
	<ul style="list-style-type: none"><li>消費された余剰クレジットが、インスタンスが 24 時間に獲得できる最大クレジット数を超えており、最大数を越えて消費された余剰クレジットは、時間の最後に課金されます。</li><li>インスタンスが停止または終了した。</li><li>インスタンスは <code>unlimited</code> から <code>standard</code> に切り替わります。</li></ul> <p>単位: クレジット (vCPU 分)</p>

### CPU クレジット使用状況の計算

インスタンスの CPU クレジット使用状況は、前述の表で説明したインスタンス CloudWatch メトリクスを使用して計算されます。

Amazon EC2 は、メトリクスを 5 分ごとに CloudWatch に送信します。前のメトリクス値の参照はいつでも、5 分前に送信された、直前のメトリクス値を意味します。

### スタンダードインスタンスの CPU クレジット使用状況の計算

- CPU クレジット残高は、CPU 利用率がベースラインを下回り、前の 5 分間に消費したクレジットが獲得したクレジットより少なかった場合に増加します。
- CPU クレジット残高は、CPU 利用率がベースラインを上回り、前の 5 分間に消費したクレジットが獲得したクレジットよりも多かった場合に減少します。

数学的に、これは次の式で表されます。

#### Example

```
CPUCreditBalance = prior CPUCreditBalance + [Credits earned per hour * (5/60) - CPUCreditUsage]
```

インスタンスのサイズは、インスタンスが 1 時間あたりに獲得できるクレジットの数と、クレジット残高に蓄積できる獲得クレジットの数を決定します。1 時間あたりに獲得するクレジット数と、各インスタンスサイズのクレジット残高制限については、「[クレジットの表 \(p. 201\)](#)」を参照してください。

#### 例

この例では、`t3.nano` インスタンスを使用します。インスタンスの `CPUCreditBalance` 値を計算するには、前述の式を次のように使用します。

- `CPUCreditBalance` – 計算する現在のクレジットバランス。
- `prior CPUCreditBalance` – 5 分前のクレジットバランス。この例では、インスタンスは 2 クレジットを蓄積しています。
- `Credits earned per hour` – `t3.nano` インスタンスは 1 時間あたり 6 クレジット獲得します。
- `5/60` – CloudWatch メトリクスのパブリッシュ間の 5 分間隔を表します。1 時間あたりに獲得するクレジットに 60 分の 5 (5 分) を掛けて、インスタンスが過去 5 分間に獲得したクレジット数を求めます。`t3.nano` インスタンスは、5 分ごとに 0.5 クレジットを獲得します。
- `CPUCreditUsage` – インスタンスが過去 5 分間に消費したクレジット数。この例では、インスタンスは過去 5 分間に 1 クレジットを消費しました。

これらの値を使用して、`CPUCreditBalance` の値を計算できます。

Example

```
CPUCreditBalance = 2 + [0.5 - 1] = 1.5
```

無制限インスタンスの CPU クレジット使用状況の計算

T3、T3a、または T2 インスタンスがベースライン以上でバーストする必要がある場合、余剰クレジットを消費する前に、蓄積されたクレジットが常に消費されます。蓄積した CPU クレジット残高を使いきると、余剰クレジットを消費して必要なだけバーストします。CPU 利用率がベースラインを下回った場合、インスタンスが獲得クレジットを蓄積する前に常に余剰クレジットが支払われます。

この 5 分間に発生するアクティビティを反映するため、次の式では *Adjusted balance* という用語を使用します。`CPUCreditBalance` および `CPUSurplusCreditBalance` の CloudWatch メトリクスの値に達するため、この値を使用します。

Example

```
Adjusted balance = [prior CPUCreditBalance - prior CPUSurplusCreditBalance] + [Credits earned per hour * (5/60) - CPUCreditUsage]
```

0 に対する *Adjusted balance* の値は、インスタンスが獲得したすべてのクレジットがバーストに消費され、余剰クレジットは消費されなかつたことを示します。結果として、`CPUCreditBalance` と `CPUSurplusCreditBalance` は 0 に設定されます。

*Adjusted balance* の正の値は、インスタンスが獲得クレジットを蓄積し、前の余剰クレジットが（ある場合）支払われたことを示します。結果として、*Adjusted balance* の値は `CPUCreditBalance` に割り当てられ、`CPUSurplusCreditBalance` は 0 に設定されます。インスタンスサイズは、蓄積可能な最大クレジット数 (p. 201) を決定します。

Example

```
CPUCreditBalance = min [max earned credit balance, Adjusted balance]  
CPUSurplusCreditBalance = 0
```

*Adjusted balance* の負の値は、インスタンスが蓄積したすべての獲得クレジットに加えて、余剰クレジットもバーストに消費されたことを示します。結果として、*Adjusted balance* の値は `CPUSurplusCreditBalance` と `CPUCreditBalance` に割り当てられ、0 に設定されます。繰り返しになりますが、インスタンスサイズは、蓄積可能な最大クレジット数 (p. 201) を決定します。

Example

```
CPUSurplusCreditBalance = min [max earned credit balance, -Adjusted balance]  
CPUCreditBalance = 0
```

消費される余剰クレジットがインスタンスに蓄積可能な最大クレジットを越えた場合、余剰クレジット残高は前述の式に示すように最大に設定されます。残りの余剰クレジットは `CPUSurplusCreditsCharged` メトリクスで示すように課金されます。

Example

```
CPUSurplusCreditsCharged = max [-Adjusted balance - max earned credit balance, 0]
```

最後に、`CPUSurplusCreditBalance` により追跡された余剰クレジットもインスタンスの終了時に課金されます。インスタンスを `unlimited` から `standard` に切り替えると、残りの `CPUSurplusCreditBalance` も課金されます。

## コンピュート最適化インスタンス

コンピューティング最適化インスタンスは、高パフォーマンスプロセッサから恩恵を受けるコンピューティングバウンドな用途に最適です。このインスタンスは、以下の用途に最適です。

- 作業負荷のバッチ処理
- メディアの変換
- 高性能なウェブサーバー
- ハイパフォーマンスコンピューティング (HPC)
- 科学的なモデル
- 専用ゲームサーバーおよび広告エンジン
- 機械学習推論やその他の大量の演算を行うアプリケーション

詳細については、「[Amazon EC2 C5 インスタンス](#)」を参照してください。

### コンテンツ

- [ハードウェア仕様 \(p. 231\)](#)
- [インスタンスのパフォーマンス \(p. 232\)](#)
- [ネットワークパフォーマンス \(p. 232\)](#)
- [SSD I/O パフォーマンス \(p. 233\)](#)
- [インスタンスの機能 \(p. 234\)](#)
- [リリースノート \(p. 235\)](#)

## ハードウェア仕様

以下に示しているのは、コンピューティング最適化インスタンスのハードウェア仕様の要約です。

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
c4.large	2	3.75
c4.xlarge	4	7.5
c4.2xlarge	8	15
c4.4xlarge	16	30
c4.8xlarge	36	60
c5.large	2	4
c5.xlarge	4	8
c5.2xlarge	8	16
c5.4xlarge	16	32
c5.9xlarge	36	72
c5.12xlarge	48	96
c5.18xlarge	72	144
c5.24large	96	192

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
c5.metal	96	192
c5d.large	2	4
c5d.xlarge	4	8
c5d.2xlarge	8	16
c5d.4xlarge	16	32
c5d.9xlarge	36	72
c5d.12xlarge	48	96
c5d.18xlarge	72	144
c5d.24large	96	192
c5d.metal	96	192
c5n.large	2	5.25
c5n.xlarge	4	10.5
c5n.2xlarge	8	21
c5n.4xlarge	16	42
c5n.9xlarge	36	96
c5n.18xlarge	72	192
c5n.metal	72	192

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

CPU オプションの指定についての詳細は、「[CPU オプションの最適化 \(p. 571\)](#)」を参照してください。

## インスタンスのパフォーマンス

EBS 最適化インスタンスは、インスタンスからの Amazon EBS I/O とその他のネットワークトラフィックとの競合を排除することによって、EBS ボリュームの安定した高パフォーマンスを実現できます。一部のコンピューティングの最適化インスタンスは、追加料金なしでデフォルトで EBS 最適化されています。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

一部のコンピューティングの最適化インスタンスでは、Linux でプロセッサの C ステートと P ステートを制御できます。C ステートは非アクティブ時のコアのスリープレベルを制御し、P ステートは希望するコアからのパフォーマンス (CPU 周波数) を制御します。詳細については、「[EC2 インスタンスタイプのプロセッサのステート制御 \(p. 561\)](#)」を参照してください。

## ネットワークパフォーマンス

サポート対象のインスタンスタイプで、拡張されたネットワーキング機能を有効にすることができます。拡張ネットワーキングでは、パケット毎秒 (PPS) が非常に大きく、ネットワークのストレスが少なく、レイテンシーが低くなります。詳細については、「[Linux の拡張ネットワーキング \(p. 737\)](#)」を参照してください。

拡張されたネットワーキングのための Elastic Network Adapter (ENA) を使用するインスタンスタイプは、高いパケット/秒パフォーマンスと一貫して低いレイテンシーを同時に実現します。ほとんどのアプリケーションでは、高いレベルのネットワークパフォーマンスが一貫して必要なわけではありませんが、データの送受信時にアクセスする帯域幅を増やすことでメリットを得られます。ENA を使用し、「最大 10 Gbps」または「最大 25 Gbps」のネットワークパフォーマンスをサポートするインスタンスサイズでは、ネットワーク I/O クレジットメカニズムを使用して、平均帯域幅使用率に基づいてインスタンスにネットワーク帯域幅を割り当てます。これらのインスタンスでは、ネットワーク帯域幅がベースライン制限を下回るとクレジットを獲得し、これらのクレジットをネットワークデータ転送を実行するときに使用できます。

以下に示しているのは、拡張ネットワーキングをサポートするコンピューティング最適化インスタンスのネットワークパフォーマンスの要約です。

インスタンスタイプ	ネットワークパフォーマンス	拡張ネットワーキング
c5.4xlarge 以下   c5d.4xlarge 以下	最大 10 Gbps	<a href="#">ENA (p. 738)</a>
c5.9xlarge   c5d.9xlarge	10 Gbps	<a href="#">ENA (p. 738)</a>
c5.12xlarge   c5d.12xlarge	12 Gbps	<a href="#">ENA (p. 738)</a>
c5n.4xlarge 以下	最大 25 Gbps	<a href="#">ENA (p. 738)</a>
c5.18xlarge   c5.24xlarge   c5.metal   c5d.18xlarge   c5d.24xlarge   c5d.metal	25 Gbps	<a href="#">ENA (p. 738)</a>
c5n.9xlarge	50 Gbps	<a href="#">ENA (p. 738)</a>
c5n.18xlarge   c5n.metal	100 Gbps	<a href="#">ENA (p. 738)</a>
c4.large	中	<a href="#">Intel 82599 VF (p. 751)</a>
c4.xlarge   c4.2xlarge   c4.4xlarge	高	<a href="#">Intel 82599 VF (p. 751)</a>
c4.8xlarge	10 Gbps	<a href="#">Intel 82599 VF (p. 751)</a>

## SSD I/O パフォーマンス

カーネルバージョン 4.4 以降の Linux AMI を使用し、インスタンスで利用可能なすべての SSD ベースのインスタンストアボリュームを使用する場合は、以下の表に示されている IOPS (4,096 バイトブロックサイズ) のパフォーマンスを得ることができます (キューの深さの飽和度において)。それ以外の場合、IOPS パフォーマンスは低下します。

インスタンスサイズ	100% のランダム読み取り時 IOPS	書き込み IOPS
c5d.large *	20,000	9,000
c5d.xlarge *	40,000	18,000
c5d.2xlarge *	80,000	37,000
c5d.4xlarge *	175,000	75,000
c5d.9xlarge	350,000	170,000

インスタンスサイズ	100% のランダム読み取り時 IOPS	書き込み IOPS
c5d.12xlarge	700,000	340,000
c5d.18xlarge	700,000	340,000
c5d.24xlarge	1,400,000	680,000
c5d.metal	1,400,000	680,000

\* これらのインスタンスの場合、最大で指定されたパフォーマンスを得ることができます。

インスタンスに SSD ベースのインスタンストアボリュームを使用するほど、アーカイブできる書き込み IOPS の数は減少します。これは、SSD コントローラーが実行する必要がある追加の作業が原因です。SSD コントローラーは、利用可能な領域を見つけ、既存のデータを再書き込みし、未使用の領域を消去して、再書き込みができるようにします。このガベージコレクションというプロセスにより、SSD への内部的な書き込み増幅が発生し、ユーザーの書き込み操作に対する SSD 書き込み操作の割合として表示されます。書き込み操作が 4,096 バイトの倍数でないか、4,096 バイトの境界に整合していない場合、パフォーマンスの低下はさらに大きくなります。少量のバイト数または整合していないバイト数で書き込む場合、SSD コントローラーは周辺のデータを読み取り、その結果を新しい場所に保存する必要があります。このパターンにより、書き込み増幅が大幅に増え、レイテンシーが増加し、I/O パフォーマンスが大きく低下します。

SSD コントローラーは、複数の方法を利用すると、書き込み増幅の影響を減らすことができます。このような方法の 1 つには、SSD インスタンストレージに領域を予約し、コントローラーが書き込み操作に利用できる領域をより効率的に管理できるようにすることです。これをオーバープロビジョニングと呼びます。インスタンスに提供された SSD ベースのインスタンストアボリュームには、オーバープロビジョニングに対して予約された領域がありません。書き込み増幅を減らすには、ボリュームの 10% を未使用的状態のままにし、SSD コントローラーがこれをオーバープロビジョニングに使用できるようにすることをお勧めしますこれにより、使用できるストレージは減りますが、ディスクが総容量に近づいた場合でもパフォーマンスを向上させることができます。

TRIM をサポートするインスタンストアボリュームの場合、TRIM コマンドを使用して、書き込んだデータが不要になったときはいつでも SSD コントローラーに通知することができます。これにより、より多くの空き領域がコントローラーに与えられ、その結果書き込み増幅が減り、パフォーマンスが向上します。詳細については、「[インスタンストアボリュームの TRIM のサポート \(p. 1087\)](#)」を参照してください。

## インスタンスの機能

コンピュート最適化インスタンスの機能の概要を以下に示します。

	EBS のみ	NVMe EBS	インスタンストア	配置グループ
C4	はい	いいえ	いいえ	あり
C5	はい	はい	いいえ	あり
C5d	いいえ	はい	NVMe *	あり
C5n	はい	はい	いいえ	あり

\* ルートデバイスボリュームは、Amazon EBS ボリュームにする必要があります。

詳細については、以下を参照してください。

- [Linux インスタンスの Amazon EBS および NVMe \(p. 1027\)](#)

- Amazon EC2 インスタンストア (p. 1076)
- プレイスマントグループ (p. 791)

## リリースノート

- C5 および C5d インスタンスは、第 1 世代 (Skylake-SP) または第 2 世代 (Cascade Lake) の 3.1 GHz Intel Xeon Platinum 8000 シリーズプロセッサーを搭載しています。
- C4、C5、C5d、および C5n インスタンスには、64 ビットの EBS-backed HVM AMI が必要です。これらのインスタンスはハイメモリであるため、そのキャパシティーを活用するには 64 ビットのオペレーティングシステムが必要です。HVM AMI は、ハイメモリインスタンスタイプの準仮想化 (PV) AMI よりも優れたパフォーマンスを提供します。さらに、拡張ネットワーキングを利用するには、HVM AMI を使用する必要があります。
- C5、C5d、および C5d インスタンスには以下の要件があります。
  - NVMe ドライバー (p. 1027)がインストールされている必要があります。
  - Elastic Network Adapter (ENA) ドライバー (p. 738)がインストールされている必要があります。

以下の Linux AMI はこれらの要件を満たしています。

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (linux-aws カーネル) 以降
- Red Hat Enterprise Linux 7.4 以降
- SUSE Linux Enterprise Server 12 SP2 以降
- CentOS 7.4.1708 以降
- FreeBSD 11.1 以降
- Debian GNU/Linux 9 以降
- C5、C5d、および C5n インスタンスは、ネットワークインターフェイス、EBS ボリューム、および NVMe インスタンストアボリュームを含め、最大 28 のアタッチをサポートしています。インスタンスごとに 1 つ以上のネットワークインターフェイスのアタッチメントがあります。
- ベアメタルインスタンスを起動すると、基盤となるサーバーが起動します。これには、すべてのハードウェアやファームウェアコンポーネントの確認が含まれます。つまり、インスタンスが実行状態になってからネットワーク経由で使用できるようになるまでに 20 分かかることがあります。
- ベアメタルインスタンスから EBS ボリュームまたはセカンダリネットワークインターフェイスをアタッチまたはデタッチするには、PCIe のネイティブホットプラグサポートが必要です。Amazon Linux 2 および最新バージョンの Amazon Linux AMI は PCIe ネイティブホットプラグをサポートしていますが、以前のバージョンではサポートされていません。次の Linux カーネル設定オプションを有効にする必要があります。

```
CONFIG_HOTPLUG_PCI_PCIE=y
CONFIG_PCIEASPM=y
```

- ベアメタルインスタンスでは、I/O ポートベースのシリアルデバイスではなく、PCI ベースのシリアルデバイスを使用しています。アップストリームの Linux カーネルと最新の Amazon Linux AMI は、このデバイスをサポートしています。また、ベアメタルインスタンスでは、システムが PCI ベースのシリアルデバイスを自動的に使用できるようにする ACPI SPCR テーブルも使用できます。最新の Windows AMI では、自動的に PCI ベースのシリアルデバイスが使用されます。
- C5、C5d、および C5n インスタンスには、API リクエストによるクリーンシャットダウンをサポートするための acpid がインストールされている必要があります。
- リージョンで起動できるインスタンスの合計数には制限があります。また、一部のインスタンスタイプにはその他の制限もあります。詳細については、「[Amazon EC2 で実行できるインスタンスの数はいくつですか？](#)」を参照してください。制限の引き上げをリクエストするには、「[Amazon EC2 インスタンス申請フォーム](#)」を使用します。

## メモリ最適化インスタンス

メモリ最適化インスタンスは、メモリ内の大きいデータセットを処理するワークロードに対して高速なパフォーマンスを実現するように設計されています。

### R4、R5、R5a、R5ad、R5d、R5dn、および R5n インスタンス

これらのインスタンスは、以下の用途に適しています。

- ・ハイパフォーマンスリレーショナル (MySQL) および NoSQL (MongoDB、Cassandra) データベース。
- ・キー値タイプのデータ (Memcached および Redis) のインメモリキャッシュを提供する分散型ウェブスケールキャッシュストア。
- ・ビジネスインテリジェンス用に最適化されたデータストレージ形式と分析機能 (SAP HANA など) を使用するインメモリデータベース。
- ・巨大な非構造化データ (金融サービス、Hadoop/Spark クラスター) のリアルタイム処理を実行するアプリケーション。
- ・ハイパフォーマンスコンピューティング (HPC) および電子設計オートメーション (EDA) アプリケーション。

r5.metal および r5d.metal インスタンスでは、プロセッサやメモリなどのホストサーバーの物理リソースにアプリケーションから直接アクセスすることができます。これらのインスタンスは、次の用途に適しています。

- ・仮想環境で利用できない、または完全にサポートされていない低レベルのハードウェア機能 (例: Intel VT) へのアクセスを必要とするワークロード
- ・ライセンスやサポートを目的として非仮想化環境で実行する必要があるアプリケーション

詳細については、「[Amazon EC2 R5 インスタンス](#)」を参照してください。

### ハイメモリインスタンス

ハイメモリインスタンス (u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal、および u-24tb1.metal) は、インスタンスあたり 6 TiB、9 TiB、12 TiB、18 TiB、および 24 TiB のメモリを提供します。これらのインスタンスは、SAP HANA の実稼働環境を含む大きなメモリ内のデータベースを実行するように設計されています。ホストハードウェアに直接アクセスできるベアメタルパフォーマンスを提供します。

### X1 インスタンス

これらのインスタンスは、以下の用途に適しています。

- ・SAP HANA などのメモリ内データベース (Business Suite S/4HANA の SAP 認定サポート、Business Suite on HANA (SoH)、Business Warehouse on HANA (BW)、および Data Mart Solutions on HANA を含む)。詳細については、「[AWS クラウドの SAP HANA](#)」を参照してください。
- ・Apache Spark や Presto などのビッグデータ処理エンジン。
- ・ハイパフォーマンスコンピューティング (HPC) アプリケーション。

詳細については、「[Amazon EC2 X1 インスタンス](#)」を参照してください。

### X1e インスタンス

これらのインスタンスは、以下の用途に適しています。

- ・高性能データベース。

- SAP HANA などのメモリ内データベース。詳細については、「[AWS クラウドの SAP HANA](#)」を参照してください。
- メモリを大量に消費するエンタープライズアプリケーション。

詳細については、「[Amazon EC2 X1e インスタンス](#)」を参照してください。

### z1d インスタンス

これらのインスタンスは、ハイコンピューティングとハイメモリの両方を提供し、以下の用途に最適です。

- 電子設計オートメーション (EDA)
- リレーションナルデータベースワークロード

z1d.metal インスタンスは、プロセッサとメモリなどのホストサーバーの物理リソースにアプリケーションが直接アクセスできるようにします。これらのインスタンスは、次の用途に適しています。

- 仮想環境で利用できない、または完全にサポートされていない低レベルのハードウェア機能 (例: Intel VT) へのアクセスを必要とするワークロード
- ライセンスやサポートを目的として非仮想化環境で実行する必要があるアプリケーション

詳細については、「[Amazon EC2 z1d インスタンス](#)」を参照してください。

### コンテンツ

- [ハードウェア仕様 \(p. 237\)](#)
- [メモリ性能 \(p. 240\)](#)
- [インスタンスのパフォーマンス \(p. 240\)](#)
- [ネットワークパフォーマンス \(p. 241\)](#)
- [SSD I/O パフォーマンス \(p. 241\)](#)
- [インスタンスの機能 \(p. 243\)](#)
- [個の vCPU のサポート \(p. 244\)](#)
- [リリースノート \(p. 244\)](#)

## ハードウェア仕様

メモリ最適化インスタンスのハードウェア仕様の要約を以下に示します。

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
r4.large	2	15.25
r4.xlarge	4	30.5
r4.2xlarge	8	61
r4.4xlarge	16	122
r4.8xlarge	32	244
r4.16xlarge	64	488
r5.large	2	16

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
メモリ最適化インスタンス

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
r5.xlarge	4	32
r5.2xlarge	8	64
r5.4xlarge	16	128
r5.8xlarge	32	256
r5.12xlarge	48	384
r5.16xlarge	64	512
r5.24xlarge	96	768
r5.metal	96	768
r5a.large	2	16
r5a.xlarge	4	32
r5a.2xlarge	8	64
r5a.4xlarge	16	128
r5a.8xlarge	32	256
r5a.12xlarge	48	384
r5a.16xlarge	64	512
r5a.24xlarge	96	768
r5ad.large	2	16
r5ad.xlarge	4	32
r5ad.2xlarge	8	64
r5ad.4xlarge	16	128
r5ad.12xlarge	48	384
r5ad.24xlarge	96	768
r5d.large	2	16
r5d.xlarge	4	32
r5d.2xlarge	8	64
r5d.4xlarge	16	128
r5d.8xlarge	32	256
r5d.12xlarge	48	384
r5d.16xlarge	64	512
r5d.24xlarge	96	768
r5d.metal	96	768

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
メモリ最適化インスタンス

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
r5dn.large	2	16
r5dn.xlarge	4	32
r5dn.2xlarge	8	64
r5dn.4xlarge	16	128
r5dn.8xlarge	32	256
r5dn.12xlarge	48	384
r5dn.16xlarge	64	512
r5dn.24xlarge	96	768
r5n.large	2	16
r5n.xlarge	4	32
r5n.2xlarge	8	64
r5n.4xlarge	16	128
r5n.8xlarge	32	256
r5n.12xlarge	48	384
r5n.16xlarge	64	512
r5n.24xlarge	96	768
u-6tb1.metal	448 *	6,144
u-9tb1.metal	448 *	9,216
u-12tb1.metal	448 *	12,288
u-18tb1.metal	448 *	18,432
u-24tb1.metal	448 *	24,576
x1.16xlarge	64	976
x1.32xlarge	128	1,952
x1e.xlarge	4	122
x1e.2xlarge	8	244
x1e.4xlarge	16	488
x1e.8xlarge	32	976
x1e.16xlarge	64	1,952
x1e.32xlarge	128	3,904
z1d.large	2	16
z1d.xlarge	4	32

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
z1d.2xlarge	8	64
z1d.3xlarge	12	96
z1d.6xlarge	24	192
z1d.12xlarge	48	384
z1d.metal	48	384

\* 各論理プロセッサは 224 コアのハイパースレッドです

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

CPU オプションの指定についての詳細は、「[CPU オプションの最適化 \(p. 571\)](#)」を参照してください。

## メモリ性能

X1 インスタンスには、300 GiB/秒の持続可能なメモリ読み取り帯域幅と、140 GiB/秒の持続可能なメモリ書き込み帯域幅を提供する、Intel Scalable Memory Buffer が含まれます。

メモリ最適化インスタンスに対して有効にできる RAM の量の詳細については、「[ハードウェア仕様 \(p. 237\)](#)」を参照してください。

メモリ最適化インスタンスにはハイメモリがあり、その処理能力を活用するためには 64 ビットの HVM AMI が必要です。HVM AMI は、メモリ最適化インスタンスの準仮想化 (PV) AMI よりも優れたパフォーマンスを提供します。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。

## インスタンスのパフォーマンス

R4 インスタンスは、最大 64 個の vCPU に対応し、E5-2686v4 ベースの 2 個の AWS-customized Intel XEON プロセッサを使用します。このプロセッサはハイメモリ帯域幅と、より大きい L3 キャッシュにより、インメモリアプリケーションのパフォーマンスを向上させます。

X1e および X1 インスタンスは、最大 128 個の vCPU に対応し、4 個の Intel Xeon E7-8880 v3 プロセッサを使用します。このプロセッサは高メモリ帯域幅と、より大きい L3 キャッシュにより、メモリ内アプリケーションのパフォーマンスを向上させます。

ハイメモリインスタンス (u-6tb1.metal、u-9tb1.metal、および u-12tb1.metal) は、ミッショングクリティカルなエンタープライズワークロードに最適化された最新世代の Intel Xeon Platinum 8176M (Skylake) プロセッサを搭載した 8 ソケットプラットフォームで動作する初のインスタンスです。

18 TB および 24 TB のメモリを使用するハイメモリインスタンス (u-18tb1.metal および u-24tb1.metal) は、第 2 世代 Intel Xeon Scalable 8280L (Cascade Lake) プロセッサを搭載した 8 ソケットプラットフォームで動作する初のインスタンスです。

メモリ最適化インスタンスは、最新の Intel AES-NI 機能を通じてより高い暗号化のパフォーマンスを実現し、Intel Transactional Synchronization Extensions (TSX) のサポートによりインメモリトランザクションデータ処理のパフォーマンスを高めます。また、Advanced Vector Extensions 2 (Intel AVX2) プロセッサ命令のサポートにより、ほとんどの整数コマンドを 256 ビットに拡大します。

一部のメモリ最適化インスタンスでは、Linux のプロセッサの C ステートと P ステートを制御できます。C ステートは非アクティブ時のコアのスリープレベルを制御し、P ステートは希望するコアからのパフォーマンス (CPU 周波数で測定) を制御します。詳細については、「[EC2 インスタンスタイプのプロセッサのステート制御 \(p. 561\)](#)」を参照してください。

## ネットワークパフォーマンス

サポート対象のインスタンスタイプで、拡張されたネットワーキング機能を有効にすることができます。拡張ネットワーキングでは、パケット毎秒 (PPS) が非常に大きく、ネットワークのストレスが少なく、レイテンシーが低くなります。詳細については、「[Linux の拡張ネットワーキング \(p. 737\)](#)」を参照してください。

拡張されたネットワーキングのための Elastic Network Adapter (ENA) を使用するインスタンスタイプは、高いパケット/秒パフォーマンスと一貫して低いレイテンシーを同時に実現します。ほとんどのアプリケーションでは、高いレベルのネットワークパフォーマンスが一貫して必要なわけではありませんが、データの送受信時にアクセスする帯域幅を増やすことでメリットを得られます。ENA を使用し、「最大 10 Gbps」または「最大 25 Gbps」のネットワークパフォーマンスをサポートするインスタンスサイズでは、ネットワーク I/O クレジットメカニズムを使用して、平均帯域幅使用率に基づいてインスタンスにネットワーク帯域幅を割り当てます。これらのインスタンスでは、ネットワーク帯域幅がベースライン制限を下回るとクレジットを獲得し、これらのクレジットをネットワークデータ転送を実行するときに使用できます。

拡張ネットワーキングをサポートするメモリ最適化インスタンスのネットワークパフォーマンスの要約を以下に示します。

インスタンスタイプ	ネットワークパフォーマンス	拡張ネットワーキング
r4.4xlarge 以下   r5.4xlarge 以下   r5a.8xlarge 以下   r5ad.4xlarge 以下   r5d.4xlarge 以下   x1e.8large 以下   z1d.3xlarge 以下	最大 10 Gbps	<a href="#">ENA (p. 738)</a>
r4.8xlarge   r5.8xlarge   r5.12xlarge   r5a.12xlarge   r5ad.12xlarge   r5d.8xlarge   r5d.12xlarge   x1.16xlarge   x1e.16xlarge   z1d.6xlarge	10 Gbps	<a href="#">ENA (p. 738)</a>
r5a.16xlarge   r5ad.16xlarge	12 Gbps	<a href="#">ENA (p. 738)</a>
r5.16xlarge   r5a.24xlarge   r5ad.24xlarge   r5d.16xlarge	20 Gbps	<a href="#">ENA (p. 738)</a>
r5dn.4xlarge 以下   r5n.4xlarge 以下	最大 25 Gbps	<a href="#">ENA (p. 738)</a>
r4.16xlarge   r5.24xlarge   r5.metal   r5d.24xlarge   r5d.metal   r5dn.8xlarge   r5n.8xlarge   u-6tb1.metal   u-9tb1.metal   u-12tb1.metal   x1.32xlarge   x1e.32xlarge   z1d.12xlarge   z1d.metal	25 Gbps	<a href="#">ENA (p. 738)</a>
r5dn.12xlarge   r5n.12xlarge	50 Gbps	<a href="#">ENA (p. 738)</a>
r5dn.16xlarge   r5n.16xlarge	75 Gbps	<a href="#">ENA (p. 738)</a>
r5dn.24xlarge   r5n.24xlarge   u-18tb1.metal   u-24tb1.metal	100 Gbps	<a href="#">ENA (p. 738)</a>

## SSD I/O パフォーマンス

カーネルバージョン 4.4 以降の Linux AMI を使用し、インスタンスで利用可能なすべての SSD ベースのインスタンストアボリュームを使用する場合は、以下の表に示されている IOPS (4,096 バイトブロックサ

イズ<sup>\*</sup> のパフォーマンスを得ることができます(キューの深さの飽和度において)。それ以外の場合、IOPS パフォーマンスは低下します。

インスタンスサイズ	100% のランダム読み取り時 IOPS	書き込み IOPS
r5ad.large *	30,000	15,000
r5ad.xlarge *	59,000	29,000
r5ad.2xlarge *	117,000	57,000
r5ad.4xlarge *	234,000	114,000
r5ad.12xlarge	700,000	340,000
r5ad.24xlarge	1,400,000	680,000
r5d.large *	30,000	15,000
r5d.xlarge *	59,000	29,000
r5d.2xlarge *	117,000	57,000
r5d.4xlarge *	234,000	114,000
r5d.8xlarge	466,666	233,333
r5d.12xlarge	700,000	340,000
r5d.16xlarge	933,333	466,666
r5d.24xlarge	1,400,000	680,000
r5d.metal	1,400,000	680,000
r5dn.large *	30,000	15,000
r5dn.xlarge *	59,000	29,000
r5dn.2xlarge *	117,000	57,000
r5dn.4xlarge *	234,000	114,000
r5dn.8xlarge	466,666	233,333
r5dn.12xlarge	700,000	340,000
r5dn.16xlarge	933,333	466,666
r5dn.24xlarge	1,400,000	680,000
z1d.large *	30,000	15,000
z1d.xlarge *	59,000	29,000
z1d.2xlarge *	117,000	57,000
z1d.3xlarge *	175,000	75,000
z1d.6xlarge	350,000	170,000
z1d.12xlarge	700,000	340,000

インスタンスサイズ	100% のランダム読み取り時 IOPS	書き込み IOPS
z1d.metal	700,000	340,000

\* これらのインスタンスの場合、最大で指定されたパフォーマンスを得ることができます。

インスタンスに SSD ベースのインスタンスストアボリュームを使用するほど、アーカイブできる書き込み IOPS の数は減少します。これは、SSD コントローラーが実行する必要がある追加の作業が原因です。SSD コントローラーは、利用可能な領域を見つけ、既存のデータを再書き込みし、未使用の領域を消去して、再書き込みができるようにします。このガベージコレクションというプロセスにより、SSD への内部的な書き込み増幅が発生し、ユーザーの書き込み操作に対する SSD 書き込み操作の割合として表示されます。書き込み操作が 4,096 バイトの倍数でないか、4,096 バイトの境界に整合していない場合、パフォーマンスの低下はさらに大きくなります。少量のバイト数または整合していないバイト数で書き込む場合、SSD コントローラーは周辺のデータを読み取り、その結果を新しい場所に保存する必要があります。このパターンにより、書き込み増幅が大幅に増え、レイテンシーが増加し、I/O パフォーマンスが大きく低下します。

SSD コントローラーは、複数の方法を利用すると、書き込み増幅の影響を減らすことができます。このような方法の 1 つには、SSD インスタンスストレージに領域を予約し、コントローラーが書き込み操作に利用できる領域をより効率的に管理できるようにすることです。これをオーバープロビジョニングと呼びます。インスタンスに提供された SSD ベースのインスタンスストアボリュームには、オーバープロビジョニングに対して予約された領域がありません。書き込み増幅を減らすには、ボリュームの 10% を未使用的状態のままにし、SSD コントローラーがこれをオーバープロビジョニングに使用できるようにすることをお勧めします。これにより、使用できるストレージは減りますが、ディスクが総容量に近づいた場合でもパフォーマンスを向上させることができます。

TRIM をサポートするインスタンスストアボリュームの場合、TRIM コマンドを使用して、書き込んだデータが不要になったときはいつでも SSD コントローラーに通知することができます。これにより、より多くの空き領域がコントローラーに与えられ、その結果書き込み増幅が減り、パフォーマンスが向上します。詳細については、「[インスタンスストアボリュームの TRIM のサポート \(p. 1087\)](#)」を参照してください。

## インスタンスの機能

メモリ最適化インスタンスの機能の概要を以下に示します。

	EBS のみ	NVMe EBS	インスタンスストア	配置グループ
R4	はい	いいえ	いいえ	あり
R5	はい	はい	いいえ	あり
R5a	はい	はい	いいえ	あり
R5ad	いいえ	はい	NVME *	あり
R5d	いいえ	はい	NVME *	はい
R5dn	いいえ	はい	NVME *	はい
R5n	はい	はい	いいえ	はい
u-6tb1.metal	あり	はい	いいえ	いいえ
u-9tb1.metal	あり	はい	いいえ	いいえ
u-12tb1.metal	あり	はい	いいえ	いいえ

	EBS のみ	NVMe EBS	インスタンスストア	配置グループ
u-18tb1.m <i>xx</i> 1	いいえ	はい	いいえ	いいえ
u-24tb1.m <i>xx</i> 1	いいえ	はい	いいえ	なし
X1	いいえ	いいえ	SSD	あり
X1e	いいえ	いいえ	SSD *	はい
z1d	いいえ	はい	NVME *	はい

\* ルートデバイスピリュームは、Amazon EBS ボリュームにする必要があります。

詳細については、以下を参照してください。

- [Linux インスタンスの Amazon EBS および NVMe \(p. 1027\)](#)
- [Amazon EC2 インスタンスストア \(p. 1076\)](#)
- [プレイスメントグループ \(p. 791\)](#)

## 個の vCPU のサポート

メモリ最適化インスタンスは多数の vCPU を提供するため、vCPU の制限が低いオペレーティングシステムで起動の問題が発生することがあります。メモリ最適化インスタンスを起動する場合は、最新の AMI を使用することをお勧めします。

以下の AMI では、メモリ最適化インスタンスの起動がサポートされています。

- [Amazon Linux 2 \(HVM\)](#)
- [Amazon Linux AMI 2016.03 \(HVM\) 以降](#)
- [Ubuntu Server 14.04 LTS \(HVM\)](#)
- [Red Hat Enterprise Linux 7.1 \(HVM\)](#)
- [SUSE Linux Enterprise Server 12 SP1 \(HVM\)](#)
- [Windows Server 2019](#)
- [Windows Server 2016](#)
- [Windows Server 2012 R2](#)
- [Windows Server 2012](#)
- [Windows Server 2008 R2 64 ビット](#)
- [Windows Server 2008 SP2 64 ビット](#)

## リリースノート

- R5 および R5d インスタンスは、第 1 世代 (Skylake-SP) または第 2 世代 (Cascade Lake) の 3.1 GHz Intel Xeon Platinum 8000 シリーズプロセッサーを搭載しています。
- R5a および R5ad インスタンスは、2.5 GHz AMD EPYC 7000 シリーズプロセッサを搭載しています。
- ハイメモリ、R5、R5a、R5ad、R5d、R5dn、R5n、および z1d インスタンスの要件は以下のとおりです。
  - [NVMe ドライバー \(p. 1027\)](#)がインストールされている必要があります。
  - [Elastic Network Adapter \(ENA\) ドライバー \(p. 738\)](#)がインストールされている必要があります。

以下の Linux AMI はこれらの要件を満たしています。

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (linux-aws カーネル) 以降
- Red Hat Enterprise Linux 7.4 以降
- SUSE Linux Enterprise Server 12 SP2 以降
- CentOS 7.4.1708 以降
- FreeBSD 11.1 以降
- Debian GNU/Linux 9 以降
- R5、R5a、R5ad、R5d、R5dn、R5n、および z1d インスタンスは、ネットワークインターフェイス、EBS ボリューム、および NVMe インスタンスストアボリュームを含め、最大 28 のアタッチをサポートしています。インスタンスごとに 1 つ以上のネットワークインターフェイスのアタッチメントがあります。たとえば、EBS のみのインスタンスに追加のネットワークインターフェイスのアタッチがない場合は、そのインスタンスに 27 個の EBS ボリュームをアタッチできます。
- u-6tb1.metal、u-9tb1.metal、および u-12tb1.metal の各インスタンスは、最大 13 の EBS ボリュームをサポートしています。u-18tb1.metal および u-24tb1.metal の各インスタンスは最大 19 の EBS ボリュームをサポートしています。
- ベアメタルインスタンスを起動すると、基盤となるサーバーが起動します。これには、すべてのハードウェアやファームウェアコンポーネントの確認が含まれます。つまり、インスタンスが実行状態になってからネットワーク経由で使用できるようになるまでに 20 分かかることがあります。
- ベアメタルインスタンスから EBS ボリュームまたはセカンダリネットワークインターフェイスをアタッチまたはデタッチするには、PCIe のネイティブホットプラグサポートが必要です。Amazon Linux 2 および最新バージョンの Amazon Linux AMI は PCIe ネイティブホットプラグをサポートしていますが、以前のバージョンではサポートされていません。次の Linux カーネル設定オプションを有効にする必要があります。

```
CONFIG_HOTPLUG_PCI_PCIE=y  
CONFIG_PCIEASPM=y
```

- ベアメタルインスタンスでは、I/O ポートベースのシリアルデバイスではなく、PCI ベースのシリアルデバイスを使用しています。アップストリームの Linux カーネルと最新の Amazon Linux AMI は、このデバイスをサポートしています。また、ベアメタルインスタンスでは、システムが PCI ベースのシリアルデバイスを自動的に使用できるようにする ACPI SPCR テーブルも使用できます。最新の Windows AMI では、自動的に PCI ベースのシリアルデバイスが使用されます。
- Windows Server 2008 SP2 64 ビット AMI を使用して、X1 インスタンス (x1.16xlarge インスタンスは除く) を起動することはできません。
- Windows Server 2008 SP2 64 ビット AMI を使用して、X1e インスタンスを起動することはできません。
- Windows Server 2008 R2 64-bit AMI の旧バージョンでは、r4.large および r4.4xlarge インスタンスを起動できません。この問題が発生した場合は、この AMI の最新バージョンに更新してください。
- リージョンで起動できるインスタンスの合計数には制限があります。また、一部のインスタンスタイプにはその他の制限もあります。詳細については、「[Amazon EC2 で実行できるインスタンスの数はいくつですか？](#)」を参照してください。制限の引き上げをリクエストするには、「[Amazon EC2 インスタンス申請フォーム](#)」を使用します。

## ストレージ最適化インスタンス

ストレージ最適化インスタンスは、ローカルストレージの大規模データセットに対する高いシーケンシャル読み取りおよび書き込みアクセスを必要とするワークロード用に設計されています。ストレージ最適化インスタンスは、数万回の低レイテンシーとランダム I/O オペレーション/秒 (IOPS) をアプリケーションに提供するように最適化されています。

## D2 インスタンス

D2 インスタンスは、次の用途に適しています。

- 超並列処理 (MPP) データウェアハウス
- MapReduce および Hadoop 分散コンピューティング
- ログまたはデータ処理アプリケーション

## H1 インスタンス

H1 インスタンスは、次の用途に適しています。

- MapReduce および分散ファイルシステムなどのデータ集約型ワークロード
- 直接アタッチされたインスタンストレージにある大量データへのシーケンシャルアクセスを必要とするアプリケーション
- 大量のデータへの高スループットアクセスを必要とするアプリケーション

## I3 および I3en インスタンス

これらのインスタンスは、以下の用途に適しています。

- 高頻度オンライントランザクション処理 (OLTP) システム
- リレーショナルデータベース
- NoSQL データベース
- メモリ内データベース (Redis など) のキャッシュ
- データウェアハウスアプリケーション
- 分散されたファイルシステム

ペアメタルインスタンスを使用すると、アプリケーションから、プロセッサとメモリなどのホストサーバーの物理リソースに直接アクセスすることができます。これらのインスタンスは、次の用途に適しています。

- 仮想環境で利用できない、または完全にサポートされていない低レベルのハードウェア機能 (例: Intel VT) へのアクセスを必要とするワークロード
- ライセンスやサポートを目的として非仮想化環境で実行する必要があるアプリケーション

詳細については、「[Amazon EC2 I3 インスタンス](#)」を参照してください。

## コンテンツ

- [ハードウェア仕様 \(p. 247\)](#)
- [インスタンスのパフォーマンス \(p. 248\)](#)
- [ネットワークパフォーマンス \(p. 248\)](#)
- [SSD I/O パフォーマンス \(p. 249\)](#)
- [インスタンスの機能 \(p. 250\)](#)
- [個の vCPU のサポート \(p. 250\)](#)
- [リリースノート \(p. 252\)](#)

## ハードウェア仕様

D2 インスタンスのプライマリデータストレージは、HDD インスタンスストアボリュームです。I3 インスタンスのプライマリデータストレージは、Non-Volatile Memory Express (NVMe) SSD インスタンスストアボリュームです。

インスタンスストアボリュームは、インスタンスの存続中のみ使用できます。インスタンスを停止または終了すると、アプリケーションとそのインスタンスストアボリュームのデータは消去されます。インスタンスストアボリュームの重要なデータは、定期的にバックアップまたはレプリケートすることをお勧めします。詳細については、「[Amazon EC2 インスタンスストア \(p. 1076\)](#)」および「[SSD インスタンスストアボリューム \(p. 1086\)](#)」を参照してください。

ストレージ最適化インスタンスのハードウェア仕様の要約を以下に示します。

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)
d2.xlarge	4	30.5
d2.2xlarge	8	61
d2.4xlarge	16	122
d2.8xlarge	36	244
h1.2xlarge	8	32
h1.4xlarge	16	64
h1.8xlarge	32	128
h1.16xlarge	64	256
i3.large	2	15.25
i3.xlarge	4	30.5
i3.2xlarge	8	61
i3.4xlarge	16	122
i3.8xlarge	32	244
i3.16xlarge	64	488
i3.metal	72	512
i3en.large	2	16
i3en.xlarge	4	32
i3en.2xlarge	8	64
i3en.3xlarge	12	96
i3en.6xlarge	24	192
i3en.12xlarge	48	384
i3en.24xlarge	96	768
i3en.metal	96	768

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

CPU オプションの指定についての詳細は、「[CPU オプションの最適化 \(p. 571\)](#)」を参照してください。

## インスタンスのパフォーマンス

Linux のインスタンスから最善のディスクスループットパフォーマンスを得るには、最新バージョンの Amazon Linux 2 または Amazon Linux AMI を使用することをお勧めします。

NVMe インスタンストアボリュームを持つインスタンスの場合は、カーネルバージョン 4.4 以降の Linux AMI を使用する必要があります。それ以外の場合、インスタンスは利用可能な最大の IOPS パフォーマンスを達成できません。

D2 インスタンスは、永続許可 (ディスクスループットと拡張性を大幅に向上させる Xen ブロックリングポートコールの拡張機能) をサポートする Linux カーネルを使用するときに、最大のディスクパフォーマンスを提供します。永続許可の詳細については、Xen プロジェクトのブログの[こちらの記事](#)を参照してください。

EBS 最適化インスタンスは、インスタンスからの Amazon EBS I/O とその他のネットワークトラフィックとの競合を排除することによって、EBS ボリュームの安定した高パフォーマンスを実現できます。一部のストレージ最適化インスタンスは、追加料金なしでデフォルトで EBS 最適化されます。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

一部のストレージ最適化インスタンスでは、Linux でプロセッサの C ステートと P ステートを制御できます。C ステートは非アクティブ時のコアのスリープレベルを制御し、P ステートは希望するコアからのパフォーマンス (CPU 周波数) を制御します。詳細については、「[EC2 インスタンスタイプのプロセッサのステート制御 \(p. 561\)](#)」を参照してください。

## ネットワークパフォーマンス

サポート対象のインスタンスタイプで、拡張されたネットワーキング機能を有効にすることができます。拡張ネットワーキングでは、パケット毎秒 (PPS) が非常に大きく、ネットワークのストレスが少なく、レイテンシーが低くなります。詳細については、「[Linux の拡張ネットワーキング \(p. 737\)](#)」を参照してください。

拡張されたネットワーキングのための Elastic Network Adapter (ENA) を使用するインスタンスタイプは、高いパケット/秒パフォーマンスと一貫して低いレイテンシーを同時に実現します。ほとんどのアプリケーションでは、高いレベルのネットワークパフォーマンスが一貫して必要なわけではありませんが、データの送受信時にアクセスする帯域幅を増やすことでメリットを得られます。ENA を使用し、「最大 10 Gbps」または「最大 25 Gbps」のネットワークパフォーマンスをサポートするインスタンスサイズでは、ネットワーク I/O クレジットメカニズムを使用して、平均帯域幅使用率に基づいてインスタンスにネットワーク帯域幅を割り当てます。これらのインスタンスでは、ネットワーク帯域幅がベースライン制限を下回るとクレジットを獲得し、これらのクレジットをネットワークデータ転送を実行するときに使用できます。

拡張ネットワーキングをサポートするストレージ最適化インスタンスのネットワークパフォーマンスの要約を以下に示します。

インスタンスタイプ	ネットワークパフォーマンス	拡張ネットワーキング
i3.4xlarge 以下	最大 10 Gbps、ネットワーク I/O クレジットメカニズムを使用	<a href="#">ENA (p. 738)</a>
i3.8xlarge   h1.8xlarge	10 Gbps	<a href="#">ENA (p. 738)</a>
i3en.3xlarge 以下	最大 25 Gbps、ネットワーク I/O クレジットメカニズムを使用	<a href="#">ENA (p. 738)</a>

インスタンスタイプ	ネットワークパフォーマンス	拡張ネットワーキング
i3.16xlarge   i3.metal   i3en.6xlarge   h1.16xlarge	25 Gbps	<a href="#">ENA (p. 738)</a>
i3en.12xlarge	50 Gbps	<a href="#">ENA (p. 738)</a>
i3en.24xlarge	100 Gbps	<a href="#">ENA (p. 738)</a>
d2.xlarge	中	<a href="#">Intel 82599 VF (p. 751)</a>
d2.2xlarge   d2.4xlarge	高	<a href="#">Intel 82599 VF (p. 751)</a>
d2.8xlarge	10 Gbps	<a href="#">Intel 82599 VF (p. 751)</a>

## SSD I/O パフォーマンス

カーネルバージョン 4.4 以降の Linux AMI を使用し、インスタンスで利用可能なすべての SSD ベースのインスタンスストアボリュームを使用する場合は、以下の表に示されている IOPS (4,096 バイトブロックサイズ) のパフォーマンスを得ることができます (キューの深さの飽和度において)。それ以外の場合、IOPS パフォーマンスは低下します。

インスタンスサイズ	100% のランダム読み取り時 IOPS	書き込み IOPS
i3.large *	100,125	35,000
i3.xlarge *	206,250	70,000
i3.2xlarge	412,500	180,000
i3.4xlarge	825,000	360,000
i3.8xlarge	1,650,000	720,000
i3.16xlarge	3,300,000	140 万
i3.metal	3,300,000	140 万
i3en.large *	42,500	32,500
i3en.xlarge *	85,000	65,000
i3en.2xlarge *	170,000	130,000
i3en.3xlarge	250,000	200,000 件の
i3en.6xlarge	500,000	400,000
i3en.12xlarge	100 万件	800,000
i3en.24xlarge	200 万件	160 万件
i3en.metal	200 万件	160 万件

\* これらのインスタンスの場合、最大で指定されたパフォーマンスを得ることができます。

SSD ベースのインスタンスストアボリュームをいっぱいにすると、得られる I/O パフォーマンスは低下します。これは、SSD コントローラーが実行する必要がある追加の作業が原因です。SSD コントローラーは、利用可能な領域を見つけ、既存のデータを再書き込みし、未使用的領域を消去して、再書き込みができるようになります。このガベージコレクションというプロセスにより、SSD への内部的な書き込み増幅が発生し、ユーザーの書き込み操作に対する SSD 書き込み操作の割合として表示されます。書き込み操作が 4,096 バイトの倍数でないか、4,096 バイトの境界に整合していない場合、パフォーマンスの低下はさらに大きくなります。少量のバイト数または整合していないバイト数で書き込む場合、SSD コントローラーは周辺のデータを読み取り、その結果を新しい場所に保存する必要があります。このパターンにより、書き込み増幅が大幅に増え、レイテンシーが増加し、I/O パフォーマンスが大きく低下します。

SSD コントローラーは、複数の方法を利用すると、書き込み増幅の影響を減らすことができます。このような方法の 1 つには、SSD インスタンスストレージに領域を予約し、コントローラーが書き込み操作に利用できる領域をより効率的に管理できるようにすることです。これをオーバープロビジョニングと呼びます。インスタンスに提供された SSD ベースのインスタンスストアボリュームには、オーバープロビジョニングに対して予約された領域がありません。書き込み増幅を減らすには、ボリュームの 10% を未使用的状態のままにし、SSD コントローラーがこれをオーバープロビジョニングに使用できるようにすることをお勧めします。これにより、使用できるストレージは減りますが、ディスクが総容量に近づいた場合でもパフォーマンスを向上させることができます。

TRIM をサポートするインスタンスストアボリュームの場合、TRIM コマンドを使用して、書き込んだデータが不要になったときはいつでも SSD コントローラーに通知することができます。これにより、より多くの空き領域がコントローラーに与えられ、その結果書き込み増幅が減り、パフォーマンスが向上します。詳細については、「[インスタンスストアボリュームの TRIM のサポート \(p. 1087\)](#)」を参照してください。

## インスタンスの機能

ストレージ最適化インスタンスの機能の概要を以下に示します。

	EBS のみ	インスタンスストア	配置グループ
D2	いいえ	HDD	はい
H1	いいえ	HDD *	はい
I3	いいえ	NVMe *	はい
I3en	いいえ	NVMe *	はい

\* ルートデバイスボリュームは、Amazon EBS ボリュームにする必要があります。

詳細については、以下を参照してください。

- [Linux インスタンスの Amazon EBS および NVMe \(p. 1027\)](#)
- [Amazon EC2 インスタンスストア \(p. 1076\)](#)
- [プレイスメントグループ \(p. 791\)](#)

## 個の vCPU のサポート

d2.8xlarge インスタンスでは 36 個の vCPU が提供されますが、これにより vCPU を 32 個に制限している一部の Linux オペレーティングシステムで問題が生じる可能性があります。d2.8xlarge インスタンスを起動する場合は、最新の AMI を使用することをお勧めします。

次の Linux AMI は、36 個の vCPU を使用した d2.8xlarge インスタンスの起動をサポートしています。

- [Amazon Linux 2 \(HVM\)](#)
- [Amazon Linux AMI 2018.03 \(HVM\)](#)

- Ubuntu Server 14.04 LTS (HVM) 以降
- Red Hat Enterprise Linux 7.1 (HVM)
- SUSE Linux Enterprise Server 12 (HVM)

アプリケーションに別の AMI を使用する必要がある場合に、d2.8xlarge インスタンスの起動が正常に完了しないとき (stopped 状態遷移に伴って起動中にインスタンスのステータスが Client.InstanceInitiatedShutdown に変更されるときなど) は、以下の手順に従って 32 個を超える vCPU をサポートするようにインスタンスを変更し、d2.8xlarge インスタンスタイプを使用できるようになります。

### 32 個の以上の vCPU をサポートするようにインスタンスを更新する

1. AMI を使用して D2 インスタンスを起動し、d2.8xlarge 以外の D2 インスタンスタイプを選択します。
2. 使用するオペレーティングシステム固有の手順に従って、カーネルを最新バージョンに更新します。たとえば、RHEL 6 の場合は、次のコマンドを使用します。

```
sudo yum update -y kernel
```

3. インスタンスを停止します。
4. (オプション) 将来的に必要な追加の d2.8xlarge インスタンスを起動するために使用できるインスタンスから AMI を作成します。
5. 停止したインスタンスのインスタンスタイプを d2.8xlarge に変更します ([Actions]、[Instance Settings]、[Change Instance Type] の順に選択し、指示に従います)。
6. インスタンスを起動します。インスタンスが正常に起動すれば、完了です。これでもインスタンスが正しく実施されない場合、以下のステップに進みます。
7. (オプショナル) インスタンスがまだ正しく実施されない場合、インスタンスのカーネルが 32 個の vCPU をサポートしていない可能性があります。ただし、vCPU を制限すると、インスタンスを起動できる場合があります。
  - a. 停止したインスタンスのインスタンスタイプを d2.8xlarge 以外の D2 インスタンスタイプのいずれかに変更します ([Actions]、[Instance Settings]、[Change Instance Type] の順に選択し、指示に従います)。
  - b. 使用するオペレーティングシステム固有の手順に従って、カーネル起動パラメータに maxcpus=32 オプションを追加します。たとえば、RHEL 6 の場合は、/boot/grub/menu.lst ファイルを編集し、最新のアクティブな kernel エントリに次のオプションを追加します。

```
default=0
timeout=1
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-504.3.3.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-504.3.3.el6.x86_64 maxcpus=32 console=ttyS0 ro
      root=UUID=9996863e-b964-47d3-a33b-3920974fdbd9 rd_NO_LUKS KEYBOARDTYPE=pc
      KEYTABLE=us LANG=en_US.UTF-8 xen_blkfront.sda_is_xvda=1 console=ttyS0,115200n8
      console=tty0 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_NO_LVM
      rd_NO_DM
initrd /boot/initramfs-2.6.32-504.3.3.el6.x86_64.img
```

- c. インスタンスを停止します。
- d. (オプション) 将来的に必要な追加の d2.8xlarge インスタンスを起動するために使用できるインスタンスから AMI を作成します。
- e. 停止したインスタンスのインスタンスタイプを d2.8xlarge に変更します ([Actions]、[Instance Settings]、[Change Instance Type] の順に選択し、指示に従います)。

- f. インスタンスを起動します。

## リリースノート

- ストレージ最適化インスタンスは、HVM AMI を使用して起動する必要があります。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。
- i3en および i3.meta1 インスタンスの要件を以下に示します。
  - [NVMe ドライバー \(p. 1027\)](#)がインストールされている必要があります。
  - [Elastic Network Adapter \(ENA\) ドライバー \(p. 738\)](#)がインストールされている必要があります。

以下の Linux AMI はこれらの要件を満たしています。

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (linux-aws カーネル) 以降
- Red Hat Enterprise Linux 7.4 以降
- SUSE Linux Enterprise Server 12 SP2 以降
- CentOS 7.4.1708 以降
- FreeBSD 11.1 以降
- Debian GNU/Linux 9 以降
- i3.meta1 インスタンスを起動すると、基盤となるサーバーが起動します。これには、すべてのハードウェアやファームウェアコンポーネントの確認が含まれます。つまり、インスタンスが実行状態になってからネットワーク経由で使用できるようになるまでに 20 分かかることがあります。
- ベアメタルインスタンスから EBS ボリュームまたはセカンダリネットワークインターフェイスをアタッチまたはデタッチするには、PCIe のネイティブホットプラグサポートが必要です。Amazon Linux 2 および最新バージョンの Amazon Linux AMI は PCIe ネイティブホットプラグをサポートしていますが、以前のバージョンではサポートされていません。次の Linux カーネル設定オプションを有効にする必要があります。

```
CONFIG_HOTPLUG_PCI_PCIE=y  
CONFIG_PCIEASPM=y
```

- ベアメタルインスタンスでは、I/O ポートベースのシリアルデバイスではなく、PCI ベースのシリアルデバイスを使用しています。アップストリームの Linux カーネルと最新の Amazon Linux AMI は、このデバイスをサポートしています。また、ベアメタルインスタンスでは、システムが PCI ベースのシリアルデバイスを自動的に使用できるようにする ACPI SPCR テーブルも使用できます。最新の Windows AMI では、自動的に PCI ベースのシリアルデバイスが使用されます。
- FreeBSD AMI では、ベアメタルインスタンスは、起動に約 1 時間かかり、口一カルの NVMe ストレージへの I/O は完了しません。回避策として、次の行を /boot/loader.conf に追加し、再起動します。

```
hw.nvme.per_cpu_io_queues="0"
```

- d2.8xlarge インスタンスタイプには 36 個の vCPU がありますが、これにより vCPU を 32 個に制限している一部の Linux オペレーティングシステムで問題が生じる可能性があります。詳細については、「[個の vCPU のサポート \(p. 250\)](#)」を参照してください。
- リージョンで起動できるインスタンスの合計数には制限があります。また、一部のインスタンスタイプにはその他の制限もあります。詳細については、「[Amazon EC2 で実行できるインスタンスの数はいくつですか?](#)」を参照してください。制限の引き上げをリクエストするには、「[Amazon EC2 インスタンス申請フォーム](#)」を使用します。

## Linux 高速コンピューティングインスタンス

高度な処理機能が必要な場合は、高速コンピューティングインスタンスを使用すると、Graphics Processing Units (GPU) や Field Programmable Gate Arrays (FPGA) などのハードウェアベースのコンピューティングアクセラレーターにアクセスできます。高速コンピューティングインスタンスでは、大量の演算を行うワークロードでさらに多くの並列処理が可能となり、より高いスループットが得られます。

GPU ベースのインスタンスでは、数千のコンピューティングコアを持つ NVIDIA GPU にアクセスできます。GPU ベースの高速コンピューティングインスタンスを使用すると、CUDA または Open Computing Language (OpenCL) パラレルコンピューティングフレームワークを活用することにより、サイエンス、エンジニアリング、およびレンダリングアプリケーションを高速化できます。また、ゲームストリーミング、3D アプリケーションストリーミング、およびその他のグラフィックスワークロードを含む、グラフィックアプリケーションにも使用できます。

FPGA ベースのインスタンスでは、数百万の並列システム論理セルを持つ大きな FPGA にアクセスできます。FPGA ベースの高速コンピューティングインスタンスを使用すると、カスタムハードウェアアクセラレーションを活用することにより、ゲノム解析、財務分析、リアルタイム動画処理、ビッグデータ解析、およびセキュリティワークロードなどのワークロードを高速化できます。Verilog や VHDL などのハードウェア記述言語を使用するか、または OpenCL パラレルコンピューティングフレームワークなどの高レベル言語を使用して、これらの加速度を開発できます。ハードウェアアクセラレーションコードを自身で作成することも、[AWS Marketplace](#) からハードウェアアクセラレーションを購入することもできます。

### Important

FPGA ベースのインスタンスは、Microsoft Windows をサポートしていません。

高速コンピューティングインスタンスをクラスター・プレイスメント・グループにクラスター化できます。クラスター・プレイスメント・グループは、1 つのアベイラビリティ・ゾーン内で、インスタンス間の低レイテンシーで高帯域幅の接続を提供します。詳細については、「[プレイスメント・グループ \(p. 791\)](#)」を参照してください。

### コンテンツ

- [高速コンピューティングインスタンス・ファミリー \(p. 253\)](#)
- [ハードウェア仕様 \(p. 255\)](#)
- [インスタンスのパフォーマンス \(p. 256\)](#)
- [ネットワークパフォーマンス \(p. 256\)](#)
- [インスタンスの機能 \(p. 257\)](#)
- [リリースノート \(p. 258\)](#)
- [Linux インスタンスへの NVIDIA ドライバーのインストール \(p. 258\)](#)
- [NVIDIA GRID 仮想アプリケーションの有効化 \(p. 264\)](#)
- [GPU 設定の最適化 \(p. 265\)](#)
- [FPGA 開発を開始する \(p. 266\)](#)
- [AWS Inferentia 開発の開始方法 \(p. 266\)](#)

Windows 高速コンピューティングインスタンスについては、「[Windows 高速コンピューティングインスタンス](#)」(Windows インスタンスの Amazon EC2 ユーザーガイド) を参照してください。

## 高速コンピューティングインスタンス・ファミリー

高速コンピューティングインスタンス・ファミリーは、ハードウェアアクセラレーターやコプロセッサーを使用して、浮動小数点数計算、グラフィック処理、データパターンマッチングのような機能を CPU で実

行されるソフトウェア以上に効率的に実行します。次の高速コンピューティングインスタンスファミリーを Amazon EC2 で起動できます。

#### F1 インスタンス

F1 インスタンスは Xilinx UltraScale+ VU9P FPGA を使用し、汎用 CPU に適さないデータフロー や高度な並列処理のような計算集約型のアルゴリズムを高速化するように設計されています。F1 インスタンスの各 FPGA には、約 250 万の論理要素と約 6,800 のデジタル信号処理 (DSP) エンジン、ローカルの 64 GiB DDR ECC 保護メモリが含まれ、専用の PCIe Gen3 x16 接続によってインスタンスに接続されています。F1 インスタンスは、ローカルの NVMe SSD ボリュームを提供します。

開発者は FPGA Developer AMI および AWS Hardware Developer Kit を使用して、F1 インスタンスで使用するカスタムハードウェアアクセラレーションを作成できます。FPGA Developer AMI には、クラウド上の FPGA 完全サイクル開発用の開発ツールが含まれます。これらのツールを使用して、開発者は F1 インスタンスの FPGA にロードできる Amazon FPGA Image (AFI) を作成し、共有できます。

詳細については、「[Amazon EC2 F1 インスタンス](#)」を参照してください。

#### P3 インスタンス

P3 インスタンスは NVIDIA Tesla V100 GPU を使用し、CUDA または OpenCL プログラミングモデルを使用するか、Machine Learning フレームワークを使用する汎用 GPU コンピューティング用に設計されています。P3 インスタンスは高帯域幅ネットワーキング、強力な半精度、単精度、および倍精度浮動小数点機能、および GPU ごとに最大 32 GiB メモリを提供し、深層学習、数値流体力学、金融工学、耐震解析、分子モデリング、ゲノム解析、レンダリング、その他サーバー側 GPU コンピューティングワークロードに最適です。Tesla V100 GPU はグラフィックモードをサポートしません。詳細については、「[Amazon EC2 P3 インスタンス](#)」を参照してください。

P3 インスタンスは NVIDIA NVLink のピアツーピア転送をサポートします。

システムのトポロジ情報を表示するには、以下のコマンドを実行します。

```
nvidia-smi topo -m
```

詳細については、[NVIDIA NVLink](#) を参照してください。

#### P2 インスタンス

P2 インスタンスは NVIDIA Tesla K80 GPU を使用し、CUDA または OpenCL プログラミングモデルを使用する汎用 GPU コンピューティング用に設計されています。P2 インスタンスは高帯域幅ネットワーキング、強力な単精度および倍精度浮動小数点機能、および GPU ごとに 12 GiB メモリを提供し、ディープラーニング、グラフデータベース、高パフォーマンスデータベース、数値流体力学、金融工学、耐震解析、分子モデリング、ゲノム解析、レンダリング、その他サーバー側 GPU コンピューティングワークロードに最適です。

P2 インスタンスは NVIDIA GPUDirect のピアツーピア転送をサポートします。

システムのトポロジ情報を表示するには、以下のコマンドを実行します。

```
nvidia-smi topo -m
```

詳細については、[NVIDIA GPUDirect](#) を参照してください。

#### G4 インスタンス

G4 インスタンスは NVIDIA Tesla GPU を使用して、汎用 GPU コンピューティング用のコスト効率とパフォーマンスに優れたプラットフォームを CUDA を通じて提供するか、グラフィックアプリケーション

を備えた機械学習フレームワークを DirectX または OpenGL を通じて提供します。G4 インスタンスは、高帯域幅ネットワーキング、強力な半精度浮動小数点機能、単精度浮動小数点機能、INT8 精度、および INT4 精度を提供します。各 GPU は 16 GiB の GDDR6 メモリを備えているため、G4 インスタンスは機械学習推論、動画トランスコード、グラフィックアプリケーション（リモートグラフィックワークステーションやクラウド内のゲームストリーミングなど）に最適です。

G4 インスタンスは、NVIDIA GRID 仮想ワークステーションをサポートします。詳細については、[NVIDIA Marketplace の提供サービス](#)を参照してください。

### G3 インスタンス

G3 インスタンスは NVIDIA Tesla M60 GPU を使用し、DirectX または OpenGL を使用してグラフィックアプリケーション向けに費用対効果の高パフォーマンスのプラットフォームを提供します。また、G3 インスタンスは、最大 4096x2160 の解像度を持つ 4 つのモニターと NVIDIA GRID 仮想アプリケーションのサポートなど、NVIDIA GRID 仮想ワークステーションの機能も提供します。G3 インスタンスは、アプリケーションの例としては、3D ビジュアライゼーション、グラフィックを強化したリモートワークステーション、3D レンダリング、動画エンコード、仮想リアリティやそのほかの大規模なパラレル処理を必要とするサーバー側のグラフィックワークロードなどのアプリケーションに最適です。

G3 インスタンスは、NVIDIA GRID 仮想ワークステーションと NVIDIA GRID 仮想アプリケーションをサポートします。これらの機能のいずれかを有効にするには、「[NVIDIA GRID 仮想アプリケーションの有効化 \(p. 264\)](#)」を参照してください。

### G2 インスタンス

G2 インスタンスは NVIDIA GRID K520 GPU を使用し、DirectX または OpenGL を使用してグラフィックアプリケーション向けに費用対効果の高パフォーマンスのプラットフォームを提供します。NVIDIA GRID GPU は、NVIDIA の高速キャプチャおよびエンコード API オペレーションもサポートします。アプリケーションのサンプルには、動画作成サービス、3D 仮想化、グラフィックを多用したストリーミングアプリケーションなどのサーバー側のグラフィックワークロードが含まれています。

## ハードウェア仕様

以下に示しているのは、高速コンピューティングインスタンスのハードウェア仕様の要約です。

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)	アクセラレータ
p2.xlarge	4	61	1
p2.8xlarge	32	488	8
p2.16xlarge	64	732	16
p3.2xlarge	8	61	1
p3.8xlarge	32	244	4
p3.16xlarge	64	488	8
p3dn.24xlarge	96	768	8
g2.2xlarge	8	15	1
g2.8xlarge	32	60	4
g3s.xlarge	4	30.5	1
g3.4xlarge	16	122	1

インスタンスタイプ	デフォルト vCPU	メモリ (GiB)	アクセラレータ
g3.8xlarge	32	244	2
g3.16xlarge	64	488	4
g4dn.xlarge	4	16	1
g4dn.2xlarge	8	32	1
g4dn.4xlarge	16	64	1
g4dn.8xlarge	32	128	1
g4dn.12xlarge	48	192	4
g4dn.16xlarge	64	256	1
f1.2xlarge	8	122	1
f1.4xlarge	16	244	2
f1.16xlarge	64	976	8

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

CPU オプションの指定についての詳細は、「[CPU オプションの最適化 \(p. 571\)](#)」を参照してください。

## インスタンスのパフォーマンス

インスタンスで最大のパフォーマンスを実現するための GPU 設定の最適化には、さまざまなものがあります。詳細については、「[GPU 設定の最適化 \(p. 265\)](#)」を参照してください。

EBS 最適化インスタンスは、インスタンスからの Amazon EBS I/O とその他のネットワークトラフィックとの競合を排除することによって、EBS ボリュームの安定した高パフォーマンスを実現できます。一部の高速コンピューティングインスタンスは、追加料金なしでデフォルトで EBS 最適化されています。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

一部の高速コンピューティングインスタンスタイプでは、Linux でプロセッサの C ステートと P ステートを制御できます。C ステートは非アクティブ時のコアのスリープレベルを制御し、P ステートは希望するコアからのパフォーマンス (CPU 周波数) を制御します。詳細については、「[EC2 インスタンスタイプのプロセッサのステート制御 \(p. 561\)](#)」を参照してください。

## ネットワークパフォーマンス

サポート対象のインスタンスタイプで、拡張されたネットワーキング機能を有効にすることができます。拡張ネットワーキングでは、パケット毎秒 (PPS) が非常に大きく、ネットワークのストレスが少なく、レイテンシーが低くなります。詳細については、「[Linux の拡張ネットワーキング \(p. 737\)](#)」を参照してください。

拡張されたネットワーキングのための Elastic Network Adapter (ENA) を使用するインスタンスタイプは、高いパケット/秒パフォーマンスと一貫して低いレイテンシーを同時に実現します。ほとんどのアプリケーションでは、高いレベルのネットワークパフォーマンスが一貫して必要なわけではありませんが、データの送受信時にアクセスする帯域幅を増やすことでメリットを得られます。ENA を使用し、「最大 10 Gbps」または「最大 25 Gbps」のネットワークパフォーマンスをサポートするインスタンスサイズでは、ネットワーク I/O クレジットメカニズムを使用して、平均帯域幅使用率に基づいてインスタンスにネットワーク帯域幅を割り当てます。これらのインスタンスでは、ネットワーク帯域幅がベースライン制限を下

回るとクレジットを獲得し、これらのクレジットをネットワークデータ転送を実行するときに使用できます。

以下に示しているのは、拡張ネットワーキングをサポートする高速コンピューティングインスタンスのネットワークパフォーマンスの要約です。

インスタンスタイプ	ネットワークパフォーマンス	拡張ネットワーキング
f1.2xlarge   f1.4xlarge   g3.4xlarge   p3.2xlarge	最大 10 Gbps	ENAs (p. 738)
g3s.xlarge   g3.8xlarge   p2.8xlarge   p3.8xlarge	10 Gbps	ENAs (p. 738)
g4dn.xlarge   g4dn.2xlarge   g4dn.4xlarge	最大 25 Gbps	ENAs (p. 738)
f1.16xlarge   g3.16xlarge   p2.16xlarge   p3.16xlarge	25 Gbps	ENAs (p. 738)
g4dn.8xlarge   g4dn.12xlarge   g4dn.16xlarge	50 Gbps	ENAs (p. 738)
p3dn.24xlarge	100 Gbps	ENAs (p. 738)

## インスタンスの機能

以下に示しているのは、高速コンピューティングインスタンスの機能の要約です。

	EBS のみ	NVMe EBS	インスタンスストア	配置グループ <sup>*</sup>
G2	なし	いいえ	SSD	あり
G3	はい	いいえ	いいえ	はい
G4	いいえ	はい	NVMe *	はい
P2	はい	いいえ	いいえ	はい
P3	p3dn.24xlarge: なし 他のすべてのサイズ: あり	p3dn.24xlarge: あり 他のすべてのサイズ: なし	p3dn.24xlarge: NVMe *	はい
F1	いいえ	いいえ	NVMe *	はい

\* ルートデバイスボリュームは、Amazon EBS ボリュームにする必要があります。

詳細については、以下を参照してください。

- Linux インスタンスの Amazon EBS および NVMe (p. 1027)
- Amazon EC2 インスタンスストア (p. 1076)
- プレイスマントグループ (p. 791)

## リリースノート

- インスタンスは、HVM AMI を使用して起動する必要があります。
- G4 インスタンスの要件を以下に示します。
  - NVMe ドライバー (p. 1027)がインストールされている必要があります。
  - Elastic Network Adapter (ENA) ドライバー (p. 738)がインストールされている必要があります。

以下の Linux AMI はこれらの要件を満たしています。

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (linux-aws カーネル) 以降
- Red Hat Enterprise Linux 7.4 以降
- SUSE Linux Enterprise Server 12 SP2 以降
- CentOS 7.4.1708 以降
- FreeBSD 11.1 以降
- Debian GNU/Linux 9 以降
- NVIDIA ドライバーがインストールされていない限り、GPU ベースのインスタンスは GPU にアクセスできません。詳細については、「[Linux インスタンスへの NVIDIA ドライバーのインストール \(p. 258\)](#)」を参照してください。
- リージョンごとに 100 AFI という制限があります。
- 実行できるインスタンス数は制限されています。詳細については、Amazon EC2 のよくある質問で「[How many instances can I run in Amazon EC2?](#)」を参照してください。これらの制限の引き上げを申請するには、[Request to Increase Amazon EC2 Instance Limit](#) というフォームを使用してください。

## Linux インスタンスへの NVIDIA ドライバーのインストール

GPU がアタッチされたインスタンス (P3 インスタンスや G4 インスタンスなど) には、適切な NVIDIA ドライバーがインストールされている必要があります。インスタンスタイプにより、パブリック NVIDIA ドライバー、または AWS カスタマーのみが使用できる Amazon S3 のドライバーをダウンロードするか、プリインストールされているドライバーで AMI を使用できます。

### 目次

- [NVIDIA ドライバーの種類 \(p. 258\)](#)
- [インスタンスタイプ別の使用可能なドライバー \(p. 259\)](#)
- [インストールオプション \(p. 259\)
  - オプション 1: NVIDIA ドライバーがインストールされた AMI \(p. 259\)
  - オプション 2: NVIDIA Tesla パブリックドライバー \(p. 260\)
  - オプション 3: GRID ドライバー \(G3 および G4 インスタンス\) \(p. 260\)
  - オプション 4: NVIDIA ゲームドライバー \(G4 インスタンス\) \(p. 262\)](#)

## NVIDIA ドライバーの種類

GPU ベースのインスタンスで使用できる主な種類の NVIDIA ドライバーを次に示します。

### Tesla ドライバー

これらのドライバーは主に、機械学習用の並列浮動小数点計算、ハイパフォーマンスコンピューティングアプリケーション用の高速フーリエ変換などの計算タスクに GPU を使用するコンピューティングワークロードを対象としています。

### GRID ドライバー

これらのドライバーは、3D モデルや高解像度動画などのコンテンツをレンダリングするプロフェッショナルな視覚化アプリケーションに最適なパフォーマンスを提供することが認定されています。GRID ドライバーを構成すると、2 つのモードをサポートできます。Quadro Virtual Workstation は、GPU あたり 4 台の 4K ディスプレイへのアクセスを提供します。GRID vApps は、RDSH アプリのホスティング機能を提供します。

### ゲームドライバー

これらのドライバーはゲーム用に最適化されており、パフォーマンスを向上させるために頻繁に更新されます。これらは GPU あたり単一の 4K ディスプレイをサポートします。

### に対してサポートされている API

- OpenCL、OpenGL、Vulkan
- NVIDIA CUDA および関連ライブラリ (cuDNN、TensorRT、nvJPEG、cuBLAS など)
- 動画エンコード用の NVENC と動画デコード用の NVDEC

## インスタンスタイプ別の使用可能なドライバー

次の表は、GPU インスタンスタイプごとにサポートされている NVIDIA ドライバーをまとめたものです。

インスタンスタイプ	Tesla ドライバー	GRID ドライバー	ゲームドライバー
G2	はい	いいえ	いいえ
G3	あり	はい	いいえ
G4	はい	はい	あり
P2	はい	いいえ	いいえ
P3	はい	はい †	いいえ

† Marketplace AMI のみを使用

## インストールオプション

次のいずれかのオプションを使用して、GPU インスタンスに必要な NVIDIA ドライバーを取得します。

### オプション

- オプション 1: NVIDIA ドライバーがインストールされた AMI (p. 259)
- オプション 2: NVIDIA Tesla パブリックドライバー (p. 260)
- オプション 3: GRID ドライバー (G3 および G4 インスタンス) (p. 260)
- オプション 4: NVIDIA ゲームドライバー (G4 インスタンス) (p. 262)

### オプション 1: NVIDIA ドライバーがインストールされた AMI

AWS と NVIDIA では、NVIDIA ドライバーがインストールされた、それぞれ異なる Amazon マシンイメージ (AMI) を提供しています。

- Tesla ドライバーを使用した Marketplace 製品
- GRID ドライバーを使用した Marketplace 製品

- ゲームドライバーを使用した Marketplace 製品

これらのAMIのいずれかを使用してインストールされたドライバーのバージョンを更新するには、バージョンの競合を避けるために、インスタンスから NVIDIA パッケージをアンインストールする必要があります。次のコマンドを使用して、NVIDIA パッケージをアンインストールします。

```
[ec2-user ~]$ sudo yum erase nvidia cuda
```

CUDA ツールキットパッケージは、NVIDIA ドライバーに依存します。NVIDIA パッケージをアンインストールすると、CUDA ツールキットが消去されます。NVIDIA ドライバーをインストールした後に、CUDA ツールキットを再インストールする必要があります。

#### オプション 2: NVIDIA Tesla パブリックドライバー

NVIDIA ドライバーをダウンロードするには

Linux インスタンスにログオンし、<http://www.nvidia.com/Download/Find.aspx> から、使用するインスタンスタイプに適した 64 ビット NVIDIA ドライバーをダウンロードします。

インスタンス	製品シリーズ	製品
G2	K シリーズ	K520
G3	M-Class	M60
G4 †	T シリーズ	T4
P2	K シリーズ	K80
P3	V シリーズ	V100

† G4 インスタンスは、ドライバーバージョン 426.00 以降が必要です。

Linux で NVIDIA ドライバーをインストールするには

ドライバーのインストールと設定の詳細については、[NVIDIA Driver Installation Quickstart Guide](#) を参照してください。

#### オプション 3: GRID ドライバー (G3 および G4 インスタンス)

これらのダウンロードは、AWS カスタマーのみが利用できます。ダウンロードすることで、ダウンロードしたソフトウェアは、NVIDIA Tesla T4 または NVIDIA Tesla M60 ハードウェアで、AMI の開発目的のみに使用することに同意したことになります。このソフトウェアをインストールすることは、[NVIDIA GRID Cloud End User License Agreement](#) の規約の遵守に同意したものと見なされます。

Linux インスタンスに NVIDIA GRID ドライバーをインストールするには

- Linux インスタンスに接続します。gcc および make をインストールします (まだインストールされていない場合)。
- パッケージのキャッシュを更新し、インスタンスに必要なパッケージの更新を取得します。
  - Amazon Linux、CentOS、Red Hat Enterprise Linux の場合:

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

- Ubuntu と Debian の場合:

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

3. (linux-aws パッケージの Ubuntu 16.04 以降) linux-aws パッケージをアップグレードして、最新バージョンを取得します。

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

4. インスタンスを再起動して、最新のカーネルバージョンを読み込みます。

```
[ec2-user ~]$ sudo reboot
```

5. 再起動後にインスタンスに再接続します。
6. 現在実行しているカーネルのバージョン用の gcc コンパイラおよびカーネルヘッダー/パッケージをインストールします。
  - Amazon Linux、CentOS、Red Hat Enterprise Linux の場合:

```
[ec2-user ~]$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

- Ubuntu と Debian の場合:

```
[ec2-user ~]$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

7. [CentOS、Red Hat Enterprise Linux、Ubuntu、Debian] NVIDIAグラフィックカード用の nouveau オープンソースドライバーを無効にします。

- a. nouveau ブラックリストファイルに /etc/modprobe.d/blacklist.conf を追加します。次のコードブロックをコピーして、ターミナルに貼り付けます。

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

- b. /etc/default/grub ファイルを編集して、次の行を追加します。

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

- c. Grub 設定を再構築します。

- CentOS と Red Hat Enterprise Linux の場合:

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Ubuntu と Debian の場合:

```
[ec2-user ~]$ sudo update-grub
```

8. 次のコマンドを使用して、GRID ドライバーインストールユーティリティをダウンロードします。

- G3 インスタンス用:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/latest/ .
```

- G4 インスタンス用:

```
[ec2-user ~]$ aws s3 cp --recursive s3://ec2-linux-nvidia-drivers/g4/latest/ .
```

GRID ドライバーの複数のバージョンがこのバケットに保存されます。使用可能なバージョンをすべて表示するには、次のコマンドを使用します。

```
[ec2-user ~]$ aws s3 ls --recursive s3://ec2-linux-nvidia-drivers/
```

9. 次のコマンドを使用して、ドライバーのインストールを実行するアクセス権限を追加します。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. 次のようにセルフインストールスクリプトを実行して、ダウンロードした GRID ドライバーをインストールします。次に例を示します。

```
[ec2-user ~]$ sudo /bin/sh ./NVIDIA-Linux-x86_64*.run
```

プロンプトが表示されたら、ライセンス契約を受諾し、必要に応じてインストールオプションを指定します (デフォルトのオプションを使用できます)。

11. インスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

12. ドライバーが機能していることを確認します。次のコマンドのレスポンスに、インストールされた NVIDIA ドライバーバージョンおよび GPU に関する詳細が表示されます。

```
[ec2-user ~]$ nvidia-smi -q | head
```

13. (オプション) 最大 4K 解像度の 4 台のディスプレイを活用するには、高性能ディスプレイプロトコル、[NICE DCV](#) を設定します。
14. (オプション) NVIDIA Quadro 仮想ワークステーションモードはデフォルトで有効になっています。RDSH アプリケーションホスティング機能用に GRID 仮想アプリケーションをアクティベート化するには、「[NVIDIA GRID 仮想アプリケーションの有効化 \(p. 264\)](#)」の GRID 仮想アプリケーションのアクティベート化手順を完了します。
15. (オプション) NVIDIA Quadro 仮想ワークステーションモードはデフォルトで有効になっています。(オプション) NVIDIA GRID 仮想アプリケーションを有効にするには、「[NVIDIA GRID 仮想アプリケーションの有効化 \(p. 264\)](#)」の GRID vApps アクティベーションステップを完了します。

#### オプション 4: NVIDIA ゲームドライバー (G4 インスタンス)

これらのドライバーは、AWS カスタマーのみが利用できます。これらをダウンロードすることで、ダウンロードしたソフトウェアは、NVIDIA Tesla T4 ハードウェアで、AMI の開発目的のみに使用することに同意したことになります。このソフトウェアをインストールすることは、[NVIDIA GRID Cloud End User License Agreement](#) の規約の遵守に同意したものと見なされます。

Linux インスタンスに NVIDIA ゲームドライバーをインストールするには

1. Linux インスタンスに接続します。gcc および make をインストールします (まだインストールされていない場合)。
2. パッケージのキャッシュを更新し、インスタンスに必要なパッケージの更新を取得します。
  - Amazon Linux、CentOS、Red Hat Enterprise Linux の場合:

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

- Ubuntu と Debian の場合:

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

- (linux-aws パッケージの Ubuntu 16.04 以降) linux-aws パッケージをアップグレードして、最新バージョンを取得します。

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

- インスタンスを再起動して、最新のカーネルバージョンを読み込みます。

```
[ec2-user ~]$ sudo reboot
```

- 再起動後にインスタンスに再接続します。
- 現在実行しているカーネルのバージョン用の gcc コンパイラおよびカーネルヘッダーパッケージをインストールします。

- Amazon Linux、CentOS、Red Hat Enterprise Linux の場合:

```
[ec2-user ~]$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

- Ubuntu と Debian の場合:

```
[ec2-user ~]$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

- [CentOS、Red Hat Enterprise Linux、Ubuntu、Debian] NVIDIAグラフィックカード用の nouveau オープンソースドライバーを無効にします。

- nouveau ブラックリストファイルに /etc/modprobe.d/blacklist.conf を追加します。次のコードブロックをコピーして、ターミナルに貼り付けます。

```
[ec2-user ~]$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivasfb
EOF
```

- /etc/default/grub ファイルを編集して、次の行を追加します。

```
GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

- Grub 設定を再構築します。

- CentOS と Red Hat Enterprise Linux の場合:

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Ubuntu と Debian の場合:

```
[ec2-user ~]$ sudo update-grub
```

- 次のコマンドを使用して、ゲームドライバーインストールユーティリティをダウンロードします。

```
[ec2-user ~]$ aws s3 cp --recursive s3://nvidia-gaming/linux/latest/ .
```

ゲームドライバーの複数のバージョンがこのバケットに保存されます。使用可能なバージョンをすべて表示するには、次のコマンドを使用します。

```
[ec2-user ~]$ aws s3 ls --recursive s3://nvidia-gaming/linux/
```

9. 次のコマンドを使用して、ドライバーのインストールを実行するアクセス権限を追加します。

```
[ec2-user ~]$ chmod +x NVIDIA-Linux-x86_64*.run
```

10. 次のコマンドを使用してインストーラを実行します。

```
[ec2-user ~]$ sudo ./NVIDIA-Linux-x86_64*.run
```

プロンプトが表示されたら、ライセンス契約を受諾し、必要に応じてインストールオプションを指定します（デフォルトのオプションを使用できます）。

11. 次のコマンドを使用して必要な設定ファイルを作成します。

```
[ec2-user ~]$ cat << EOF | sudo tee -a /etc/nvidia/gridd.conf
vGamingMarketplace=2
EOF
```

12. 次のコマンドを使用して認証ファイルをダウンロードし、名前を変更します。

```
[ec2-user ~]$ sudo curl -o /etc/nvidia/GridSwCert.txt "https://s3.amazonaws.com/nvidia-gaming/GridSwCert-Linux.cert"
```

13. インスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

14. (オプション) 最大 4K 解像度の 1 台のディスプレイを活用するには、高性能ディスプレイプロトコル、[NICE DCV](#) を設定します。
15. (オプション) 最大 4K 解像度の 4 台のディスプレイを活用するには、高性能ディスプレイプロトコル、[NICE DCV](#) を設定します。

## NVIDIA GRID 仮想アプリケーションの有効化

G3 および G4 インスタンスで GRID 仮想アプリケーションを有効にするには (NVIDIA GRID 仮想ワークステーションはデフォルトで有効になっています)、/etc/nvidia/gridd.conf ファイルでドライバーのプロダクトタイプを定義する必要があります。

Linux インスタンスで GRID 仮想アプリケーションを有効にするには

1. 提供されるテンプレートファイルから /etc/nvidia/gridd.conf ファイルを作成します。

```
[ec2-user ~]$ sudo cp /etc/nvidia/gridd.conf.template /etc/nvidia/gridd.conf
```

2. お好きなテキストエディタで /etc/nvidia/gridd.conf ファイルを開きます。
3. FeatureType 行を見つけ、それを 0 に等しく設定します。次に、IgnoreSP=TRUE の行を追加します。

```
FeatureType=0
IgnoreSP=TRUE
```

4. ファイルを保存して終了します。
5. インスタンスを再起動し、新しい設定を取得します。

```
[ec2-user ~]$ sudo reboot
```

## GPU 設定の最適化

G3、G4、P2、P3、および P3dn の各インスタンスで最大のパフォーマンスを実現するために GPU 設定を最適化する方法がいくつかあります。デフォルトでは、NVIDIA ドライバーは自動ブースト機能を使用しますが、これは GPU クロック速度に左右されます。自動ブースト機能を無効にし、GPU クロック速度を最大周波数に設定することで、安定して GPU インスタンスで最大パフォーマンスを実現できます。次の手順では、GPU 設定を永続的に設定し、自動ブースト機能を無効化して、GPU クロック速度を最大周波数に設定します。

### GPU 設定を最適化するには

1. GPU 設定を永続的になるように設定します。このコマンドの実行には数分かかることがあります。

```
[ec2-user ~]$ sudo nvidia-persistenced
```

2. インスタンスのすべての GPU で自動ブースト機能を無効にします。

```
[ec2-user ~]$ sudo nvidia-smi --auto-boost-default=0
```

#### Note

P3、P3dn、および G4 の各インスタンスの GPU は autoboot をサポートしていません。

3. すべての GPU クロック速度を最大周波数に設定します。次のコマンドで指定されるメモリとグラフィッククロック速度を使用します。

#### Note

一部のバージョンの NVIDIA ドライバーでは、アプリケーションのクロック速度を設定できないため、"Setting applications clocks is not supported for GPU ..." エラーがスローされますが、無視できます。

- G3 インスタンス:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,1177
```

- G4 インスタンス:

```
[ec2-user ~]$ sudo nvidia-smi -ac 5001,1590
```

- P2 インスタンス:

```
[ec2-user ~]$ sudo nvidia-smi -ac 2505,875
```

- P3 および P3dn インスタンス:

```
[ec2-user ~]$ sudo nvidia-smi -ac 877,1530
```

## FPGA 開発を開始する

[FPGA Developer AMI](#) は、AFI を開発、テスト、および構築するためのツールを提供します。FPGA Developer AMI は、32 GB 以上のシステムメモリを備える任意の EC2 インスタンス (例: C5、M4、R4 インスタンス) で使用できます。

詳細については、「[AWS FPGA Hardware Development Kit](#)」を参照してください。

## AWS Inferentia 開発の開始方法

[AWS Deep Learning AMI](#) には、AWS Inferentia を使用して機械学習アプリケーションを開発、テスト、構築するためのツールが用意されています。すべての Amazon EC2 Inf1 インスタンスで AWS Inferentia 開発用の深層学習 AMI を使用できます。

詳細については、[AWS Neuron](#) のドキュメントを参照してください。

## インスタンスタイプの検索

インスタンスの起動前には、使用するインスタンスタイプを選択しなければなりません。選択するインスタンスタイプは、起動するインスタンスに求める機能に応じて変えることができます。インスタンスに求める機能の具体例には、以下に関する機能があります。

- リージョン
- アーキテクチャー: 32-bit (i386)、64-bit (x86\_64) または 64-bit ARM (arm64)
- コンピューティング
- メモリ
- ストレージ
- ネットワークパフォーマンス

## Amazon EC2コンソールを使用したインスタンスタイプの検索

インスタンスタイプは、Amazon EC2 コンソールを使って検索できます。インスタンスタイプページを使うと、使用可能なすべてのインスタンスタイプを横断的に検索できます。

1. [Amazon EC2 コンソール](#) を開きます。
2. ナビゲーションバーから、インスタンスを起動するリージョンを選択します。お客様は場所に関係なく、使用できるリージョンをどれでも選択できます。
3. ナビゲーションペインで、[インスタンスタイプ] を選択します。
4. (オプション) プリファレンスアイコンを選ぶ方法で、表示するインスタンスタイプ属性 (オンデマンド Linux 料金など) を選択します。代わりに、一覧からインスタンスタイプを選択し、[詳細] タブにすべての属性が表示されるようにすることもできます。
5. (オプション) [フィルター] オプションを使用して、一覧表示されるインスタンスタイプを、興味のあるインスタンスタイプに限定します。具体例を挙げると、8 超の vCPU を持つておらず、ハイバネーションをサポートしているインスタンスタイプをすべて一覧表示するといったことができます。
6. (オプション) 複数のインスタンスタイプを選択して、並べて比較できる形ですべての属性が [詳細] ペインに表示されるようにします。
7. (オプション) インスタンスタイプの一覧を保存して後日見直せるようにするには、[一覧のダウンロード (CSV)] を選択します。選択すると、カンマ区切り値 (.csv) 形式のファイルをダウンロードできます。このファイルには、設定したフィルターに合致するすべてのインスタンスタイプが含まれるほか、表に表示されているすべての属性に合致するインスタンスタイプがあればそのタイプも含まれます。

## AWS CLIを使用したインスタンスタイプの検索

Amazon EC2のAWS CLIコマンドを使用すると、必要なインスタンスタイプのみを一覧表示できます。必要なインスタンスタイプが見つかったら、その ID を記録して、インスタンスの起動に使用できるようにしておきます。詳細については、AWS Command Line Interface ユーザーガイドの[AWS CLI を使用したインスタンスの起動](#)を参照してください。

`describe-instance-types`コマンドは、フィルタリングパラメータをサポートしています。たとえば、以下のフィルターを使用すると、48 個の vCPU を持つインスタンスタイプのみを表示できます。

```
aws ec2 describe-instance-types --filters "Name=vcpu-info.default-vcpus,Values=48"
```

`describe-instance-type-offerings`コマンドは、フィルタリングパラメータをサポートしています。たとえば、`--location-type` パラメーターを使用すると、アベイラビリティーゾーンの提供内容を表示できます。

```
aws ec2 describe-instance-type-offerings --location-type "availability-zone"
```

前のコマンドに以下のフィルターを追加すると、`us-east-1a`アベイラビリティーゾーンにおいて提供されるインスタンスタイプのみ表示できます。

```
--filters "Name=location,Values=us-east-1a"
```

## インスタンスタイプを変更する

ニーズが変わるために、インスタンスの利用率が高すぎたり（インスタンスタイプが小さすぎる）、低すぎたりする（インスタンスタイプが大きすぎる）ことに気付く場合があります。このような場合は、インスタンスのサイズを変更できます。たとえば、`t2.micro` インスタンスがワークロードに対して小さすぎる場合は、そのインスタンスをワークロードに適した別のインスタンスタイプに変更できます。

以前の世代のインスタンスタイプから現行世代のインスタンスタイプに移行し、たとえば IPv6 のサポートなどの機能を活用することもできます。

インスタンスのルートデバイスが EBS ボリュームの場合は、インスタンスタイプを変更するだけでインスタンスのサイズを変更できます。この処理はサイズ変更と呼ばれます。インスタンスのルートデバイスがインスタンストアボリュームの場合、必要なインスタンスタイプで新しいインスタンスにアプリケーションを移行する必要があります。ルートデバイスボリュームの詳細については、[ルートデバイスのストレージ \(p. 96\)](#) を参照してください。

インスタンスのサイズを変更すると、インスタンスの設定と互換性のあるインスタンスタイプを選択する必要があります。使用するインスタンスタイプに既存のインスタンス設定との互換性がない場合、必要なインスタンスタイプで新しいインスタンスにアプリケーションを移行する必要があります。

### Important

インスタンスのサイズを変更すると、サイズ変更したインスタンスには通常、元のインスタンスの起動時に指定したのと同じ数のインスタンストアボリュームが設定されます。NVMe インスタンストアボリューム（デフォルトで使用可能）をサポートするインスタンスタイプでは、AMI によっては、サイズ変更されたインスタンスに追加のインスタンストアボリュームが存在する可能性があります。それ以外の場合は、新しいインスタンスを起動するときに必要なインスタンストアボリュームの数を指定して、アプリケーションを新しいインスタンスタイプのインスタンスに手動で移行できます。

### コンテンツ

- インスタンスのサイズ変更の互換性 (p. 268)
- Amazon EBS-Backed インスタンスのサイズ変更 (p. 269)
- Instance Store-Backed インスタンスの移行 (p. 270)
- 新しいインスタンス設定への移行 (p. 270)

## インスタンスのサイズ変更の互換性

現在のインスタンスタイプおよび使用する新しいインスタンスタイプが次のように互換性がある場合にのみ、インスタンスのサイズを変更することができます。

- 仮想化タイプ: Linux AMI は、準仮想化 (PV) またはハードウェア仮想マシン (HVM) のいずれかの仮想化タイプを使用します。PV AMI から起動されたインスタンスのサイズを、HVM のみであるインスタンスタイプに変更することはできません。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。インスタンスの仮想化タイプを確認するには、Amazon EC2 コンソールで [Instances] 画面の詳細ペインの [Virtualization] フィールドを参照します。
- アーキテクチャー: AMI はプロセッサのアーキテクチャーに固有であるため、プロセッサアーキテクチャーが現在のインスタンスタイプと同じインスタンスタイプを選択する必要があります。例:
  - A1 インスタンスは、Arm アーキテクチャベースのプロセッサをサポートする唯一のインスタンスです。AMI アーキテクチャベースのプロセッサでインスタンスタイプをサイズ変更している場合、AMI アーキテクチャベースのプロセッサをサポートするインスタンスタイプに制限されます。
  - 32 ビット AMI をサポートするのは以下のインスタンスタイプのみです。t2.nano、t2.micro、t2.small、t2.medium、c3.large、t1.micro、m1.small、m1.medium、および c1.medium。32 ビットインスタンスのサイズを変更する場合は、これらのインスタンスタイプに制限されます。
- ネットワーク: 一部のインスタンスタイプは、VPC で起動する必要があります。そのため、EC2-Classic プラットフォームでは、デフォルト以外の VPC がない場合にインスタンスのサイズを VPC でのみ利用可能なインスタンスタイプに変更することはできません。インスタンスが VPC 内にあるかどうかを確認するには、Amazon EC2 コンソールで [Instances] 画面の詳細ペインの [VPC ID] 値を確認します。詳細については、「[EC2-Classic の Linux インスタンスから VPC の Linux インスタンスへの移行 \(p. 825\)](#)」を参照してください。
- 拡張ネットワーク: [拡張ネットワーク \(p. 737\)](#)をサポートするインスタンスタイプでは、必要なドライバーがインストールされていなければなりません。たとえば、A1, C5, C5d, C5n, F1, G3, G4, H1, I3, I3en, Inf1, m4.16xlarge, M5, M5a, M5ad, M5d, M5dn, M5n, P2, P3, R4, R5, R5a, R5ad, R5d, R5dn, R5n, T3, T3a, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, u-24tb1.metal, X1, X1e, and z1d インスタンスタイプでは、Elastic Network Adapter (ENA) ドライバーがインストールされた EBS-backed AMI が必要です。既存のインスタンスを、拡張ネットワークをサポートするインスタンスタイプにサイズ変更するには、[ENA ドライバー \(p. 738\)](#)または [ixgbefv ドライバー \(p. 751\)](#)を必要に応じてインスタンスにインストールしてください。
- NVMe: [Nitro ベースのインスタンス \(p. 187\)](#)では、EBS ボリュームは NVMe ブロックデバイスとして公開されます。NVMe をサポートしないインスタンスタイプから NVMe をサポートするインスタンスタイプにインスタンスをサイズ変更する場合、まずインスタンスに [NVMe ドライバー \(p. 1027\)](#)をインストールする必要があります。ブロックデバイスマッピングで指定したデバイス名は、NVMe デバイス名 (/dev/nvme[0-26]n1) を使用して名称変更されます。したがって、/etc/fstab を使用してブート時にファイルシステムをマウントするには、デバイス名の代わりに UUID/Label を使用する必要があります。
- AMI: 拡張ネットワークと NVMe をサポートするインスタンスタイプで必要な AMI については、次のマニュアルの「リリースノート」を参照してください。
  - [汎用インスタンス \(p. 190\)](#)
  - [コンピュート最適化インスタンス \(p. 231\)](#)
  - [メモリ最適化インスタンス \(p. 236\)](#)
  - [ストレージ最適化インスタンス \(p. 245\)](#)

## Amazon EBS–Backed インスタンスのサイズ変更

Amazon EBS–Backed インスタンスタイプを変更するには、そのインスタンスを停止する必要があります。インスタンスを停止して再度起動するときは、以下に注意してください：

- インスタンスは新しいハードウェアに移動されますが、インスタンス ID は変更されません。
- インスタンスにパブリック IPv4 アドレスがある場合には、このアドレスは解放されて、新しいパブリック IPv4 アドレスになります。インスタンスは、プライベート IPv4 アドレス、Elastic IP アドレス、および IPv6 アドレスを保持します。
- インスタンスが Auto Scaling グループにある場合、Amazon EC2 Auto Scaling サービスはインスタンスを異常と判断して停止し、場合によってはそれを終了して代わりのインスタンスを起動します。インスタンスのサイズを変更するときに、そのグループのスケーリングプロセスを中断することで、これを行なうことができます。詳細については、『Amazon EC2 Auto Scaling ユーザーガイド』の「[スケーリングプロセスの中止と再開](#)」を参照してください。
- インスタンスがクラスター・プレイスメント・グループ ([p. 792](#)) にあり、インスタンスタイプの変更後にインスタンスの起動に失敗する場合は、次の操作をお試しください。クラスター・プレイスメント・グループのすべてのインスタンスを停止し、影響を受けたインスタンスのインスタンスタイプを変更します。次に、クラスター・プレイスメント・グループのすべてのインスタンスを再起動します。
- インスタンスが停止している間のダウントIMEを予定しておいてください。インスタンスを停止し、サイズ変更を行うと、数分かかります。インスタンスを再起動すると、アプリケーションの起動スクリプトによってかかる時間が変動する場合があります。

詳細については、「[インスタンスの停止と起動 \(p. 529\)](#)」を参照してください。

AWS マネジメントコンソール を使って Amazon EBS–Backed インスタンスのサイズを変更するには、次の手順を行います。

Amazon EBS–Backed インスタンスのサイズを変更するには

1. (オプション) 新しいインスタンスのタイプに既存のインスタンスにインストールされていないドライバーが必要な場合は、インスタンスに接続してドライバーをインストールする必要があります。詳細については、「[インスタンスのサイズ変更の互換性 \(p. 268\)](#)」を参照してください。
2. Amazon EC2 コンソールを開きます。
3. ナビゲーションペインで、[インスタンス] を選択します。
4. インスタンスを選択し、[Actions]、[Instance State]、[Stop] の順に選択します。
5. 確認ダイアログボックスで [Yes, Stop] を選択します。インスタンスが停止するまで、数分かかる場合があります。
6. インスタンスが選択された状態で、[Actions]、[Instance Settings]、[Change Instance Type] の順に選択します。インスタンスの状態が `stopped` ではない場合、このアクションは無効になります。
7. [Change Instance Type] ダイアログボックスで、次の操作を行います。
  - a. [インスタンスタイプ] から、使用するインスタンスタイプを選択します。使用するインスタンスタイプがリストに表示されない場合は、インスタンスの設定と互換性がありません (仮想化タイプが原因の場合など)。詳細については、「[インスタンスのサイズ変更の互換性 \(p. 268\)](#)」を参照してください。
  - b. (オプション) 選択したインスタンスタイプが EBS–最適化をサポートしている場合は、[EBS-optimized] を選択して EBS 最適化を有効にするか、[EBS-optimized] を選択解除して EBS 最適化を無効にします。選択したインスタンスタイプがデフォルトで EBS に最適化される場合、[EBS に最適化] が選択されており、選択解除することはできません。
  - c. [Apply] を選択して、新しい設定を受け入れます。
8. 停止されているインスタンスを再起動するには、インスタンスを選択後、[Actions]、[Instance State]、[Start] の順に選択します。

9. 確認ダイアログボックスで [Yes, Start] を選択します。インスタンスが *running* 状態になるまで、数分かかる場合があります。
10. (トラブルシューティング) インスタンスが起動しない場合、新しいインスタンスタイプの要件の 1 つが満たされていない可能性があります。詳細については、「[Why is my Linux instance not booting after I changed its type?](#)」を参照してください。

## Instance Store-Backed インスタンスの移行

1 つの instance store-backed インスタンスから別のインスタンスタイプの instance store-backed インスタンスにアプリケーションを移動する場合は、インスタンスからイメージを作成し、このイメージから必要なインスタンスタイプで新しいインスタンスを起動することにより、移行する必要があります。インスタンスでホストしているアプリケーションをユーザーが中断なく継続して使用できるようにするには、元のインスタンスに関連付けられた Elastic IP アドレスを新しいインスタンスに関連付ける必要があります。その後、元のインスタンスを終了できます。

Instance Store-Backed インスタンスを移行するには

1. 永続的ストレージに保持する必要がある、インスタンストアボリュームのデータをバックアップします。保持する必要がある EBS ボリュームのデータを移行するには、ボリュームのスナップショットを作成するか（「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」を参照）、またはインスタンスからボリュームをデタッチして後で新しいインスタンスにアタッチできるようにします（「[インスタンスからの Amazon EBS ボリュームのデタッチ \(p. 967\)](#)」を参照）。
2. Instance Store-Backed インスタンスから AMI を作成するには、「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」に記載された前提条件と手順に従います。インスタンスから AMI を作成したら、この手順に戻ります。
3. ナビゲーションペインで Amazon EC2 コンソールを開き、[AMI] を選択します。フィルターリストで [自己所有] を選択し、前のステップで作成したイメージを選択します。[AMI Name] は、イメージを登録したときに指定した名前であり、[Source] は Amazon S3 バケットです。

### Note

前のステップで作成した AMI が表示されない場合は、AMI を作成したリージョンを選択していることを確認します。

4. [Launch] を選択します。インスタンスに対してオプションを指定する場合は、使用する新しいインスタンスタイプを選択してください。使用するインスタンスタイプを選択できない場合は、作成した AMI の構成と互換性がありません（仮想化タイプが原因の場合など）。元のインスタンスからデタッチした EBS ボリュームを指定することもできます。

インスタンスが *running* 状態になるまで、数分かかる場合があります。

5. (オプション) 不要になった場合は、起動したインスタンスを終了できます。インスタンスを選択し、終了しようとしているのが新しいインスタンスではなく元のインスタンスであることを確認します（名前や起動時間を確認するなど）。[アクション]、[インスタンスの状態]、[終了] の順に選択します。

## 新しいインスタンス設定への移行

インスタンスの現在の設定に使用する新しいインスタンスタイプとの互換性がない場合、そのインスタンスタイプにインスタンスのサイズを変更することはできません。代わりに、使用する新しいインスタンスタイプと互換性がある設定で新しいインスタンスにアプリケーションを移行できます。

PV AMI から起動されたインスタンスから HVM のみのインスタンスタイプに移動する場合、一般的な手順は次のとおりです。

互換性のあるインスタンスへアプリケーションを移行するには

1. 永続的ストレージに保持する必要がある、インスタンストアボリュームのデータをバックアップします。保持する必要がある EBS ボリュームのデータを移行するには、ボリュームのスナップショット

を作成するか（「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」を参照）、またはインスタンスからボリュームをデタッチして後で新しいインスタンスにアタッチできるようにします（「[インスタンスからの Amazon EBS ボリュームのデタッチ \(p. 967\)](#)」を参照）。

2. 新しいインスタンスを起動して、以下のものを選択します。
  - HVM AMI
  - HVM のみのインスタンスタイプ。
  - Elastic IP アドレスを使用している場合は、元のインスタンスを現在実行している VPC を選択します。
  - 元のインスタンスからデタッチして新しいインスタンスにアタッチする EBS ボリューム、または作成したスナップショットに基づいた新しい EBS ボリューム。
  - 同じトラフィックが新しいインスタンスに到達できるようにする場合は、元のインスタンスと関連付けられるセキュリティグループを選択します。
3. アプリケーションと必要なソフトウェアをインスタンスにインストールします。
4. 元のインスタンスのインスタンスストアボリュームからバックアップしたデータを復元します。
5. Elastic IP アドレスを使用している場合、以下のように新しく起動したインスタンスにそのアドレスを割り当てます。
  - a. ナビゲーションペインで [Elastic IP] を選択します。
  - b. 元のインスタンスに関連付ける Elastic IP アドレスを選択して、[アクション]、[アドレスの関連付けの解除] の順に選択します。確認を求めるメッセージが表示されたら、[Disassociate address] を選択します。
  - c. Elastic IP アドレスがまだ選択された状態で、[アクション]、[アドレスの関連付け] の順に選択します。
  - d. [Instance] から新しいインスタンスを選択し、[Associate] を選択します。
6. (オプション) 不要になった場合は、元のインスタンスを終了できます。インスタンスを選択し、終了しようとしているのが新しいインスタンスではなく元のインスタンスであることを確認します（名前や起動時間を確認するなど）。[Actions]、[Instance State]、[Terminate] の順に選択します。

## インスタンスタイプに関する推奨事項の取得

AWS Compute Optimizer は、パフォーマンスの向上、コストの削減、またはその両方に役立つ Amazon EC2 インスタンスの推奨事項を提供します。これらの推奨事項を使用して、新しいインスタンスタイプに移行するかどうかを判断できます。

推奨事項を作成するために、Compute Optimizer は既存のインスタンスの仕様と使用率メトリクスを分析します。次に、コンパイルされたデータを使用して、既存のワークロードを処理するのに最適な Amazon EC2 インスタンスタイプを推奨します。推奨事項は、時間あたりのインスタンス料金とともに返されます。

このトピックでは、Amazon EC2 コンソールで推奨事項を表示する方法について説明します。詳細については、[AWS Compute Optimizer ユーザーガイド](#) を参照してください。

### Note

Compute Optimizer から推奨事項を取得するには、まず Compute Optimizer にオプトインする必要があります。詳細については、AWS Compute Optimizer ユーザーガイドの「[AWS Compute Optimizer の開始方法](#)」を参照してください。

### 目次

- [制約事項 \(p. 272\)](#)
- [結果 \(p. 272\)](#)
- [推奨事項の表示 \(p. 272\)](#)

- 推奨事項の評価に関する考慮事項 (p. 273)

## 制約事項

Compute Optimizer は現在、M、C、R、T、X のインスタンスタイプの推奨事項を生成します。他のインスタンスタイプは Compute Optimizer によって考慮されません。他のインスタンスタイプを使用している場合、Compute Optimizer の推奨事項ビューに推奨事項は表示されません。これらのインスタンスタイプや他のインスタンスタイプについては、「[インスタンスタイプ \(p. 183\)](#)」を参照してください。

## 結果

Compute Optimizer は、EC2 インスタンスの検出結果を以下のように分類します。

- プロビジョニング不足 – EC2 インスタンスは、インスタンス仕様の CPU、メモリ、ネットワークなどの 1 つ以上の要素がワークロードのパフォーマンス要件を満たしていない場合に、プロビジョニング不足と見なされます。EC2 インスタンスがプロビジョニング不足である場合、アプリケーションのパフォーマンスが低下することがあります。
- 過剰プロビジョニング – EC2 インスタンスは、インスタンス仕様の CPU、メモリ、ネットワークなどの 1 つ以上の要素をサイズダウンしてもワークロードのパフォーマンス要件を満たす場合や、どの仕様要素もプロビジョニング不足でない場合に、過剰プロビジョニングと見なされます。EC2 インスタンスの過剰プロビジョニングは、余分なインフラストラクチャコストを発生させる場合があります。
- 最適化 – EC2 インスタンスは、インスタンス仕様の CPU、メモリ、ネットワークなどのすべての要素がワークロードのパフォーマンス要件を満たし、インスタンスが過剰プロビジョニングされていない場合に、最適化されていると見なされます。最適化された EC2 インスタンスは、最適なパフォーマンスとインフラストラクチャコストでワークロードを実行します。最適化されたインスタンスとして、Compute Optimizer は新世代のインスタンスタイプを推奨する場合があります。
- なし – このインスタンスに対する推奨事項はありません。この結果になる可能性があるのは、Compute Optimizer にオプトインしてから 12 時間未満である場合、インスタンスの実行時間が 30 時間未満である場合、またはインスタンスタイプが Compute Optimizer でサポートされていない場合です。詳細については、前セクションの [制約事項 \(p. 272\)](#) を参照してください。

## 推奨事項の表示

Compute Optimizer にオプトインすると、EC2 コンソールで EC2 インスタンスについて Compute Optimizer によって生成される結果を表示できます。その後、Compute Optimizer コンソールにアクセスして、推奨事項を表示できます。最近オプトインした場合は、結果が EC2 コンソールに反映されるまで最大 12 時間かかることがあります。

EC2 コンソールを使用して EC2 インスタンスの推奨事項を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[説明] タブで [Finding (結果)] フィールドを確認します。[View detail (詳細を表示)] を選択します。

インスタンスは Compute Optimizer で開き、[Current] というラベルが付いています。最大 3 つの異なるインスタンスタイプが [Option 1 (オプション 1)]、[Option 2 (オプション 2)]、[Option 3 (オプション 3)] というラベル付きで推奨されます。ウィンドウの下半分には、現在のインスタンスの最新の CloudWatch メトリクスデータとして、CPU 使用率、メモリ使用率、ネットワーク入力、ネットワーク出力が表示されます。

4. (オプション) Compute Optimizer コンソールで、設定 () アイコンを選択してテーブル内の表示列を変更するか、現在および推奨のインスタンスタイプの購入オプション別の一般料金情報を表示します。

#### Note

リザーブドインスタンスを購入した場合、オンデマンドインスタンスはリザーブドインスタンスとして請求される場合があります。現在のインスタンスタイプを変更する前に、まずリザーブドインスタンスの使用率とカバレッジに対する影響を評価します。

推奨事項の 1 つを使用するかどうかを決定します。最適化の目的を、パフォーマンスの向上、コストの削減、またはこの両方のいずれにするかを決定します。詳細については、AWS Compute Optimizer ユーザーガイドの「[リソースの推奨事項の表示](#)」を参照してください。

Compute Optimizer コンソールを使用して、すべてのリージョンのすべての EC2 インスタンスに関する推奨事項を表示するには

1. <https://console.aws.amazon.com/compute-optimizer/> で、Compute Optimizer コンソールを開きます。
2. [View recommendations for all EC2 instances (すべての EC2 インスタンスの推奨事項を表示)] を選択します。
3. 推奨事項ページでは、次のアクションを実行できます。
  - a. 1 つ以上の AWS リージョンに対する推奨事項をフィルタリングするには、[Filter by one or more Regions (1 つ以上のリージョンでフィルター)] テキストボックスにリージョンの名前を入力するか、表示されるドロップダウンリストで 1 つ以上のリージョンを選択します。
  - b. 別のアカウントのリソースに対する推奨情報を表示するには、[Account (アカウント)] を選択し、別のアカウント ID を選択します。
  - c. 選択したフィルタをクリアするには、[Clear filters (フィルターのクリア)] を選択します。
  - d. 現在および推奨のインスタンスタイプに表示される購入オプションを変更するには、設定 () アイコンを選択し、[On-Demand Instances (オンデマンドインスタンス)]、[Reserved Instances, standard 1-year no upfront (リザーブドインスタンス、標準 1 年間前払いなし)]、[Reserved Instances, standard 3-year no upfront (リザーブドインスタンス、標準 3 年間前払いなし)] のいずれかを選択します。
  - e. 追加の推奨事項や使用率メトリックスの比較などの詳細を表示するには、目的のインスタンスの横に表示される結果 ([Under-provisioned (プロビジョニング不足)]、[Over-provisioned (過剰プロビジョニング)]、または [Optimized (最適化)]) を選択します。詳細については、AWS Compute Optimizer ユーザーガイドの「[リソースの詳細の表示](#)」を参照してください。

## 推奨事項の評価に関する考慮事項

インスタンスタイプを変更する前に、次の点を考慮してください。

- 推奨情報は使用状況を予測するものではありません。推奨事項は、直近の 14 日間の使用履歴に基づいています。将来のリソースニーズを満たすことが予想されるインスタンスタイプを必ず選択します。
- グラフ化されたメトリクスを参考にして、実際の使用量がインスタンスの容量よりも低いかどうかを判断します。メトリクスデータ (平均、ピーク、パーセンタイル) を CloudWatch で表示し、EC2 インスタンスの推奨事項をさらに評価することもできます。たとえば、一日の CPU パーセンテージメトリクスがどのように変化するか、ピークに対応する必要があるかどうかに注目します。詳細については、Amazon CloudWatch ユーザーガイドの「[使用可能なメトリクスの表示](#)」を参照してください。
- Compute Optimizer は、バーストパフォーマンスインスタンス (T3、T3a、および T2 インスタンス) の推奨事項を提供する場合があります。定期的にベースラインを超えてバーストする場合は、新しいインスタンスタイプの vCPU に基づいて引き続きバーストを実行できることを確認します。詳細については、「[バースト可能パフォーマンスインスタンスの CPU クレジットおよびベースラインパフォーマンス \(p. 200\)](#)」を参照してください。

- リザーブドインスタンスを購入した場合、オンデマンドインスタンスはリザーブドインスタンスとして請求される場合があります現在のインスタンスタイプを変更する前に、まずリザーブドインスタンスの使用率とカバレッジに対する影響を評価します。
- 可能であれば、新世代のインスタンスへの交換を検討します。
- 別のインスタンスファミリーに移行する場合は、仮想化、アーキテクチャー、ネットワークタイプなどの点で、現在のインスタンスタイプと新しいインスタンスタイプに互換性があることを確認してください。詳細については、「[インスタンスのサイズ変更の互換性 \(p. 268\)](#)」を参照してください。
- 最後に、推奨事項ごとに提供されるパフォーマンスリスク評価を検討します。パフォーマンスリスクは、推奨されるインスタンスタイプがワークロードのパフォーマンス要件を満たすかどうかを検証するために費やす必要のある作業量を示します。また、変更前と変更後に厳格な負荷テストおよびパフォーマンステストを行うことをお勧めします。

EC2 インスタンスのサイズを変更する際には、他の考慮事項があります。詳細については、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。

#### その他のリソース

- [インスタンスタイプ \(p. 183\)](#)
- [AWS Compute Optimizer ユーザーガイド](#)

## インスタンス購入オプション

Amazon EC2 には、ニーズに基づいてコストを最適化するための以下の購入オプションがあります。

- [オンデマンドインスタンス] – 起動するインスタンスに対して秒単位でお支払いいただきます。
- Savings Plans – 1~3 年の期間、1 時間に USD で、一定の使用量を守ることにより Amazon EC2 コストを削減します。
- リザーブドインスタンス – 1~3 年の期間、インスタンスタイプとリージョンを含む一定のインスタンス設定を守ることにより Amazon EC2 コストを削減します。
- スケジュールされたインスタンス – 1 年の期間、指定された定期的なスケジュールで常に使用できるインスタンスを購入します。
- スポットインスタンス – 未使用的 EC2 インスタンスをリクエストして、Amazon EC2 コストを大幅に削減します。
- Dedicated Hosts – 完全にインスタンスの実行専用の物理ホストに対してお支払いいただき、既存のソケット単位、コア単位、または VM 単位のソフトウェアライセンスを持ち込んでコストを削減できます。
- ハードウェア専有インスタンス – シングルテナントハードウェアで実行されるインスタンスに対して、時間単位でお支払いいただきます。
- キャパシティーの予約 – 特定のアベイラビリティゾーンの EC2 インスタンスに対して任意の期間キャパシティーを予約します。

キャパシティーの予約が連続して必要な場合、特定のアベイラビリティゾーン用にリザーブドインスタンスまたはキャパシティーの予約を購入します。または、スケジュールされたインスタンスを購入します。スポットインスタンスは、アプリケーションを実行する時間に柔軟性がある場合や、それらを中断できる場合に、費用効率の高い選択肢です。Dedicated Hosts またはハードウェア専有インスタンスは、既存のサーバー範囲内のソフトウェアライセンスを使用することにより、コンプライアンス要件を満たし、コストを削減するのに役立ちます。詳細については、「[Amazon EC2 料金表](#)」を参照してください。

Savings Plans の詳細については、[AWS Savings Plans ユーザーガイド](#)を参照してください。

#### コンテンツ

- [インスタンスのライフサイクルの決定 \(p. 275\)](#)
- [オンデマンドインスタンス \(p. 276\)](#)
- [リザーブドインスタンス \(p. 279\)](#)
- [スケジュールされたリザーブドインスタンス \(p. 317\)](#)
- [スポットインスタンス \(p. 320\)](#)
- [Dedicated Hosts \(p. 395\)](#)
- [ハードウェア専有インスタンス \(p. 425\)](#)
- [オンデマンドキャパシティー予約 \(p. 431\)](#)

## インスタンスのライフサイクルの決定

インスタンスのライフサイクルは起動時に開始され、終了時に終了されます。選択する購入のオプションにより、インスタンスのライフサイクルに影響があります。たとえば、起動時に オンデマンドインスタンスが実行され、終了時に実行が終了されます。スポットインスタンスは、キャパシティーが利用可能で、上限料金がスポット料金より高い限り実行されます。スケジュールされたインスタンスは、スケジュールされた期間中に起動できます。Amazon EC2 はインスタンスを起動し、期間終了の 3 分前に終了します。

次の手順を使用して、インスタンスのライフサイクルを決定します。

コンソールを使用してインスタンスのライフサイクルを決定するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択します。
4. [Description] タブで [Tenancy] を見つけます。値が host の場合、インスタンスは Dedicated Host で実行されています。値が dedicated の場合、インスタンスは ハードウェア専有インスタンスで実行されています。
5. [Description] タブで [Lifecycle] を見つけます。値が spot の場合、インスタンスは スpotトインスタンスで実行されています。値が scheduled の場合、インスタンスはスケジュールされたインスタンスです。値が normal の場合、インスタンスは オンデマンドインスタンスまたは リザーブドインスタンスです。
6. (オプション) リザーブドインスタンスを購入し、適用されていることを確認するには、Amazon EC2 の使用状況レポートを確認できます。詳細については、「[Amazon EC2 使用状況レポート \(p. 1132\)](#)」を参照してください。

AWS CLIを使用してインスタンスのライフサイクルを決定するには

次の `describe-instances` コマンドを使用します。

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0
```

インスタンスが Dedicated Host で実行されている場合、出力には次の情報が含まれます。

```
"Tenancy": "host"
```

インスタンスが ハードウェア専有インスタンスの場合、出力には次の情報が含まれます。

```
"Tenancy": "dedicated"
```

インスタンスが スポットインスタンス の場合、出力には次の情報が含まれます。

```
"InstanceLifecycle": "spot"
```

インスタンスがスケジュールされたインスタンスの場合、出力には次の情報が含まれます。

```
"InstanceLifecycle": "scheduled"
```

それ以外の場合、出力には `InstanceLifecycle` は含まれません。

## オンデマンドインスタンス

オンデマンドインスタンス は、オンデマンドで使用するインスタンスです。そのライフサイクルを完全に制御でき、いつ起動、停止、休止、開始、再起動、または終了するかを決定できます。

オンデマンドインスタンス を購入するときに要求される長期的なコミットメントはありません。ご利用のオンデマンドインスタンス が `running` 状態になっている 秒数に対してのみお支払いいただきます。実行している オンデマンドインスタンス に対する秒あたりの料金は固定されており、[オンデマンド料金](#)ページに記載されています。

短期間、不規則なワークロードがあり中断できないアプリケーションには、オンデマンドインスタンス の使用をお勧めします。

オンデマンドインスタンスを大幅に削減するには、[AWS Savings Plans](#)、[スポットインスタンス](#) (p. 320)、または[リザーブドインスタンス](#) (p. 279)を使用してください。

### 目次

- [オンデマンドインスタンス の使用](#) (p. 276)
- [オンデマンドインスタンス の制限](#) (p. 277)
  - [必要な vCPU の数の計算](#) (p. 277)
  - [制限の引き上げのリクエスト](#) (p. 279)
  - [オンデマンドインスタンス の制限と使用量のモニタリング](#) (p. 279)
- [AWS のサービスの料金の照会](#) (p. 279)

## オンデマンドインスタンス の使用

次の方法で オンデマンドインスタンス を使用できます。

- [インスタンスの起動](#) (p. 448)
- [Linux インスタンスへの接続](#) (p. 505)
- [インスタンスの停止と起動](#) (p. 529)
- [Linux インスタンスの休止](#) (p. 532)
- [インスタンスの再起動](#) (p. 542)
- [インスタンスのリタイア](#) (p. 543)
- [インスタンスの終了](#) (p. 545)
- [インスタンスの復旧](#) (p. 551)
- [Amazon Linux インスタンスを設定する](#) (p. 552)
- [EC2 Linux インスタンスを特定する](#) (p. 623)

Amazon EC2 を初めて使用する場合は、[Amazon EC2 の使用を開始する方法 \(p. 1\)](#) を参照してください。

## オンデマンドインスタンス の制限

リージョンごと、AWS アカウントごとに実行できるオンデマンドインスタンスの数には制限があります。オンデマンドインスタンスの制限は、インスタンスタイプに関係なく、実行中のオンデマンドインスタンスで使用している仮想中央演算装置 (vCPU、virtual central processing unit) の数で管理されます。

オンデマンドインスタンス制限には、以下の表に示す 5 つがあります。各制限は、1 つ以上のインスタンスマリナーの vCPU 制限を指定します。さまざまなインスタンスマリナー、世代、およびサイズの詳細については、[Amazon EC2 インスタンスタイプ](#) を参照してください。

オンデマンドインスタンス 制限の名前	デフォルトの vCPU 制限
オンデマンド標準 (A, C, D, H, I, M, R, T, Z) インスタンスの実行	1152 vCPUs
オンデマンド F インスタンスの実行	128 vCPUs
オンデマンド G インスタンスの実行	128 vCPUs
オンデマンド Inf インスタンスの実行	128 vCPUs
オンデマンド P インスタンスの実行	128 vCPUs
オンデマンド X インスタンスの実行	128 vCPUs

### Note

新規の AWS アカウントにおける初期制限値は、ここに記載されている制限値より低い場合があります。

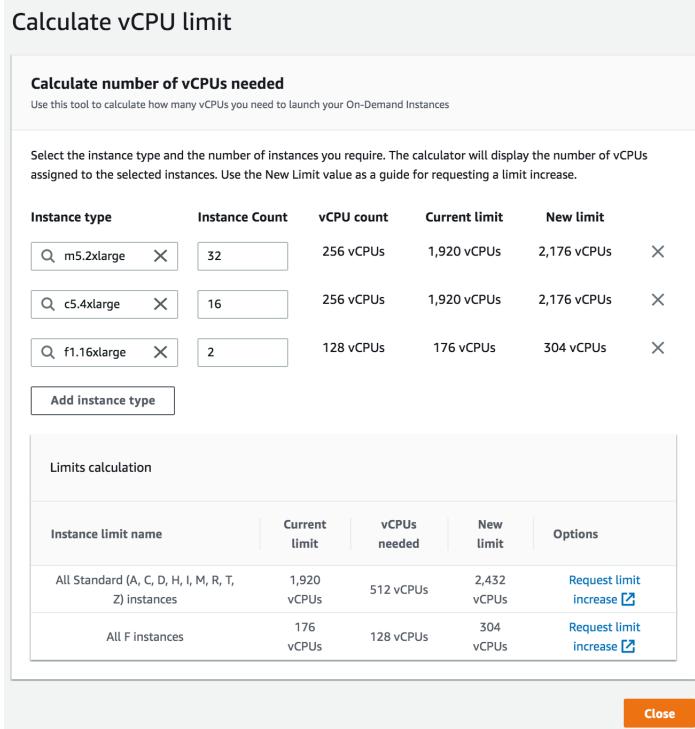
vCPU 制限では、変化するアプリケーションのニーズに合わせて、任意の組み合わせのインスタンスタイプを起動するために必要な vCPU の数で制限を利用できます。たとえば、256 vCPU のスタンダードインスタンス制限では、32 個の m5.2xlarge インスタンス ( $32 \times 8$  vCPU) または 16 個の c5.4xlarge インスタンス ( $16 \times 16$  vCPU)、または合計 256 個の vCPU になる任意のスタンダードインスタンスのタイプとサイズの組み合わせを起動できます。詳細については、「[EC2 オンデマンドインスタンス の制限](#)」を参照してください。

## 必要な vCPU の数の計算

vCPU 制限計算ツールを使用して、アプリケーションのニーズに必要な vCPU の数を求めることができます。

計算ツールを使用する際は、次の点に留意してください。このツールでは、現在の制限に達していることが前提となっています。[Instance Count (インスタンス数)] に入力する値は、現在の制限で許可されている値に、起動する必要があるインスタンスの数を足したものになります。計算ツールでは、現在の制限を [Instance Count (インスタンス数)] に追加することで新しい制限を求めます。

次のスクリーンショットは、vCPU 制限計算ツールを示しています。



Close

以下のコントロールと情報を表示して使用できます。

- [インスタンスタイプ] – vCPU 制限計算ツールに追加するインスタンスタイプ。
- [インスタンス数] – 選択したインスタンスタイプに必要なインスタンスの数。
- [vCPU count (vCPU 数)] – [Instance count (インスタンス数)] に対応する vCPU の数。
- [Current limit (現在の制限)] – インスタンスタイプが属する制限タイプの現在の制限。この制限は、同じ制限タイプのすべてのインスタンスタイプに適用されます。たとえば、前掲のスクリーンショットでは、m5.2xlarge と c5.4xlarge の現在の制限は 1,920 個の vCPU で、これは、すべてのスタンダードインスタンスの制限に属するすべてのインスタンスタイプに対する制限です。
- [New limit (新しい制限)] – vCPU 数で示される新しい制限 ([vCPU count (vCPU 数)] と [Current limit (現在の制限)] を足すことで計算されます)。
- [X] – 行を削除するには、[X] を削除します。
- [Add instance type (インスタンスタイプの追加)] – 計算ツールに別のインスタンスタイプを追加するには、[Add instance type (インスタンスタイプの追加)] を選択します。
- [Limits calculation (制限の計算)] – 現在の制限、必要な vCPU 数、および制限タイプに対する新しい制限を表示します。
  - [Instance limit name (インスタンス制限名)] – 選択したインスタンスタイプに対応する制限タイプ。
  - [Current limit (現在の制限)] – 制限タイプに対する現在の制限。
  - [vCPUs needed (必要な vCPU 数)] – [Instance count (インスタンス数)] に指定されているインスタンスの数に対応する vCPU の数。すべてのスタンダードインスタンス制限タイプについて、この制限タイプのすべてのインスタンスタイプに対する [vCPU 数] の値を追加することで、必要な vCPU 数が計算されます。
  - [New limit (新しい制限)] – 新しい制限は、[Current limit (現在の制限)] および [vCPUs needed (必要な vCPU 数)] を足すことで計算されます。
  - [オプション] – [制限引き上げをリクエスト] を選択して、対応する制限タイプの制限引き上げをリクエストします。

### 必要な vCPU の数を計算するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、リージョンを選択します。
3. 左側にあるナビゲータから、[制限] を選択します。
4. [Calculate vCPU limit (vCPU 制限の計算)] を選択します。
5. [Add instance type (インスタンスタイプの追加)] を選択し、必要なインスタンスタイプを選択して、必要なインスタンスの数を指定します。さらにインスタンスタイプを追加するには、もう一度 [Add instance type (インスタンスタイプの追加)] を選択します。
6. 必要な新しい制限の [Limits calculation (制限の計算)] を表示します。
7. 計算ツールの使用を完了したら、[閉じる] を選択します。

### 制限の引き上げのリクエスト

各 オンデマンドインスタンス 制限タイプに対する制限の引き上げは、[Limits \(制限\) ページ](#)または Amazon EC2 コンソールの vCPU 制限計算ツールからリクエストできます。ユースケースを使用して、AWS サポートセンターの [制限引き上げフォーム](#)の必須フィールドを記入します。[Primary Instance Type (プライマリインスタンスタイプ)] で、vCPU 制限計算ツールの [Instance limit name (インスタンス制限名)] に対応する制限タイプを選択します。新しい制限の値には、vCPU 制限計算ツールの [新しい制限] 列に表示される値を使用します。制限引き上げのリクエストの詳細については、「[Amazon EC2 サービスの制限 \(p. 1130\)](#)」を参照してください。

### オンデマンドインスタンス の制限と使用量のモニタリング

オンデマンドインスタンス の制限を確認して管理するには、Amazon EC2 コンソールの [制限ページ](#)、Service Quotas コンソールの [Amazon EC2 サービスのクォータページ](#)、および AWS Trusted Advisor コンソールの [サービスの制限ページ](#)を参照できます。詳細については、「[Amazon EC2 サービスの制限 \(p. 1130\)](#)」(Linux インスタンス用 Amazon EC2 ユーザーガイド)、「[サービスクォータの表示](#)」(サービスクォータユーザーガイド)、および [AWS Trusted Advisor](#) を参照してください。

Amazon CloudWatch のメトリクス統合では、制限に対して EC2 の使用量をモニタリングできます。制限に近づいたときに警告を発するようにアラームを設定することもできます。詳細については、「[Amazon CloudWatch アラームの使用](#)」(サービスクォータユーザーガイド) を参照してください。

### AWS のサービスの料金の照会

Price List Service API または AWS Price List API を使用して、オンデマンドインスタンス の料金を照会できます。詳細については、AWS Billing and Cost Management ユーザーガイドの「[AWS Price List API の使用](#)」を参照してください。

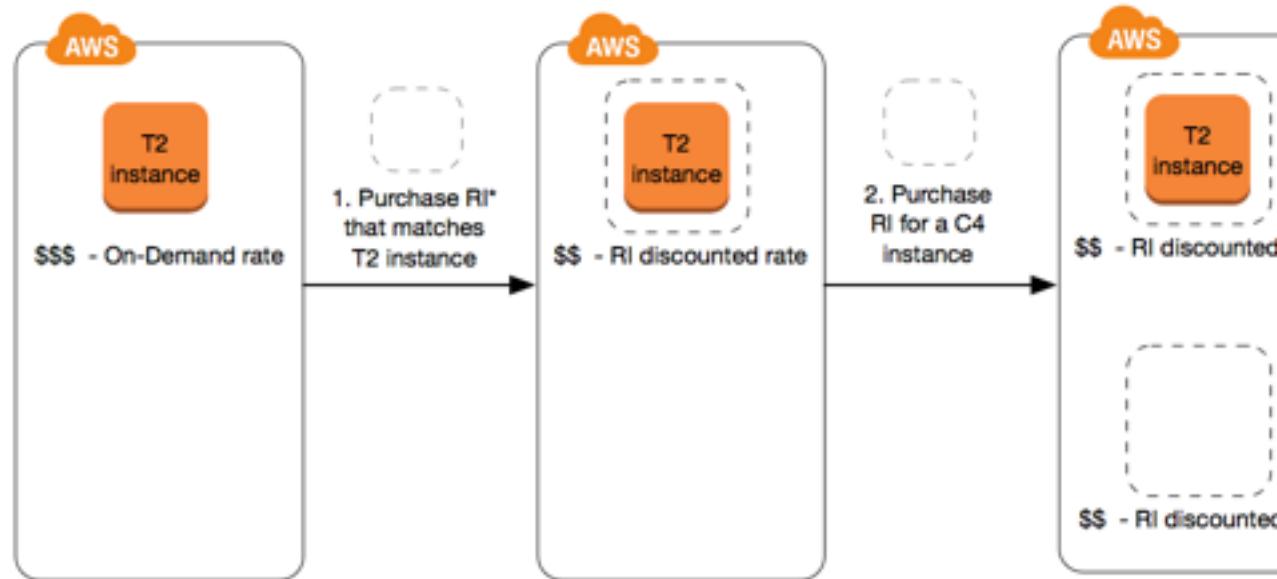
### リザーブドインスタンス

リザーブドインスタンスでは、オンデマンドインスタンスの料金と比べて Amazon EC2 コストの大額な割引を受けられます。リザーブドインスタンスは物理インスタンスではありませんが、請求の割引はアカウントでのオンデマンドインスタンスの使用に適用されます。請求割引のメリットを得るには、これらのオンデマンドインスタンスは、インスタンスタイプやリージョンなどの特定の属性に一致する必要があります。

Savings Plans でも、オンデマンドインスタンスの料金と比べて Amazon EC2 コストの大額な割引を受けられます。Savings Plans では、1 時間に 1 USD 単位で一定の使用量を守ります。これによって、特定のインスタンス設定を使用することにコミットせずに、ニーズに最も適したインスタンス設定を使用して、コストを削減し続ける柔軟性が得られます。詳細については、「[AWS Savings Plans ユーザーガイド](#)」を参照してください。

## リザーブドインスタンスの概要

次の図では、リザーブドインスタンス の購入と使用の基本的な概要を示します。



\*RI = Reserved Instance

このシナリオでは、現在オンデマンドレートを支払っているアカウントの オンデマンドインスタンス (T2) を実行しています。実行しているインスタンスの属性を一致する リザーブドインスタンス を購入すると、料金上の利点が即時適用されます。次に、C4 インスタンスに リザーブドインスタンス を購入します。この リザーブドインスタンス に属性が一致するアカウントで実行しているインスタンスはありません。この最終ステップでは、C4 リザーブドインスタンス の属性と一致するインスタンスを起動すると、料金上の利点が独自適用されます。

## リザーブドインスタンス 料金を決定するキー変数

リザーブドインスタンス 料金は、次のキー変数によって決まります。

### インスタンスの属性

リザーブドインスタンス には、その料金を決める 4 つのインスタンス属性があります。また、これらの属性は リザーブドインスタンス がアカウント内で実行中のインスタンスにどのように適用されるかを決定します。

- ・ インスタンスタイプ: たとえば、`m4.large`。これは、インスタンスマージャー (`m4`) とインスタンスサイズ (`large`) で構成されます。
- ・ リージョン: リザーブドインスタンスが購入されているリージョン。
- ・ テナント: インスタンスが共有 (デフォルト) または単一のテナント (専用) のハードウェアで実行されるかについて。詳細については、「[ハードウェア専有インスタンス \(p. 425\)](#)」を参照してください。
- ・ プラットフォーム: オペレーティング システム。たとえば、Windows や Linux/Unix。詳細については、「[プラットフォームの選択 \(p. 294\)](#)」を参照してください。

リザーブドインスタンス は自動的に更新されません。有効期限が切れても、引き続き EC2 インスタンスを使用できますが、オンデマンド価格が課金されます。上記の例では、T2 および C4 インスタンスを対象

とする リザーブドインスタンス の期限が切れた場合、インスタンスが終了するまでオンデマンドレートの支払いに戻るか、あるいはインスタンスの属性に一致する新しい リザーブドインスタンス を購入します。

## コミットメント期間

1 年あるいは 3 年のコミットメントで リザーブドインスタンス を購入することができます。3 年のコミットメントには大幅な割引が提供されます。

- 1 年: 1 年は 31536000 秒 (365 日) として定義されます。
- 3 年: 3 年は 94608000 秒 (1095 日) として定義されます。

## 支払いオプション

リザーブドインスタンス では次の支払いオプションが用意されています。

- すべて前払い: 期間の開始時に全額が支払われ、使用時間数に関係なく、残りの期間に他のコストや追加時間課金は生じません。
- 一部前払い: 料金の一部を前払いする必要があります、期間内の残りの時間は、リザーブドインスタンス が使用されたかどうかにかかわらず、割引された時間料金で請求されます。
- 前払いなし: リザーブドインスタンス が使用されたかどうかにかかわらず、期間内のすべての時間は割引時間料金での請求となります。前払い料金は必要ではありません。

### Note

前払いなしの リザーブドインスタンス は、予約の全期間について毎月支払いを行う契約義務に基づいています。そのため、前払いなしの リザーブドインスタンス を購入するには、請求履歴に問題がないことが必須となります。

一般的には、リザーブドインスタンス の前払い額を高く設定するほど、より多くの費用を節約できます。また、より安価で短期間のサードパーティの販売者提供の リザーブドインスタンス を リザーブドインスタンスマーケットプレイス で見つけることもできます。詳細については、「[リザーブドインスタンスマーケットプレイス \(p. 299\)](#)」を参照してください。

## 提供クラス

コンピューティングに変更が必要な場合、提供クラスによって、リザーブドインスタンス を変更または交換することができます。

- スタンダード: 最大の割引を提供しますが、変更のみを行うことができます。
- コンバータブル: スタンダード リザーブドインスタンス より少ない割引ですが、異なるインスタンス属性を使用する別のコンバータブル リザーブドインスタンス と交換できます。コンバータブル リザーブドインスタンス は変更することもできます。

詳細については、「[リザーブドインスタンス のタイプ \(提供しているクラス\) \(p. 283\)](#)」を参照してください。

リザーブドインスタンス を購入した後で購入をキャンセルすることはできません。ただし、ユーザーのニーズが変更した場合、リザーブドインスタンス を [変更 \(p. 306\)](#)、[交換 \(p. 313\)](#)、[売却 \(p. 299\)](#) できる場合もあります。

料金の詳細については、「[Amazon EC2 リザーブドインスタンス料金表](#)」を参照してください。

## リザーブドインスタンス の制限

1か月当たりに購入できる リザーブドインスタンス の数は制限されています。リージョンごとに 1か月当たり 20 個の [リージョン \(p. 284\)](#) リザーブドインスタンス に追加して、アベイラビリティーゾーンごとに 1か月当たり 20 個の [ゾーン \(p. 284\)](#) リザーブドインスタンス を購入できます。

たとえば、リージョンに 3 つのアベイラビリティーゾーンがある場合、リザーブドインスタンス の制限は 1か月当たり 80 です。つまり、そのリージョンのリージョン リザーブドインスタンス 20 と、3 つのアベイラビリティーゾーンそれぞれで 20 のゾーン リザーブドインスタンス ( $20 \times 3 = 60$ ) です。

リージョン リザーブドインスタンス では、オンデマンドインスタンス の実行に割引が適用されます。デフォルトの オンデマンドインスタンス の制限は 20 です。リージョン オンデマンドインスタンス を購入すると、リザーブドインスタンス の実行制限を超えることはできません。たとえば、すでに オンデマンドインスタンス を 20 回実行していて、20 のリージョン リザーブドインスタンス を購入した場合、20 のリージョン リザーブドインスタンス には 20 回の オンデマンドインスタンス の実行に割引が適用されます。さらに多くのリージョン リザーブドインスタンス を購入した場合は、オンデマンドインスタンス の制限に達しているため、さらにインスタンスを起動することはできません。

リージョン リザーブドインスタンス を購入する前に、オンデマンドインスタンス の制限数が所有する予定のリージョン リザーブドインスタンス の数に一致するかそれを超えることを確認してください。必要に応じて、さらにリージョン リザーブドインスタンス を購入する前に オンデマンドインスタンス の制限数の増加を依頼してください。

ゾーン リザーブドインスタンス (特定のアベイラビリティーゾーンで購入された リザーブドインスタンス) は、容量の予約と割引を提供します。ゾーン リザーブドインスタンス を購入することで、実行中の オンデマンドインスタンス の制限を超えることができます。たとえば、すでに 20 の オンデマンドインスタンス を実行していて、20 のゾーン リザーブドインスタンス を購入した場合は、ゾーン リザーブドインスタンス の仕様に一致する 20 の オンデマンドインスタンス をさらに起動して、合計 40 実行インスタンスを実行できます。

Amazon EC2 コンソールで制限情報を確認できます。詳細については、「[現在の制限を表示する \(p. 1131\)](#)」を参照してください。

## リージョンおよびゾーン リザーブドインスタンス (スコープ)

リザーブドインスタンス の購入時に リザーブドインスタンス のスコープを決定します。スコープは、リージョンあるいはゾーンのいずれかになります。

- リージョナル: リージョン用に リザーブドインスタンス を購入する場合、これはリージョナル リザーブドインスタンス と呼ばれます。
- ゾーン: 特定のアベイラビリティーゾーン用に リザーブドインスタンス を購入する場合、これはゾーン リザーブドインスタンス と呼ばれます。

## リージョンとゾーン の リザーブドインスタンス の違い

次のテーブルでは、リージョン リザーブドインスタンス とゾーン リザーブドインスタンス の主な違いをいくつか示しています。

	リージョン リザーブドインスタンス	ゾーン リザーブドインスタンス
アベイラビリティーゾーンの柔軟性	指定するリージョン内のすべてのアベイラビリティーゾーンにおけるインスタンスの使用に対して、リザーブドインスタンス 割引が適用されます。	アベイラビリティーゾーンの柔軟性なし — リザーブドインスタンス 割引は、指定したアベイラビリティーゾーン内ののみのインス

	リージョン リザーブドインスタンス	ゾーン リザーブドインスタンス
		タンスの使用に対して適用されます。
キャパシティの予約	キャパシティの予約なし — リージョン リザーブドインスタンスでは、キャパシティの予約が提供されません。	ゾーン リザーブドインスタンスは、指定したアベイラビリティーゾーンでキャパシティの予約を提供します。
インスタンスサイズの柔軟性	インスタンスファミリー内のインスタンスの使用に対して、サイズを問わず、リザーブドインスタンス割引が適用されます。Amazon Linux/Unix リザーブドインスタンスのデフォルトテナントのみでサポートされます。詳細については、「 <a href="#">正規化係数によって決定されたインスタンスサイズの柔軟性 (p. 284)</a> 」を参照してください。	インスタンスサイズの柔軟性なし — リザーブドインスタンス割引は、指定されたインスタンスタイプとサイズにおけるインスタンスの使用に対してのみ適用されます。

詳細な説明と例については、「[リザーブドインスタンス がどのように適用されるか \(p. 284\)](#)」を参照してください。

## リザーブドインスタンス のタイプ (提供しているクラス)

リザーブドインスタンス を購入するとき、スタンダードあるいはコンバータブルの提供クラスから選択できます。リザーブドインスタンス は、期間内で単一のインスタンスタイプ、プラットフォーム、スコープ、テナントに適用されます。コンピューティングに変更が必要な場合、提供クラスによって、リザーブドインスタンス を変更または交換することができます。提供クラスには追加の制約あるいは制限がある場合があります。

以下に、スタンダード提供クラスとコンバータブル提供クラスの違いを示します。

スタンダード リザーブドインスタンス	コンバータブルリザーブドインスタンス
インスタンスサイズなどの一部の属性は期間中に変更できますが、インスタンスファミリーは変更できません。スタンダードリザーブドインスタンス は交換できず、変更のみができます。詳細については、「 <a href="#">リザーブドインスタンス の変更 (p. 306)</a> 」を参照してください。	期間内で、インスタンスファミリー、インスタンスタイプ、プラットフォーム、スコープやテナントなどの新しい属性の別のコンバータブルリザーブドインスタンス に交換することができます。詳細については、「 <a href="#">コンバータブルリザーブドインスタンス の交換 (p. 313)</a> 」を参照してください。また、コンバータブルリザーブドインスタンス の一部の属性を変更することもできます。詳細については、「 <a href="#">リザーブドインスタンス の変更 (p. 306)</a> 」を参照してください。
リザーブドインスタンスマーケットプレイス で販売できます。	リザーブドインスタンスマーケットプレイス で販売できません。

スタンダードと コンバータブルリザーブドインスタンス は、特定のアベイラビリティーゾーン (ゾーン リザーブドインスタンス) 内のインスタンス、またはリージョン (リージョン リザーブドインスタンス) 内

のインスタンスに適用するために購入できます。詳細な説明と例については、「[リザーブドインスタンスがどのように適用されるか \(p. 284\)](#)」を参照してください。

日、週、または月ベースで処理能力の予約購入を繰り返す場合は、スケジュールされたリザーブドインスタンスで、ニーズが満たされている可能性があります。詳細については、「[スケジュールされたリザーブドインスタンス \(p. 317\)](#)」を参照してください。

## リザーブドインスタンスがどのように適用されるか

リザーブドインスタンスを購入し、リザーブドインスタンスの仕様と一致するインスタンスをすでに実行している場合、料金上の利点は即時適用されます。インスタンスを再起動する必要はありません。使用可能な実行中のインスタンスが存在しない場合、インスタンスを起動して、リザーブドインスタンスに指定した条件と一致していることを確認します。詳細については、「[リザーブドインスタンスを使用する \(p. 299\)](#)」を参照してください。

リザーブドインスタンスは、提供タイプ(スタンダードまたはコンバティブル)に関係なく適用方法は同じであり、属性が一致する実行中のオンデマンドインスタンスに自動的に適用されます。

## ゾーンリザーブドインスタンスがどのように適用されるか

特定のアベイラビリティゾーンに割り当てられたリザーブドインスタンスでは、そのアベイラビリティゾーンの一一致するインスタンスの使用に対してリザーブドインスタンス割引が適用されます。たとえば、us-east-1a のアベイラビリティゾーンで、デフォルトテナントの c4.xlarge Linux/Unix スタンダード リザーブドインスタンスを 2 つ購入すると、us-east-1a のアベイラビリティゾーンで実行している 2 つまでのデフォルトテナントの c4.xlarge Linux/Unix インスタンスでリザーブドインスタンス割引を利用できます。実行中のインスタンスの属性(テナント、プラットフォーム、アベイラビリティゾーン、インスタンスタイプ、およびインスタンスサイズ)は、リザーブドインスタンスの属性と一致する必要があります。

## リージョンリザーブドインスタンスがどのように適用されるか

リージョンリザーブドインスタンスはリージョンでの購入となり、アベイラビリティゾーンの柔軟性を提供します。このリージョン内のすべてのアベイラビリティゾーンにおけるインスタンスの使用に対して、リザーブドインスタンス割引が適用されます。

また、リージョンリザーブドインスタンスは、インスタンスマトリクス内のインスタンスの使用に対してサイズを問わずにリザーブドインスタンス割引が適用される、インスタンスサイズの柔軟性も提供します。

### インスタンスサイズの柔軟性における制限

インスタンスサイズの柔軟性は、次のリザーブドインスタンスには適用されません。

- 特定のアベイラビリティゾーン(ゾーンリザーブドインスタンス)用に購入されたリザーブドインスタンス
- 専有テナントを使用するリザーブドインスタンス
- Windows Server、Windows Server with SQL Standard、Windows Server with SQL Server Enterprise、Windows Server with SQL Server Web、RHEL、および SLES 用のリザーブドインスタンス
- G4 インスタンス用のリザーブドインスタンス

### 正規化係数によって決定されたインスタンスサイズの柔軟性

インスタンスサイズの柔軟性は、インスタンスサイズの正規化係数によって決定されます。割引は、リージョン内のアベイラビリティゾーンで、予約したインスタンスサイズによって、同じインスタンスマトリクス

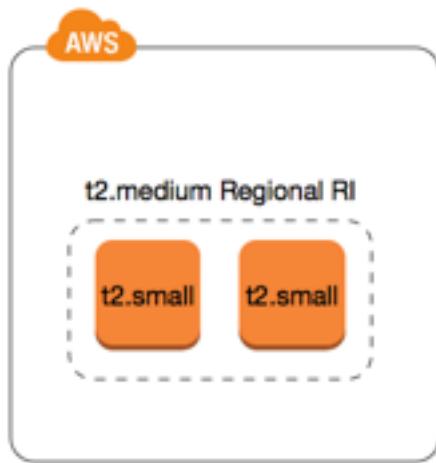
ミラーで実行中のインスタンスに完全または部分的に適用されます。一致する必要がある属性は、インスタンスファミリー、テナント、プラットフォームのみです。

インスタンスサイズの柔軟性は、インスタンスファミリー内の最も小さいインスタンスサイズから大きいインスタンスサイズへ、正規化係数に基づいて適用されます。

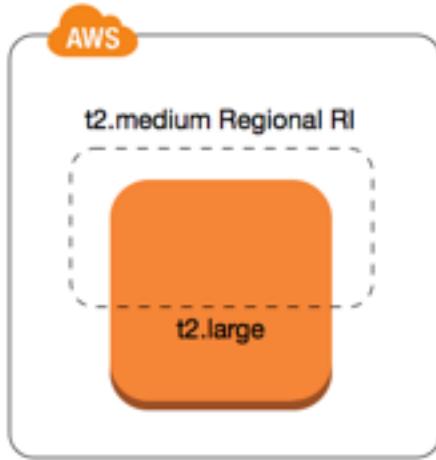
次の表は、インスタンスファミリー内のさまざまなサイズおよび対応する 1 時間あたりの正規化係数の一覧です。このスケールを使用して、リザーブドインスタンス の割引料金をインスタンスファミリーの正規化された使用に適用します。

インスタンスサイズ	正規化係数
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
3xlarge	24
4xlarge	32
6xlarge	48
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

たとえば、t2.medium インスタンスには 2 の正規化係数があります。米国東部（バージニア北部）で t2.medium デフォルトテナント Amazon Linux/Unix リザーブドインスタンス を購入し、このリージョンのアカウントで 2 つの t2.small インスタンスを実行している場合、料金上の利点はどちらのインスタンスにも完全に適用されます。



または、米国東部（バージニア北部）リージョンのアカウントで実行している 1 つの t2.large インスタンスがある場合、料金上の利点はこのインスタンスの使用の 50% に適用されます。



正規化係数は、リザーブドインスタンス の変更時にも適用されます。詳細については、「[リザーブドインスタンス の変更 \(p. 306\)](#)」を参照してください。

#### ペアメタルインスタンスの正規化係数

また、インスタンスサイズの柔軟性は、インスタンスマルチー内のペアメタルインスタンスにも適用されます。ペアメタルインスタンスで共有するテナントを使用したリージョンの Amazon Linux/Unix リザーブドインスタンスがある場合、同じインスタンスマルチー内で リザーブドインスタンス 割引の特典を受けることができます。逆の場合も同様です。同じマルチー内のインスタンスでペアメタルインスタンスとしてテナントを共有している、リージョン Amazon Linux/Unix リザーブドインスタンスがある場合、ペアメタルインスタンスで リザーブドインスタンス 割引の特典を受けることができます。

ペアメタルインスタンスは、同じインスタンスマルチーないの最大のインスタンスと同じサイズです。たとえば、i3.metal は i3.16xlarge と同じサイズであるため、同じ正規化係数があります。

#### Note

.metal インスタンスサイズには、単一の正規化係数がありません。特定のインスタンスマルチーに基づいて異なります。

ペアメタルインスタンスサイズ	正規化係数
c5.metal	192
i3.metal	128
r5.metal	192
r5d.metal	192
z1d.metal	96
m5.metal	192
m5d.metal	192

たとえば、1つの i3.metal インスタンスには 128 の正規化係数があります。米国東部（バージニア北部）で i3.metal デフォルトテナント Amazon Linux/Unix リザーブドインスタンスを購入した場合、料金上のメリットは次のようにになります。

- そのリージョン内のアカウントで実行している 1 つの i3.16xlarge がある場合、料金上のメリットは i3.16xlarge インスタンス全体に適用されます (i3.16xlarge 正規化係数 = 128)。
- あるいは、そのリージョン内のアカウントで実行している 2 つの i3.8xlarge がある場合、料金上のメリットは両方の i3.8xlarge インスタンス全体に適用されます (i3.8xlarge 正規化係数 = 64)。
- あるいは、そのリージョン内のアカウントで実行している 4 つの i3.4xlarge がある場合、料金上のメリットは 4 つの i3.4xlarge インスタンス全体に適用されます (i3.4xlarge 正規化係数 = 32)。

逆の場合も同様です。たとえば、米国東部（バージニア北部）で 2 つの i3.8xlarge デフォルトテナント Amazon Linux/Unix リザーブドインスタンスを購入し、そのリージョンで実行している 1 つの i3.metal インスタンスがある場合、料金上のメリットは i3.metal インスタンス全体に適用されます。

## リザーブドインスタンスの適用例

リザーブドインスタンスの適用方法を以下のシナリオで示します。

### Example シナリオ 1: 単一アカウントのリザーブドインスタンス

アカウント A で以下のオンデマンドインスタンスを実行しているとします。

- us-east-1a アベイラビリティーゾーンで 4 つのデフォルトテナントの m3.large Linux インスタンス
- us-east-1b アベイラビリティーゾーンで 2 つのデフォルトテナントの m4.xlarge Amazon Linux インスタンス
- us-east-1c アベイラビリティーゾーンで 1 つのデフォルトテナントの c4.xlarge Amazon Linux インスタンス

アカウント A で以下のリザーブドインスタンスを購入するとします。

- us-east-1a アベイラビリティーゾーンで 4 つのデフォルトテナントの m3.large Linux リザーブドインスタンス (キャパシティーの予約あり)
- us-east-1 リージョンで 4 つのデフォルトテナントの m4.large Amazon Linux リザーブドインスタンス
- us-east-1 リージョンで 1 つのデフォルトテナントの c4.large Amazon Linux リザーブドインスタンス

リザーブドインスタンスの利点は以下のように適用されます。

- 4 つの m3.large リージョン リザーブドインスタンス の割引とキャパシティーの予約は、属性 (インスタンスサイズ、リージョン、プラットフォーム、テナンシー) が一致する 4 つの m3.large インスタンスによって使用されます。
- m4.large リージョン リザーブドインスタンス は、デフォルトテナンシーの Amazon Linux リザーブドインスタンス であるため、アベイラビリティーゾーンおよびインスタンスサイズの柔軟性を提供します。

1 つの m4.large は、4 つの正規化された単位/時間に相当します。

4 つの m4.large リージョン リザーブドインスタンス を購入したので、合計で 16 の正規化された単位/時間 (4x4) に相当します。アカウント A で実行している 2 つの m4.xlarge インスタンスは、16 の正規化された単位/時間 (2x8) に相当します。この場合、4 つの m4.large リージョン リザーブドインスタンス は、2 つの m4.xlarge インスタンスの使用時間全体に対して請求のメリットを提供します。

- us-east-1 の c4.large リージョン リザーブドインスタンス は、デフォルトテナンシーのリージョンの Amazon Linux リザーブドインスタンス であるため、アベイラビリティーゾーンおよびインスタンスサイズの柔軟性を提供し、c4.xlarge インスタンスに適用されます。c4.large インスタンスは 4 つの正規化された単位/時間に相当し、c4.xlarge インスタンスは 8 つの正規化された単位/時間に相当します。

この場合、c4.large リージョン リザーブドインスタンス は、c4.xlarge の使用の一部に対してメリットを提供します。これは、この c4.large リザーブドインスタンス は 4 つの正規化された単位/時間に相当しますが、c4.xlarge インスタンスは 8 つの正規化された単位/時間を必要とするためです。したがって、c4.large リザーブドインスタンス の請求割引は、c4.xlarge の使用の 50% に適用されます。c4.xlarge の残りの使用はオンデマンド価格で課金されます。

#### Example シナリオ 2: 連結アカウントのリージョン リザーブドインスタンス

リザーブドインスタンス は、最初に購入アカウント内の使用に適用され、次に組織内の他のアカウントの該当する使用に適用されます。詳細については、「[リザーブドインスタンス および一括請求 \(コンソリデータイッドビリング\)](#) (p. 291)」を参照してください。サイズの柔軟性を提供するリージョン リザーブドインスタンス の場合、インスタンスマリーニー内のインスタンスサイズに関係なく、利点がインスタンスに適用されます。

アカウント A (購入しているアカウント) で以下の オンデマンドインスタンス を実行しているとします。

- us-east-1a アベイラビリティーゾーンで 2 つのデフォルトテナンシーの m4.xlarge Linux インスタンス
- us-east-1b アベイラビリティーゾーンで 1 つのデフォルトテナンシーの m4.2xlarge Linux インスタンス
- us-east-1a アベイラビリティーゾーンで 2 つのデフォルトテナンシーの c4.xlarge Linux インスタンス
- us-east-1b アベイラビリティーゾーンで 1 つのデフォルトテナンシーの c4.2xlarge Linux インスタンス

別のお客様は、アカウント B—(連結アカウント) で以下の オンデマンドインスタンス を実行しています。

- us-east-1a アベイラビリティーゾーンで 2 つのデフォルトテナンシーの m4.xlarge Linux インスタンス

アカウント A で以下のリージョン リザーブドインスタンス を購入するとします。

- us-east-1 リージョンで 4 つのデフォルトテナンシーの m4.xlarge Linux リザーブドインスタンス
- us-east-1 リージョンで 2 つのデフォルトテナンシーの c4.xlarge Linux リザーブドインスタンス

リージョンリザーブドインスタンスの利点は以下のように適用されます。

- 4つの m4.xlarge リザーブドインスタンス の割引は、アカウント A の 2つの m4.xlarge インスタンスと 1つの m4.2xlarge インスタンスによって使用されます。3つのインスタンスすべてにおいて、属性(インスタンスマニマー、リージョン、プラットフォーム、テナンシー)が一致しています。アカウント B(リンクされたアカウント)にリザーブドインスタンス にも一致する 2つの m4.xlarge がある場合でも、この割引は購入したアカウント(アカウント A)内のインスタンスにまず適用されます。このリザーブドインスタンス はリージョン リザーブドインスタンス であるため、キャパシティの予約はありません。
- c4.xlarge リザーブドインスタンス インスタンスよりもインスタンスサイズが小さいため、2つの c4.xlarge リザーブドインスタンスの割引は、2つの c4.2xlarge インスタンスに適用されます。このリザーブドインスタンス はリージョン リザーブドインスタンス であるため、キャパシティの予約はありません。

#### Example シナリオ 3: 連結アカウントのゾーン リザーブドインスタンス

通常、アカウントで所有されている リザーブドインスタンス が、そのアカウントでの使用に最初に適用されます。ただし、組織の他のアカウントに特定のアベイラビリティーゾーン(ゾーン リザーブドインスタンス)の未使用的 リザーブドインスタンス がある場合は、これらがアカウントで所有されているリージョン リザーブドインスタンスより先に適用されます。これは、リザーブドインスタンス の使用率を最大限に高めて請求額を下げるための処置です。請求の目的では、組織内のすべてのアカウントが 1つのアカウントとして扱われます。次の例が、この説明に役立つ場合があります。

アカウント A(購入しているアカウント) で以下の オンデマンドインスタンス を実行しているとします。

- us-east-1a アベイラビリティーゾーンで 1つのデフォルトテナンシーの m4.xlarge Linux インスタンス

お客様は、別の連結アカウント B で以下の オンデマンドインスタンス を実行しています。

- us-east-1b アベイラビリティーゾーンで 1つのデフォルトテナンシーの m4.xlarge Linux インスタンス

アカウント A で以下のリージョン リザーブドインスタンス を購入するとします。

- us-east-1 リージョンで 1つのデフォルトテナンシーの m4.xlarge Linux リザーブドインスタンス

ユーザーが連結アカウント C で以下のゾーン リザーブドインスタンス も購入するとします。

- us-east-1a アベイラビリティーゾーンで 1つのデフォルトテナンシーの m4.xlarge Linux リザーブドインスタンス

リザーブドインスタンス の利点は以下のように適用されます。

- アカウント C で所有されている m4.xlarge ゾーン リザーブドインスタンス の割引はアカウント A の m4.xlarge の使用に適用されます。
- アカウント A で所有されている m4.xlarge リージョン リザーブドインスタンス の割引はアカウント B の m4.xlarge の使用に適用されます。
- アカウント A で所有されている リージョン リザーブドインスタンス は、最初にアカウント A での使用に適用されます。アカウント C で所有されているゾーン リザーブドインスタンス は使用されず、アカウント B での使用はオンデマンド価格で課金されます。

詳細については、「[Billing and Cost Management レポートの リザーブドインスタンス](#)」を参照してください。

## 課金の仕組み

すべての リザーブドインスタンス の料金は、オンデマンドインスタンスの料金から割引された額になります。リザーブドインスタンス では、実際の使用に関係なく、全期間の料金をお支払いいただきます。リザーブドインスタンス に特定された支払いオプション (p. 281)によって、リザーブドインスタンス の支払い方法を前払い、一部前払い、月ごとから選択できます。

リザーブドインスタンス 期間が終了すると、EC2 インスタンスの使用についてオンデマンド価格が課金されます。リザーブドインスタンス の購入を最大 3 年先までキューに入れることができます。これにより、サービスを切れ目なく利用できます。詳細については、「[購入をキューに入れる \(p. 294\)](#)」を参照してください。

AWS の無料利用枠は、新規の AWS アカウントで使用できます。AWS の無料利用枠を使用して Amazon EC2 インスタンスを実行している場合、リザーブドインスタンス を購入すると、そのリザーブドインスタンス は標準の料金ガイドラインに基づいて課金されます。詳細については、「[AWS 無料利用枠](#)」を参照してください。

### コンテンツ

- [使用料の請求 \(p. 290\)](#)
- [請求の表示 \(p. 291\)](#)
- [リザーブドインスタンス および一括請求 \(コンソリデータイッドビリング\) \(p. 291\)](#)
- [リザーブドインスタンス 割引料金範囲 \(p. 291\)](#)

## 使用料の請求

リザーブドインスタンス は、選択した期間内の 1 時間ごとに請求されます。インスタンスが実行中であるかどうかは関係しません。各時間は、標準の 24 時間の時計の正時 (毎時ゼロ分ゼロ秒) に開始します。たとえば、1:00:00 ~ 1:59:59 が 1 時間です。インスタンステータスの詳細については、「[インスタンスのライフサイクル \(p. 443\)](#)」を参照してください。

リザーブドインスタンス の料金上の特典は秒単位課金で実行中のインスタンスに適用されます。秒単位の請求は、オープンソースの Linux ディストリビューション (Amazon Linux、Ubuntu など) を使用するインスタンスで使用できます。時間単位の請求は、Linux の商用ディストリビューション (Red Hat Enterprise Linux、SUSE Linux Enterprise Server など) で使用できます。

リザーブドインスタンス の料金上の利点は、1 時間当たり最大 3600 秒 (1 時間) のインスタンス使用に適用できます。複数のインスタンスを同時に実行できますが、リザーブドインスタンス 割引の特典を受けることができるのは 1 時間あたり合計 3600 秒までです。インスタンスの使用が 1 時間あたり 3600 秒を超えると、オンデマンドレートで課金されます。

たとえば、1 つの m4.xlarge リザーブドインスタンス を購入し、4 つの m4.xlarge インスタンスを 1 時間に同時に実行する場合、1 つのインスタンスには リザーブドインスタンス の 1 時間分の使用料が、他の 3 つのインスタンスにはオンデマンドの 3 時間分の使用料が課金されます。

一方、1 つの m4.xlarge リザーブドインスタンス を購入し、4 つの m4.xlarge インスタンスを同じ 1 時間に内にそれぞれ 15 分 (900 秒) ずつ実行した場合、インスタンスの合計実行時間は 1 時間となり、リザーブドインスタンス の使用料が 1 時間分課金されるだけで、オンデマンドの使用料は課金されません。

	1:00	1:15	1:30	1:45
Instance 1	■			
Instance 2		■		
Instance 3			■	
Instance 4				■

複数の対象インスタンスが同時に実行されている場合、リザーブドインスタンス の料金上の特典は 1 時間あたり 3600 秒まで同時にすべてのインスタンスに適用されます。3600 秒を超えると、オンデマンド価格が適用されます。



[Billing and Cost Management](#) コンソール上の [Cost Explorer] は、オンデマンドインスタンスの実行に対する節約を分析することができます。リザーブドインスタンスについてのよくある質問には、表示価格の計算例があります。

AWS アカウントを解約すると、リソースのオンデマンド課金は停止します。ただし、アカウントにリザーブドインスタンスがある場合には、その期限が切れるまで続けて課金されます。

## 請求の表示

[AWS Billing and Cost Management](#) コンソールで、アカウントへの請求および料金を確認できます。

- ・[ダッシュボード] には、アカウント利用料の概要が表示されます。
- ・[請求書] ページの [明細] で、[Elastic Compute Cloud] セクションと リザーブドインスタンスの請求情報を取得するリージョンを展開します。

請求額をオンラインで表示することも、CSV ファイルとしてダウンロードすることもできます。

AWS のコストと使用状況リポートを使用して、リザーブドインスタンスの使用を追跡することもできます。詳細については、『AWS Billing and Cost Management ユーザーガイド』のコストと使用状況レポートの「リザーブドインスタンス」を参照してください。

## リザーブドインスタンス および一括請求 (コンソリデータイドビリング)

購入アカウントが、1つの一括請求の支払いアカウントに請求される一連のアカウントの一部である場合、リザーブドインスタンスの料金面でのメリットを広範囲に利用できます。すべてのメンバーアカウントのインスタンス使用量が月次で支払いアカウントに集約されます。さまざまな役割を持つチームやグループがある企業にとって特に便利です。したがって、請求書の計算には通常のリザーブドインスタンスのロジックが適用されます。詳細については、『AWS Organizations ユーザーガイド』の「Consolidated Billing and AWS Organizations」を参照してください。

リザーブドインスタンスを購入したアカウントを解約すると、リザーブドインスタンスが期限切れになるか、解約したアカウントが完全に削除されるまで、支払いアカウントにリザーブドインスタンスの請求が継続されます。解約したアカウントは 90 日後に完全に削除されます。削除後、メンバーアカウントはリザーブドインスタンスの請求割引を受けられなくなります。アカウントの解約の詳細については、AWS Organizations ユーザーガイドの「AWS アカウントの解約」を参照してください。

## リザーブドインスタンス 割引料金範囲

割引料金範囲が適用されると、そのアカウントは、以降、その範囲レベル内で行われるリザーブドインスタンス 購入の前払い料金およびインスタンス使用料に対して、自動的に割引を受けます。割引を受けるためには、リージョンのリザーブドインスタンスの表示価格が 500,000 USD 以上である必要があります。

以下のルールが適用されます。

- ・料金範囲およびそれに関連する割引は、Amazon EC2 スタンダード リザーブドインスタンスの購入にのみ適用されます。
- ・料金範囲は、SQL Server Standard、SQL Server Web、および SQL Server Enterprise を使用する Windows 用のリザーブドインスタンスには適用されません。

- 料金範囲は、SQL Server Standard、SQL Server Web、および SQL Server Enterprise を使用する Linux 用の リザーブドインスタンス には適用されません。
- 料金範囲の割引は、AWS での購入にのみ適用されます。これは、サードパーティの リザーブドインスタンス の購入には適用されません。
- 料金範囲割引は現在、コンバータブルリザーブドインスタンス の購入には適用されません。

#### トピック

- [リザーブドインスタンス の料金割引の計算 \(p. 292\)](#)
- [割引範囲での購入 \(p. 292\)](#)
- [購入料金範囲 \(p. 293\)](#)
- [料金範囲の一括請求 \(p. 293\)](#)

#### リザーブドインスタンス の料金割引の計算

リージョンのすべての リザーブドインスタンス の合計表示価格を計算することによって、アカウントの料金範囲を決定できます。各予約の時間当たりの定期料金を期間の合計時間数に掛けて、購入時に「[リザーブドインスタンス の料金表ページ](#)」に表示される割引されていない前払い料金(固定料金とも呼ばれる)を加算します。表示価格は割引前料金(一般料金)に基づいているため、従量制割引の適用を受けた場合または リザーブドインスタンス を購入した後の値下げ分は反映されません。

```
List value = fixed price + (undiscounted recurring hourly price * hours in term)
```

たとえば、1年間の一部前払い t2.small リザーブドインスタンス の場合、前払い料金は 60.00 ドル、時間料金は 0.007 ドルと仮定します。これにより、表示価格は 121.32 ドルとなります。

```
121.32 = 60.00 + (0.007 * 8760)
```

Amazon EC2 コンソールを使用して リザーブドインスタンス の固定料金を表示するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Reserved Instances] を選択します。
- 右上の [列の表示/非表示] (歯車の形のアイコン) を選択して、[前払い価格] 列を表示します。

リザーブドインスタンス の固定料金をコマンドラインを使用して表示するには

- [describe-reserved-instances \(AWS CLI\)](#)
- [Get-EC2ReservedInstance \(AWS Tools for Windows PowerShell\)](#)
- [DescribeReservedInstances \(Amazon EC2 API\)](#)

#### 割引範囲での購入

リザーブドインスタンス を購入すると、割引料金範囲に該当する部分のリザーブドインスタンスに対応する割引があれば、Amazon EC2 によって自動的に適用されます。特に何かを行う必要はなく、どの Amazon EC2 ツールを使用しても リザーブドインスタンス を購入できます。詳細については、「[リザーブドインスタンス の購入 \(p. 293\)](#)」を参照してください。

リージョンでのアクティブな リザーブドインスタンス の表示価格が割引料金範囲に該当した場合、そのリージョンでの リザーブドインスタンス は、以降、すべての購入が割引料金で課金されます。リージョンの リザーブドインスタンス の1回の購入でしきい値を超える場合は、そのご購入分のうち、割引範囲のしきい値を超える部分が割引になります。購入プロセス中に作成された一時的な リザーブドインスタンス ID の詳細については「[購入料金範囲 \(p. 293\)](#)」を参照してください。

表示価格が割引範囲の金額を下回った場合は（一部のリザーブドインスタンスの有効期限が切れた場合など）、以降、そのリージョンで購入されるリザーブドインスタンスには割引が適用されません。ただし、もともと割引料金範囲で購入されたリザーブドインスタンスに対しては、引き続き割引が適用されます。

リザーブドインスタンスを購入すると、次の4つのいずれかの状況になります。

- ・割引なし—1つのリージョンでのリザーブドインスタンスの購入が、まだ割引しきい値より下である。
- ・一部割引—1つのリージョンでのリザーブドインスタンスの購入により、最初の割引範囲のしきい値を超える。割引なしが1つ以上の予約に適用され、割引料金が残りの予約に適用されます。
- ・完全割引—リージョン内の購入全体が1つの割引範囲に完全に含まれ、適切に割引されます。
- ・2つの割引料金—1つのリージョンでのリザーブドインスタンスの購入が、下の割引範囲から上の割引範囲まで及ぶ。2つの異なる料金が課金されます。1つ以上の予約により低い割引率が適用され、残りの予約により高い割引率が適用されます。

## 購入料金範囲

割引料金範囲に到達した場合、その購入に対して複数のエントリが表示されます。通常の料金が課金される部分と、割引料金が適用されて課金される部分です。

リザーブドインスタンスサービスによって複数のリザーブドインスタンスIDが生成されます。これは、割引が適用されない範囲と割引範囲、または複数の割引範囲に購入がまたがるためです。範囲内の予約のセットにはそれぞれIDがあります。この結果、購入CLIコマンドまたはAPIアクションによって返されるIDは、新しいリザーブドインスタンスの実際のIDとは異なるものになります。

## 料金範囲の一括請求

一括請求アカウントはリージョン内のメンバーアカウントの表示価格を集計します。一括請求アカウントのすべてのアクティブなリザーブドインスタンスの表示価格が割引料金範囲に達すると、以降（その一括請求アカウントの表示価格が割引価格範囲のしきい値を超えている限り）、一括請求アカウント内のアカウントで購入されたリザーブドインスタンスには割引が適用されます。詳細については、「[リザーブドインスタンスおよび一括請求（コンソリデーティッドビリング）\(p. 291\)](#)」を参照してください。

## リザーブドインスタンスの購入

リザーブドインスタンスを購入するには、AWSとサードパーティ販売者でリザーブドインスタンス製品を検索し、希望するものと完全に一致するタイプが見つかるまで検索パラメータを調整します。

購入するリザーブドインスタンスを検索する際、提供タイプの費用の見積もりが表示されます。購入手続きに進むと、AWSは自動的に購入価格に上限価格を指定します。リザーブドインスタンスの合計コストが、指定した金額を超えることはありません。

何らかの理由により料金が上がったり変更された場合、購入は完了しません。購入時に、選択した内容と同じような内容で価格が安い製品があった場合、AWSはその製品をその安い価格で販売します。

購入を承認する前に、購入を検討しているリザーブドインスタンスの詳細を点検して、すべてのパラメータが正しいことを確認してください。リザーブドインスタンスを購入した後では（リザーブドインスタンスマーケットプレイスでサードパーティの販売者から、またはAWSのいずれの場合も）、購入をキャンセルすることはできません。

### Note

リザーブドインスタンスを購入および変更するには、アベイラビリティーゾーンの表示など、IAMユーザーアカウントに適切なアクセス許可があることを確認します。詳細については、「[AWS CLIまたはAWS SDKで使用するサンプルポリシー](#)」および「[Amazon EC2 コンソールで使用するサンプルポリシー](#)」を参照してください。

### タスク

- ・[プラットフォームの選択 \(p. 294\)](#)

- [購入をキューに入れる \(p. 294\)](#)
- [スタンダード リザーブドインスタンス の購入 \(p. 295\)](#)
- [コンバータブルリザーブドインスタンス の購入 \(p. 296\)](#)
- [リザーブドインスタンス を表示する \(p. 298\)](#)
- [キューに入れた購入予約のキャンセル \(p. 298\)](#)
- [リザーブドインスタンス の更新 \(p. 299\)](#)
- [リザーブドインスタンス を使用する \(p. 299\)](#)

## プラットフォームの選択

Amazon EC2 は、次の Linux プラットフォームを リザーブドインスタンス でサポートしています。

- Linux/UNIX
- Linux with SQL Server Standard
- Linux with SQL Server Web
- Linux with SQL Server Enterprise
- SUSE Linux
- Red Hat Enterprise Linux

リザーブドインスタンス を購入する際、インスタンスのオペレーティングシステムを表すプラットフォームに対するサービスを選択する必要があります。

- SUSE Linux および RHEL ディストリビューションでは、これらの特定のプラットフォーム (SUSE Linux または Red Hat Enterprise Linux プラットフォーム) 用のサービスを選択する必要があります。
- その他のすべての Linux ディストリビューション (Ubuntu を含む) の場合は、Linux/UNIX プラットフォームに対するサービスを選択します。
- 既存の RHEL サブスクリプションを持ち込む場合は、Red Hat Enterprise Linux プラットフォーム用のサービスではなく、Linux/UNIX プラットフォーム用のサービスを選択する必要があります。

Windows でサポートされているプラットフォームについては、Windows インスタンスの Amazon EC2 ユーザーガイドの「[プラットフォームの選択](#)」を参照してください。

リザーブドインスタンス を購入し、請求製品コードを使用して AMI から起動した オンデマンドインスタンス に適用した場合、一致する請求製品コードが リザーブドインスタンス にあることを確認してください。一致する請求製品コードなしで リザーブドインスタンス を購入した場合、リザーブドインスタンス は オンデマンドインスタンス に適用されません。AMI 請求コードの取得方法の詳細については、「[請求情報の取得 \(p. 161\)](#)」を参照してください。

## 購入をキューに入れる

デフォルトでは、リザーブドインスタンス は購入するとすぐに実行されます。別の方法として、将来の日の購入予約をキューに入れることができます。たとえば、既存の リザーブドインスタンス が期限切れになる頃の購入予約をキューに入れることができます。これにより、サービスを切れ目なく利用できます。

リザーブドインスタンス の購入予約をキューに入れる場合、リージョンは指定できますが、ゾーンを指定した リザーブドインスタンス の購入予約や、他の販売者からの リザーブドインスタンス の購入予約を行なうことはできません。購入予約は 3 年先までキューに入れることができます。予約した日時に、デフォルトの支払い方法を使用して購入が実行されます。支払いが正常に行われると、支払い特典が適用されます。

キューに入れた購入予約は Amazon EC2 コンソールで確認できます。キューに入れた購入予約のステータスは [queued] になります。キューに入れた購入予約は、予約日の前にいつでもキャンセルできます。詳細については、「[キューに入れた購入予約のキャンセル \(p. 298\)](#)」を参照してください。

## スタンダード リザーブドインスタンス の 購入

スタンダード リザーブドインスタンス を特定のアベイラビリティゾーンで購入し、キャパシティーの予約ができます。または、キャパシティの予約を見送り、リージョンのスタンダード リザーブドインスタンス を購入することもできます。

コンソールを使用してスタンダード リザーブドインスタンス を購入するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [リザーブドインスタンス] を選択し、[リザーブドインスタンス の購入] を選択します。
3. [提供クラス] で [スタンダード] を選択し、スタンダード リザーブドインスタンス を表示します。
4. キャパシティーの予約を購入するには、購入画面の右上で [Only show offerings that reserve capacity] を選択します。リージョン リザーブドインスタンス を購入するには、チェックボックスを選択しないままにします。
5. 必要に応じて他の設定を選択してから、[Search] を選択します。

リザーブドインスタンスマーケットプレイスからスタンダード リザーブドインスタンス を購入するには、[サードパーティ] を検索結果の [販売者] 列から検出します。[期間] 列には標準以外の期間が表示されます。

6. 購入する リザーブドインスタンス を選択し、数量を入力して、[Add to Cart (カートに追加)] を選択します。
7. 選択した リザーブドインスタンス の要約を確認するには、[View Cart (カートを見る)] を選択します。
8. [Order On (注文日)] が [Now] の場合は、購入が即座に実行されます。購入予約をキューに入れるには、[Now] を選択して日付を選択します。カート内の有効なサービスごとに別の日付を選択できます。購入予約は、ブラウザのタイムゾーンで、選択した日付の 00:00 までキュー内に残ります。
9. 注文を確定するには、[Order (注文)] を選択します。

注文時に、選択したインスタンスと同等でより安価なインスタンスがある場合、AWS はより安価なインスタンスを販売します。

10. 注文のステータスは [State] 列に表示されます。注文が確定されると、[State] の値が [payment-pending] から [active] に変わります。リザーブドインスタンス が active の場合、使用準備が完了しています。

### Note

ステータスが `retired` になると、AWS は支払いを受信していない場合があります。

AWS CLI を使用してスタンダード リザーブドインスタンス を購入するには

1. `describe-reserved-instances-offerings` コマンドを使用して、利用できる リザーブドインスタンス を見つけます。スタンダード リザーブドインスタンス のみを返すには、`standard` を `--offering-class` パラメータに指定します。追加のパラメータを適用して結果を絞り込むことができます。たとえば、Linux/UNIX のデフォルトテナンシーのリージョナル `t2.large` リザーブドインスタンス を 1 年間の期間だけで購入するには:

```
aws ec2 describe-reserved-instances-offerings --instance-type t2.large --offering-class standard --product-description "Linux/UNIX" --instance-tenancy default --filters Name=duration,Values=31536000 Name=scope,Values=Region
```

リザーブドインスタンスマーケットプレイス だけで リザーブドインスタンス を探すには、marketplace フィルタを使用します。期間が 1 年間あるいは 3 年間より短い場合があるため、リクエストに期間は指定しません。

```
aws ec2 describe-reserved-instances-offerings --instance-type t2.large --offering-class standard --product-description "Linux/UNIX" --instance-tenancy default --filters Name=marketplace,Values=true
```

ニーズに合う リザーブドインスタンス が見つかったら、提供 ID を書き留めます。次に例を示します。

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

- purchase-reserved-instances-offering コマンドを使用して、リザーブドインスタンスを購入します。前のステップで取得した リザーブドインスタンス 提供 ID を指定し、予約するインスタンスの数を指定する必要があります。

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 --instance-count 1
```

デフォルトでは、購入は即座に実行されます。別の方法として、購入予約をキューに入れるには、次のパラメータを前の呼び出しに追加します。

```
--purchase-time "2020-12-01T00:00:00Z"
```

- describe-reserved-instances コマンドを使用して、リザーブドインスタンスを購入します。

```
aws ec2 describe-reserved-instances
```

または、以下の AWS Tools for Windows PowerShell コマンドを使用します。

- Get-EC2ReservedInstancesOffering
- New-EC2ReservedInstance
- Get-EC2ReservedInstance

購入の完了後、リザーブドインスタンスの仕様と一致するインスタンスをすでに実行している場合は、支払い特典が即座に適用されます。インスタンスを再起動する必要はありません。適切な実行中のインスタンスが存在しない場合、インスタンスを起動して、リザーブドインスタンスに指定した条件と一致していることを確認します。詳細については、「[リザーブドインスタンスを使用する \(p. 299\)](#)」を参照してください。

実行しているインスタンスにどのように リザーブドインスタンス が適用されるかについての例は、「[リザーブドインスタンスがどのように適用されるか \(p. 284\)](#)」を参照します。

## コンバーティブルリザーブドインスタンスの購入

コンバーティブルリザーブドインスタンスを特定のアベイラビリティゾーンで購入し、キャパシティーの予約ができます。または、キャパシティの予約を見送り、リージョン コンバーティブルリザーブドインスタンスを購入することもできます。

コンソールを使用して コンバーティブルリザーブドインスタンスを購入するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで [リザーブドインスタンス] を選択し、[リザーブドインスタンスの購入] を選択します。
- [提供クラス] で [コンバーティブル] を選択し、コンバーティブルリザーブドインスタンスを表示します。

4. キャパシティーの予約を購入するには、購入画面の右上で [Only show offerings that reserve capacity] を選択します。リージョン リザーブドインスタンス を購入するには、チェックボックスを選択しないままにします。
  5. 必要に応じて他の設定を選択してから、[Search] を選択します。
  6. 購入する コンバティブルリザーブドインスタンス を選択し、数量を入力して、[Add to Cart (カートに追加)] を選択します。
  7. 選択したリザーブドインスタンスの要約を確認するには、[View Cart (カートを見る)] を選択します。
  8. [Order On (注文日)] が [Now] の場合は、購入が即座に実行されます。購入予約をキューに入れるには、[Now] を選択して日付を選択します。カート内の有効なサービスごとに別の日付を選択できます。購入予約は、ブラウザのタイムゾーンで、選択した日付の 00:00 までキュー内に残ります。
  9. 注文を確定するには、[Order (注文)] を選択します。
- 注文時に、選択したインスタンスと同等でより安価なインスタンスがある場合、AWS はより安価なインスタンスを販売します。
10. 注文のステータスは [State] 列に表示されます。注文が確定されると、[State] の値が [payment-pending] から [active] に変わります。リザーブドインスタンス が active の場合、使用準備が完了しています。

#### Note

ステータスが `retired` になると、AWS は支払いを受信していない場合があります。

#### AWS CLI を使用して コンバティブルリザーブドインスタンス を購入するには

1. `describe-reserved-instances-offerings` コマンドを使用して、利用できる リザーブドインスタンス を見つけます。コンバティブルリザーブドインスタンスだけを返すには、`convertible` を `--offering-class` パラメータに指定します。追加のパラメータを適用して結果を絞り込むことができます。たとえば、Linux/UNIX のデフォルトテナンシーのリージョナル `t2.large` リザーブドインスタンス を購入するには:

```
aws ec2 describe-reserved-instances-offerings --instance-type t2.large --offering-class convertible --product-description "Linux/UNIX" --instance-tenancy default --filters Name=scope,Values=Region
```

ニーズに合う リザーブドインスタンス が見つかったら、提供 ID を書き留めます。次に例を示します。

```
"ReservedInstancesOfferingId": "bec624df-a8cc-4aad-a72f-4f8abc34caf2"
```

2. `purchase-reserved-instances-offering` コマンドを使用して、リザーブドインスタンス を購入します。前のステップで取得した リザーブドインスタンス 提供 ID を指定し、予約するインスタンスの数を指定する必要があります。

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-id bec624df-a8cc-4aad-a72f-4f8abc34caf2 --instance-count 1
```

デフォルトでは、購入は即座に実行されます。別の方法として、購入予約をキューに入れるには、次のパラメータを前の呼び出しに追加します。

```
--purchase-time "2020-12-01T00:00:00Z"
```

3. `describe-reserved-instances` コマンドを使用して、リザーブドインスタンス を購入します。

```
aws ec2 describe-reserved-instances
```

または、以下の AWS Tools for Windows PowerShell コマンドを使用します。

- [Get-EC2ReservedInstancesOffering](#)
- [New-EC2ReservedInstance](#)
- [Get-EC2ReservedInstance](#)

リザーブドインスタンスの仕様と一致するインスタンスをすでに実行している場合、料金上の利点は即時適用されます。インスタンスを再起動する必要はありません。適切な実行中のインスタンスが存在しない場合、インスタンスを起動して、リザーブドインスタンスに指定した条件と一致していることを確認します。詳細については、「[リザーブドインスタンスを使用する \(p. 299\)](#)」を参照してください。

実行しているインスタンスにどのようにリザーブドインスタンスが適用されるかについての例は、「[リザーブドインスタンスがどのように適用されるか \(p. 284\)](#)」を参照します。

## リザーブドインスタンスを表示する

Amazon EC2 コンソールあるいはコマンドラインツールを使用して、購入したリザーブドインスタンスを表示できます。

リザーブドインスタンスをコンソールで表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Reserved Instances] を選択します。
3. アクティブおよびリタイアされたリザーブドインスタンスが一覧表示されます。[状態] 列には状態が表示されます。
4. ユーザーがリザーブドインスタンスマーケットプレイスの販売者の場合、[出品] タブには [リザーブドインスタンスマーケットプレイス \(p. 299\)](#) で一覧表示される予約の状態が表示されます。詳細については、「[リザーブドインスタンスの出品状態 \(p. 304\)](#)」を参照してください。

コマンドラインを使用してリザーブドインスタンスを表示するには

- [describe-reserved-instances](#) (AWS CLI)
- [Get-EC2ReservedInstance](#) (Tools for Windows PowerShell)

## キューに入れた購入予約のキャンセル

購入予約は 3 年先までキューに入れることができます。キューに入れた購入予約は、予約日の前にいつでもキャンセルできます。

キューに入れた購入をキャンセルするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Reserved Instances] を選択します。
3. 1 つまたは複数のリザーブドインスタンスを選択します。
4. [アクション]、[キュー入りリザーブドインスタンスの削除] の順に選択します。
5. 確認を求めるメッセージが表示されたら、[Yes, Delete] を選択します。

コマンドラインを使用してキューに入れた購入予約をキャンセルするには

- [delete-queued-reserved-instances](#) (AWS CLI)
- [Remove-EC2QueuedReservedInstance](#) (Tools for Windows PowerShell)

## リザーブドインスタンス の更新

リザーブドインスタンス は有効期限が切れる前に更新できます。リザーブドインスタンス を更新すると、現在の リザーブドインスタンス が期限切れになるまで、同じ設定の リザーブドインスタンス の購入予約がキューに入れられます。

キューに入れた購入予約を使用して リザーブドインスタンス を更新するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Reserved Instances] を選択します。
3. 1 つまたは複数の リザーブドインスタンス を選択します。
4. [Actions (アクション)]、[Renew Reserved Instances (リザーブドインスタンスの更新)] の順に選択します。
5. 注文を確定するには、[Order (注文)] を選択します。

## リザーブドインスタンス を使用する

リザーブドインスタンス は、仕様の一一致する実行中の オンデマンドインスタンス に自動的に適用されます。リザーブドインスタンス の仕様と一致する実行中の オンデマンドインスタンス が存在しない場合、必要な仕様が搭載されるインスタンスを起動するまで、リザーブドインスタンス は未使用となります。

リザーブドインスタンス の料金上の利点を利用してインスタンスを起動する場合、起動時に以下を必ず指定してください。

- プラットフォーム: リザーブドインスタンス のプラットフォーム (製品の説明) と一致するAmazon Machine Image (AMI) を選択する必要があります。たとえば、Linux/UNIX を指定する場合、Amazon Linux AMI または Ubuntu AMI からインスタンスを起動できます。
- インスタンスタイプ: リザーブドインスタンス と同じインスタンスタイプを指定します。たとえば、t2.large など。
- アベイラビリティーゾーン: 特定のアベイラビリティーゾーンに リザーブドインスタンス を購入する場合、同じアベイラビリティーゾーンでインスタンスを起動する必要があります。リージョン リザーブドインスタンス を購入した場合、どのアベイラビリティーゾーンでもインスタンスを起動できます。
- テナント: インスタンスのテナントは リザーブドインスタンス のテナントを一致する必要があります。たとえば、dedicated や shared など。詳細については、「[ハードウェア専有インスタンス \(p. 425\)](#)」を参照してください。

詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。実行しているインスタンスにどのように リザーブドインスタンス が適用されるかについての例は、「[リザーブドインスタンス がどのように適用されるか \(p. 284\)](#)」を参照します。

Amazon EC2 Auto Scaling または他の AWS サービスを使用して、リザーブドインスタンス のメリットを利用する オンデマンドインスタンス を起動できます。詳細については、「[Amazon EC2 Auto Scaling ユーザーガイド](#)」を参照してください。

## リザーブドインスタンスマーケットプレイス

リザーブドインスタンスマーケットプレイス は、サードパーティーや AWS のお客様が購入した、さまざまな期間と料金オプションの未使用スタンダード リザーブドインスタンス の販売をサポートするプラットフォームです。たとえば、ビジネスの必要性が変更した場合や不要なキャパシティーがある場合に、新しいインスタンスタイプに変更し、期間が終了する前にプロジェクトを終わらせるときに、新しい AWS リージョンにインスタンスを移動したあとで リザーブドインスタンス を販売することができます。

リザーブドインスタンスマーケットプレイス で未使用の リザーブドインスタンス を販売する場合は、特定の要件基準を満たす必要があります。

## コンテンツ

- [リザーブドインスタンスマーケットプレイス で販売する \(p. 300\)](#)
- [リザーブドインスタンス Marketplace からの購入 \(p. 305\)](#)

## リザーブドインスタンスマーケットプレイス で販売する

リザーブドインスタンスマーケットプレイス で リザーブドインスタンス を出品するとすぐに、購入者側から見えるようになります。すべての リザーブドインスタンス は、残り期間や時間料金別にグループ化されます。

AWS は購入者のリクエストを満たすため、指定されたグループで最も前払い料金が低い リザーブドインスタンス を最初に販売します。次に、購入者の注文全体が満たされるまで、次に最も低い価格の リザーブドインスタンス を販売します。AWS は次にトランザクションを処理し、リザーブドインスタンス の所有権を購入者に移します。

出品した リザーブドインスタンス は、売れるまでお客様の所有です。販売後は、予約済みのキャパシティーと割引使用料金は使用できません。インスタンスを使用し続ける場合、AWS は リザーブドインスタンス が売却された時間から起算したオーデマンド価格を課金します。

### 目次

- [制約と制限 \(p. 300\)](#)
- [販売者として登録する \(p. 301\)](#)
- [支払い用の銀行口座 \(p. 301\)](#)
- [税金情報 \(p. 302\)](#)
- [リザーブドインスタンス の価格決定 \(p. 302\)](#)
- [リザーブドインスタンス の出品 \(p. 303\)](#)
- [リザーブドインスタンス の出品状態 \(p. 304\)](#)
- [出品のライフサイクル \(p. 304\)](#)
- [リザーブドインスタンス が売却された後 \(p. 305\)](#)
- [支払いを受け取る \(p. 305\)](#)
- [購入者と共有する情報 \(p. 305\)](#)

### 制約と制限

未使用的リザーブドインスタンスを販売できるようになる前に、販売者として リザーブドインスタンスマーケットプレイス に登録する 必要があります。詳細については、[販売者として登録する \(p. 301\)](#) を参照してください。

リザーブドインスタンス の販売時に次の制約と制限が適用されます：

- リザーブドインスタンスマーケットプレイス で販売できるのは、Amazon EC2 スタンダードリザーブドインスタンスのみになります。Amazon EC2コンバータイブルリザーブドインスタンスは販売できません。Amazon RDS や Amazon ElastiCache などの他の AWS 用のリザーブドインスタンスは販売できません。
- スタンダード リザーブドインスタンス の有効期間が 1 か月以上残っている必要があります。
- デフォルトで無効になっているリージョン (アジアパシフィック (香港) および 中東 (バーレーン)) では、スタンダードリザーブドインスタンスは販売できません。
- リザーブドインスタンスマーケットプレイス で許容される最低販売価格は、0.00 USD です。
- リザーブドインスタンスマーケットプレイス では、リザーブドインスタンス の前払いなし、一部前払い、全額前払いには対応していません。リザーブドインスタンス の前払いがある場合は、AWS が前払

い料金を受け取り、予約がアクティブになってから(所有してから)30日以上経過しないと販売できません。

- リザーブドインスタンスマーケットプレイスの出品内容を直接変更することはできません。ただし、最初に出品をキャンセルしてから、新しいパラメータで別の出品を作成することはできます。詳細については、[リザーブドインスタンスの価格決定 \(p. 302\)](#) を参照してください。出品する前にリザーブドインスタンスを変更することもできます。詳細については、[リザーブドインスタンスの変更 \(p. 306\)](#) を参照してください。
- AWSは、リザーブドインスタンスマーケットプレイスで販売する各スタンダード リザーブドインスタンスに対して、前払い価格の総額の12%をサービス料として課金します。前払い価格は、販売者がスタンダード リザーブドインスタンスに課金する価格です。

## 販売者として登録する

### Note

アカウントを販売者として登録できるのは、AWSアカウントルートユーザーのみです。

リザーブドインスタンスマーケットプレイスで販売するには、まず販売者として登録することが必要です。登録時には以下の情報を指定します。

- 銀行情報 — AWSでは、予約の販売時に集金された金額をお支払いするために、お客様の銀行情報が必要です。住所が米国内の銀行でなければなりません。詳細については、「[支払い用の銀行口座 \(p. 301\)](#)」を参照してください。
- 税金情報 — 必要な税金報告義務を判断するために、販売者は必ず、税金情報の質問に回答する必要があります。詳細については、「[税金情報 \(p. 302\)](#)」を参照してください。

記入済みの販売者登録がAWSに受領されると、登録を確認する電子メールが届いて、リザーブドインスタンスマーケットプレイスでの販売が可能になったことが伝えられます。

## 支払い用の銀行口座

AWSでは、リザーブドインスタンスの販売時に集金された金額をお支払いするために、お客様の銀行情報が必要です。住所が米国内の銀行でなければなりません。

支払い用のデフォルトの銀行口座を登録するには

- [リザーブドインスタンスマーケットプレイス 販売者登録] ページを開き、AWS認識情報でサインインします。
- [Manage Bank Account] ページで、支払いを受け取る銀行に関する以下の情報を提供します。
  - 銀行口座の名義
  - 支店コード
  - アカウント番号
  - 銀行口座の種類

### Note

法人の銀行口座を使用する場合は、口座に関する情報をFAX(1-206-765-3424)で送信するよう指示されます。

登録後、指定された銀行口座がデフォルトとして設定され、銀行の確認待ちとなります。新しい銀行口座を確認するには、最長で2週間かかります。この間は支払金を受け取ることができません。確立済みの口座の場合、支払い完了まで通常およそ2日かかります。

## 支払い用のデフォルトの銀行口座を変更するには

- [リザーブドインスタンスマーケットプレイス 販売者登録] ページで、登録時に使用したアカウントを使ってサインインします。
- [Manage Bank Account] ページで、必要に応じて新規口座のアカウントを追加するか、デフォルトの銀行口座を変更します。

## 税金情報

リザーブドインスタンス の販売には、取引関連の税金 (消費税または付加価値税など) がかかることがあります。取引関連の税金が適用されるかどうかについては、税務部、法務部、財務部、または経理部に確認する必要があります。お客様は、取引関連の税金を収集し、該当する税務署に納める役割を担います。

販売者登録手続きの一環として、「[販売者登録ポータル](#)」の Tax Interview を完了させる必要があります。このインタビューでは、税金情報を収集し、IRS フォーム W-9、W-8BEN、または W-8BEN-E に入力します。このフォームは、必要な税金報告義務を明確にするために使用されます。

Tax Interview の一環として入力する税金情報は、個人または法人であるか、また、米国人または非米国人 (米国企業または非米国企業) かによって異なる場合があります。Tax interview の記入を行う際は、次に注意してください。

- AWS が提供する情報 (このトピックの情報を含む) は、税金、法律、またはその他の専門的なアドバイスではありません。IRS のレポート要件がビジネスに及ぼす影響について知りたい場合、またはご質問がある場合は、税金、法律、またはその他の専門家にお問い合わせください。
- IRS のレポート要件をできるだけ効率的に満たすには、Tax interview の中で要求されたすべての質問に答え、情報を入力します。
- 答えを確認します。綴りを間違ったり、誤った税金識別番号を入力したりしないようにします。これらのミスがあると、誤った税金フォームが生成されます。

Tax Interview の回答と IRS 報告のしきい値に基づいて、Amazon は Form 1099-K を提出する場合があります。Amazon は、お客様の税金口座がしきい値レベルに達した年の翌年の 1 月 31 日までに、フォーム 1099-K のコピーを郵送します。たとえば、税金口座が 2018 年にしきい値に達した場合は、2019 年の 1 月 31 日までにフォーム 1099-K が郵送されます。

IRS の要件とフォーム 1099-K の詳細については、「[IRS](#)」のウェブサイトを参照してください。

## リザーブドインスタンス の価格決定

販売する リザーブドインスタンス に指定できるのは前払い料金のみです。前払い料金は、購入者が リザーブドインスタンス を購入する際に支払う一括払いの料金です。

以下は留意すべき重要な制限です。

- 1 年間に 50,000 USD までの リザーブドインスタンス を販売できます。これよりも多く販売するには、「[Amazon EC2 リザーブドインスタンス 上限緩和申請](#)」フォームにご記入ください。
- 最低価格は 0 USD です。リザーブドインスタンスマーケットプレイス で許容される最低販売価格は、0.00 USD です。

出品内容を直接変更することはできません。ただし、最初に出品をキャンセルしてから、新しいパラメータで別の出品を作成することはできます。

出品は、active 状態であればいつでもキャンセルできます。既にマッチングされていたり、販売処理が行われている出品はキャンセルできません。出品したインスタンスの一部がマッチングされている場合にその出品をキャンセルすると、マッチングされていない残りのインスタンスが出品から削除されます。

リザーブドインスタンス の価値は時間が経つにつれて低下するため、AWS ではデフォルトで、1か月ごとに同じ割合で低下するような価格設定を行っています。ただし、予約を販売する時期に基づいて、異なる前払い価格を設定できます。

たとえば、リザーブドインスタンス の残りの期間が 9か月の場合、顧客が残り 9か月の リザーブドインスタンス を購入する場合、受領する額を指定できます。残りが 5か月である別の価格を設定し、さらに残りが 1か月の別の価格を設定することができます。

### リザーブドインスタンス の出品

登録済みの販売者の場合、販売する リザーブドインスタンス を 1つまたは複数選択できます。1件のリストにすべてまとめて販売することも、個別に販売することもできます。さらに、インスタンスタイプ、プラットフォーム、スコープのすべての設定で リザーブドインスタンス を出品できます。

コンソールによって、提示価格が決定します。お客様の リザーブドインスタンス に一致するサービスを確認し、最低価格のサービスに合わせます。それ以外の場合、残り時間の リザーブドインスタンス のコストに基づいて提示価格を計算します。計算後の値が 1.01 USD 未満の場合、提示価格は 1.01 USD です。

出品をキャンセルする場合、その一部が既に売れているとき、売却済みの部分についてのキャンセルは無効です。まだ売っていない部分のみが リザーブドインスタンスマーケットプレイス からなくなります。

AWS マネジメントコンソール を使用して リザーブドインスタンス を リザーブドインスタンスマーケットプレイス に出品するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Reserved Instances] を選択します。
3. 出品する リザーブドインスタンス を選択し、[リザーブドインスタンス の出品] を選択します。
4. [Configure Your Reserved Instances Listing (リザーブドインスタンス出品の設定)] ページで、販売するインスタンス数および残り期間に対する前払い価格を該当列に設定します。[Months Remaining] 列の隣にある矢印を選択して、残りの有効期間における予約の価値の変化状況を確認します。
5. 上級ユーザーが価格をカスタマイズする場合は、今後の月に異なる価格を入力できます。デフォルトの直線形の価格減少に戻すには、[Reset] を選択します。
6. 出品の設定が終了したら、[Continue] を選択します。
7. [Confirm Your Reserved Instances Listing (リザーブドインスタンス出品の確認)] ページに表示された出品詳細を確認し、問題がなければ [List Reserved Instance (リザーブドインスタンスの出品)] を選択します。

出品したインスタンスをコンソールで表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Reserved Instances] を選択します。
3. 出品した リザーブドインスタンス を選択し、[My Listings (自分の出品)] を選択します。

AWS CLI を使用して リザーブドインスタンスマーケットプレイス の リザーブドインスタンス を管理するには

1. `describe-reserved-instances` コマンドを使用して、リザーブドインスタンス の一覧を取得します。
2. 出品する リザーブドインスタンス の ID を書き留めて、`create-reserved-instances-listing` を呼び出します。リザーブドインスタンス の ID、インスタンスの数、価格体系を指定する必要があります。
3. ユーザーの出品を表示するには、`describe-reserved-instances-listings` コマンドを使用します。
4. 出品をキャンセルするには、`cancel-reserved-instances-listings` コマンドを使用します。

## リザーブドインスタンス の出品状態

リザーブドインスタンス ページの [出品] タブの [出品状態] には、出品の現在状況が表示されます。

[Listing State] に表示される情報は、リザーブドインスタンスマーケットプレイスへのお客様の出品の状態に関するものです。これは、[リザーブドインスタンス] ページの [State] 列に表示される状態情報とは異なります。この [State] 情報は、お客様の予約に関するものです。

- [アクティブ] — 購入できます。
- [cancelled (キャンセル済み)] — 出品がキャンセルされ、リザーブドインスタンスマーケットプレイスでの購入ができません。
- [closed (クローズ)] — リザーブドインスタンス は出品されていません。リザーブドインスタンス は、出品が完了したため [closed] になっている可能性があります。

## 出品のライフサイクル

出品したすべてのインスタンスがマッチングされて売れるとき、[My Listings] タブに表示される [Total instance count] が [Sold] の下に表示された数と同じになります。出品で残っている [Available] インスタンスがなくなり、[Status] が [closed] になります。

出品の一部だけが売れた場合、AWS は出品されている リザーブドインスタンス を取り下げ、残りの リザーブドインスタンス と同数の リザーブドインスタンス を作成します。したがって、出品 ID とその ID の出品は、販売中の予約が少なくなっていますがアクティブのままでです。

この出品内の リザーブドインスタンス の売却は今後、この方法で行われます。出品内のすべての リザーブドインスタンス が売れるとき、AWS でこの出品が [closed] としてマークされます。

たとえば、出品数が 5 の リザーブドインスタンス ID 5ec28771-05ff-4b9b-aa31-9e57dexample の出品を作成するとします。

コンソールの [Reserved Instance] ページの [My Listings] タブに、次のように出品が表示されます。

リザーブドインスタンス の出品 ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 0
- Available = 5
- Status = active

購入者が 2 つの予約を購入すると、3 つの予約が販売用に残ります。一部分が売れたため、AWS では引き続き販売される残りの予約に相当する、3 つの新しい予約が作成されます。

お客様の出品は、[My Listings (自分の出品)] タブで次のように表示されます。

リザーブドインスタンス の出品 ID 5ec28771-05ff-4b9b-aa31-9e57dexample

- Total reservation count = 5
- Sold = 2
- Available = 3
- Status = active

出品をキャンセルする場合、その一部が既に売れているとき、売却済みの部分についてのキャンセルは無効です。まだ売れていない部分のみが リザーブドインスタンスマーケットプレイス からなくなります。

## リザーブドインスタンスが売却された後

リザーブドインスタンスが売却されると、AWS から E メールの通知が送信されます。何らかのアクティビティがあった日ごとに、毎日のすべてのアクティビティをキャプチャした 1 通の E メール通知が送信されます。たとえば、出品の作成、出品の販売、AWS から口座への送金などです。

コンソールの リザーブドインスタンス の出品の状態を追跡するには、[リザーブドインスタンス]、[My Listings (自分の出品)] の順に選択します。[My Listings] タブには、[Listing State] の値が含まれています。また、期間、出品価格、および出品されているインスタンスの中で使用可能、売却済み、およびキャンセルされたものがいくつあるかの詳細といった情報が示されます。また、[describe-reserved-instances-listings](#) コマンドを適切なフィルタで使用することで、リザーブドインスタンスの出品に関する情報を取得できます。

## 支払いを受け取る

AWS が購入者からの支払い金を受領するとすぐに、販売された リザーブドインスタンス に登録されている所有者アカウントの E メールアドレスに、メッセージが送信されます。

AWS は指定された銀行口座に、自動決済機関 (ACH) の電子送金を送信します。通常、この送金は、リザーブドインスタンス の売却後 1~3 日の間に行われます。支払いは、1 日に 1 回行われます。送金が行われると、支払い報告がメールで届きます。AWS が銀行からの検証結果を受領するまで、支払い金を受け取れないことに注意してください。これには最大 2 週間かかることがあります。

販売した リザーブドインスタンス は、ユーザーの リザーブドインスタンス の詳細に引き続き表示されます。

販売者は リザーブドインスタンス に対する現金の支払いを、振込によって直接各自の銀行口座で受け取ります。AWS は、リザーブドインスタンスマーケットプレイス で販売する各スタンダード リザーブドインスタンス に対して、前払い価格の総額の 12% をサービス料として課金します。

## 購入者と共有する情報

リザーブドインスタンスマーケットプレイス で販売する場合、AWS は米国の規制に従って、お客様の正式な会社名を購入者のステートメントに示します。さらに、請求書またはその他の税金関連の理由で、購入者から販売者に連絡したいとの要望が AWS サポートにあった場合、購入者から販売者に直接連絡できるよう、AWS は必要に応じて販売者の E メールアドレスを購入者に提供する場合があります。

同様の理由で、販売者には購入者の郵便番号および国情報が支払いレポートによって提供されます。販売者として、この情報を、国に支払わなければならない取引税（売上税や付加価値税など）に添付する必要がある場合があります。

AWS は税金に関する助言を行うことはできませんが、お客様が追加情報を必要としているときお客様の税務専門家が判断した場合は、[AWS カスタマーサポートにご連絡ください](#)。

## リザーブドインスタンス Marketplace からの購入

不要になった リザーブドインスタンス を販売するサードパーティの販売者から リザーブドインスタンスマーケットプレイス で リザーブドインスタンス を購入できます。これを行うには、Amazon EC2 コンソールまたはコマンドラインツールを使用します。このプロセスは、AWS から リザーブドインスタンス を購入するプロセスに類似しています。詳細については、「[リザーブドインスタンス の購入 \(p. 293\)](#)」を参照してください。

リザーブドインスタンスマーケットプレイス で購入した リザーブドインスタンス と AWS から直接購入した リザーブドインスタンス の間にはいくつかの違いがあります。

- ・ **期間** — サードパーティ販売者から購入した リザーブドインスタンス は、残り期間が完全な標準期間よりも短くなっています。AWS の完全な標準期間は 1 年または 3 年間です。
- ・ **前払い価格** — サードパーティの リザーブドインスタンス は、さまざまな前払い価格で販売されます。使用料金または定期的に支払う料金は、リザーブドインスタンス を最初に AWS から購入したときに設定された料金と同じ金額です。

- リザーブドインスタンスのタイプリザーブドインスタンス — リザーブドインスタンスマーケットプレイスで購入できるのは、Amazon EC2 スタンダード リザーブドインスタンスのみです。コンバーティブルリザーブドインスタンス、Amazon RDS、および Amazon ElastiCache リザーブドインスタンスはリザーブドインスタンスマーケットプレイスでは購入できません。

お客様に関する基本情報(郵便番号や国情報など)は、販売者と共有されます。

この情報を使用して、販売者は、国に支払う必要な取引税(売上税や付加価値税など)を計算し、支払いレポートとして提示します。まれに、AWSが販売者にEメールアドレスを提供する必要がある場合があります。これは、販売者が、販売に関する質問があり、それに関して連絡できるようにするために(たとえば税務上の質問など)。

同様の理由で、AWSは購入者の請求書に販売者の正式名を記載します。税金または関連する理由で販売者の情報が必要な場合は、[AWS サポート](#)までお問い合わせください。

## リザーブドインスタンスの変更

ニーズが変化したときは、スタンダードまたはコンバーティブルリザーブドインスタンスを変更し、引き続き料金上の利点を得られます。アベイラビリティゾーン、インスタンスサイズ(同じインスタンスファミリー内で)やリザーブドインスタンスのスコープなどの属性を変更できます。

### Note

また、別の構成で、別のコンバーティブルリザーブドインスタンスのコンバーティブルリザーブドインスタンスに交換することもできます。詳細については、「[コンバーティブルリザーブドインスタンスの交換\(p. 313\)](#)」を参照してください。

すべてのリザーブドインスタンス、またはそのサブセットを変更できます。元のリザーブドインスタンスを2つ以上の新しいリザーブドインスタンスに分割できます。たとえば、us-east-1aに10のインスタンスを予約があり、そのうち5つのインスタンスをus-east-1bに移動する場合、結果的にこの変更は2つの新しい予約をリクエストします。us-east-1aでの5つのインスタンス用の予約とus-east-1bでの5つのインスタンス用の予約です。

また、2つ以上のリザーブドインスタンスを単一のリザーブドインスタンスにマージすることもできます。たとえば、それぞれに1つのインスタンスがある4つのt2.smallリザーブドインスタンスがある場合、これらをマージして1つのt2.largeリザーブドインスタンスを作成できます。詳細については、「[インスタンスサイズの変更のサポート\(p. 308\)](#)」を参照してください。

変更後、リザーブドインスタンスの利点は、リザーブドインスタンスの新しいパラメータと一致するインスタンスのみに適用されます。たとえば、予約のアベイラビリティゾーンを変更する場合、キャパシティーの予約と料金上の利点は、新しいアベイラビリティゾーン内のインスタンスの使用に対して自動的に適用されます。新しいパラメータに一致しないインスタンスは、他に適用可能な予約がない場合、オンデマンド価格で課金されます。

変更リクエストが成功した場合。

- 変更後の予約がすぐに有効になり、変更リクエストが完了した時刻から、割引料金が新しいインスタンスに適用されます。たとえば、午後9時15分に予約の変更が成功した場合、割引料金は午後9時00分から新しいインスタンスに移ります。変更されたリザーブドインスタンスの発行日は[describe-reserved-instances](#)コマンドを使用して取得できます。
- 元の予約は終了します。その終了日は新しい予約の開始日であり、新しい予約の終了日は元のリザーブドインスタンスの終了日と同じです。有効期限のうち16ヶ月が残っている3年の予約を正常に変更した場合、変更後の予約は16ヶ月の予約であり、終了日は変更前の予約と同じです。
- 変更後の予約の固定価格は0USDであり、元の予約の固定価格ではありません。
- 変更後の予約の固定価格はアカウントに適用される割引料金範囲の計算に影響を与えません。割引範囲の計算は元の予約の固定価格に基づきます。

変更リクエストに失敗した場合、リザーブドインスタンスは元の設定を維持し、別の変更リクエストをすぐに利用できます。

変更に手数料は必要なく、新しく課金されたり、請求書が届いたりすることはありません。

予約の変更是必要に応じて何度も行うことができますが、変更を送信後に保留中の変更リクエストを変更またはキャンセルすることはできません。変更が完了した後は、必要に応じて別の変更リクエストを送信して、実行した変更をロールバックできます。

## コンテンツ

- [変更の要件と制限 \(p. 307\)](#)
- [インスタンスサイズの変更のサポート \(p. 308\)](#)
- [変更リクエストの送信 \(p. 311\)](#)
- [変更リクエストのトラブルシューティング \(p. 312\)](#)

## 変更の要件と制限

以下のように、これらの属性を変更できます。

変更可能な属性	サポートされているプラットフォーム	制約事項
同じリージョン内でアベイラビリティーゾーンを変更する	Linux と Windows	–
スコープをアベイラビリティーゾーンからリージョンに、またはその逆に変更する	Linux と Windows	スコープをアベイラビリティーゾーンからリージョンに変更した場合、キャパシティーの予約の利点を失います。 スコープをリージョンからアベイラビリティーゾーンに変更する場合、アベイラビリティーゾーンの柔軟性とインスタンスサイズの柔軟性(適用される場合)を失います。詳細については、「 <a href="#">リザーブドインスタンスがどのように適用されるか (p. 284)</a> 」を参照してください。
同じインスタンスファミリー内でインスタンスサイズを変更する	Linux/UNIX のみ  リザーブドインスタンスのインスタンスサイズの柔軟性は、他のプラットフォーム (Linux with SQL Server Standard、Linux with SQL Server Web、Linux with SQL Server Enterprise、Red Hat Enterprise Linux、SUSE Linux、Windows、Windows with SQL Standard、Windows with SQL Server Enterprise、および Windows with SQL Server Web) では利用できません。	予約ではデフォルトのテナントを使用する必要があります。使用できる他のサイズがないため、一部のインスタンスファミリーはサポートされません。詳細については、「 <a href="#">インスタンスサイズの変更のサポート (p. 308)</a> 」を参照してください。

変更可能な属性	サポートされているプラットフォーム	制約事項
ネットワークを EC2-Classic から Amazon VPC に、またはその逆に変更する	Linux と Windows	ネットワークプラットフォームは AWS アカウントで利用できる必要があります。2013 年 12 月 4 日より後に AWS アカウントを作成した場合、EC2-Classic はサポートされません。

## 要件

変更リクエストは、変更後の設定(該当する場合)に対して十分なリザーブドインスタンス容量があり、以下の条件が満たされている場合に Amazon EC2 で処理されます。

- 購入と同時期またはその前にリザーブドインスタンスを変更できないこと
- リザーブドインスタンスがアクティブであること
- 保留中の変更リクエストがないこと
- リザーブドインスタンスがリザーブドインスタンスマーケットプレイスに出品されていないこと
- アクティブな予約のインスタンスサイズのフットプリントと変更後の設定が一致していることが必要です。詳細については、「[インスタンスサイズの変更のサポート \(p. 308\)](#)」を参照してください。
- 入力リザーブドインスタンスはすべてスタンダードリザーブドインスタンスあるいはすべてコンバティブルリザーブドインスタンスであり、両方のタイプが混ざっていないこと
- スタンダードリザーブドインスタンスの場合、入力リザーブドインスタンスは同じ時間内に期限切れとなること
- リザーブドインスタンスは G4 インスタンスではありません。

## インスタンスサイズの変更のサポート

プラットフォームが Linux/UNIX で、インスタンスファミリーに複数のサイズがある場合は、リザーブドインスタンスのインスタンスサイズを変更できます。

### Note

インスタンスはファミリー(ストレージまたは CPU 容量に基づく)、タイプ(特定のユースケース用に設計)、およびサイズによってグループ分けされています。たとえば、c4 インスタンスファミリーはコンピューティング最適化ファミリーに含まれ、複数のサイズで利用できます。c3 インスタンスは同じファミリーに含まれますが、c4 インスタンスを c3 インスタンスに変更することはできません。これは、ハードウェア仕様が異なるためです。インスタンスタイプの詳細については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

各インスタンスファミリーでは 1 つのサイズしか使用できないため、以下のインスタンスタイプのリザーブドインスタンスのインスタンスサイズを変更することはできません。

- cc2.8xlarge
- cr1.8xlarge
- hs1.8xlarge
- t1.micro

各リザーブドインスタンスにはインスタンスサイズのフットプリントがあり、これはインスタンスタイプの正規化係数と予約に含まれるインスタンスの数によって決まります。リザーブドインスタンスを変更す

る場合、変更後の設定のフットプリントは元の設定のフットプリントと一致する必要があり、一致しないと変更リクエストは処理されません。

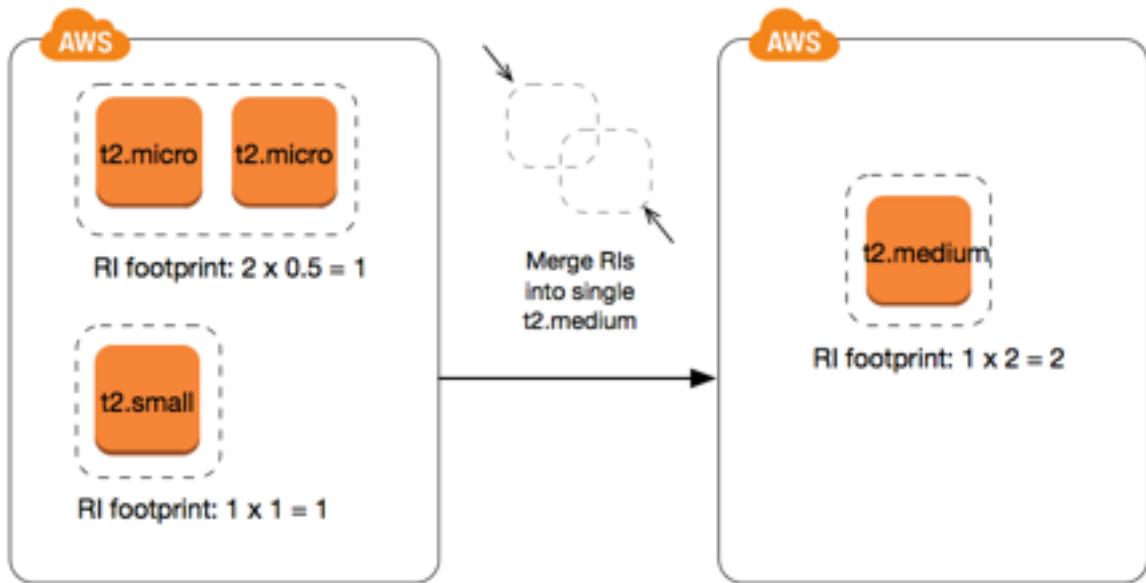
正規化係数は、インスタンスファミリー内のインスタンスサイズ（たとえば、m1 インスタンスファミリー内の m1.xlarge インスタンス）に基づきます。これは同じインスタンスファミリー内でのみ意味を持ちます。インスタンスファミリーを 1 つの型から別の型に変更することはできません。Amazon EC2 コンソールでは、正規化係数はユニットで測定されます。次の表では、インスタンスファミリー内で適用される正規化係数を示します。

インスタンスサイズ	正規化係数
nano	0.25
micro	0.5
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64
9xlarge	72
10xlarge	80
12xlarge	96
16xlarge	128
18xlarge	144
24xlarge	192
32xlarge	256

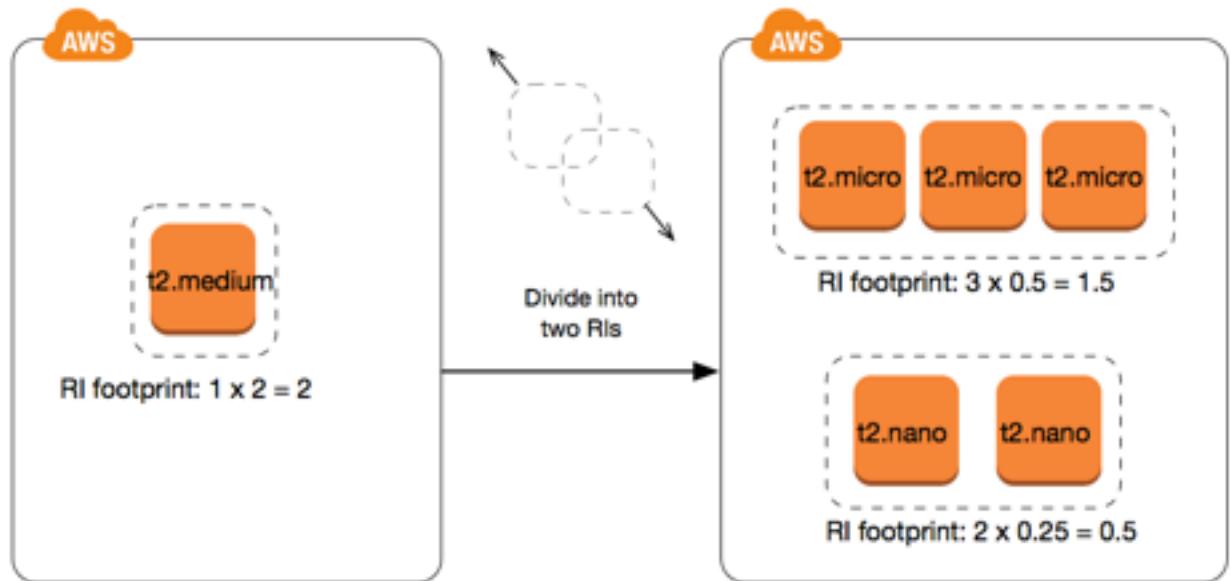
リザーブドインスタンスでインスタンスサイズのフットプリントを計算するには、インスタンスの数に正規化係数を掛けます。たとえば、t2.medium の正規化係数は 2 であるため、t2.medium インスタンス 4 個の予約は 8 ユニットのフットプリントを持ちます。

予約のインスタンスサイズのフットプリントが同じである場合は、同じインスタンスファミリー内で（たとえば、T2 インスタンスファミリー内など）、予約を異なるインスタンスサイズとして割り当てることができます。たとえば、1 つの t2.large (1 x 4) インスタンスの予約を 4 つの t2.small (4 x 1) インスタンスに分割したり、4 つの t2.small インスタンスの予約を 1 つの t2.large インスタンスに結合したりできます。ただし、2 つの t2.small (2 x 1) インスタンスの予約を 1 つの t2.large (1 x 4) インスタンスに変更することはできません。これは、現在の予約の既存のインスタンスサイズのフットプリントが、提案された予約よりも小さいためです。

次の例では、2 つの t2.micro インスタンスの予約（1 つに対するフットプリントが与えられる）と 1 つの t2.small インスタンスの予約（1 つに対するフットプリントが与えられる）があります。両方の予約を单一の t2.medium インスタンスにマージします。2 つの元の予約のインスタンスサイズフットプリントの組み合わせは、変更された予約のフットプリントと同等になります。



また、予約を変更して 2 つ以上の予約に分割することもできます。次の例では、t2.medium インスタンスへの予約があります。この予約を 2 つの t2.nano インスタンスのある予約と、3 つの t2.micro インスタンスがある予約に分割します。



### ペアメタルインスタンスの正規化係数

同じファミリー内で .metal リザーブドインスタンスを別のサイズに変更でき、また同様に、同じファミリー内の別のサイズの リザーブドインスタンスを .metal リザーブドインスタンスに変更できます。ペアメタルインスタンスは、同じインスタンスファミリーないの最大のインスタンスと同じサイズです。たとえば、i3.metal は i3.16xlarge と同じサイズであるため、同じ正規化係数があります。

#### Note

.metal インスタンスサイズには、単一の正規化係数がありません。特定のインスタンスファミリーに基づいて異なります。

ペアメタルインスタンスサイズ	正規化係数
c5.metal	192
i3.metal	128
r5.metal	192
r5d.metal	192
z1d.metal	96
m5.metal	192
m5d.metal	192

たとえば、1つの i3.metal インスタンスには 128 の正規化係数があります。i3.metal デフォルトテンプレート Amazon Linux/Unix リザーブドインスタンスを購入する場合、次のように予約を分割できます。

- i3.16xlarge は i3.metal インスタンスと同じサイズであるため、その正規化係数は 128 (128/1) です。1つの i3.metal インスタンスの予約は、1つの i3.16xlarge インスタンス内で変更できます。
- i3.8xlarge は i3.metal インスタンスの半分のサイズであるため、その正規化係数は 64 (128/2) です。1つの i3.metal インスタンスの予約は、2つの i3.8xlarge インスタンスに分割できます。
- i3.4xlarge は i3.metal インスタンスの 4 分の 1 のサイズであるため、その正規化係数は 32 (128/4) です。1つの i3.metal インスタンスの予約は、4つの i3.4xlarge インスタンスに分割できます。

## 変更リクエストの送信

リザーブドインスタンスを変更する前に、適用される「制約 (p. 307)」を必ず確認してください。インスタンスのサイズを変更する前に、変更する予約のすべてのインスタンスサイズフットプリント (p. 308) を計算し、その結果が希望する設定の全インスタンスサイズフットプリントと一致することを確認してください。

AWS マネジメントコンソールを使用して リザーブドインスタンスを変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [リザーブドインスタンス] ページで、変更する リザーブドインスタンスを 1 つ以上選択し、[アクション]、[リザーブドインスタンスの変更] の順に選択します。

### Note

リザーブドインスタンスがアクティブ状態ではない場合、または変更できない場合は、[リザーブドインスタンスの変更] が無効となります。

3. 変更テーブルの最初のエントリには、選択した リザーブドインスタンスの属性とその上部に少なくとも 1 つのターゲット設定が表示されます。[単位] 列には全インスタンスサイズのフットプリントが表示されます。追加する各新規設定で [追加] を選択します。各設定で必要に応じて属性を変更し、[続行] を選択します。
  - [スコープ]: 設定の適用先が 1 つのアベイラビリティゾーンまたはリージョン全体のどちらであるかを選択します。
  - [アベイラビリティゾーン]: 必要なアベイラビリティゾーンを選択します。リージョン リザーブドインスタンスには適用されません。
  - [インスタンスタイプ]: 必要なインスタンスタイプを選択します。組み合わせた設定は、元の設定のインスタンスサイズのフットプリントと等しくなければなりません。

- [カウント]: インスタンス数を指定します。リザーブドインスタンスを複数の設定に分割するには、カウントを減らし、[追加]を選択して、追加する設定のカウントを指定します。たとえば、カウントが 10 の設定が 1 つある場合、そのカウントを 6 に変更し、カウントが 4 の設定を別に追加できます。このプロセスでは、新しいリザーブドインスタンスがアクティブになった後で、元のリザーブドインスタンスを終了させます。
4. ターゲット設定の指定を完了したときに、変更の選択を確認するには、[Submit Modifications] を選択します。
  5. 変更リクエストのステータスは、リザーブドインスタンス画面の [状態] 列で確認できます。有効な状態には以下のものがあります。
    - アクティブ（変更の保留）—元のリザーブドインスタンスの移行状態
    - リタイヤ（変更の保留）—新しいリザーブドインスタンスを作成中の元のリザーブドインスタンスの移行状態
    - リタイヤー—リザーブドインスタンスは正常に変更され、置き換えられました
    - アクティブ—次のいずれかを選択します。
      - 正常な変更リクエストにより新しいリザーブドインスタンスが作成されました
      - 変更リクエストが失敗したため、元のリザーブドインスタンスです

コマンドラインを使用してリザーブドインスタンスを変更するには

1. リザーブドインスタンスを変更するには、次のコマンドの 1 つを使用できます。
  - [modify-reserved-instances](#) (AWS CLI)
  - [Edit-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)
2. 変更リクエスト (processing、fulfilled、または failed) のステータスを取得するには、以下のコマンドから 1 つを使用します。
  - [describe-reserved-instances-modifications](#) (AWS CLI)
  - [Get-EC2ReservedInstancesModification](#) (AWS Tools for Windows PowerShell)

## 変更リクエストのトラブルシューティング

リクエストしたターゲット設定が一意であれば、リクエストが処理されるメッセージを受信します。この時点では、Amazon EC2 は変更リクエストのパラメータが有効であることを確認しています。まだ、処理中に容量が利用できないために変更リクエストが失敗する可能性があります。

場合によって、確認の代わりに変更リクエストが不完全または失敗したことを示すメッセージが表示されることがあります。メッセージの情報を参考にして、別の変更リクエストを再送信します。リクエストを送信する前に、適用される制約 ([p. 307](#)) を必ず確認してください。

選択されたリザーブドインスタンスに変更できないものがあります

Amazon EC2 は変更できないリザーブドインスタンスを示します。このようなメッセージを受け取ったら、Amazon EC2 コンソールの [リザーブドインスタンス] ページ リザーブドインスタンスについての詳細情報を確認します。

変更リクエストの処理中にエラーが発生しました

送信したリザーブドインスタンス変更リクエストをすべて処理できません。変更している予約の数によつては、メッセージが異なる場合があります。

Amazon EC2 は変更リクエストを処理できない理由を示します。たとえば、変更しているリザーブドインスタンスの 1 つ以上のサブセットに同じターゲット設定 (アベイラビリティーゾーンとプラットフォームの組み合わせ) を指定したような場合です。予約のインスタンス詳細が一致し、変更対象のすべてのサブセットのターゲット設定が一意であることを確認して、変更リクエストの再送信を試みます。

## コンバータブルリザーブドインスタンス の交換

また、インスタンスファミリー、オペレーティングシステム、およびテナンシーを含む別の構成で、1つ以上の別のコンバータブルリザーブドインスタンスのコンバータブルリザーブドインスタンスに交換することもできます。交換先のコンバータブルリザーブドインスタンスが交換元のコンバータブルリザーブドインスタンスと同等あるいはそれ以上の値である限り、交換の実行回数に制限はありません。

コンバータブルリザーブドインスタンスを交換する場合、現在の予約のインスタンスの数はターゲットのコンバータブルリザーブドインスタンスの設定と同じあるいはそれ以上の値を含有する数のインスタンスと交換されます。Amazon EC2は、交換で受け取ったリザーブドインスタンスの数を計算します。

### 目次

- コンバータブルリザーブドインスタンス 交換の要件 (p. 313)
- コンバータブルリザーブドインスタンス 交換の計算 (p. 314)
- コンバータブルリザーブドインスタンス のマージ (p. 315)
- コンバータブルリザーブドインスタンス の一部の交換 (p. 315)
- 交換リクエストの送信 (p. 316)

### コンバータブルリザーブドインスタンス 交換の要件

以下の条件を満たしている場合に、Amazon EC2では交換リクエストが処理されます。コンバータブルリザーブドインスタンスは次のとおりである必要があります:

- アクティブ
- 保留中の以前の交換リクエストがないこと

以下のルールが適用されます。

- コンバータブルリザーブドインスタンスは AWSによって現在提供されている別のコンバータブルリザーブドインスタンスにのみ交換することができます。
- コンバータブルリザーブドインスタンスは特定のリージョンと関連付けられ、予約の期間中は固定されます。コンバータブルリザーブドインスタンスを別のリージョンのコンバータブルリザーブドインスタンスと交換することはできません。
- 1つのコンバータブルリザーブドインスタンスで1つ以上のコンバータブルリザーブドインスタンスを一度に交換することができます。
- コンバータブルリザーブドインスタンスの一部を交換するには、2つ以上の予約に変更して、1つ以上の予約を新しいコンバータブルリザーブドインスタンスに交換することができます。詳細については、「[コンバータブルリザーブドインスタンス の一部の交換 \(p. 315\)](#)」を参照してください。リザーブドインスタンスの変更の詳細については、「[リザーブドインスタンス の変更 \(p. 306\)](#)」を参照してください。
- 全額前払い コンバータブルリザーブドインスタンスは一部前払い コンバータブルリザーブドインスタンスに交換できます。その逆も可能です。

#### Note

交換に対する前払いの合計(差額)が 0.00 USD 未満の場合、差額が 0.00 USD 以上になるように、AWSによって自動的にコンバータブルリザーブドインスタンスのインスタンス数が提供されます。

#### Note

新しいコンバータブルリザーブドインスタンスの合計価格(前払い料金 + 時間料金 × 残り時間数)が交換元のコンバータブルリザーブドインスタンスの合計価格未満の場合、交換元の

コンバータブルリザーブドインスタンス の合計価格以上になるように AWS によって自動的にコンバータブルリザーブドインスタンス のインスタンス数が提供されます。

- より有利な料金を利用するため、前払いなし コンバータブルリザーブドインスタンス を全額前払いまたは一部前払い コンバータブルリザーブドインスタンス に交換できます。
- 全額前払いまたは一部前払い コンバータブルリザーブドインスタンス を前払いなし コンバータブルリザーブドインスタンス に交換することはできません。
- 前払いなし コンバータブルリザーブドインスタンス を別の前払いなし コンバータブルリザーブドインスタンス に交換できます。ただし、新しい コンバータブルリザーブドインスタンス の時間料金が交換元の コンバータブルリザーブドインスタンス の時間料金以上である場合に限ります。

#### Note

新しい コンバータブルリザーブドインスタンス の合計価格 (時間料金 × 残り時間数) が交換元の コンバータブルリザーブドインスタンス の合計価格未満の場合、交換元の コンバータブルリザーブドインスタンス の合計価格以上になるように AWS によって自動的にコンバータブルリザーブドインスタンス のインスタンス数が提供されます。

- 有効期限の異なる複数の コンバータブルリザーブドインスタンス を交換すると、新しい コンバータブルリザーブドインスタンス の有効期限は、将来の最も遅い日付になります。
- 1つの コンバータブルリザーブドインスタンス を交換する場合は、新しい コンバータブルリザーブドインスタンス と同じ期間 (1年または3年) が必要です。異なる期間を持つ複数の コンバータブルリザーブドインスタンス をマージすると、新しい コンバータブルリザーブドインスタンス の期間は3年となります。詳細については、「[コンバータブルリザーブドインスタンス のマージ \(p. 315\)](#)」を参照してください。

## コンバータブルリザーブドインスタンス 交換の計算

コンバータブルリザーブドインスタンス の交換は無料です。ただし、交換元の コンバータブルリザーブドインスタンス と交換先の コンバータブルリザーブドインスタンス の按分計算での前払い額に差額があれば、その差額を支払う必要があります。

それぞれの コンバータブルリザーブドインスタンス に定価があります。交換元と交換先の コンバータブルリザーブドインスタンス で定価が比較され、交換の結果として得られる コンバータブルリザーブドインスタンス の数が決まります。

たとえば、定価 35 USD の 1つの コンバータブルリザーブドインスタンス を定価 10 USD の新しいインスタンスタイプに交換するとします。

\$35/\$10 = 3.5

コンバータブルリザーブドインスタンス を 10 USD の 3つの コンバータブルリザーブドインスタンス に交換できます。半個単位で購入することはできないため、余り分を補うには、コンバータブルリザーブドインスタンス を追加購入する必要があります。

3.5 = 3 whole ##### + 1 additional ####.

4つ目の コンバータブルリザーブドインスタンス で、終了日が他の 3つのものと同じものだとすると、一部前払いまたは全額前払いの コンバータブルリザーブドインスタンス を交換する場合、その 4つ目のリザーブドインスタンス の料金を差額として支払います。交換元の コンバータブルリザーブドインスタンス の前払い額のうち 500 USD が残っており、交換先の コンバータブルリザーブドインスタンス の料金が按分計算で 600 USD になるとすると、100 USD が請求されます。

\$600 prorated upfront cost of new reservations - \$500 remaining upfront cost of original reservations = \$100 difference.

## コンバーティブルリザーブドインスタンス のマージ

2つ以上の コンバーティブルリザーブドインスタンス をマージする場合、新しい コンバーティブルリザーブドインスタンス の期間は元の コンバーティブルリザーブドインスタンス と同じであるか、元の コンバーティブルリザーブドインスタンス の最長の期間でなければなりません。新しい コンバーティブルリザーブドインスタンス の有効期間は、将来の最も長い有効期間となります。

たとえば、アカウントに以下の コンバーティブルリザーブドインスタンス があるとします。

リザーブドインスタンス ID	期間	有効期限日
aaaa1111	1年	2018-12-31
bbbb2222	1年	2018-07-31
cccc3333	3年	2018-06-30
dddd4444	3年	2019-12-31

- aaaa1111 と bbbb2222 をマージして、それらを 1 年の コンバーティブルリザーブドインスタンス と交換できます。それらを 3 年の コンバーティブルリザーブドインスタンス に交換することはできません。新しい コンバーティブルリザーブドインスタンス 有効期限は 2018-12-31 です。
- bbbb2222 と cccc3333 をマージして、それらを 3 年の コンバーティブルリザーブドインスタンス と交換できます。それらを 1 年の コンバーティブルリザーブドインスタンス に交換することはできません。新しい コンバーティブルリザーブドインスタンス 有効期限は 2018-07-31 です。
- cccc3333 と dddd4444 をマージして、それらを 3 年の コンバーティブルリザーブドインスタンス と交換できます。それらを 1 年の コンバーティブルリザーブドインスタンス に交換することはできません。新しい コンバーティブルリザーブドインスタンス 有効期限は 2019-12-31 です。

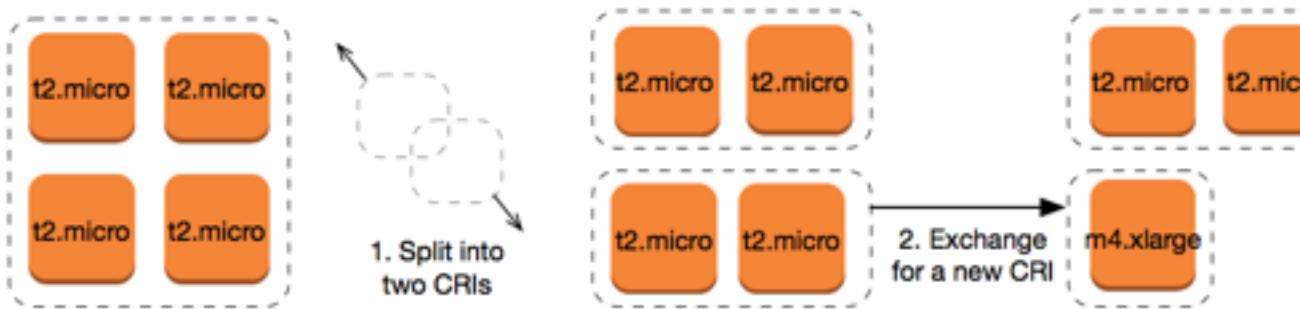
## コンバーティブルリザーブドインスタンス の一部の交換

変更プロセスを使用して、コンバーティブルリザーブドインスタンス をより小さい予約に分割し、新しい予約のうちの 1 つ以上を新しい コンバーティブルリザーブドインスタンス と交換することができます。次の例はそれを行う方法を示しています。

Example 例: 複数のインスタンスを持つ コンバーティブルリザーブドインスタンス

この例では、この例では、予約に 4 つのインスタンスがある t2.micro コンバーティブルリザーブドインスタンス があります。t2.micro インスタンスに対して 2 つの m4.xlarge インスタンスを交換するには:

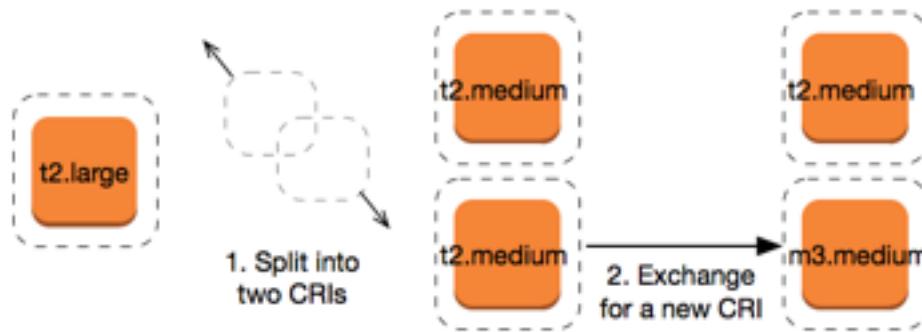
- t2.micro コンバーティブルリザーブドインスタンス を変更するには、それぞれ 2 つの t2.micro コンバーティブルリザーブドインスタンス に分割します。
- 新しい t2.micro コンバーティブルリザーブドインスタンス のいずれかを m4.xlarge コンバーティブルリザーブドインスタンス と交換します。



Example 例: 1 つのインスタンスを持つ コンバーティブルリザーブドインスタンス

この例では、t2.large コンバーティブルリザーブドインスタンス があります。小さな t2.medium インスタンスと m3.medium インスタンスに変更するには:

1. t2.large コンバーティブルリザーブドインスタンス を変更するには、2 つの t2.medium コンバーティブルリザーブドインスタンス に分割します。1 つの t2.large インスタンスに対して、2 つの t2.medium インスタンスと同じインスタンスサイズのフットプリントが含まれます。
2. 新しい t2.medium コンバーティブルリザーブドインスタンス のいずれかを m3.medium コンバーティブルリザーブドインスタンス と交換します。



詳細については、「[インスタンスサイズの変更のサポート \(p. 308\)](#)」および「[交換リクエストの送信 \(p. 316\)](#)」を参照してください。

## 交換リクエストの送信

Amazon EC2 コンソールまたはコマンドラインツールを使って、コンバーティブルリザーブドインスタンス を交換できます。

### コンソールを使用した コンバーティブルリザーブドインスタンス の交換

コンバーティブルリザーブドインスタンス 提供タイプを検索し、検索された選択肢の中から新しい設定を選択できます。

Amazon EC2 コンソールを使用して コンバーティブルリザーブドインスタンス を交換するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [リザーブドインスタンス] を選択し、交換する コンバーティブルリザーブドインスタンス を選び、[アクション]、[リザーブドインスタンス の交換] の順に選択します。
3. ドロップダウンメニューから希望する設定の属性を選び、[提供タイプの検索] を選択します。

- 新しいコンバーティブルリザーブドインスタンスを選択します。[インスタンス数]列には、交換で受け取るリザーブドインスタンスの数が表示されます。ニーズに応じるコンバーティブルリザーブドインスタンスを選択したら、[交換]を選択します。

交換元のリザーブドインスタンスが消え、新しいリザーブドインスタンスがAmazon EC2に表示されます。このプロセスが反映されるまでには数分かかることがあります。

#### コマンドラインインターフェイスを使用したコンバーティブルリザーブドインスタンスの交換

コンバーティブルリザーブドインスタンスを交換するには、まず自分のニーズに合ったターゲットコンバーティブルリザーブドインスタンスを見つけます。

- [describe-reserved-instances-offerings](#) (AWS CLI)
- [Get-EC2ReservedInstancesOffering](#) (Tools for Windows PowerShell)

交換で受け取るリザーブドインスタンスの数と交換時の差額の起案額を含む交換の見積りを取得します。

- [get-reserved-instances-exchange-quote](#) (AWS CLI)
- [GetEC2-ReservedInstancesExchangeQuote](#) (Tools for Windows PowerShell)

これで、交換を実行できます。

- [accept-reserved-instances-exchange-quote](#) (AWS CLI)
- [Confirm-EC2ReservedInstancesExchangeQuote](#) (Tools for Windows PowerShell)

## スケジュールされたリザーブドインスタンス

スケジュールされたリザーブドインスタンス(スケジュールされたインスタンス)によって、1年間にわたり毎日、毎週、または毎月ベースの指定された開始時間および期間で繰り返しキャパシティー予約を購入できます。あらかじめキャパシティーを予約しておき、必要なときに使用できるようにします。料金は、インスタンスを使用しなくても、インスタンスがスケジュールされた時間に対して支払います。

スケジュールされたインスタンスは、継続的には実行されないが定期的なスケジュールで実行されるワークフローに適しています。たとえば、営業時間中に実行するアプリケーションや週末に実行するバック処理に、スケジュールされたインスタンスを使用できます。

キャパシティーの予約が連続して必要な場合、リザーブドインスタンスがニーズを満たし、コストを削減できる可能性があります。詳細については、「[リザーブドインスタンス \(p. 279\)](#)」を参照してください。インスタンスをいつ実行するか特に決めていない場合、スポットインスタンスが要件を満たし、コストを削減できる可能性があります。詳細については、「[スポットインスタンス \(p. 320\)](#)」を参照してください。

### コンテンツ

- [スケジュールされたインスタンスとは何ですか？ \(p. 318\)](#)
- [スケジュールされたインスタンスのためのサービスにリンクされたロール \(p. 318\)](#)
- [スケジュールされたインスタンスの購入 \(p. 319\)](#)
- [スケジュールされたインスタンスの起動 \(p. 320\)](#)
- [スケジュールされたインスタンスの制限 \(p. 320\)](#)

## スケジュールされたインスタンスとは何ですか？

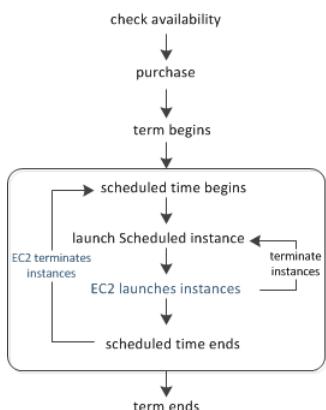
Amazon EC2 は、スケジュールされた EC2 インスタンスに使用するプールを別個に、各アベイラビリティーゾーンに設定します。各プールはインスタンスタイプ、オペレーティングシステム、ネットワークの特定の組み合わせをサポートします。

開始するには、使用可能なスケジュールを探す必要があります。複数のプールまたは単一のプールを検索できます。適したスケジュールを見つけたら、それを購入します。

スケジュールされた期間中に、購入したスケジュールの属性（インスタンスタイプ、アベイラビリティーゾーン、ネットワーク、およびプラットフォーム）に一致する起動設定を使用して、スケジュールされたインスタンスを起動する必要があります。すると、Amazon EC2 は指定された起動仕様に基づき、ユーザーに代わって EC2 インスタンスを起動します。Amazon EC2 は、現在のスケジュールされた期間の終了までに EC2 インスタンスが終了し、予約されている他のスケジュールされたインスタンスでキャパシティーを利用可能にする必要があります。したがって、Amazon EC2 は現在のスケジュールされた期間の終了 3 分前に EC2 インスタンスを終了します。

スケジュールされたインスタンスを停止または再起動することはできませんが、必要に応じて手動で終了することはできます。現在のスケジュールされた期間が終わる前にスケジュールされたインスタンスを終了した場合、数分後にもう一度起動できます。そうでない場合は、次のスケジュールされた期間まで待つ必要があります。

次の図は、スケジュールされたインスタンスのライフサイクルを示しています。



## スケジュールされたインスタンスのためのサービスにリンクされたロール

スケジュールされたインスタンスを購入すると、Amazon EC2 によってサービスにリンクされたロールが作成されます。サービスにリンクされたロールには、Amazon EC2 がユーザーに代わって他の AWS サービスを呼び出すために必要なすべての権限が含まれます。詳細については、『IAM ユーザーガイド』の「[サービスにリンクされたロールの使用](#)」を参照してください。

Amazon EC2 は、[AWSServiceRoleForEC2ScheduledInstances] というサービスにリンクされたロールを使用して、次のアクションを実行します。

- `ec2:TerminateInstances` - スケジュール完了後にスケジュールされたインスタンスを終了する
- `ec2:CreateTags` - スケジュールされたインスタンスにシステムタグを追加する

Amazon EC2 がこのサービスにリンクされたロールのサポートを開始した 2017 年 10 月以前にスケジュールされたインスタンスを購入した場合は、Amazon EC2 が AWS アカウントで [AWSServiceRoleForEC2ScheduledInstances] ロールを作成しています。詳細については、『IAM ユーザーガイド』の「[AWS アカウントに新しいロールが表示される](#)」を参照してください。

スケジュールされたインスタンスを使用する必要がなくなった場合は、  
[AWSServiceRoleForEC2ScheduledInstances] ロールを削除することをお勧めします。このロールがアカウントから削除された後で、スケジュールされたインスタンスを購入すると、Amazon EC2 はロールを再度作成します。

## スケジュールされたインスタンスの購入

スケジュールされたインスタンスを購入するには、スケジュールされたリザーブドインスタンスの予約 ウィザードを使用できます。

### Warning

スケジュールされたインスタンスの購入後、購入をキャンセル、変更、または再販売することはできません。

スケジュールされたインスタンス (コンソール) を購入するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[INSTANCES] の下にある [Scheduled Instances] を選択します。現在選択されているリージョンではスケジュールされたインスタンスがサポートされない場合、このページは利用できません。[詳細はこちら \(p. 320\)](#)
3. [Purchase Scheduled Instances] を選択します。
4. [Find available schedules] ページで、以下の操作を実行します。

a. [Create a schedule] の下で、[Starting on] から開始日を、[Recurring] からスケジュールの繰り返し (Daily、Weekly、または Monthly) を、[for duration] から最小期間を選択します。最小期間に指定された値が、スケジュールされたインスタンスの最低限必要な使用率 (1 年当たり 1,200 時間) を満たすことを、コンソールによって確認されることに注意してください。

#### Create a schedule

Starting on	<input type="text"/>	for duration	4		hours
<input type="checkbox"/> +/- 2 hours					
Recurring	Daily				

- b. [Instance details] で、[Platform] からオペレーティングシステムとネットワークを選択します。結果を絞り込むには、[Instance type] で 1 つ以上のインスタンスタイプを選択するか、[Availability Zone] で 1 つ以上のアベイラビリティーゾーンを選択します。

#### Instance details

Platform	Linux/UNIX (Amazon VPC)	Instance type	Any
Availability Zone	Any		

- c. [Find schedules] を選択します。
- d. [Available schedules] で、1 つまたは複数のスケジュールを選択します。選択した各スケジュールに対してインスタンス数を設定し、[Add to Cart] を選択します。
- e. カートは、ページの下部に表示されます。スケジュールをカートに追加およびカートから削除し終えたら、[Review and purchase] を選択します。
5. [Review and purchase] ページで、選択を確認し、必要に応じて編集します。完了したら、[Purchase] を選択します。

スケジュールされたインスタンス (AWS CLI) を購入するには

[describe-scheduled-instance-availability](#) コマンドを使用して、ニーズを満たす利用可能なスケジュールをリスト表示し、次に [purchase-scheduled-instances](#) コマンドを使用して購入を完了します。

## スケジュールされたインスタンスの起動

スケジュールされたインスタンスを購入すると、スケジュールされた期間中に起動できるようになります。

スケジュールされたインスタンス (コンソール) を起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[INSTANCES] の下にある [Scheduled Instances] を選択します。現在選択されているリージョンではスケジュールされたインスタンスがサポートされない場合、このページは利用できません。[詳細はこちら \(p. 320\)](#)
3. スケジュールされたインスタンスを選択し、[Launch Scheduled Instances] を選択します。
4. [Configure] ページで、スケジュールされたインスタンスの起動仕様を入力し、[Review] を選択します。

### Important

起動指定は、購入したスケジュールのインスタンスタイプ、アベイラビリティーゾーン、ネットワーク、およびプラットフォームに一致する必要があります。

5. [Review] ページで、起動設定を確認し必要に応じて変更します。完了したら、[Launch] を選択します。

スケジュールされたインスタンス (AWS CLI) を起動するには

[describe-scheduled-instances](#) コマンドを使用してスケジュールされたインスタンスをリスト表示し、次に [run-scheduled-instances](#) コマンドを使用して、スケジュールされた期間中にスケジュールされた各インスタンスを起動します。

## スケジュールされたインスタンスの制限

スケジュールされたインスタンスには以下の制限が適用されます。

- サポートされているインスタンスタイプは次のタイプのみです: C3、C4、M4 および R3。
- 必要な期間は 365 日 (1 年) です。
- 最低限必要な使用率は、1 年当たり 1,200 時間です。
- 事前に 3 か月までのスケジュールされたインスタンスを購入できます。
- 現在、これらの機能は、米国東部 (バージニア北部)、米国西部 (オレゴン)、欧州 (アイルランド) の各リージョンで利用できます。

## スポットインスタンス

スポットインスタンス は、オンデマンド価格より低価で利用できる未使用的 EC2 インスタンスです。スポットインスタンス では未使用的 EC2 インスタンスを静止状態割引でリクエストできるため、Amazon EC2 のコストを大幅に削減できます。スポットインスタンス の時間単位の使用料金はスポット料金と呼ばれます。各アベイラビリティーゾーンにおけるそれぞれのインスタンスタイプのスポット料金は、Amazon EC2 によって設定され、スポットインスタンス の長期供給と需要に基づいて徐々に調整されます。スポットインスタンス は、キャパシティーが使用可能でリクエストの 1 時間あたりの上限価格がスポット料金を超えるたびに実行されます。

スポットインスタンス は、アプリケーションを実行する時間に柔軟性がある場合や、アプリケーションを中断できる場合に、費用効率の高い選択肢です。たとえば、スポットインスタンス は、データ分

析、バッチジョブ、バックグラウンド処理、およびオプションタスクに適しています。詳細については、「[Amazon EC2 スpotトインスタンス](#)」を参照してください。

## トピック

- [概念 \(p. 321\)](#)
- [開始方法 \(p. 322\)](#)
- [関連サービス \(p. 323\)](#)
- [価格決定と削減額 \(p. 323\)](#)

## 概念

スポットインスタンス の使用を開始する前に、以下の概念を理解しておく必要があります。

- **スポットインスタンス プール** – 同様のインスタンスタイプ (`m5.large` など)、オペレーティングシステム、アベイラビリティゾーン、ネットワークプラットフォームの一連の使われていない EC2 インスタンスです。
- **スポット料金** – 時間当たりの スpotトインスタンス の現在の料金です。
- **スポットインスタンス リクエスト** – (またはスポット入札) とは、スポットインスタンス に対して支払う 1 時間あたりの上限料金を提供します。上限料金を指定しない場合、デフォルトの上限料金はオンデマンド価格となります。お客様のリクエストの時間あたりの上限料金がスポット料金を超える場合で、容量がご利用可能な場合、Amazon EC2 はお客様のリクエストを受理します。スポットインスタンス リクエストは、1 回限りまたは永続的です。Amazon EC2 は、リクエストに関連付けられた スpotトインスタンス が終了した後、自動的に永続スポットリクエストを再送信します。スポットインスタンス リクエストにはオプションで スpotトインスタンス の継続期間を指定できます。
- **スポットフリート** – 指定した条件によって起動された一連の スpotトインスタンス です。スポットフリート は必要条件に合った スpotトインスタンス プールを選択して、フリートのターゲット容量を満たすまで スpotトインスタンス を起動します。デフォルトでは、スポットフリート は、フリートの スpotトインスタンス が削除された後に代替インスタンスを作成することによってターゲット容量が維持されるように設定されています。インスタンスの削除後に保持されないワンタイムリクエストとして スpotトフリート を送信することもできます。オンデマンドインスタンス リクエストに スpotトフリート リクエストを含めることができます。
- **スポットインスタンス の中断** – スpotト料金がお客様のリクエストの上限料金を超えた場合、または容量が使用できなくなった場合、Amazon EC2 は スpotトインスタンス を終了、停止、または休止状態にします。Amazon EC2 では、スポットインスタンス の中断通知が表示されます。それにより、インスタンスの停止前に 2 分の警告期間が与えられます。

## スポットインスタンス と オンデマンドインスタンス の主な違い

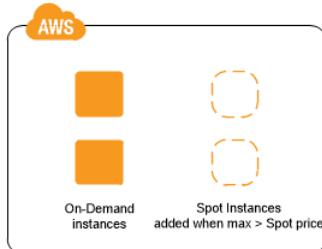
次のテーブルは、スポットインスタンス と オンデマンドインスタンス の主な違いをまとめたものです。

	スポットインスタンス	オンデマンドインスタンス
作成時刻	スポットリクエストがアクティブであり、容量が利用可能である場合に限り、即時に起動できます。	手動で起動リクエストを実行し、容量が利用可能である場合に限り、即時に起動できます。
使用可能な容量	容量が利用可能ではない場合、スポットリクエストは、容量が利用可能になるまで継続して自動的に起動リクエストを実行します。	起動リクエストを行うときに容量が利用可能でない場合は、容量不足エラー (ICE) が表示されます。

	スポットインスタンス	オンデマンドインスタンス
時間料金	スポットインスタンス の時間単位の使用料金は、オンデマンドに応じて異なります。	オンデマンドインスタンス の時間単位の使用料金は固定です。
インスタンスの中断	Amazon EBS-Backedスポットインスタンスは停止して起動することができます。また、キャパシティーが利用できなくなった場合、スポット料金が上限価格を超過する場合、またはスポットインスタンスへの需要が増大した場合に、Amazon EC2 スポットサービスは個別のスポットインスタンスを中止 (p. 385) できます。	お客様は、いつオンデマンドインスタンスが中止 (停止、休止、または終了) されるかを決定します。

## スポットインスタンスを使用した戦略

アプリケーションのコンピューティングリソースの最低保証レベルを維持する 1 つの戦略としては、オンデマンドインスタンス のコアグループを起動して、機会が発生したときにスポットインスタンスでコンピューティングリソースを補完する方法があります。



もう 1 つの戦略は、指定された期間 (スポットブロックとも呼ばれます) でスポットインスタンス を起動することです。これは、中断されず、選択した期間継続して実行されるように設計されています。まれに Amazon EC2 のキャパシティーの都合でスポットブロックの中止が発生する場合があります。この場合、インスタンスを削除する前に 2 分間の警告が与えられ、使用した場合でも、削除したインスタンスについては課金されません。詳細については、「[スポットインスタンスの継続期間の定義 \(p. 336\)](#)」を参照してください。

## 開始方法

最初に必要なことは、Amazon EC2 を使用するようにセットアップすることです。また、スポットインスタンス を起動する前に、オンデマンドインスタンス を起動した経験があると役立ちます。

### 起動と実行

- [Amazon EC2 でのセットアップ \(p. 22\)](#)
- [Amazon EC2 Linux インスタンスの開始方法 \(p. 26\)](#)

### スポットの基本

- [スポットインスタンスの仕組み \(p. 324\)](#)
- [スポットフリートの詳細 \(p. 326\)](#)

### スポットインスタンスの操作

- [中止に対する準備 \(p. 388\)](#)

- [スポットインスタンス リクエストを作成する \(p. 339\)](#)
- [リクエストステータス情報の取得 \(p. 383\)](#)

### スポットフリート の操作

- [スポットフリート の前提条件 \(p. 346\)](#)
- [スポットフリート リクエストを作成する \(p. 350\)](#)

## 関連サービス

Amazon EC2 を使用して スポットインスタンス を直接プロビジョニングすることができます。また、AWS の他のサービスを使用して スポットインスタンス をプロビジョニングすることもできます。詳細については、次のドキュメントを参照してください。

### Amazon EC2 Auto Scaling および スポットインスタンス

Amazon EC2 Auto Scaling で スポットインスタンス を起動できるように、支払う予定の上限価格を 使用して起動テンプレートまたは起動設定を作成できます。詳細については、『Amazon EC2 Auto Scaling ユーザーガイド』の「Auto Scaling グループでの スポットインスタンス の起動」および「複数のインスタンスタイプと購入オプションを使用する」を参照してください。

### Amazon EMR および スポットインスタンス

シナリオによっては、Amazon EMR クラスターで スポットインスタンス を実行すると便利な場合があります。詳細については、『Amazon EMR 管理ガイド』の「[スポットインスタンス](#)」および「[スポットインスタンス はどのような場合に使用しますか?](#)」を参照してください。

### AWS CloudFormation テンプレート

AWS CloudFormation によって、JSON 形式のテンプレートを利用して、AWS リソースのコレクションを作成および管理できます。AWS CloudFormation テンプレートには、お支払いいただく上料金が含まれます。詳細については、「[EC2 スポットインスタンス の更新 - Auto Scaling と CloudFormation の統合](#)」を参照してください。

### AWS SDK for Java

Java プログラミング言語を使用して、スポットインスタンス を管理できます。詳細については、「[チュートリアル: Amazon EC2 スポットインスタンス](#)」と「[チュートリアル: Amazon EC2 スポットリクエストの高度な管理](#)」を参照してください。

### AWS SDK for .NET

.NET プログラミング環境を使用して、スポットインスタンス を管理できます。詳細については、「[チュートリアル: Amazon EC2 スポットインスタンス](#)」を参照してください。

## 価格決定と削減額

スポットインスタンス はスポット料金で課金されます。これは Amazon EC2 によって設定され、スポットインスタンス の長期供給と需要に基づいて徐々に調整されます。お客様のリクエストの時間あたりの上料金が現在のスポット料金を超える場合で、容量がご利用可能な場合、Amazon EC2 はお客様のリクエストを受理します。スポットインスタンス は、お客様が自らスポットインスタンス を終了するか、容量が使用できなくなるか、スポット料金が上限料金を超えるか、縮小時に Amazon EC2 Auto Scaling グループ のインスタンスが削除されるまで実行されます。

継続期間を事前定義した スポットインスタンス には、固定時間単価が使用され、この価格は スポットインスタンス の実行中は有効なままになります。

お客様 または Amazon EC2 が実行中の スポットインスタンス を中断した場合、使用したオペレーティングシステムおよび スポットインスタンス を中断したユーザーに応じて、使用した時間 (秒) または 1 時間

の料金が請求されます。詳細については、「[中断された スポットインスタンスの請求 \(p. 391\)](#)」を参照してください。

## 料金の表示

AWS リージョンおよびインスタンスタイプごとの最新の(5分ごとに更新される)最低スポット料金を確認するには、「[スポットインスタンス 料金表](#)」ページを参照してください。

過去3か月間のスポット価格の履歴を表示するには、Amazon EC2 コンソールを使用するか、`describe-spot-price-history` コマンド(AWS CLI)を使用します。詳細については、「[スポットインスタンスの価格設定履歴 \(p. 333\)](#)」を参照してください。

AWS アカウントごとに、個別にアベイラビリティーボーンがコードにマッピングされます。したがって、アカウント間で同じアベイラビリティーボーンコード(たとえば、us-west-2a)に対して結果が異なる場合があります。

## 削減額の表示

単一の スポットフリート またはすべての スpotインスタンス に対して スpotインスタンス を使用することによって得られた削減額を表示することができます。過去1時間または過去3日間の削減状況を表示でき、vCPU 時間あたりの平均コストとメモリ(GiB) 時間あたりの平均コストも確認できます。削減額が予想されますが、使用状況に対する請求の調整が含まれていないため、実際の削減額と異なる場合があります。削減額情報の表示の詳細については、「[スポットインスタンス 購入による削減額 \(p. 333\)](#)」を参照してください。

## 請求書の表示

請求書を確認するには、[AWS アカウントアクティビティページ](#)を参照してください。請求書には、料金の明細が記載された使用状況レポートへのリンクが記載されています。詳細については、「[AWS アカウント請求](#)」を参照してください。

AWS の請求、アカウント、イベントについてご質問がある場合は、[AWS サポートにお問い合わせください](#)。

## スポットインスタンス の仕組み

スポットインスタンス を使用するには、スポットインスタンス リクエストまたはスポットフリート リクエストを作成します。このリクエストには、1インスタンスあたり1時間あたりの支払い上限料金(デフォルトはオンデマンド価格)とその他の制約(インスタンスタイプやアベイラビリティーボーンなど)を含めます。上限料金が指定されたインスタンスの現在のスポット料金を超える場合、リクエストは即座に受理されます。それ以外の場合、上限料金がスポット料金を超える場合、リクエストは受理されません。スポットインスタンスは、お客様がスポットインスタンスを停止または終了するか、Amazon EC2 によってスポットインスタンスが中断される(スポットインスタンスの中止とも呼ばれる)まで実行されます。

スポットインスタンス を使用するときは、中断に備える必要があります。Amazon EC2 は、スポット料金が上限料金を上回ったとき、スポットインスタンス に対する需要が上昇したとき、スポットインスタンス の供給が低下したときに、スポットインスタンス を中断する可能性があります。Amazon EC2 では、スポットインスタンス が中断され、スポットインスタンス の中断通知が表示されます。それにより、Amazon EC2 の中断前にインスタンスに2分の警告期間が与えられます。スポットインスタンス の削除保護を有効にすることはできません。詳細については、「[スポットインスタンス の中断 \(p. 385\)](#)」を参照してください。

Amazon EBS-backed インスタンスは、停止、起動、再起動、または終了できます。スポットサービスは、中断したときにスポットインスタンスを停止、終了、または休止状態にすることができます。

### 目次

- [起動グループでの スpotインスタンス の起動 \(p. 325\)](#)

- [アベイラビリティーゾーングループでの スポットインスタンス の起動 \(p. 325\)](#)
- [VPC での スポットインスタンス の起動 \(p. 325\)](#)

## 起動グループでの スポットインスタンス の起動

スポットインスタンスリクエストで起動グループを指定することによって、すべてを起動できる場合にのみ一連のスポットインスタンスを起動するよう Amazon EC2 に指示できます。また、スポットサービスが起動グループ内のインスタンスの 1 つを終了する必要がある場合（スポット料金が上限料金を超えた場合など）、すべてのインスタンスを終了する必要があります。ただし、お客様が起動グループ内の 1 つ以上のインスタンスを終了する場合、Amazon EC2 は起動グループ内のその他のインスタンスを終了しません。

このオプションは便利な場合もありますが、この制約を追加することによって、スポットインスタンスリクエストが受理される可能性は低くなり、スポットインスタンスが削除される可能性が高くなります。たとえば、起動グループに複数のアベイラビリティーゾーンのインスタンスが含まれるとします。これらのアベイラビリティーゾーンのいずれかのキャパシティーが減少して使用できなくなった場合、Amazon EC2 は起動グループのすべてのインスタンスを終了します。

以前に成功したリクエストと同じ（既存の）起動グループを指定する別の正常なスポットインスタンスリクエストを作成する場合、新しいインスタンスはこの起動グループに追加されます。したがって、この起動グループ内のインスタンスが終了されると、起動グループ内のすべてのインスタンスが終了します。これには、最初のリクエストと 2 番目リクエストによって起動されたすべてのインスタンスが含まれます。

## アベイラビリティーゾーングループでの スポットインスタンス の起動

同じアベイラビリティーゾーン内の一連のスポットインスタンスを起動するようにスポットサービスに通知するには、スポットインスタンスリクエストでアベイラビリティーゾーングループを指定します。Amazon EC2 は、アベイラビリティーゾーングループ内のすべてのインスタンスを同時に中断する必要がありません。Amazon EC2 がアベイラビリティーゾーングループ内のいずれかのインスタンスを中断する場合、他のインスタンスはそのまま実行されます。

このオプションは便利な場合もありますが、この制約を追加することによって、スポットインスタンスリクエストが受理される可能性が低くなる場合があることに注意してください。

アベイラビリティーゾーングループを指定したが、スポットインスタンスリクエストでアベイラビリティーゾーンを指定しない場合、その結果は、指定したネットワークによって異なります。

### デフォルト VPC

Amazon EC2 は、指定されたサブネットのアベイラビリティーゾーンを使用します。サブネットを指定しなかった場合は、アベイラビリティーゾーンとそのデフォルトのサブネットが選択されますが、最低価格のゾーンではない可能性があります。アベイラビリティーゾーンのデフォルトのサブネットを削除した場合は、別のサブネットを指定する必要があります。

### デフォルトではない VPC

Amazon EC2 は、指定されたサブネットのアベイラビリティーゾーンを使用します。

## VPC での スポットインスタンス の起動

スポットインスタンスのサブネットを指定するのと同じ方法で、オンデマンドインスタンスのサブネットを指定します。

- デフォルトの上限価格（オンデマンド価格）を使用するか、VPC 内のスポットインスタンスのスポット料金履歴に基づいて上限価格を設定する必要があります。
- [デフォルトの VPC] 特定の低価格のアベイラビリティーゾーンでスポットインスタンスを起動する場合、スポットインスタンスリクエストで対応するサブネットを指定する必要があります。サブネットを指定しなかった場合、Amazon EC2 によってサブネットが選択されますが、このサブネットのアベイラビリティーゾーンのスポット料金は最低ではない可能性があります。

- [デフォルト以外の VPC] スポットインスタンス のサブネットを指定する必要があります。

## スポットフリート の詳細

スポットフリート は、スポットインスタンス (オプションでは オンデマンドインスタンス) のコレクションまたはフリートです。

スポットフリート は、スポットフリート リクエストで指定した容量ターゲットを満たすような スポットインスタンスと オンデマンドインスタンス の数を起動しようと試みます。スポットインスタンスへのリクエストは、利用可能な容量があり、リクエストで指定した上限料金がスポット料金を超えている場合に達成されます。また、スポットインスタンス が中断した場合、スポットフリート はターゲット容量フリーを維持しようとします。

フリートに対する 1 時間あたりの支払い上限料金を設定し、上限料金に達するまで スpotフリート で インスタンスを起動することもできます。支払い上限料金に達すると、ターゲット容量に満たない場合でも、フリートはインスタンスの起動を停止します。

スポットインスタンス プールは、同様のインスタンスタイプ (`m5.large` など)、オペレーティングシステム、アベイラビリティーゾーン、ネットワークプラットフォームの一連の使われていない EC2 インスタンスです。スポットフリート のリクエストを行う場合に複数の起動条件を含めることができます。これにはインスタンスタイプ、AMI、アベイラビリティーゾーン、またはサブネットがあります。スポットフリート は、スポットフリート リクエストとその スpotフリート リクエストの設定を含む起動条件に基づいてリクエストを満たすために使用されるスポットインスタンス プールを選択します。スポットインスタンス は選択されたプールから取得されます。

### コンテンツ

- [スポットフリート でのオンデマンド \(p. 326\)](#)
- [スポットインスタンス の配分戦略 \(p. 327\)](#)
- [スポット料金の優先 \(p. 328\)](#)
- [使用量の管理 \(p. 328\)](#)
- [スポットフリート インスタンスの分量指定 \(p. 329\)](#)
- [チュートリアル: スpotフリート を使ってインスタンスの分量を指定する \(p. 330\)](#)

## スポットフリート でのオンデマンド

インスタンスのキャパシティーを常に確保するには、オンデマンドキャパシティーのリクエストを スpotフリート リクエストに含めることができます。スポットフリート リクエストでは、希望するターゲットキャパシティーとそのキャパシティーのうちどのくらいが オンデマンドであるかを指定します。このバランスは、利用可能な Amazon EC2 キャパシティーと可用性がある場合に起動されるスポットキャパシティーで構成されます。たとえば、スポットフリート リクエストでターゲットキャパシティーを 10、オンデマンドキャパシティーを 8 と指定すると、Amazon EC2 は 8 キャパシティーユニットを オンデマンドとして、2 キャパシティーユニット (10-8=2) を スpot として起動します。

### オンデマンド容量に基づくインスタンスタイプの優先順位付け

スポットフリート で オンデマンド容量を達成する場合、デフォルトで最低価格のインスタンスタイプが最初に起動されます。OnDemandAllocationStrategy を prioritized に設定すると、スポットフリート は 優先度に従って、オンデマンド容量を達成するために最初に使用するインスタンスタイプを決定します。優先度は起動テンプレートの上書きに割り当てられ、最も高い優先度が最初に起動されます。

たとえば、3 つの起動テンプレートの上書きにそれぞれ異なるインスタンスタイプとして `c3.large`、`c4.large`、`c5.large` を設定したとします。`c5.large` の オンデマンド価格は、`c4.large` より低価格です。`c3.large` が最低価格です。順番の決定に優先度を使用しない場合、フリートは オンデマンド容量を達成するために最初に `c3.large` を起動し、次に `c5.large` を起動します。`c4.large`

のリザーブドインスタンスは未使用のことが多いため、起動テンプレートの上書きの優先度を設定し、c4.large、c3.large、c5.large の順にすることができます。

## スポットインスタンスの配分戦略

スポットフリートのスポットインスタンスの配分戦略は、起動条件によるスポットインスタンスプールからどのようにスポットフリートのリクエストを満たすかについて決定します。以下に、スポットフリートリクエストで指定できる配分戦略を示します。

`lowestPrice`

スポットインスタンスは、最低価格のプールから取得されます。これはデフォルトの戦略です。

`diversified`

スポットインスタンスはすべてのプールに分散されます。

`capacityOptimized`

スポットインスタンスは、起動するインスタンスの数に最適な容量のスポットインスタンスプールから取得されます。

`InstancePoolsToUseCount`

スポットインスタンスは、指定した数のスポットプールに分散されます。このパラメータは `lowestPrice` と組み合わせて使用する場合にのみ有効です。

## ターゲット容量を維持する

スポット料金やスポットインスタンスプールの使用可能な容量の変動に伴ってスポットインスタンスが終了すると、`maintain`型のスポットフリートによって代替スポットインスタンスが起動されます。配分戦略が `lowestPrice` である場合、スポット群は、スポット料金が現在最低値のプールに代替インスタンスを起動します。配分戦略が `diversified` である場合には、フリートは残りのプールに代替スポットインスタンスを分散します。配分戦略が `lowestPrice` と `InstancePoolsToUseCount` の組み合わせである場合、フリートは最低価格のスポットプールを選択し、指定した数のスポットプールでスポットインスタンスを起動します。

## コスト最適化のためのスポットフリートの設定

スポットインスタンスの使用コストを最適化するには、`lowestPrice` 配分戦略を指定し、スポットフリートが現在のスポット料金に基づいてインスタンスタイプおよびアベイラビリティゾーンの最も安価な組み合わせを自動的にデプロイするようにします。

オンデマンドインスタンスのターゲット容量では、スポットフリートはスポットインスタンスの配分戦略 (`lowestPrice`、`capacityOptimized`、または `diversified`) を引き続き採用しながら、公開オンデマンド価格に基づいて最もコストが低いインスタンスタイプを常に選択します。

## コスト最適化と分散のためのスポットフリートの設定

安価で同時に分散型のスポットインスタンスのフリートを作成するには、`lowestPrice` 配分戦略を `InstancePoolsToUseCount` と組み合わせて使用します。スポットフリートは、現在のスポット料金に基づく最も安いインスタンスタイプとアベイラビリティゾーンの組み合わせを、指定した数のスポットプールに自動的にデプロイします。この組み合わせを使用することで、最も高価なスポットインスタンスを回避できます。

## 容量最適化のためのスポットフリートの設定

スポットインスタンスでは、価格は需要と供給の長期的な傾向に基づいて時間の経過とともに緩やかに変動しますが、容量はリアルタイムで変動します。`capacityOptimized` 戰略では、リアルタイムの容量データを調べ、可用性の最も高いプールを予測することで、そのプールからスポットインスタンスを自動的に起動します。この戦略は、作業の再開とチェックポイントの設定に関連する中断のコスト

が高くなる可能性のあるワークロード (ビッグデータと分析、画像とメディアのレンダリング、機械学習、ハイパフォーマンスコンピューティングなど) に適しています。中断の可能性を低くすることにより、capacityOptimized 戦略ではワークロードの全体的なコストを削減できます。

### 適切な配分戦略の選択

ユースケースに基づいて スポットフリート を最適化できます。

フリートが小さい場合、または短時間の実行である場合、すべてのインスタンスが単一の スポットインスタンス プールにあるとしても、スポットインスタンス が中断されることがあります。これにより、lowestPrice 戰略は、低コストを提供している期間に条件に合いやすくなります。

フリートが大サイズ、または長期間実行される場合には、複数のプールに スポットインスタンス を分散することでフリートの可用性を改善できます。たとえば、スポットフリート リクエストの条件が 10 プールとして、ターゲット容量が 100 インスタンスとすると、フリートはプールごとに 10 個の スポットインスタンス を起動します。1 つのプールのスポット料金がこのプールの上限料金を超える場合、フリートの 10% のみに影響がおよびます。この戦略を使用すると、いずれのプールにおいても経時にフリートが受けるスポット料金の上昇の影響を減少させます。

diversified 戰略では、スポットフリート は、[オンデマンド価格](#)以上のスポット料金のいずれのプールにも スポットインスタンス を起動しません。

安価で分散型のフリートを作成するには、lowestPrice 戰略を InstancePoolsToUseCount と組み合わせて使用します。スポットインスタンス には少数または多数のスポットプールを選択して割り当てることができます。たとえば、バッチ処理を実行する場合は、少数のスポットプール (InstancePoolsToUseCount=2 など) を指定することをお勧めします。これにより、キューのコンピューティング性能を常に確保しながら、削減額を最大化できます。ウェブサービスを実行する場合は、多数のスポットプール (InstancePoolsToUseCount=10 など) を指定し、スポットインスタンス が一時的に使用不可になった場合の影響を最小限に抑えることをお勧めします。

作業の再開とチェックポイント設定に関連する中断に伴うコストが高くなる可能性があるワークロードをフリートで実行している場合は、capacityOptimized 戰略を使用します。この戦略では中断の可能性を低くすることにより、ワークロードの全体的なコストを削減できます。

### スポット料金の優先

各 スポットフリート リクエストには、グローバルな最大価格を含めることも、デフォルト (オンデマンド価格) を使用することもできます。スポットフリート は、これを各起動仕様のデフォルトの最高価格として使用します。

任意で 1 つまたは複数の起動条件に上限料金を指定することができます。これは、起動条件に指定された料金です。起動条件に特定の料金が含まれる場合、スポットフリート は起動条件の上限料金としてこの料金を使用し、全体の上限料金に優先することになります。特定の上限料金を含まないそのほかの起動条件は、全体の上限料金を引き続き使用することにご注意ください。

### 使用量の管理

ターゲット容量または支払い上限料金に達すると、スポットフリート はインスタンスの起動を停止します。フリートに支払う 1 時間あたりの料金を管理するには、スポットインスタンス の SpotMaxTotalPrice と オンデマンドインスタンス の OnDemandMaxTotalPrice を指定できます。上限の合計料金に達すると、ターゲット容量に満たない場合でも、スポットフリート はインスタンスの起動を停止します。

以下の例は、2 つの異なるシナリオを示しています。最初の例では、ターゲット容量に達すると、スポットフリート はインスタンスの起動を停止します。2 番目の例では、支払い上限料金に達すると、スポットフリート はインスタンスの起動を停止します。

例: ターゲット容量に達したときにインスタンスの起動を停止する

m4.large オンデマンドインスタンス に対するリクエストの内容が以下のとおりとします。

- ・ オンデマンド料金: 1 時間あたり 0.10 USD
- ・ OnDemandTargetCapacity: 10
- ・ OnDemandMaxTotalPrice: 1.50 USD

スポットフリートは 10 オンデマンドインスタンスを起動します。合計料金 1.00 USD (10 インスタンス × 0.10 USD) は OnDemandMaxTotalPrice (1.50 USD) を超えないためです。

例: 上限の合計料金に達したときにインスタンスの起動を停止する

m4.large オンデマンドインスタンスに対するリクエストの内容が以下のとおりとします。

- ・ オンデマンド料金: 1 時間あたり 0.10 USD
- ・ OnDemandTargetCapacity: 10
- ・ OnDemandMaxTotalPrice: 0.80 USD

スポットフリートがオンデマンドターゲット容量 (10 オンデマンドインスタンス) を起動した場合、1 時間あたりの合計コストは 1.00 USD になります。これは OnDemandMaxTotalPrice として指定した料金 (0.80 USD) を超えます。支払い可能な額を超えないように、スポットフリートは 8 オンデマンドインスタンス (オンデマンドターゲット容量未満) を起動します。これを超えて起動すると、OnDemandMaxTotalPrice を超えてしまいます。

## スポットフリート インスタンスの分量指定

スポットインスタンスのフリートをリクエストするとき、それぞれのインスタンスタイプがアプリケーションのパフォーマンスに応じるように容量ユニットを定義し、また、インスタンス分量指定を利用してスポットインスタンス プールごとに上限価格を調整できます。

デフォルトでは、指定したスポット料金は 1 インスタンス時間当たりの入札料金を表します。インスタンスの分量指定機能を使用すると、指定した料金は ユニット時間ごとの料金となります。ユニット時間あたりの入札価格は、インスタンスタイプの入札価格をそのユニットの数で割って計算できます。スポットフリートでは、ターゲット容量をインスタンス分量で割ることで起動するスポットインスタンスの数を計算します。その結果が整数でなければ、スポットフリートはその数を次の整数に切り上げ、これによりフリートのサイズがターゲット容量以上になります。起動されたインスタンスの容量がリクエストされたターゲット容量を超えた場合でも、スポットフリートは起動仕様で指定したどのプールでも選択できます。

以下の表では、スポットフリート リクエストのターゲット容量が 10 の場合のユニットあたりの料金を算定する例を示します。

インスタンスタイプ	インスタンスの分量	インスタンス時間あたりのスポット料金	ユニット時間あたりの価格	起動されたインスタンスの数
r3.xlarge	2	0.05 USD	.025 (.05 ÷ 2)	5 (10 ÷ 2)

インスタンスタイプ	インスタンスの分量	インスタンス時間あたりのスポット料金	ユニット時間あたりの価格	起動されたインスタンスの数
r3.8xlarge	8	0.10 USD	.0125 (.10 ÷ 8)	2 (10 ÷ 8、結果切り上げ)

次に示すように、スポットフリートを使用して、受理時のユニットごとの最低価格のプールに指定するターゲット容量をプロビジョニングします。

1. スポットフリートのターゲット容量を、インスタンス(デフォルト)あるいは仮想 CPU、メモリ、ストレージまたはスループットからご希望のユニットで設定します。
2. ユニットあたりの料金を設定します。
3. 各起動設定で、インスタンスタイプがターゲット容量に対して必要なユニット数である分量を指定します。

#### インスタンスの分量指定例

次の設定のスポットフリートを検討します。

- ターゲット容量 24
- `r3.2xlarge` のインスタンスタイプの起動条件と分量 6
- `c3.xlarge` のインスタンスタイプの起動条件と分量 5

分量とは、インスタンスタイプがターゲット容量に対して必要なユニット数を表します。最初の起動条件がユニットあたりの料金を最低値で提供する場合(インスタンス時間あたりの `r3.2xlarge` の料金を 6 で割ったもの)、スポットフリートはこれらのインスタンスから 4 つを起動します(24 を 6 で割ったもの)。

2 番目の起動条件がユニットあたりの料金を最低値で提供する場合(インスタンス時間あたりの `c3.xlarge` の料金を 5 で割ったもの)、スポットフリートはこれらのインスタンスから 5 つを起動します(24 を 5 で割ったもの、結果が切り上げられる)。

#### インスタンスの分量指定と配分戦略

次の設定のスポットフリートを検討します。

- ターゲット容量 30
- `c3.2xlarge` のインスタンスタイプの起動条件と分量 8
- `m3.xlarge` のインスタンスタイプの起動条件と分量 8
- `r3.xlarge` のインスタンスタイプの起動条件と分量 8

スポットフリートは、4 つのインスタンスを起動します(30 を 8 出割ったもの、結果を切り上げ)。`lowestPrice` 戰略では、すべての 4 つのインスタンスはユニットあたりの最低価格を提供するプールから取得されます。`diversified` 戰略では、スポットフリートは 3 プールごとに 1 つのインスタンスを起動し、そしてプールのいずれかから取得された 4 つ目のインスタンスがユニットあたりの最低スポット料金を提供することになります。

## チュートリアル: スpotフリートを使ってインスタンスの分量を指定する

このチュートリアルでは、サンプル株式会社という名の架空会社で、インスタンス分量指定を使ったスポットフリートリクエストのプロセスを説明します。

### 目的

製薬会社であるサンプル株式会社は、癌と闘うために使用される可能性のある化合物を選別するために Amazon EC2 の計算処理能力を利用したいと考えています。

### 計画

サンプル株式会社はまず、「[Spot Best Practices](#)」を参照します。次に、サンプル株式会社はスポットフリートに関する以下の要件を確認します。

## インスタンスタイプ

サンプル株式会社には、60 GB 以上のメモリと 8 つの仮想 CPU (vCPU) で最適に実行される、計算能力とメモリに負担がかかるアプリケーションがあります。同社は、できるだけ低価格でアプリケーション用のこれらのリソースを最大化したいと考えています。サンプル株式会社は、以下のいずれかの EC2 インスタンスタイプがそのニーズを満たすと判断します。

インスタンスタイプ	メモリ (GiB)	vCPU
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

## ユニット単位の目標容量

インスタンスの分量指定を使用すると、ターゲット容量はインスタンスの数 (デフォルト)、またはコア (vCPU)、メモリ (GiB) とストレージ (GB) との要素の組み合わせで表すことができます。アプリケーションのベース (60 GB の RAM と 8 個の vCPU) を 1 ユニットとして考えることで、サンプル株式会社はこの量の 20 倍で十分ニーズに合うと決定します。これにより、会社は スポットフリート リクエストのターゲット容量を 20 に設定します。

## インスタンスの分量

ターゲット容量の決定後、サンプル株式会社はインスタンスの分量を計算します。各インスタンスタイプのインスタンスの分量を計算することは、以下のように、ターゲット容量に達するために必要な各インスタンスタイプのユニットの数を決定することです。

- r3.2xlarge (61.0 GB、8 個の vCPU) = 1/20 ユニット
- r3.4xlarge (122.0 GB、16 個の vCPU) = 2/20 ユニット
- r3.8xlarge (244.0 GB、32 個の vCPU) = 4/20 ユニット

これよりサンプル株式会社は、1、2 と 4 のインスタンス分量を スポットフリート リクエストのそれぞれの起動設定に割り当てます。

## ユニット時間あたりの価格

サンプル株式会社は、料金の出発点としてインスタンス時間あたりの「[オンデマンド料金](#)」を使用します。最近のスポット料金または 2 つの組み合わせを使用することもできます。ユニット時間あたりの料金を計算するために、インスタンス時間あたりの出発点の料金を分量で割ります。例:

インスタンスタイプ	オンデマンド価格	インスタンスの分量	ユニット時間あたりの価格
r3.2xLarge	\$0.7	1	\$0.7
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

サンプル株式会社は、ユニット時間あたりのグローバルな料金として 0.7 USD を使用し、3 つのインスタンスタイプすべてで競争力を高めることができます。また、r3.8xlarge の起動条件のなかで、1 ユニッ

ト時間あたりの全体料金を 0.7 USD、そして 1 ユニット時間あたりの指定入力料金を 0.9 USD とすることもできます。

## アクセス許可の確認

スポットフリート のリクエストを作成する前に、サンプル株式会社は必要アクセス許可の IAM ロールがあることを確認します。詳細については、「[スポットフリート の前提条件 \(p. 346\)](#)」を参照してください。

## リクエストを作成する

サンプル株式会社は、スポットフリート リクエストのために次の設定の config.json ファイルを作成します。

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "SubnetId": "subnet-482e4972",  
            "WeightedCapacity": 1  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.4xlarge",  
            "SubnetId": "subnet-482e4972",  
            "WeightedCapacity": 2  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.8xlarge",  
            "SubnetId": "subnet-482e4972",  
            "SpotPrice": "0.90",  
            "WeightedCapacity": 4  
        }  
    ]  
}
```

サンプル株式会社は、次の `request-spot-fleet` コマンドを使用して スpotフリート リクエストを作成します。

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

詳細については、「[スポットフリート リクエスト \(p. 345\)](#)」を参照してください。

## 受理

配分戦略は、スポットインスタンス が取得される スpotインスタンス プールを決定します。

`lowestPrice` 戰略 (デフォルトの戦略) では、受理時にユニットあたりのスポット料金が最低値であるプールからスポットインスタンスが取得されます。20 ユニットの容量を提供するためには、20 の `r3.2xlarge` インスタンス ( $20 \div 1$ )、10 の `r3.4xlarge` インスタンス ( $20 \div 2$ )、あるいは 5 の `r3.8xlarge` インスタンス ( $20 \div 4$ ) が スpotフリート から起動されることになります。

サンプル株式会社が `diversified` 戰略を採用する場合、スポットインスタンス は 3 つのすべてのプールから取得されます。スポットフリート は、6 つの `r3.2xlarge` インスタンス (6 ユニットを提供)、3 つの

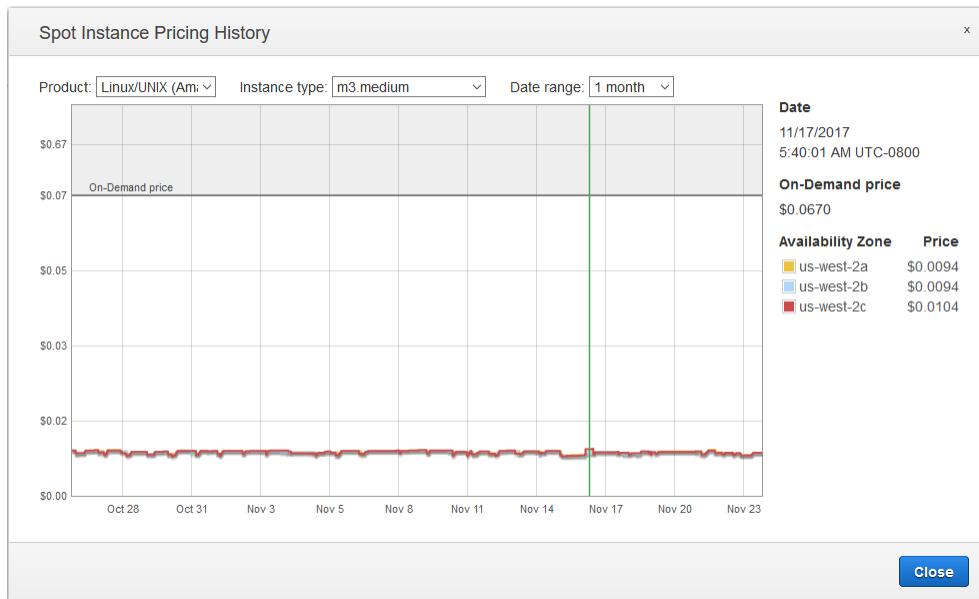
r3.4xlarge インスタンス (6 ユニットを提供)、そして 2 つの r3.8xlarge インスタンス (8 ユニットを提供) の全部で 20 ユニットを起動します。

## スポットインスタンス の価格設定履歴

スポットインスタンス をリクエストするときは、デフォルトの上限価格 (オンデマンド価格) を使用することをお勧めします。上限料金を指定する場合は、スポット料金の履歴を確認してから指定することをお勧めします。インスタンスタイプ、オペレーティングシステム、アベイラビリティゾーンでフィルタリングして、過去 90 日間のスポット料金履歴を表示できます。

スポット料金履歴を表示するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。
3. スポットインスタンス を初めて使用する場合は、ウェルカムページを参照してください。[Get started] を選択して、画面下部までスクロールし、[キャンセル] を選択します。
4. [Pricing History] を選択します。
5. 料金履歴を表示するオペレーティングシステム ([製品])、[インスタンスタイプ]、[日付範囲] を選択します。グラフ上にピントを移動すると、選択した日付範囲の特定の時刻の料金が表示されます。



6. (オプション) 特定のアベイラビリティゾーンのスポット料金履歴を確認するには、リストからゾーンを選択します。別の製品、インスタンスタイプまたは日付範囲を選択することもできます。

コマンドラインを使用してスポット料金履歴を表示するには

次のコマンドの 1 つを使用できます。詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [describe-spot-price-history](#) (AWS CLI)
- [Get-EC2SpotPriceHistory](#) (AWS Tools for Windows PowerShell)

## スポットインスタンス 購入による削減額

フリートあたりレベルの スポットインスタンス またはすべての実行中の スポットインスタンス に関する使用状況と削減額の情報を表示できます。フリートあたりのレベルでは、使用状況と削減額の情報にフ

リートが起動および終了するすべてのインスタンスが含まれます。この情報は、過去 1 時間または過去 3 日間から表示できます。

次の「スポットリクエスト」ページのスクリーンショットは、スポットフリート のスポットの使用状況と削減額の情報を示します。



#### Details

c3.large (8)	1152 vCPU-hours	2160 mem(GiB)-hours	\$16.82 total	72% savings
c4.xlarge (6)	1728 vCPU-hours	3240 mem(GiB)-hours	\$26.48 total	69% savings
t2.medium (3)	432 vCPU-hours	864 mem(GiB)-hours	\$3.00 total	70% savings

表示できる使用状況と削減額の情報は次のとおりです。

- スポットインスタンス – スpot フリート が起動および終了する スポットインスタンス の数。削減額 の要約を表示した場合、その数字は実行中のすべての スポットインスタンス を表します。
- vCPU-hours – 選択した時間枠ですべての スポットインスタンス で使用される vCPU 時間数。
- Mem(GiB)-hours – 選択した時間枠ですべての スポットインスタンス で使用される GiB 時間数。
- On-Demand total – これらのインスタンスを オンデマンドインスタンス として起動した場合、選択した 時間枠で支払った合計金額。
- Spot total – 選択した時間枠で支払う合計金額。
- Savings – オンデマンド価格を支払わないことで節約される割合。
- Average cost per vCPU-hour – 選択した時間枠ですべての スポットインスタンス で vCPU を使用する 1 時間あたりの平均コスト。次の式で計算されます: Average cost per vCPU-hour = Spot total / vCPU-hours
- Average cost per mem(GiB)-hour – 選択した時間枠ですべての スポットインスタンス で GiB を使用す る 1 時間あたりの平均コスト。次の式で計算されます: Average cost per mem(GiB)-hour = Spot total / Mem(GiB)-hours
- Details テーブル – スpot フリート を構成するさまざまなインスタンスタイプ（インスタンスタイプあ たりのインスタンス数は括弧で囲まれます）。削減額の要約を表示した場合、その数字は実行中のすべて の スポットインスタンス から成ります。

削減額情報は、Amazon EC2 コンソールからのみ表示できます。

スspot フリート の削減額情報を表示するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。
3. スpot フリート リクエストを選択して、[Savings] を選択します。
4. デフォルトでは、過去 3 日間の使用状況と削減額の情報が表示されます。[last hour] または [last three days] を選択できます。1 時間未満前に起動された スpot フリート の場合は、その時間の削減見込 み額が表示されます。

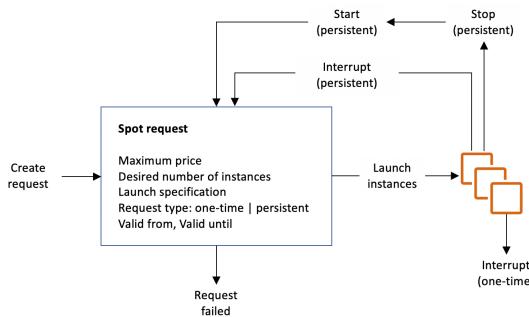
実行中のすべての スpot インスタンス の削減額情報を表示するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。
3. [Savings Summary] を選択します。

## スポットインスタンス リクエスト

スポットインスタンスを使用するには、希望のインスタンスの数、インスタンスタイプ、アベイラビリティーゾーン、インスタンス時間あたりに支払える上限価格（入札価格）を含むスポットインスタンスリクエストを作成します。時間あたりの上限料金が現在のスポット料金を超える場合で、容量がご利用可能な場合、Amazon EC2 は即時にお客様のリクエストを受理します。それ以外の場合、Amazon EC2 は、リクエストが受理できるようになるか、お客様がリクエストをキャンセルするまで待機します。

次の図に、スポットリクエストのしくみを示します。リクエストタイプ（ワンタイムまたは永続）によって、Amazon EC2 がスポットインスタンスを中断したとき、またはお客様がスポットインスタンスを停止した場合に、リクエストが再度開かれるかどうかが決まります。リクエストが永続リクエストの場合、スポットインスタンスの中断後、リクエストが再度開かれます。リクエストが永続的で、お客様がスポットインスタンスを停止した場合、リクエストはスポットインスタンスを起動した後にのみ開かれます。



### 目次

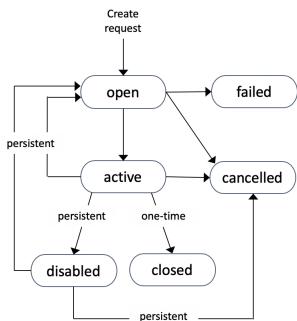
- [スポットインスタンス リクエストの状態 \(p. 335\)](#)
- [スポットインスタンス の継続期間の定義 \(p. 336\)](#)
- [スポットインスタンス のテナントの指定 \(p. 337\)](#)
- [スポットインスタンス リクエスト用のサービスにリンクされたロール \(p. 338\)](#)
- [スポットインスタンス リクエストを作成する \(p. 339\)](#)
- [実行中の スpotインスタンス の検索 \(p. 340\)](#)
- [スポットインスタンス リクエストのタグ付け \(p. 341\)](#)
- [スポットインスタンス リクエストのキャンセル \(p. 341\)](#)
- [スポットインスタンスの停止 \(p. 342\)](#)
- [スポットインスタンスの起動 \(p. 342\)](#)
- [スポットインスタンス の終了 \(p. 343\)](#)
- [スポットリクエストの起動仕様の例 \(p. 343\)](#)

## スポットインスタンス リクエストの状態

スポットインスタンス リクエストは、次に示す状態のいずれかになります。

- **open** – リクエストは受理されるまで待機状態です。
- **active** – リクエストは受理された状態であり、関連付けられたスポットインスタンスが存在します。
- **failed** – リクエストの 1 つ以上のパラメータが正しくありません。
- **closed** – スpotインスタンス は中止または終了されました。
- **disabled** – お客様がスポットインスタンスを停止しました。
- **cancelled** – お客様がリクエストをキャンセルしたか、リクエストの有効期限が切れました。

次の図は、リクエストの状態の遷移を示しています。遷移はリクエストのタイプ(ワンタイムまたは永続)によって異なります。



ワンタイム スポットインスタンス リクエストは、Amazon EC2 が スポットインスタンス を起動するか、リクエストの有効期限が切れるか、またはお客様がリクエストをキャンセルするまでアクティブ状態です。スポット料金が上限価格を上回った場合または容量が利用できない場合、スポットインスタンス は終了し、スポットインスタンス リクエストはクローズされます。

永続 スポットインスタンス リクエストは、リクエストが受理されても、リクエストの有効期限が切れるか、お客様がリクエストをキャンセルするまでアクティブ状態です。スポット料金が上限価格を上回った場合または容量が利用できない場合、スポットインスタンス は中断されます。インスタンスが中断された後、上限料金がスポット料金を超えるか、容量が再び利用可能になると、スポットインスタンス が開始されるか(停止している場合)、再開されます(休止状態の場合)。キャバシティーが利用可能で、上限価格が現在のスポット料金を超える場合は、スポットインスタンスを停止して再び起動できます。スポットインスタンスが終了した場合(スポットインスタンスが停止状態か実行状態かに関係なく)、スポットインスタンスリクエストが再び開かれ、Amazon EC2 が新しいスポットインスタンスを起動します。詳細については、「[スポットインスタンスの停止 \(p. 342\)](#)」、「[スポットインスタンスの起動 \(p. 342\)](#)」、および「[スポットインスタンスの終了 \(p. 343\)](#)」を参照してください。

スポットインスタンス リクエストのステータスと、入札ステータスによって起動された スポットインスタンス のステータスを追跡できます。詳細については、「[スポットリクエストステータス \(p. 379\)](#)」を参照してください。

## スポットインスタンス の継続期間の定義

継続期間(スポットブロックとも呼ばれます)が定義された スpotインスタンス は、中断されることなく、選択した期間にわたって継続して実行されます。このようなインスタンスは、バッチ処理、エンコードとレンダリング、モデリングと解析、継続的な統合など、完了までに一定の時間かかるジョブに最適です。

継続期間として 1 時間、2 時間、3 時間、4 時間、5 時間、または 6 時間を使用できます。支払い額はこの指定した継続期間によって決まります。1 時間または 6 時間の継続期間の現在の価格を確認するには、「[スポットインスタンス の料金表](#)」ページを参照してください。これらの価格を使用して 2、3、4、5 時間の継続期間のコストを見積もることができます。継続期間を指定したリクエストが履行されると、スポットインスタンス の価格は固定され、インスタンスの終了時まで有効なままになります。インスタンスが実行している時間(または 1 時間未満)ごとに、この価格で請求されます。インスタンスの実行時間が 1 時間未満の場合は、最も近い秒に切り上げて請求されます。

スポットリクエストで継続期間を定義すると、各 スpotインスタンス にインスタンス ID が割り当てられると同時に継続期間が開始されます。スポットインスタンス は手動終了されるか継続期間が終了するまで実行されます。継続期間の終了時、Amazon EC2 では スpotインスタンス が終了対象としてマークされ、スポットインスタンス の終了通知が表示されます。それにより、インスタンスの終了前に 2 分の警告期間が与えられます。まれに Amazon EC2 のキャバシティーの都合でスポットブロックの中止が発生する場合があります。この場合、インスタンスを削除する前に 2 分間の警告が与えられ、使用した場合でも、削除したインスタンスについて課金されません。

継続期間が定義された スpotインスタンス を起動するには(コンソール)

詳細については、「[スポットフリート リクエストを作成する \(p. 350\)](#)」を参照してください。

継続期間が定義された スポットインスタンス を起動するには (AWS CLI)

スポットインスタンス の継続期間を指定するには、`--block-duration-minutes` オプションを付けて `request-spot-instances` コマンドを実行します。たとえば、以下のコマンドでは、継続期間が 2 時間の スポットインスタンス を起動するスポットリクエストが作成されます。

```
aws ec2 request-spot-instances --instance-count 5 --block-duration-minutes 120 --type "one-time" --launch-specification file://specification.json
```

継続期間が定義された スポットインスタンス のコストを取得するには (AWS CLI)

継続期間を指定した スポットインスタンス の固定費を取得するには、`describe-spot-instance-requests` コマンドを使用します。この情報は `actualBlockHourlyPrice` フィールドにあります。

## スポットインスタンス のテナントの指定

スポットインスタンス は、シングルテナントのハードウェアで実行できます。ハードウェア専有 スポットインスタンス は、他の AWS アカウントに属するインスタンスから物理的に分離されます。詳細については、「[ハードウェア専有インスタンス \(p. 425\)](#)」および「[Amazon EC2 ハードウェア専有インスタンス](#)」の製品ページを参照してください。

ハードウェア専有 スポットインスタンス を実行するには、次のいずれかを実行します。

- スポットインスタンス のリクエストを作成するときに、`dedicated` のテナントを指定します。詳細については、「[スポットインスタンス リクエストを作成する \(p. 339\)](#)」を参照してください。
- スポットインスタンス のインスタンスのテナントを持つ VPC の `dedicated` をリクエストします。 詳細については、「[インスタンスのテナント属性が専有である VPC を作成する \(p. 428\)](#)」を参照してください。`default` のインスタンスのテナントを使用して VPC でインスタンスをリクエストすると、`dedicated` のテナントで スポットインスタンス をリクエストすることができません。

次のインスタンスタイプは、ハードウェア専有 スポットインスタンス をサポートします。

### 現行世代

- `c4.8xlarge`
- `d2.8xlarge`
- `i3.16xlarge`
- `m4.10xlarge`
- `m4.16xlarge`
- `p2.16xlarge`
- `r4.16xlarge`
- `x1.32xlarge`

### 前の世代

- `c3.8xlarge`
- `cc2.8xlarge`
- `cr1.8xlarge`
- `g2.8xlarge`
- `i2.8xlarge`
- `r3.8xlarge`

## スポットインスタンス リクエスト用のサービスにリンクされたロール

Amazon EC2 は、ユーザーに代わって AWS の他のサービスを呼び出すために必要なアクセス許可を持つサービスにリンクされたロールを使用します。サービスにリンクされたロールは、AWS のサービスに直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは、AWS のサービスにアクセス許可を委任するためのセキュアな方法を提供します。これは、リンクされたサービスのみが、サービスにリンクされたロールを引き受けることができるためです。詳細については、IAM ユーザーガイドの「[サービスにリンクされたロールの使用](#)」を参照してください。

Amazon EC2 は、AWSServiceRoleForEC2Spot という、サービスにリンクされたロールを使用して、ユーザーの代わりに スポットインスタンス を起動して管理します。

### AWSServiceRoleForEC2Spot によって付与されるアクセス許可

Amazon EC2 は、AWSServiceRoleForEC2Spot という、サービスにリンクされたロールを使用して、次のアクションを実行します。

- `ec2:DescribeInstances` – スポットインスタンス を記述
- `ec2:StopInstances` – スポットインスタンス を停止
- `ec2:StartInstances` – スポットインスタンス を起動

### サービスにリンクされたロールの作成

ほとんどの状況では、サービスにリンクされたロールを手動で作成する必要はありません。コンソールを使用して スポットインスタンス を初めてリクエストしたときに、サービスにリンクされたロールとして AWSServiceRoleForEC2Spot が Amazon EC2 で自動的に作成されます。

Amazon EC2 がこのサービスにリンクされたロールのサポートを開始した 2017 年 10 月以前にアクティブな スポットインスタンス リクエストを行った場合は、Amazon EC2 が AWS アカウントで [AWSServiceRoleForEC2Spot] ロールを作成しています。詳細については、IAM ユーザーガイドの「[AWS アカウントに新しいロールが表示される](#)」を参照してください。

AWS CLI または API を使用して スポットインスタンス をリクエストする前にこのロールが存在していることを確認します。ロールを作成するには、IAM コンソールを次のように使用します。

サービスにリンクされたロール AWSServiceRoleForEC2Spot を手動で作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [Roles (ロール)] を選択します。
3. [ロールの作成] を選択します。
4. [Select type of trusted entity (信頼されたエンティティのタイプを選択)] ページで、[EC2]、[EC2 - Spot Instances (EC2 - スポットインスタンス)]、[Next: Permissions (次の手順: アクセス許可)] の順に選択します。
5. 次のページで、[Next:Review (次の手順: 確認)] を選択します。
6. [確認] ページで、[ロールの作成] を選択します。

スポットインスタンス を使用する必要がなくなった場合は、[AWSServiceRoleForEC2Spot] ロールを削除することをお勧めします。このロールがアカウントから削除された後で、Amazon EC2 をリクエストすると、スポットインスタンス はロールを再度作成します。

### 暗号化された AMI および EBS スナップショット用に CMK へのアクセスを付与する

暗号化された AMI (p. 151) または暗号化された Amazon EBS スナップショット (p. 1014) を スポットインスタンス で指定し、カスタマーマネジメント CMK (カスタマーマスターキー) を暗号化に使用する場合は、CMK

を使用するアクセス許可を AWSServiceRoleForEC2Spot ロールに付与して Amazon EC2 がユーザーに代わって スポットインスタンス を起動できるようにする必要があります。これを行うには、次の手順で示すように、CMK に付与を追加する必要があります。

アクセス権限を設定するときは、付与がキー policy の代わりになります。詳細については、「[許可の使用](#)」と「[AWS KMS でのキー policy の使用](#)」(AWS Key Management Service Developer Guide) を参照してください。

CMK を使用するアクセス許可を AWSServiceRoleForEC2Spot ロールに付与するには

- `create-grant` コマンドを使用して CMK に付与を追加し、プリンシパル (サービスにリンクされたロール AWSServiceRoleForEC2Fleet) を指定します。このプリンシパルには、付与が許可するオペレーションを実行するためのアクセス許可が提供されます。CMK を指定するには、key-id パラメータと CMK の ARN を使用します。プリンシパルを指定するには、grantee-principal パラメータとサービスにリンクされたロール AWSServiceRoleForEC2Spot の ARN を使用します。

次の例は、読みやすいようにフォーマットされています。

```
aws kms create-grant \
--region us-east-1 \
--key-id arn:aws:kms:us-east-1:44445556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
--grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Spot \
--operations "Decrypt" "Encrypt" "GenerateDataKey" "GenerateDataKeyWithoutPlaintext" \
"CreateGrant" "DescribeKey" "ReEncryptFrom" "ReEncryptTo"
```

## スポットインスタンス リクエストを作成する

スポットインスタンス をリクエストするプロセスは、オンデマンドインスタンス を起動するプロセスに似ています。スポットインスタンス リクエストを送信した後で、上限料金などのスポットリクエストのパラメータを変更することはできません。

一度に複数の スpotインスタンス をリクエストした場合、Amazon EC2 では個々の スpotインスタンス リクエストが作成され、各リクエストのステータスを個別に追跡できます。スポットインスタンス リクエストの追跡については、「[スポットリクエストステータス \(p. 379\)](#)」を参照してください。

### 前提条件

開始する前に、上限価格、必要な スpotインスタンス の数、使用するインスタンスタイプを決定します。スポット料金の傾向を確認するには、「[スポットインスタンス の価格設定履歴 \(p. 333\)](#)」を参照してください。

スポットインスタンス リクエストを作成するには (コンソール)

詳細については、「[スポットフリート リクエストを作成する \(p. 350\)](#)」を参照してください。

スポットインスタンス リクエストを作成するには (AWS CLI)

ワンタイムリクエストを作成するには、以下の `request-spot-instances` コマンドを使用します。

```
aws ec2 request-spot-instances --instance-count 5 --type "one-time" --launch-specification
file://specification.json
```

永続リクエストを作成するには、以下の `request-spot-instances` を使用します。

```
aws ec2 request-spot-instances --instance-count 5 --type "persistent" --launch-
specification file://specification.json
```

以下のコマンドで使用する起動仕様ファイルの例については、「[スポットリクエストの起動仕様の例 \(p. 343\)](#)」を参照してください。コンソールから起動仕様ファイルをダウンロードする場合、`request-spot-fleet` コマンドを使用する必要があります (コンソールは、スポットフリートを使用してスポットリクエストを指定します)。

上限料金がスポット料金を超える場合、Amazon EC2 は スポットインスタンスを起動します。スポットインスタンスは中断されるか手動終了されるまで実行されます。スポットインスタンスのリクエストを監視するには、以下の `describe-spot-instance-requests` コマンドを使用します。

```
aws ec2 describe-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

## 実行中の スポットインスタンス の検索

上限料金がスポット料金を超える場合、Amazon EC2 は スポットインスタンスを起動します。スポットインスタンスは中断されるか手動終了されるまで実行されます。上限価格がスポット料金と厳密に等しい場合、需要に応じて、スポットインスタンスは実行されたままとなる可能性があります。

実行中の スポットインスタンス を検索するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。

スポットインスタンス リクエストとスポットフリート リクエストの両方を参照することができます。スポットインスタンス リクエストが実行された場合、[容量] は スポットインスタンスの ID です。スポットフリートの場合、[容量] はリクエスト容量のうち受理された量を示しています。スポットフリートのインスタンスの ID を表示するには、拡張矢印を選択するか、フリートを選択して [インスタンス] を選択します。

### Note

スポットインスタンスリクエストはすぐにタグ付けされず、一定期間はスポットフリートリクエスト (SFR) と別々に表示される場合があります。

3. または、ナビゲーションペインで [Instances] を選択します。右上隅にある [Show/Hide] アイコンを選択し、[Lifecycle] を選択します。各インスタンスの [ライフサイクル] は、normal、spot、または scheduled のいずれかです。

実行中の スポットインスタンス を検索するには (AWS CLI)

スポットインスタンスを一覧表示するには、以下のように `--query` オプションを指定して `describe-spot-instance-requests` コマンドを実行します。

```
aws ec2 describe-spot-instance-requests --query "SpotInstanceRequests[*].{ID:InstanceId}"
```

出力例を次に示します。

```
[  
  {  
    "ID": "i-1234567890abcdef0"  
  },  
  {  
    "ID": "i-0598c7d356eba48d7"  
  }  
]
```

または、以下のように `--filters` オプションを指定して `describe-instances` コマンドを実行しても、スポットインスタンスを一覧表示できます。

```
aws ec2 describe-instances --filters "Name=instance-lifecycle,Values=spot"
```

## スポットインスタンス リクエストのタグ付け

スポットインスタンス リクエストを分類および管理しやすくするため、任意のメタデータでタグ付けすることができます。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。

タグは、作成後にスポットインスタンス リクエストに割り当てることができます。スポットインスタンス リクエストに作成するタグは、そのリクエストにのみ適用されます。これらのタグは、リクエストを受理するためにスポットサービスが起動する スpotトインスタンス には自動的に追加されません。スポットインスタンス が起動された後、スポットインスタンス に自分でタグを追加する必要があります。

AWS CLI を使用して スpotトインスタンス リクエストや スpotトインスタンス にタグを追加するには リソースにタグを追加するには、以下の [create-tags](#) コマンドを使用します。

```
aws ec2 create-tags --resources sir-08b93456 i-1234567890abcdef0 --tags  
Key=purpose,Value=test
```

## スポットインスタンス リクエストのキャンセル

スポットインスタンスが不要になった場合には、それをキャンセルすることができます。キャンセルできるのは、open、active、または disabled の状態にある スpotトインスタンス リクエストのみです。

- リクエストがまだ受理されておらず、インスタンスが起動されていない場合、スspotトインスタンス リクエストは open 状態にあります。
- リクエストが受理され、結果としてスspotトインスタンスが起動された場合、スspotトインスタンス リクエストは active 状態にあります。
- スspotトインスタンス リクエストは、お客様がスspotトインスタンスを停止した場合、disabled 状態にあります。

スspotトインスタンス リクエストが active 状態で、関連付けられているスspotトインスタンスが実行されている場合、またはスspotトインスタンス リクエストが disabled 状態で、関連付けられているスspotトインスタンスが停止している場合、リクエストをキャンセルしてもインスタンスは終了しません。スspotトインスタンス の終了の詳細については、「[スspotトインスタンス の終了 \(p. 343\)](#)」を参照してください。

### スspotトインスタンス リクエストをキャンセルするには (コンソール)

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Spot Requests] を選択し、スspotトリクエストを選択します。
- [Actions]、[Cancel spot request] の順に選択します。
- (オプション) 関連付けられたスspotトインスタンスを使い終わったら、スspotトインスタンス を終了できます。ナビゲーションペインで、[Instances] を選択し、インスタンスを選択した後で、[Actions]、[Instance State]、[Terminate] の順に選択します。

### スspotトインスタンス リクエストをキャンセルするには (AWS CLI)

- 指定したスspotトリクエストをキャンセルするには、以下の [cancel-spot-instance-requests](#) コマンドを使用します。

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

## スポットインスタンスの停止

今すぐスポットインスタンスは必要ないが、Amazon EBS ボリューム内に保持されているデータを失わず、以後で再起動する必要がある場合は、それらを停止できます。スポットインスタンスを停止する手順は、オンデマンドインスタンスを停止する手順と似ています。スポットインスタンスを停止できるのは、スポットインスタンスが `persistent` のスポットインスタンスリクエストから起動された場合だけです。

### Note

スポットインスタンスが停止している間、インスタンスの属性の一部は変更できますが、インスタンスタイプを変更することはできません。

停止しているスポットインスタンスの使用料またはデータ転送料は課金されませんが、Amazon EBS ボリュームのストレージに対しては課金されます。

### 制約事項

- スpotトインスタンスは、それがフリートまたは起動グループ、アベイラビリティゾーニングループ、またはスポットブロックの一部である場合、停止できません。

### スポットインスタンスを停止するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Instances (インスタンス)] を選択し、スポットインスタンスを選択します。
3. [Actions]、[Instance State]、[Stop] の順に選択します。

### スポットインスタンスを停止するには (AWS CLI)

- 次の `stop-instances` コマンドを使用して、1つまたは複数のスポットインスタンスを手動で停止します。

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

## スポットインスタンスの起動

以前に停止したスポットインスタンスは起動できます。スポットインスタンスを起動する手順は、オンデマンドインスタンスを起動する手順と似ています。

### 前提条件

スポットインスタンスを起動できるのは、次の場合のみです。

- スpotトインスタンスが手動で停止された。
- スpotトインスタンスは EBS-backed インスタンスである。
- スpotトインスタンスキャパシティーが利用可能である。
- スpotト料金が上限価格より低くなっている。

### 制約事項

- スpotトインスタンスは、それがフリートまたは起動グループ、アベイラビリティゾーニングループ、またはスポットブロックの一部である場合、起動できません。

### スポットインスタンスを起動するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで [Instances (インスタンス)] を選択し、スポットインスタンスを選択します。
3. [Actions]、[Instance State]、[Start] の順に選択します。

スポットインスタンスを起動するには (AWS CLI)

- 次の `start-instances` コマンドを使用して、1つまたは複数のスポットインスタンスを手動で起動します。

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

## スポットインスタンス の終了

スポットインスタンスリクエストが `active` で、関連付けられているスポットインスタンスが実行されている場合、またはスポットインスタンスリクエストが `disabled` で、関連付けられているスポットインスタンスが停止している場合、リクエストをキャンセルしてもインスタンスは終了しません。実行中のスポットインスタンスを手動で終了する必要があります。永続スポットリクエストによって起動された実行中または停止中のスポットインスタンスを終了する場合、新しいスポットインスタンスが起動できるように、スポットリクエストは `open` 状態に戻ります。永続スポットリクエストをキャンセルして、スポットインスタンスを終了するには、まずスポットリクエストをキャンセルしてからスポットインスタンスを終了する必要があります。そうしないと、永続スポットリクエストによって新しいインスタンスが起動される場合があります。スポットインスタンスリクエストのキャンセルの詳細については、「[スポットインスタンス リクエストのキャンセル \(p. 341\)](#)」を参照してください。

スポットインスタンス を手動で終了するには (AWS CLI)

- スpotトインスタンス を手動で終了するには、次の `terminate-instances` コマンドを使用します。

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0 i-0598c7d356eba48d7
```

## スポットリクエストの起動仕様の例

以下の例で示しているのは、スポットインスタンス リクエストを作成するための `request-spot-instances` コマンドで使用できる起動設定です。詳細については、「[スポットインスタンス リクエストを作成する \(p. 339\)](#)」を参照してください。

1. スpotトインスタンス を起動する (p. 343)
2. 指定したアベイラビリティーゾーンで スpotトインスタンス を起動する (p. 344)
3. 指定したサブネットで スpotトインスタンス を起動する (p. 344)
4. 専有 スpotトインスタンス を起動 (p. 345)

### 例 1: スpotトインスタンス の起動

以下の例には、アベイラビリティーゾーンまたはサブネットは含まれていません。Amazon EC2 によってアベイラビリティーゾーンが選択されます。Amazon EC2 によって、指定したアベイラビリティーゾーンのデフォルトのサブネットでインスタンスが起動されます。

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "m3.medium",  
    "IamInstanceProfile": {
```

```
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
}
```

### 例 2: 指定したアベイラビリティゾーンで スポットインスタンス を起動する

以下の例には、アベイラビリティゾーンが含まれています。Amazon EC2 によって、指定したアベイラビリティゾーンのデフォルトのサブネットでインスタンスが起動されます。

```
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],
    "InstanceType": "m3.medium",
    "Placement": {
        "AvailabilityZone": "us-west-2a"
    },
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
}
```

### 例 3: 指定したサブネットで スポットインスタンス を起動する

次の例にはサブネットが含まれています。Amazon EC2 は、指定されたサブネット内のインスタンスを起動します。デフォルト以外の VPC である場合、インスタンスにはデフォルトでパブリック IPv4 アドレスは割り当てられません。

```
{
    "ImageId": "ami-1a2b3c4d",
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],
    "InstanceType": "m3.medium",
    "SubnetId": "subnet-1a2b3c4d",
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
}
```

デフォルト以外の VPC である場合、インスタンスにパブリック IPv4 アドレスを割り当てるには、以下の例に示しているように AssociatePublicIpAddress フィールドを指定します。ネットワークインターフェイスの指定時には、例 3 に示している SubnetId および SecurityGroupIds フィールドではなく、ネットワークインターフェイスを使用して、サブネット ID およびセキュリティグループ ID を含める必要があります。

```
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "InstanceType": "m3.medium",
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "SubnetId": "subnet-1a2b3c4d",
            "Groups": [ "sg-1a2b3c4d" ],
            "AssociatePublicIpAddress": true
        }
    ],
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
    }
}
```

#### 例 4: 専有 スポットインスタンス を起動する

次の例では、`dedicated` のテナントで スポットインスタンス をリクエストします。ハードウェア専用 スポットインスタンス は、VPC で起動する必要があります。

```
{  
    "ImageId": "ami-1a2b3c4d",  
    "KeyName": "my-key-pair",  
    "SecurityGroupIds": [ "sg-1a2b3c4d" ],  
    "InstanceType": "c3.8xlarge",  
    "SubnetId": "subnet-1a2b3c4d",  
    "Placement": {  
        "Tenancy": "dedicated"  
    }  
}
```

## スポットフリート リクエスト

スポットフリート を使用するには、目標とするキャパシティー、オプションのオンデマンド部分、インスタンスの 1 つ以上の起動仕様、支払う予定のある上限価格などを指定した スpotトフリート リクエストを作成します。Amazon EC2 は、スポット料金の変更に応じて スpotトフリート の目標容量を維持しようとします。詳細については、「[スポットフリート の詳細 \(p. 326\)](#)」を参照してください。

スポットフリート リクエストには、`request` および `maintain` の 2 つのタイプがあります。スポットフリート を作成し、希望する容量のワンタイムリクエストを送信するか、ターゲット容量の継続した維持を要求します。どちらのリクエストタイプも、スポットフリート の分散戦略の恩恵を受けます。

ワンタイムリクエストを作成すると、スポットフリート は必要なリクエストを行いますが、容量が低下した場合は スpotトインスタンス の補充を試みません。使用可能な容量がない場合、スポットフリート は代替スポットプールでリクエストを送信しません。

ターゲット容量を維持するために、スポットフリート はこのターゲット容量を満たすのに必要なリクエストを行い、中断されたインスタンスを自動的に補充します。

送信後にワンタイムリクエストのターゲット容量を変更することはできません。ターゲット容量を変更するには、リクエストを変更し、新しいリクエストを送信します。

スポットフリート リクエストは、期限切れになるかお客様によってキャンセルされるまで、アクティブのままになります。スポットフリート リクエストのキャンセル時には、スポットフリート のキャンセルによって スpotトフリート の スpotトインスタンス を終了するかどうかを指定できます。

各起動仕様には、Amazon EC2 によるインスタンスの起動に必要な情報 (AMI、インスタンスタイプ、サブネットまたはアベイラビリティーボーン、そして 1 つ以上のセキュリティグループ) を指定します。

### コンテンツ

- [スポットフリート リクエストの状態 \(p. 346\)](#)
- [スポットフリート の前提条件 \(p. 346\)](#)
- [スポットフリート ユーザーと IAM ユーザー \(p. 347\)](#)
- [スポットフリート ヘルスチェック \(p. 348\)](#)
- [スポットフリート リクエストの準備 \(p. 348\)](#)
- [スポットフリート リクエスト用のサービスにリンクされたロール \(p. 349\)](#)
- [スポットフリート リクエストを作成する \(p. 350\)](#)
- [スポットフリート のタグ付け \(p. 354\)](#)
- [スポットフリート のモニタリング \(p. 359\)](#)
- [スポットフリート リクエストの変更 \(p. 360\)](#)
- [スポットフリート リクエストのキャンセル \(p. 361\)](#)

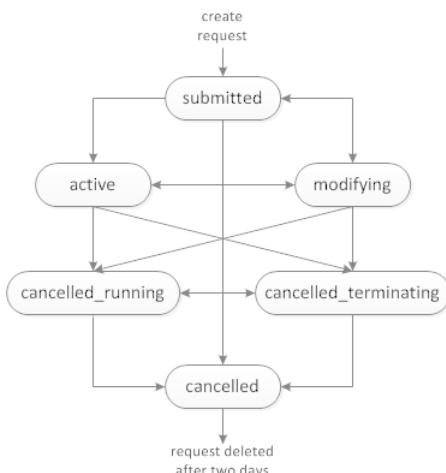
- スポットフリート 設定例 (p. 362)

## スポットフリート リクエストの状態

スポットフリート リクエストは、次に示す状態のいずれかになります。

- submitted – スpotフリート リクエストは評価中です。Amazon EC2 は目標数のインスタンスを起動する準備をしています。
- active – スpotフリートは検証済みです。Amazon EC2 は実行中の スpotインスタンス をターゲット数分、確保しようとしています。リクエストは、変更またはキャンセルされるまで、この状態のままになります。
- modifying – スpotフリート リクエストは変更中です。リクエストは、変更が完全に処理されるか、スspotフリート がキャンセルされるまで、この状態のままになります。ワンタイム request を変更することはできません。この状態は、そのようなスspotリクエストには適用されません。
- cancelled\_running – スpotフリート はキャンセルされ、追加の スpotインスタンス は起動されません。その既存の スpotインスタンス は、中断または終了されるまで実行され続けます。リクエストは、すべてのインスタンスが中断されるか終了されるまで、この状態のまになります。
- cancelled\_terminating – スpotフリート はキャンセルされました。スspotインスタンス は終了中です。リクエストは、すべてのインスタンスが終了されるまで、この状態のまになります。
- cancelled – スpotフリート はキャンセルされました。実行中の スpotインスタンス はありません。スspotフリート リクエストは、そのインスタンスが終了されてから 2 日後に削除されます。

次の図は、リクエストの状態の遷移を示しています。スspotフリート の制限を超えた場合、リクエストはすぐにキャンセルされます。



## スspotフリート の前提条件

Amazon EC2 コンソールを使用して スspotフリート を作成した場合、お客様の代わりにインスタンスのリクエスト、起動、削除、タグ付けを行うアクセス許可を スspotフリート に与える aws-ec2-spot-fleet-tagging-role という名前のロールが作成されます。このロールは、スspotフリート リクエストを作成するときに選択されます。代わりに AWS CLI または API を使用する場合、このロールが存在していることを確認する必要があります。スspotインスタンス リクエスト ウィザード (ウィザードの 2 ページ目に進むとロールが作成されます) または、次のように、IAM コンソールを使用できます。

### Important

フリート内のインスタンスにタグ付けすることを選択し、ターゲット容量を維持することを選択した場合 (スspotフリート リクエストのタイプは maintain)、IAM ユーザーと

IamFleetRole のアクセス許可の違いにより、フリート内のインスタンスのタグ付け動作に整合性がなくなる可能性があります。IamFleetRole に CreateTags アクセス許可が含まれていない場合、フリートによって起動されたインスタンスの一部がタグ付けされていない可能性があります。AWS はこの不整合の修正に取り組んでいますが、フリートによって起動されたすべてのインスタンスがタグ付けされるようにするために、IamFleetRole には aws-ec2-spot-fleet-tagging-role ロールを使用することをお勧めします。または、既存のロールを使用するには、AmazonEC2SpotFleetTaggingRole AWS 管理ポリシーを既存のロールにアタッチします。それ以外の場合は、既存のポリシーに CreateTags アクセス許可を手動で追加する必要があります。

### スポットフリートの IAM ロールを作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [Roles (ロール)] を選択します。
3. [Select type of trusted entity] ページの [AWS Service] で [EC2]、[EC2 - Spot Fleet Tagging]、[Next: Permissions] の順に選択します。
4. [Attached permissions policy] ページで、[Next: Review] を選択します。
5. [Review] ページで、ロールの名前 (たとえば、**aws-ec2-spot-fleet-tagging-role**) を入力し、[Create role] を選択します。

### スポットフリートユーザーと IAM ユーザー

IAM ユーザーがスポットフリートを作成または管理する場合、必ず次のようにして必要なアクセス許可を付与してください。

#### IAM ユーザーにスポットフリートのアクセス許可を付与するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで、[Policies]、[Create policy] の順に選択します。
3. [Create policy] ページで、[JSON] を選択し、テキストを以下に置き換えて [Review policy] を選択します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:/*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam>ListRoles",
                "iam>PassRole",
                "iam>ListInstanceProfiles"
            ],
            "Resource": "*"
        }
    ]
}
```

ec2:/\* は、IAM ユーザーにすべての Amazon EC2 API アクションを呼び出すアクセス許可を付与します。特定の Amazon EC2 API アクションに制限するには、代わりにこれらのアクションを指定します。

IAM ユーザーは、既存の IAM ロールを列挙する `iam>ListRoles` アクション、スポットフリート ロールを指定する `iamPassRole` アクション、および既存のインスタンスプロファイルを列挙する `iamListInstanceProfiles` アクションを呼び出すには、アクセス許可が必要です。

(オプション) IAM ユーザーが IAM コンソールを使用してロールまたはインスタンスプロファイルを作成できるようにするには、次のアクションをポリシーに追加する必要があります。

- `iamAddRoleToInstanceProfile`
  - `iamAttachRolePolicy`
  - `iamCreateInstanceProfile`
  - `iamCreateRole`
  - `iamGetRole`
  - `iamListPolicies`
4. [Review policy] ページでポリシー名と説明を入力し、[Create policy] を選択します。
  5. ナビゲーションペインで [Users] を選択し、ユーザーを選択します。
  6. [Permissions]、[Add permissions] の順に選択します。
  7. [Attach existing policies directly] を選択します。以前に作成したポリシーを選択し、[Next: Review] を選択します。
  8. [Add permissions] を選択します。

## スポットフリート ヘルスチェック

スポットフリート は、2 分ごとにフリートの スポットインスタンス のヘルステータスをチェックします。インスタンスのヘルステータスは `healthy` または `unhealthy` です。スポットフリート は、Amazon EC2 が提供するステータスチェックを使用してインスタンスの正常性状態を判断します。インスタンスステータスチェックとシステムヘルスチェックのいずれかのステータスが、連続した 3 回のヘルスチェックで `impaired` である場合、インスタンスのヘルステータスは `unhealthy` になります。それ以外の場合、ヘルステータスは `healthy` になります。詳細については、「[インスタンスのステータスチェック \(p. 628\)](#)」を参照してください。

異常なインスタンスは置き換えるよう スpotフリート を設定できます。ヘルスチェックの置換を有効にすると、ヘルステータスが `unhealthy` と報告された後でインスタンスが置き換えられます。異常なインスタンスを置き換えている間、数分間にわたり スpotフリート がターゲット容量を下回る場合があります。

### 要件

- ヘルスチェックの置換は、1 回限りの スpotフリート ではなく、ターゲット容量を維持する スpotフリート でのみサポートされます。
- 作成時ののみ異常なインスタンスを置き換えるよう スpotフリート を設定できます。
- IAM ユーザーは、`ec2DescribeInstanceStatus` アクションを呼び出すアクセス許可を持っている場合のみ、ヘルスチェックの置換を使用できます。

## スポットフリート リクエストの準備

スポットフリート リクエストを作成する前に、「[ベストプラクティス](#)」を確認してください。スポットフリート リクエストを計画するときにこれらのベストプラクティスを使用して、できるだけ低価格でインスタンスのタイプをプロビジョニングできるようにします。また、次のことをお勧めします。

- 目的のターゲット容量のワンタイムリクエストを送信する スpotフリート と、ターゲット容量の継続した維持を行うスspotフリートのどちらを作成するかを決定します。
- アプリケーションの要件を満たすインスタンスタイプを決定します。

- スポットフリート リクエストの目標容量を決定します。インスタンスまたはカスタムユニットでターゲット容量を設定できます。詳細については、「[スポットフリート インスタンスの分量指定 \(p. 329\)](#)」を参照してください。
- スpotフリート のターゲットキャパシティーのどの部分がオンデマンドキャパシティーとなるかを決定します。オンデマンドキャパシティーに対して 0 を指定できます。
- インスタンス分量指定を使用している場合は、ユニット当りの料金を決定します。インスタンス時間当たりの料金の計算は、インスタンス時間当たりの料金をそのインスタンスが表すユニット数(または分量)で割って算出しますインスタンス分量指定を使用する場合、ユニット当りのデフォルトの料金は 1 インスタンス時間当りの料金となります。
- スpotフリート のリクエストに対して可能なオプションを確認します。詳細については、『AWS CLI Command Reference』の [request-spot-fleet](#) コマンドを参照してください。その他の例については、「[スポットフリート 設定例 \(p. 362\)](#)」を参照してください。

## スポットフリート リクエスト用のサービスにリンクされたロール

Amazon EC2 は、ユーザーに代わって AWS の他のサービスを呼び出すために必要なアクセス許可を持つサービスにリンクされたロールを使用します。サービスにリンクされたロールは、AWS のサービスに直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは、AWS のサービスにアクセス許可を委任するためのセキュアな方法を提供します。これは、リンクされたサービスのみが、サービスにリンクされたロールを引き受けることができるためです。詳細については、IAM ユーザーガイドの「[サービスにリンクされたロールの使用](#)」を参照してください。

Amazon EC2 は、[AWSServiceRoleForEC2SpotFleet](#) という、サービスにリンクされたロールを使用して、ユーザーの代わりに スpotインスタンス を起動して管理します。

### AWSServiceRoleForEC2SpotFleet によって付与されるアクセス許可

Amazon EC2 は、[\[AWSServiceRoleForEC2SpotFleet\]](#) という、サービスにリンクされたロールを使用して、次のアクションを実行します。

- `ec2:RequestSpotInstances` - スpotインスタンス のリクエスト
- `ec2:TerminateInstances` - スpotインスタンス の終了
- `ec2:DescribeImages` - スpotインスタンス の Amazon Machine Image (AMI) の説明
- `ec2:DescribeInstanceStatus` - スpotインスタンス のステータスの説明
- `ec2:DescribeSubnets` - スpotインスタンス のサブネットの説明
- `ec2:CreateTags` - スpotインスタンス へのシステムタグの追加

### サービスにリンクされたロールの作成

ほとんどの状況では、サービスにリンクされたロールを手動で作成する必要はありません。コンソールを使用して スpotフリート を初めてリクエストしたときに、サービスにリンクされたロールとして [AWSServiceRoleForEC2SpotFleet](#) が Amazon EC2 で自動的に作成されます。

Amazon EC2 がこのサービスにリンクされたロールのサポートを開始した 2017 年 10 月より前にアクティブな スpotフリート リクエストを行った場合は、Amazon EC2 が AWS アカウントで [AWSServiceRoleForEC2SpotFleet](#) ロールを作成しています。詳細については、IAM ユーザーガイドの「[AWS アカウントに新しいロールが表示される](#)」を参照してください。

AWS CLI または API を使って スpotフリート を作成する前にこのロールが存在していることを確認します。ロールを作成するには、IAM コンソールを次のように使用します。

サービスにリンクされたロール [AWSServiceRoleForEC2SpotFleet](#) を手動で作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [Roles (ロール)] を選択します。

3. [ロールの作成] を選択します。
4. [Select type of trusted entity] ページで、[EC2]、[EC2 - Spot Fleet]、[Next: Permissions] の順に選択します。
5. 次のページで、[Next:Review (次の手順: 確認)] を選択します。
6. [確認] ページで、[ロールの作成] を選択します。

スポットフリートを使用する必要がなくなった場合は、[AWSServiceRoleForEC2SpotFleet] ロールを削除することをお勧めします。このロールがアカウントから削除された後で、Amazon EC2 をリクエストすると、スポットフリートはロールを再度作成します。

### 暗号化された AMI および EBS スナップショット用に CMK へのアクセスを付与する

暗号化された AMI (p. 151) または暗号化された Amazon EBS スナップショット (p. 1014) をスポットフリート リクエストで指定し、カスタマー管理 CMK (カスタマーマスターキー) を暗号化に使用する場合は、CMK を使用するアクセス許可を AWSServiceRoleForEC2SpotFleet ロールに付与して Amazon EC2 がユーザーに代わって スpotインスタンスを起動できるようにする必要があります。これを行うには、次の手順で示すように、CMK に付与を追加する必要があります。

アクセス権限を設定するときは、付与がキー ポリシーの代わりになります。詳細については、「[許可の使用](#)」と「[AWS KMS でのキー ポリシーの使用](#)」(AWS Key Management Service Developer Guide) を参照してください。

CMK を使用するアクセス許可を AWSServiceRoleForEC2SpotFleet ロールに付与するには

- `create-grant` コマンドを使用して CMK に付与を追加し、プリンシパル (サービスにリンクされたロール AWSServiceRoleForEC2SpotFleet) を指定します。このプリンシパルには、付与が許可するオペレーションを実行するためのアクセス許可が提供されます。CMK を指定するには、key-id パラメータと CMK の ARN を使用します。プリンシパルを指定するには、grantee-principal パラメータとサービスにリンクされたロール AWSServiceRoleForEC2SpotFleet の ARN を使用します。

次の例は、読みやすいようにフォーマットされています。

```
aws kms create-grant \
--region us-east-1 \
--key-id arn:aws:kms:us-
east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab \
--grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2SpotFleet \
--operations "Decrypt" "Encrypt" "GenerateDataKey"
"GenerateDataKeyWithoutPlaintext" "CreateGrant" "DescribeKey" "ReEncryptFrom"
"ReEncryptTo"
```

## スポットフリート リクエストを作成する

AWS マネジメントコンソールを使用して、アプリケーションまたはタスクのニーズのみと最低限のコンピューティング仕様を選択し、スポットフリートを迅速に作成します。Amazon EC2 はユーザーのニーズに最適なフリートを設定し、ベストプラクティスに従います。詳細については、「[スポットフリート リクエストをすばやく作成する \(コンソール\) \(p. 350\)](#)」を参照してください。それ以外の場合は、デフォルト設定のいかかを変更できます。詳細については、「[定義済みパラメータを使用してスポットフリートを作成する \(コンソール\) \(p. 351\)](#)」を参照してください。

### スポットフリート リクエストをすばやく作成する (コンソール)

スポットフリート リクエストをすばやく作成するには、以下の手順を実行します。

推奨設定を使用して スpotフリート リクエストを作成するには (コンソール)

1. スpotコンソール (<https://console.aws.amazon.com/ec2spot>) を開きます。

2. スポットを初めて使用する場合は、ウェルカムページが表示されるので、そこで [Get started] を選択します。それ以外の場合は、[スポットインスタンスのリクエスト] を選択します。
3. [Tell us your application or task need] については、[Flexible workloads]、[Load balancing workloads]、[Big data workloads]、または [Defined duration workloads] を選択します。
4. [Configure your instances] の下の [Minimum compute unit] で、アプリケーションまたはタスクで必要な最低限のハードウェア仕様 (vCPU、メモリ、ストレージ) を選択して、[as specs] または [as an instance type] を指定します。
  - [as specs] については、必要な vCPU 数とメモリ量を指定します。
  - [as an instance type] については、デフォルトのインスタンスタイプをそのまま使用するか、[Change instance type] を選択して別のインスタンスタイプを選択します。
5. [Tell us how much capacity you need] の下の [Total target capacity] で、ターゲット容量を要求するための単位数を指定します。インスタンスまたは vCPU を選択できます。
6. アプリケーションまたはタスクの選択に基づいた推奨される [Fleet request settings] を確認して、[Launch] を選択します。

### 定義済みパラメータを使用して スポットフリートを作成する (コンソール)

定義するパラメータを使用して、スポットフリートを作成できます。

### 定義済みパラメータを使用して スポットフリート リクエストを作成するには (コンソール)

1. スポットコンソール (<https://console.aws.amazon.com/ec2spot>) を開きます。
2. スポットを初めて使用する場合は、ウェルカムページが表示されるので、そこで [Get started] を選択します。それ以外の場合は、[スポットインスタンスのリクエスト] を選択します。
3. [Tell us your application or task need] については、[Flexible workloads]、[Load balancing workloads]、[Big data workloads]、または [Defined duration workloads] を選択します。
4. [Configure your instances] で、以下を実行します。
  - a. (オプション) [スポットフリート] [Launch template] で、起動テンプレートを選択します。起動テンプレートでは Amazon Machine Image (AMI) を指定する必要があります。起動テンプレートを指定した場合、スポットフリートを使用して AMI を上書きできないためです。

#### Important

[オプションのオンデマンド部分] を指定する場合、起動テンプレートを選択する必要があります。

- b. [AMI] では、AWS に用意されたベーシック AMI のいずれかを選択するか、[Search for AMI] を選択してユーザー コミュニティの AMI、AWS Marketplace の AMI、または独自の AMI を使用します。
- c. [Minimum compute unit] で、アプリケーションまたはタスクで必要な最低限のハードウェア仕様 (vCPU、メモリ、ストレージ) を選択して、[as specs] または [as an instance type] を指定します。
  - [as specs] については、必要な vCPU 数とメモリ量を指定します。
  - [as an instance type] については、デフォルトのインスタンスタイプをそのまま使用するか、[Change instance type] を選択して別のインスタンスタイプを選択します。
- d. (オプション) [Network] で、既存の VPC を使用するか、新しい VPC を作成するかを選択します。

[既存の VPC] VPC を選択します。

[新しい VPC] [新しい VPC の作成] を選択して Amazon VPC コンソールにアクセスします。完了したら、ウィザードに戻ってリストを更新します。

- e. (オプション) [Availability Zones] では、AWS で自動的にスポットインスタンスのアベイラビリティーフィールドを選択するか、または 1つ以上のアベイラビリティーフィールドを指定します。

アベイラビリティーフィールドに複数のサブネットがある場合、[Subnet] から適切なサブネットを選択します。サブネットを追加するには、[Create new subnet] を選択して Amazon VPC にアクセスします。完了したら、ウィザードに戻ってリストを更新します。

- f. (オプション) [Key pair name] で、既存のキーペアを使用するか、新しいキーペアを作成するかを選択します。

[Existing key pair] キーペアを選択します。

[New key pair] [Create new key pair] を選択して Amazon VPC コンソールにアクセスします。完了したら、ウィザードに戻ってリストを更新します。

5. (オプション) [Additional configurations] で、以下を実行します。

- a. (オプション) ストレージを追加するには、インスタンスタイプに応じて追加のインスタンストアボリュームまたは Amazon EBS ボリュームを指定します。
- b. (オプション) Amazon EBS 最適化を有効にするには、[EBS-optimized] で [Launch EBS-optimized instances] を選択します。
- c. (オプション) インスタンス用の一時ブロックレベルストレージを追加するには、[Instance store] で [Attach at launch] を選択します。
- d. (オプション) デフォルトでは、インスタンスに対して基本モニタリングが有効になります。詳細モニタリングを有効にするには、[Monitoring (モニタリング)] で [Enable CloudWatch detailed monitoring] を選択します。
- e. (オプション) 異常なインスタンスを置き換えるには、[Health check (ヘルスチェック)] で [Replace unhealthy instances (異常なインスタンスを置き換える)] を選択します。このオプションを有効にするには、まず [Maintain target capacity (ターゲット容量を維持する)] を選択する必要があります。
- f. (オプション) ハードウェア専有 スpotトインスタンスを実行するには、[テナンシー] で [専有 - ハードウェア専有インスタンスの実行] を選択します。
- g. (オプション) [Security groups] で、1つ以上のセキュリティグループを選択するか、新しいセキュリティグループを作成します。

[Existing security group] 1つ以上のセキュリティグループを選択します。

[New security group] [Create new security group] を選択して、Amazon VPC コンソールに移動します。完了したら、ウィザードに戻ってリストを更新します。

- h. (オプション) インスタンスにインターネットからアクセスできるようにするには、[Auto-assign IPv4 Public IP] で [Enable] を選択します。
- i. (オプション) IAM ロールを指定して スpotトインスタンスを起動するには、[IAM instance profile] でロールを選択します。
- j. (オプション) 起動スクリプトを実行するには、スクリプトを [User data] にコピーします。
- k. (オプション) タグを追加するには、[Add new tag] を選択し、そのタグのキーと値を入力します。各タグについて、これを繰り返します。

タグごとに、インスタンスと スpotトフリート リクエストに同じタグを付けるには、[Instance tags (インスタンスタグ)] と [Fleet tags (フリートタグ)] の両方が選択されていることを確認します。フリートによって起動されたインスタンスのみにタグ付けするには、[Fleet tags (フリートタグ)] をクリアします。スspotトフリート リクエストのみにタグ付けするには、[Instance tags (インスタンスタグ)] をクリアします。

6. [Tell us how much capacity you need] で、以下を実行します。

- a. [Total target capacity] で、ターゲットキャパシティにリクエストする単位数を指定します。インスタンスまたは vCPU を選択できます。ターゲット容量を 0 に指定して後で容量を追加できるようにするには、[Maintain target capacity] を選択します。

- b. (オプション) [Optional On-Demand portion] でリクエストするオンデマンド単位数を指定します。この数は、[Total target capacity] より少なくする必要があります。Amazon EC2 は差異を計算し、この差をリクエストするスポット単位に割り当てます。

Important

オプションのオンデマンド部分を指定する場合、最初に起動テンプレートを選択する必要があります。

- c. (オプション) デフォルトでは、スポットサービスは中断されるとスポットインスタンスを削除します。ターゲット容量を維持するには、[Maintain target capacity] を選択します。これで、中断時にスポットインスタンスを終了、停止、または休止するように指定できます。これを行うには、[Interruption behavior] から対応するオプションを選択します。

7. [Fleet request settings] で、以下を実行します。

- a. アプリケーションまたはタスクの選択に基づいて、フリートリクエストおよびフリートの配分戦略を確認します。インスタンスタイプまたは配分戦略を変更するには、[Apply recommendations] をオフにします。

- b. (オプション) インスタンスタイプを削除するには、[Fleet request] で [Remove] を選択します。インスタンスタイプを追加するには、[Select instance types] を選択します。

- c. [オプション] [Fleet allocation strategy] で、お客様のニーズに合った戦略を選択します。詳細については、「[スポットインスタンス の配分戦略 \(p. 327\)](#)」を参照してください。

8. [Additional request details] で、以下を実行します。

- a. 追加リクエストの詳細を確認します。変更するには、[Apply defaults] をオフにします。

- b. (オプション) [IAM fleet role] で、デフォルトのロールを使用するか、または別のロールを選択できます。ロールの変更後にデフォルトのロールを使用するには、[Use default role] を選択します。

- c. (オプション) [Maximum price] では、デフォルトの上限料金(オンデマンド料金)を使用するか、支払う予定の上限料金を指定することができます。上限価格が選択したインスタンスタイプのスポット料金より低い場合、スポットインスタンスは起動されません。

- d. (オプション) 特定の期間中のみ有効なリクエストを作成するには、[Request valid from] および [Request valid until] を編集します。

- e. (オプション) デフォルトでは、リクエストの有効期限が切れるとスポットインスタンスは終了します。リクエストの有効期限が切れた後も実行し続ける場合、[Terminate the instances when the request expires] をオフにします。

- f. (オプション) ロードバランサーを使用するスポットインスタンスを登録するには、[Receive traffic from one or more load balancers] を選択して、1つ以上のクラシックロードバランサーまたはターゲットグループを選択します。

9. (オプション) AWS CLI で使用される起動設定のコピーをダウンロードするには、[JSON config] を選択します。

10. [Launch] を選択します。

スポットフリート リクエストタイプは `fleet` です。リクエストが実行されると、タイプ `instance` のリクエストが追加されます。このとき、状態は `active` になり、ステータスは `fulfilled` になります。

AWS CLI を使用して スポットフリート リクエストを作成するには

- ・ スpotフリート リクエストを作成するには、以下の `request-spot-fleet` コマンドを使用します。

```
aws ec2 request-spot-fleet --spot-fleet-request-config file://config.json
```

設定ファイルの例については、「[スポットフリート 設定例 \(p. 362\)](#)」を参照してください。

出力例を次に示します。

```
{  
    "SpotFleetRequestId": "sfr-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE"  
}
```

## スポットフリート のタグ付け

スポットフリート リクエストを分類および管理しやすくするために、カスタムメタデータでタグ付けすることができます。スポットフリート リクエストへのタグの割り当ては、リクエストの作成時または作成後に行うことができます。Amazon EC2 コンソールまたはコマンドラインツールを使用してタグを割り当てることができます。

スポットフリート リクエストにタグ付けすると、スポットフリート によって起動されたインスタンスは自動的にタグ付けされません。スポットフリート によって起動されたインスタンスには明示的にタグ付けする必要があります。タグは、スポットフリート リクエストにのみ、フリートによって起動されたインスタンスにのみ、またはその両方に割り当てる 것을 선택できます。

タグの仕組みの詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。

### 前提条件

リソースにタグ付けするアクセス許可を IAM ユーザーに付与します。詳細については、「[例: リソースのタグ付け \(p. 873\)](#)」を参照してください。

リソースにタグ付けするアクセス許可を IAM ユーザーに付与するには

以下を含む IAM ポリシーを作成します。

- `ec2:CreateTags` アクション。これにより、タグを作成するアクセス許可が IAM ユーザーに付与されます。
- `ec2:RequestSpotFleet` アクション。これにより、スポットフリート リクエストを作成するアクセス許可が IAM ユーザーに付与されます。
- `Resource` で、`"*"` を指定する必要があります。これにより、ユーザーはすべてのリソースタイプにタグ付けできます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "TagSpotFleetRequest",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags",  
                "ec2:RequestSpotFleet"  
            ],  
            "Resource": "*"  
        }  
    ]
```

### Important

現在、`spot-fleet-request` リソースに対するリソースレベルのアクセス許可はサポートされていません。リソースとして `spot-fleet-request` を指定した場合、フリートにタグ付けしようとすると、不正な例外が発生します。以下の例は、ポリシーを設定しない方法を示しています。

```
{
```

```
"Effect": "Allow",
"Action": [
    "ec2:CreateTags",
    "ec2:RequestSpotFleet"
],
"Resource": "arn:aws:ec2:us-east-1:11122223333:spot-fleet-request/*"
```

## コンソールを使用して新しいスポットフリート リクエストにタグ付けするには

- 「[定義済みパラメータを使用してスポットフリートを作成する \(コンソール\) \(p. 351\)](#)」の手順に従います。
- タグを追加するには、[追加設定] を展開し、[新規タグの追加] を選択して、タグのキーと値を入力します。各タグについて、これを繰り返します。

タグごとに、スポットフリート リクエストとインスタンスに同じタグを付けることができます。両方にタグ付けするには、[Instance tags (インスタンスタグ)] と [Fleet tags (フリートタグ)] の両方が選択されていることを確認します。スポットフリート リクエストのみにタグ付けするには、[Instance tags (インスタンスタグ)] をクリアします。フリートによって起動されたインスタンスのみにタグ付けするには、[Fleet tags (フリートタグ)] をクリアします。

- 必須フィールドに入力してスポットフリート リクエストを作成し、[起動] を選択します。詳細については、「[定義済みパラメータを使用してスポットフリートを作成する \(コンソール\) \(p. 351\)](#)」を参照してください。

## AWS CLI を使用して新しいスポットフリート リクエストにタグ付けするには

スポットフリート リクエストの作成時にタグ付けするには、以下のように スpotフリート リクエスト設定を定義します。

- SpotFleetRequestConfig で、スポットフリート リクエストのタグを指定します。
- ResourceType で、spot-fleet-request を指定します。別の値を指定すると、フリートリクエストは失敗します。
- Tags で、キーと値のペアを指定します。キーと値のペアは複数指定できます。

以下の例では、スポットフリート リクエストに 2 つのタグ (Key=Environment and Value=Production と Key=Cost-Center and Value=123) が付けられています。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IAMFleetRole": "arn:aws:iam::11122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large"
            }
        ],
        "SpotPrice": "5",
        "TargetCapacity": 2,
        "TerminateInstancesWithExpiration": true,
        "Type": "maintain",
        "ReplaceUnhealthyInstances": true,
        "InstanceInterruptionBehavior": "terminate",
        "InstancePoolsToUseCount": 1,
        "TagSpecifications": [
            {
                "ResourceType": "spot-fleet-request",
                "Tags": [
                    {
                        "Key": "Environment",
                        "Value": "Production"
                    },
                    {
                        "Key": "Cost-Center",
                        "Value": "123"
                    }
                ]
            }
        ]
    }
}
```

```
    "Tags": [
      {
        "Key": "Environment",
        "Value": "Production"
      },
      {
        "Key": "Cost-Center",
        "Value": "123"
      }
    ]
}
```

新しい スポットフリート リクエストと、AWS CLI を使用して起動されるインスタンスにタグ付けするには

スポットフリート リクエストが作成されるときにそのリクエストにタグ付けし、フリートによってインスタンスが起動されるときにそのインスタンスにタグ付けするには、スポットフリート リクエスト設定を以下のように指定します。

#### スポットフリート リクエストタグ:

- `SpotFleetRequestConfig` で、スポットフリート リクエストのタグを指定します。
- `ResourceType` で、`spot-fleet-request` を指定します。別の値を指定すると、フリートリクエストは失敗します。
- `Tags` で、キーと値のペアを指定します。キーと値のペアは複数指定できます。

#### インスタンスタグ:

- `LaunchSpecifications` で、インスタンスのタグを指定します。
- `ResourceType` で、`instance` を指定します。別の値を指定すると、フリートリクエストは失敗します。
- `Tags` で、キーと値のペアを指定します。キーと値のペアは複数指定できます。

または、スポットフリート リクエストで参照される[起動テンプレート \(p. 456\)](#)でインスタンスのタグを指定できます。

以下の例では、スポットフリート リクエストに 2 つのタグ (`Key=Environment` and `Value=Production` と `Key=Cost-Center` and `Value=123`) が付けられています。フリートによって起動されるインスタンスには、1 つのタグ (スポットフリート リクエストのタグの 1 つと同じ `Key=Cost-Center` and `Value=123`) が付けられます。

```
{
  "SpotFleetRequestConfig": {
    "AllocationStrategy": "lowestPrice",
    "ExcessCapacityTerminationPolicy": "default",
    "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
      {
        "ImageId": "ami-0123456789EXAMPLE",
        "InstanceType": "c4.large",
        "TagSpecifications": [
          {
            "ResourceType": "instance",
            "Tags": [
              {
                "Key": "Environment",
                "Value": "Production"
              },
              {
                "Key": "Cost-Center",
                "Value": "123"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
        "Key": "Cost-Center",
        "Value": "123"
    }
}
]
],
"SpotPrice": "5",
"TargetCapacity": 2,
"TerminateInstancesWithExpiration": true,
>Type": "maintain",
"ReplaceUnhealthyInstances": true,
"InstanceInterruptionBehavior": "terminate",
"InstancePoolsToUseCount": 1,
"TagSpecifications": [
{
    "ResourceType": "spot-fleet-request",
    "Tags": [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Cost-Center",
            "Value": "123"
        }
    ]
}
]
}
```

AWS CLI を使用して スポットフリート によって起動されたインスタンスにタグ付けするには

フリートによってインスタンスが起動されるときにタグ付けするには、スポットフリート リクエストで参照される[起動テンプレート \(p. 456\)](#)でタグを指定するか、以下のように スポットフリート リクエスト設定でタグを指定できます。

- `LaunchSpecifications` で、インスタンスのタグを指定します。
- `ResourceType` で、`instance` を指定します。別の値を指定すると、フリートリクエストは失敗します。
- `Tags` で、キーと値のペアを指定します。キーと値のペアは複数指定できます。

以下の例では、フリートによって起動されるインスタンスに 1 つのタグ (`Key=Cost-Center` and `Value=123`) が付けられています。

```
{
    "SpotFleetRequestConfig": {
        "AllocationStrategy": "lowestPrice",
        "ExcessCapacityTerminationPolicy": "default",
        "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
        "LaunchSpecifications": [
            {
                "ImageId": "ami-0123456789EXAMPLE",
                "InstanceType": "c4.large",
                "TagSpecifications": [
                    {
                        "ResourceType": "instance",
                        "Tags": [
                            {
                                "Key": "Cost-Center",

```

```
        "Value": "123"
    }
}
],
"SpotPrice": "5",
"TargetCapacity": 2,
"TerminateInstancesWithExpiration": true,
"Type": "maintain",
"ReplaceUnhealthyInstances": true,
"InstanceInterruptionBehavior": "terminate",
"InstancePoolsToUseCount": 1
}
}
```

AWS CLI を使用して既存の スポットフリート リクエストにタグ付けするには

[create-tags](#) コマンドを使用して、既存のリソースにタグ付けできます。以下の例では、既存の スポットフリート リクエストに 1 つのタグ (Key=purpose and Value=test) が付けられています。

```
aws ec2 create-tags \
--resources sfr-11112222-3333-4444-5555-66666EXAMPLE \
--tags Key=purpose,Value=test
```

コンソールを使用して既存の スポットフリート リクエストにタグ付けするには

スポットフリート リクエストを作成したら、コンソールを使用してフリートリクエストにタグを追加できます。

1. スポットコンソール (<https://console.aws.amazon.com/ec2spot>) を開きます。
2. スポットフリート リクエストを選択します。
3. [Tags (タグ)] タブを選択してから、[タグの作成] を選択します。

コンソールを使用して スポットフリート リクエストタグを表示するには

1. スポットコンソール (<https://console.aws.amazon.com/ec2spot>) を開きます。
2. スポットフリート リクエストを選択してから、[Tags (タグ)] タブを選択します。

スポットフリート リクエストタグの情報を取得するには

[describe-tags](#) コマンドを使用して、指定したリソースのタグを表示します。以下の例では、指定した スポットフリート リクエストのタグの情報を取得します。

```
aws ec2 describe-tags \
--filters "Name=resource-id,Values=sfr-11112222-3333-4444-5555-66666EXAMPLE"
```

```
{
  "Tags": [
    {
      "Key": "Environment",
      "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
      "ResourceType": "spot-fleet-request",
      "Value": "Production"
    },
    {
      "Key": "Environment",
      "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
      "ResourceType": "spot-fleet-request",
      "Value": "Production"
    }
  ]
}
```

```
        "Key": "Another key",
        "ResourceId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
        "ResourceType": "spot-fleet-request",
        "Value": "Another value"
    }
]
}
```

スポットフリート リクエストの情報を取得することで、スポットフリート リクエストのタグを表示することもできます。

[describe-spot-fleet-requests](#) コマンドを使用して、指定した スポットフリート リクエストの設定を表示します。これには、フリートリクエストに指定されたタグが含まれます。

```
aws ec2 describe-spot-fleet-requests \
--spot-fleet-request-ids sfr-11112222-3333-4444-5555-66666EXAMPLE
```

```
{
    "SpotFleetRequestConfigs": [
        {
            "ActivityStatus": "fulfilled",
            "CreateTime": "2020-02-13T02:49:19.709Z",
            "SpotFleetRequestConfig": {
                "AllocationStrategy": "capacityOptimized",
                "OnDemandAllocationStrategy": "lowestPrice",
                "ExcessCapacityTerminationPolicy": "Default",
                "FulfilledCapacity": 2.0,
                "OnDemandFulfilledCapacity": 0.0,
                "IamFleetRole": "arn:aws:iam::111122223333:role/aws-ec2-spot-fleet-tagging-role",
                "LaunchSpecifications": [
                    {
                        "ImageId": "ami-0123456789EXAMPLE",
                        "InstanceType": "c4.large"
                    }
                ],
                "TargetCapacity": 2,
                "OnDemandTargetCapacity": 0,
                "Type": "maintain",
                "ReplaceUnhealthyInstances": false,
                "InstanceInterruptionBehavior": "terminate"
            },
            "SpotFleetRequestId": "sfr-11112222-3333-4444-5555-66666EXAMPLE",
            "SpotFleetRequestState": "active",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                },
                {
                    "Key": "Another key",
                    "Value": "Another value"
                }
            ]
        }
    ]
}
```

## スポットフリート のモニタリング

上限料金がスポット料金を超える場合、容量が利用可能な場合、スポットフリートは スポットインスタンスを起動します。スポットインスタンスは中断されるか手動終了されるまで実行されます。

## スポットフリートを監視するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。
3. スポットフリート リクエストを選択します。設定の詳細を表示するには、[Description] を選択します。
4. スポットフリート の スpotインスタンスを一覧表示するには、[Instances] を選択します。
5. スポットフリート の履歴を表示するには、[History] を選択します。

## スポットフリートを監視するには (AWS CLI)

スポットフリート リクエストの詳細を表示するには、以下の [describe-spot-fleet-requests](#) コマンドを使用します。

```
aws ec2 describe-spot-fleet-requests
```

指定した スポットフリート の スpotインスタンス の 詳細を表示するには、以下の [describe-spot-fleet-instances](#) コマンドを使用します。

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fdb2ce-aa30-494c-8788-1cee4EXAMPLE
```

指定した スポットフリート リクエストの履歴を表示するには、以下の [describe-spot-fleet-request-history](#) コマンドを使用します。

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fdb2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-18T00:00:00Z
```

## スポットフリート リクエストの変更

以下のタスクを完了するように、アクティブな スポットフリート リクエストを変更できます。

- ターゲット容量とオンデマンド部分を増やす
- ターゲット容量とオンデマンド部分を減らす

### Note

ワンタイム スポットフリート リクエストは変更できません。スポットフリート リクエストを作成したときに [Maintain target capacity (ターゲット容量を維持する)] を選択した場合にのみ、スポットフリート リクエストを修正することができます。

ターゲット容量を増やすと、スポットフリート は スpotインスタンス を追加で起動します。ターゲット容量を減らすと、スポットフリート は オンデマンドインスタンス を追加で起動します。

ターゲット容量を増やす場合、スポットフリート は、スポットフリート リクエストの配分戦略に従って追加の スpotインスタンス を起動します。配分戦略が `lowestPrice` の場合、スポットフリート は、スポットフリート リクエストの最低価格の スpotインスタンス プールからインスタンスを起動します。配分戦略が `diversified` の場合、スポットフリート は、スポットフリート リクエストの プールにインスタンスを分散します。

ターゲット容量を減らす場合、スポットフリート は 新しいターゲット容量を超えるすべてのオーブンリクエストをキャンセルします。スポットフリート の サイズが新しいターゲット容量に達するとスポット群の スpotインスタンス が終了されるようにリクエストできます。配分戦略が `lowestPrice` である場合

は、スポットフリート の最低単価のインスタンスが終了されます。配分戦略が diversified である場合は、スポットフリート のプール全体でインスタンスが終了されます。あるいは、スポットフリート の現在のサイズを保持するようにリクエストすることもできますが、中断または手動終了された スポットインスタンスへの置き換えはできません。

ターゲット容量が減ったためにスポットフリート によってインスタンスが削除される場合、インスタンスは スポットインスタンス の中断通知を受け取ります。

#### スポットフリート リクエストを変更するには (コンソール)

1. スポットコンソール (<https://console.aws.amazon.com/ec2spot/home/fleet>) を開きます。
2. スpotフリート リクエストを選択します。
3. [Actions]、[Modify target capacity] の順に選択します。
4. [Modify target capacity] で、以下の操作を実行します。
  - a. 新しいターゲット容量とオンデマンド部分を入力します。
  - b. (オプション) ターゲット容量を小さくしてもスポット群の現在のサイズを保持する場合は、[Terminate instances] をオフにします。
  - c. [Submit] を選択します。

AWS CLI を使用して スpotフリート リクエストを変更するには

次の `modify-spot-fleet-request` コマンドを使用して、指定する スpotフリート リクエストのターゲット容量を更新します。

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fdbd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity 20
```

前のコマンドを以下のように変更することで、結果的にいずれの スpotインスタンス も終了せずに、指定した スpotフリート のターゲット容量を減らすことができます。

```
aws ec2 modify-spot-fleet-request --spot-fleet-request-id sfr-73fdbd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity 10 --excess-capacity-termination-policy NoTermination
```

#### スポットフリート リクエストのキャンセル

スポットフリート を使用しなくなったら、スポットフリート リクエストをキャンセルできます。これにより、スポットフリート に関連付けられているすべてのスポットリクエストがキャンセルされるため、そのスポットフリート の新しいスポットインスタンスは起動されなくなります。スポットフリート の スpotインスタンス を終了するかどうか指定する必要があります。インスタンスを終了する場合、スポットフリート リクエストは `cancelled_terminating` 状態になります。それ以外の場合、スポットフリート リクエストは `cancelled_running` 状態になり、インスタンスは中断または手動終了されるまで、引き続き実行されます。

#### スポットフリート リクエストをキャンセルするには (コンソール)

1. スポットコンソール (<https://console.aws.amazon.com/ec2spot/home/fleet>) を開きます。
2. スpotフリート リクエストを選択します。
3. [Actions]、[Cancel spot request] の順に選択します。
4. [スポットリクエストをキャンセルする] で、スポットフリート のキャンセルを確認します。スポット群の現在のサイズを保持するには、[Terminate instances] をオフにします。準備ができたら、[Confirm] を選択します。

AWS CLI を使用して スポットフリート をキャンセルするには

指定した スポットフリート リクエストをキャンセルし、インスタンスを終了するには、以下の [cancel-spot-fleet-requests](#) コマンドを使用します。

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fb2ce-  
aa30-494c-8788-1cee4EXAMPLE --terminate-instances
```

出力例を次に示します。

```
{  
    "SuccessfulFleetRequests": [  
        {  
            "SpotFleetRequestId": "sfr-73fb2ce-aa30-494c-8788-1cee4EXAMPLE",  
            "CurrentSpotFleetRequestState": "cancelled_terminating",  
            "PreviousSpotFleetRequestState": "active"  
        }  
    ],  
    "UnsuccessfulFleetRequests": []  
}
```

前のコマンドを以下のように変更することで、インスタンスを終了せずに、指定した スポットフリート リクエストをキャンセルできます。

```
aws ec2 cancel-spot-fleet-requests --spot-fleet-request-ids sfr-73fb2ce-  
aa30-494c-8788-1cee4EXAMPLE --no-terminate-instances
```

出力例を次に示します。

```
{  
    "SuccessfulFleetRequests": [  
        {  
            "SpotFleetRequestId": "sfr-73fb2ce-aa30-494c-8788-1cee4EXAMPLE",  
            "CurrentSpotFleetRequestState": "cancelled_running",  
            "PreviousSpotFleetRequestState": "active"  
        }  
    ],  
    "UnsuccessfulFleetRequests": []  
}
```

## スポットフリート 設定例

以下の例で示しているのは、スポットフリート リクエストを作成するための [request-spot-fleet](#) コマンドで使用できる起動設定です。詳細については、「[スポットフリート リクエストを作成する \(p. 350\)](#)」を参照してください。

### Note

スポットフリート では、ネットワークインターフェイス ID を起動仕様に指定できません。起動仕様から NetworkInterfaceID パラメータを必ず省略してください。

1. リージョンの最低価格のアベイラビリティーゾーンあるいはサブネットで スpotトインスタンス を起動する (p. 363)
2. 指定リストから最低価格のアベイラビリティーゾーンあるいはサブネットで スpotトインスタンス を起動する (p. 363)
3. 指定されたリストから最低価格のインスタンスタイプを使用して スpotトインスタンス を起動する (p. 365)

4. リクエストの料金を上書きする (p. 366)
5. 分散配分戦略を使用して スポットフリート を起動する (p. 367)
6. インスタンスの分量指定を使用して スポットフリート を起動する (p. 369)
7. オンデマンドキャバシティーで スポットフリート を起動する (p. 370)

**例 1: リージョンの最低価格のアベイラビリティーゾーンあるいはサブネットで スポットインスタンス を起動する**

以下の例では、アベイラビリティーゾーンまたはサブネットを使用しない 1 つの起動仕様を指定しています。スポットフリートはデフォルトのサブネットを持つ最低価格のアベイラビリティーゾーンでインスタンスを起動します。お支払いいただく料金はオンデマンド価格を上回ってしません。

```
{  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "KeyName": "my-key-pair",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "m3.medium",  
            "IamInstanceProfile": {  
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"  
            }  
        }  
    ]  
}
```

**例 2: 指定したリスト内で最低価格のアベイラビリティーゾーンまたはサブネットで スポットインスタンス を起動する**

以下の例では、アベイラビリティーゾーン/サブネットは異なるがインスタンスタイプおよび AMI は同じ、2 つの起動仕様を指定しています。

**アベイラビリティーゾーン**

スポットフリートは、指定した最低価格帯のアベイラビリティーゾーンのデフォルトサブネットにあるインスタンスを起動します。

```
{  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "KeyName": "my-key-pair",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "m3.medium",  
            "Placement": {  
                "AvailabilityZone": "us-west-2a, us-west-2b"  
            },  
        }  
    ]  
}
```

```
        "IamInstanceProfile": {
            "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
        }
    ]
}
```

## Subnets

デフォルトのサブネットまたはデフォルト以外のサブネットを指定できますが、デフォルト以外のサブネットは、デフォルトの VPC またはデフォルト以外の VPC 内から選択できます。スポットサービスは、最低価格のアベイラビリティゾーンにあるいずれかのサブネットでインスタンスを起動します。

1つのスポットフリート リクエストで、同じアベイラビリティゾーンから複数の異なるサブネットを指定することはできません。

```
{
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "KeyName": "my-key-pair",
            "SecurityGroups": [
                {
                    "GroupId": "sg-1a2b3c4d"
                }
            ],
            "InstanceType": "m3.medium",
            "SubnetId": "subnet-a61dafcf, subnet-65ea5f08",
            "IamInstanceProfile": {
                "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
            }
        }
    ]
}
```

インスタンスがデフォルトの VPC で起動される場合は、デフォルトでパブリック IPv4 アドレスが割り当てられます。インスタンスがデフォルト以外の VPC で起動される場合は、デフォルトでパブリック IPv4 アドレスは割り当てられません。起動仕様でネットワークインターフェイスを使用して、デフォルト以外の VPC で起動されるインスタンスにパブリック IPv4 アドレスを割り当てます。ネットワークインターフェイスの指定時、ネットワークインターフェイスを使用してサブネット ID とセキュリティグループ ID を含める必要があります。

```
...
{
    "ImageId": "ami-1a2b3c4d",
    "KeyName": "my-key-pair",
    "InstanceType": "m3.medium",
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "SubnetId": "subnet-1a2b3c4d",
            "Groups": [ "sg-1a2b3c4d" ],
            "AssociatePublicIpAddress": true
        }
    ],
    "IamInstanceProfile": {
        "Arn": "arn:aws:iam::880185128111:instance-profile/my-iam-role"
    }
}
```

...

### 例 3: 指定したリスト内で最低価格のインスタンスタイプを使用して スポットインスタンス を起動する

次の例では、同じ AMI と アベイラビリティゾーンまたはサブネットで、複数の異なるインスタンスタイプを使用する 2 つの起動設定を指定します。スポットフリートは、指定された最低価格のインスタンスタイプを使用してインスタンスを起動します。

#### アベイラビリティゾーン

```
{  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "cc2.8xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            }  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "r3.8xlarge",  
            "Placement": {  
                "AvailabilityZone": "us-west-2b"  
            }  
        }  
    ]  
}
```

#### サブネット

```
{  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "cc2.8xlarge",  
            "SubnetId": "subnet-1a2b3c4d"  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "SecurityGroups": [  
                {  
                    "GroupId": "sg-1a2b3c4d"  
                }  
            ],  
            "InstanceType": "r3.8xlarge",  
            "SubnetId": "subnet-1a2b3c4d"  
        }  
    ]  
}
```

```
        "GroupId": "sg-1a2b3c4d"
    }
],
"InstanceType": "r3.8xlarge",
"SubnetId": "subnet-1a2b3c4d"
}
}
```

#### 例 4. リクエストの料金を上書きする

オンデマンド価格であるデフォルトの上限料金を使用することをお勧めします。必要に応じて、フリートリクエストの上限料金と個々の起動条件の上限料金を指定することができます。

以下の例は、フリートリクエストの上限料金と、3 つの起動条件のうちの 2 つの上限料金を指定しています。フリートリクエストの上限料金は、上限料金を指定しないすべての起動条件に適用されます。スポットフリートは、最低価格のインスタンスタイプを使用してインスタンスを起動します。

#### アベイラビリティーゾーン

```
{
    "SpotPrice": "1.00",
    "TargetCapacity": 30,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "SpotPrice": "0.10"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.4xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "SpotPrice": "0.20"
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.8xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

#### サブネット

```
{
    "SpotPrice": "1.00",
    "TargetCapacity": 30,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.2xlarge",
            "SubnetId": "subnet-1a2b3c4d",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

```
        "SpotPrice": "0.10"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.4xlarge",
        "SubnetId": "subnet-1a2b3c4d",
        "SpotPrice": "0.20"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c3.8xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    }
]
```

#### 例 5: 分散配分戦略を使用して スポットフリート を起動する

次の例では、`diversified` の配分戦略を使用します。これらの起動仕様では、インスタンスタイプは異なりますが、AMI およびアベイラビリティーゾーン/サブネットは同じです。スポットフリートは、3 つの起動条件について 30 個のインスタンスを分配します。これで、タイプごとに 10 個のインスタンスが配分されます。詳細については、「[スポットインスタンス の配分戦略 \(p. 327\)](#)」を参照してください。

##### アベイラビリティーゾーン

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        }
    ]
}
```

##### サブネット

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
```

```
"LaunchSpecifications": [
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c4.2xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "m3.2xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "r3.2xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    }
]
```

アベイラビリティーゾーンの 1 つで機能停止が発生した場合にスポットリクエストが EC2 のキャパシティーによって満たされる可能性を高めるためのベストプラクティスは、ゾーン間で多様化することです。このシナリオでは、使用可能な各アベイラビリティーゾーンを起動仕様に含めます。また、毎回同じサブネットを使用するのではなく、3 つの固有のサブネット（それぞれ異なるゾーンへのマッピング）を使用してください。

### アベイラビリティーゾーン

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c4.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2a"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "m3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            }
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2c"
            }
        }
    ]
}
```

### サブネット

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 30,
    "AllocationStrategy": "diversified",
```

```
"IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
"LaunchSpecifications": [
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "c4.2xlarge",
        "SubnetId": "subnet-1a2b3c4d"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "m3.2xlarge",
        "SubnetId": "subnet-2a2b3c4d"
    },
    {
        "ImageId": "ami-1a2b3c4d",
        "InstanceType": "r3.2xlarge",
        "SubnetId": "subnet-3a2b3c4d"
    }
]
```

#### 例 6: インスタンスの重み付けを使用して スポットフリート を起動する

次の例では、インスタンス分量指定を使っています。これは、料金が 1 インスタンス時間当たりではなく、1 ユニット時間当たりであることを意味します。それぞれの起動設定には、異なるインスタンスタイプおよび異なる分量がリストされます。スポットフリートは 1 ユニット時間当たり最低価格のインスタンスタイプを選択します。スポットフリートでは、ターゲット容量をインスタンス分量で割ることで起動するスポットインスタンス の数を計算します。その結果が整数でなければ、スポットフリートはその数を次の整数に切り上げ、これによりフリートのサイズがターゲット容量以上になります。

r3.2xlarge のリクエストが成功すると、スポットはこれらのインスタンスのうち、4 つをプロビジョニングします。3.33 インスタンスまで 20 を 6 で割り、そして残りの 4 つのインスタンスを切り上げます。

c3.xlarge のリクエストが成功すると、スポットはこれらのインスタンスのうち、7 つをプロビジョニングします。6.66 インスタンスまで 20 を 3 で割り、そして残りの 7 つのインスタンスを切り上げます。

詳細については、「[スポットフリート インスタンスの分量指定 \(p. 329\)](#)」を参照してください。

#### アベイラビリティーゾーン

```
{
    "SpotPrice": "0.70",
    "TargetCapacity": 20,
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",
    "LaunchSpecifications": [
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "r3.2xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "WeightedCapacity": 6
        },
        {
            "ImageId": "ami-1a2b3c4d",
            "InstanceType": "c3.xlarge",
            "Placement": {
                "AvailabilityZone": "us-west-2b"
            },
            "WeightedCapacity": 3
        }
    ]
}
```

## サブネット

```
{  
    "SpotPrice": "0.70",  
    "TargetCapacity": 20,  
    "IamFleetRole": "arn:aws:iam::123456789012:role/aws-ec2-spot-fleet-tagging-role",  
    "LaunchSpecifications": [  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "r3.2xlarge",  
            "SubnetId": "subnet-1a2b3c4d",  
            "WeightedCapacity": 6  
        },  
        {  
            "ImageId": "ami-1a2b3c4d",  
            "InstanceType": "c3.xlarge",  
            "SubnetId": "subnet-1a2b3c4d",  
            "WeightedCapacity": 3  
        }  
    ]  
}
```

## 例 7: オンデマンドキャパシティーで スポットフリート を起動する

インスタンスのキャパシティーを常に確保するには、オンデマンドキャパシティーのリクエストを スポットフリート リクエストに含めることができます。オンデマンドリクエストは、容量がある限り、常に実行されます。ターゲットキャパシティーは、キャパシティーと可用性がある場合にスポットとして実行されます。

次の例では、希望するターゲットキャパシティーを 10 とし、そのうち 5 をオンデマンドキャパシティーとして指定する必要があります。スポットキャパシティーは指定しません。これは、ターゲットキャパシティーからオンデマンドキャパシティーを引いたバランスとなります。Amazon EC2 は、利用可能な Amazon EC2 容量と可用性がある場合、オンデマンドとして 5 つの容量単位を、5 つの容量単位 ( $10 - 5 = 5$ ) をスポットとして起動します。

詳細については、「[スポットフリート でのオンデマンド \(p. 326\)](#)」を参照してください。

```
{  
    "IamFleetRole": "arn:aws:iam::781603563322:role/aws-ec2-spot-fleet-tagging-role",  
    "AllocationStrategy": "lowestPrice",  
    "TargetCapacity": 10,  
    "SpotPrice": null,  
    "ValidFrom": "2018-04-04T15:58:13Z",  
    "ValidUntil": "2019-04-04T15:58:13Z",  
    "TerminateInstancesWithExpiration": true,  
    "LaunchSpecifications": [],  
    "Type": "maintain",  
    "OnDemandTargetCapacity": 5,  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-0dbb04d4a6cca5ad1",  
                "Version": "2"  
            },  
            "Overrides": [  
                {  
                    "InstanceType": "t2.medium",  
                    "WeightedCapacity": 1,  
                    "SubnetId": "subnet-d0dc51fb"  
                }  
            ]  
        }  
    ]  
}
```

}

## スポットフリート の CloudWatch メトリクス

Amazon EC2 は、スポットフリート をモニタリングするために使用できる Amazon CloudWatch メトリクスを提供します。

### Important

正確性を確実にするため、これらのメトリクスを使用する際は詳細モニタリングを有効にするごとをお勧めします。詳細については、「[インスタンスの詳細モニタリングの有効化または無効化 \(p. 642\)](#)」を参照してください。

Amazon EC2 によって提供される CloudWatch メトリクスの詳細については、「[CloudWatch を使用したインスタンスのモニタリング \(p. 642\)](#)」を参照してください。

## スポットフリート のメトリクス

AWS/EC2Spot 名前空間には、次のメトリクスに加えて、スポット群のスポットインスタンス 用の CloudWatch メトリクスが含まれます。詳細については、「[インスタンスマトリクス \(p. 644\)](#)」を参照してください。

AWS/EC2Spot 名前空間には、次のメトリクスが含まれます。

メトリクス	説明
AvailableInstancePoolsCount	スポットフリート のリクエストで指定される スpotトインスタンス のプール。 単位: カウント
BidsSubmittedForCapacity	Amazon EC2 が スpotトフリート リクエストを送信した容量。 単位: カウント
EligibleInstancePoolCount	スポットフリート のリクエストで指定され、Amazon EC2 がリクエストを落札できる スpotトインスタンス のプール。Amazon EC2 は、スspotトインスタンス に支払う最大価格がスspotト価格よりも小さいか、スspotト価格が オンデマンドインスタンス の価格よりも大きいプールではリクエストを処理しません。 単位: カウント
FulfilledCapacity	Amazon EC2 が落札した容量。 単位: カウント
MaxPercentCapacityAllocation	スポットフリート のリクエストで指定されたすべての スpotトフリート のプールでの PercentCapacityAllocation の最大値。 単位: パーセント
PendingCapacity	TargetCapacity と FulfilledCapacity の違い。 単位: カウント
PercentCapacityAllocation	指定されたディメンションの スpotトインスタンス のプールに割り当てられた容量。すべての スpotト

メトリクス	説明
	インスタンスのプールにわたる最大値を取得するには、 <code>MaxPercentCapacityAllocation</code> を使用します。 単位: パーセント
<code>TargetCapacity</code>	スポットフリートのリクエストのターゲット容量。 単位: カウント
<code>TerminatingCapacity</code>	プロビジョニングされた容量が目標の容量より大きいために終了した容量。 単位: カウント

メトリクスの測定単位が `Count` である場合、最も有用な統計は `Average` です。

## スポットフリート ディメンション

スポットフリートのデータをフィルタリングするには、次のディメンションを使用します。

ディメンション	説明
<code>AvailabilityZone</code>	アベイラビリティーゾーン別にデータをフィルタリングします。
<code>FleetRequestId</code>	スポットフリートのリクエスト別にデータをフィルタリングします。
<code>InstanceType</code>	インスタンスタイプ別にデータをフィルタリングします。

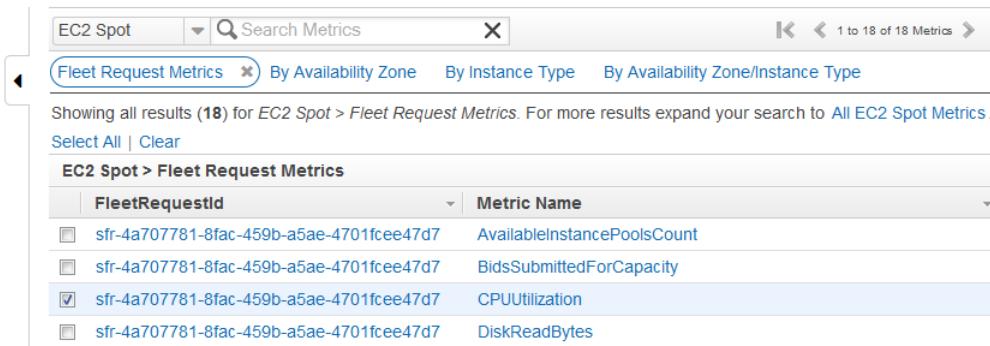
## スポットフリートの CloudWatch メトリクスを表示する

Amazon CloudWatch コンソールを使用して、スポットフリートの CloudWatch メトリクスを表示できます。これらのメトリクスは、モニタリング用のグラフのように表示されます。これらのグラフでは、スポットフリートがアクティブの場合にデータポイントが表示されます。

メトリクスはまず名前空間ごとにグループ化され、次に各名前空間内の種々のディメンションの組み合わせごとにグループ化されます。たとえば、すべての スポットフリートのメトリクスを表示するか、スポットフリートリクエスト ID、インスタンスタイプ、またはアベイラビリティーゾーン別にスポットフリートのメトリクスグループを表示できます。

### スポットフリート メトリクスを表示するには

1. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
2. ナビゲーションペインの [Metrics] で、[EC2 Spot] 名前空間を選択します。
3. (オプション) ディメンション別にメトリクスをフィルタするには、次のいずれかを選択します。
  - [Fleet Request Metrics] — スポットフリートリクエスト別にグループ化
  - [By Availability Zone] — スポットフリートリクエストおよびアベイラビリティーゾーン別にグループ化
  - [By Instance Type] — スポットフリートリクエストおよびインスタンスタイプ別にグループ化
  - [アベイラビリティーゾーン/インスタンスタイプ] — スポットフリートリクエスト、アベイラビリティーゾーン、およびインスタンスタイプ別にグループ化
4. メトリクスのデータを表示するには、メトリクスの横にあるチェックボックスをオンにします。



## スポットフリート のオートスケーリング

自動スケーリングは、需要に応じて スポットフリート のターゲット容量を自動的に増減する機能です。スポットフリート は、1 つ以上のスケーリングポリシーに応答して、選択する範囲内でインスタンスを起動(スケールアウト) するか、インスタンスを削除(スケールイン) できます。

スポットフリート は、以下のタイプの自動スケーリングをサポートします。

- [Target tracking scaling \(p. 375\)](#) – 特定のメトリクスのターゲット値に基づいて、フリートの現在の容量を増減させます。これはサーモスタッフで家の温度を管理する方法と似ています(温度を選択すれば、後はサーモスタッフがすべてを実行する)。
- [Step scaling \(p. 376\)](#) – アラーム超過のサイズに応じて変動する一連のスケーリング調整値(ステップ調整値)に基づいて、グループの現在の容量を増減させます。
- [Scheduled scaling \(p. 378\)](#) – 日付と時刻に基づいて、フリート現在の容量を増減させます。

インスタンスの重み付け (p. 329)を使用している場合は、必要に応じて スポットフリート ターゲット容量を超える場合があることに注意してください。具体的には、取得済み容量が浮動小数点数となってもターゲット容量は整数でなければならないために、スポットフリート はその数を次の整数に切り上げます。アラームがトリガーされたときにスケーリングポリシーの結果を確認する際は、このような動作を考慮に入れる必要があります。たとえば、ターゲット容量が 30、取得済み容量が 30.1 で、スケーリングポリシーが 1 を減算するとします。アラームがトリガーされると、自動スケーリングプロセスは 30.1 から 1 を減算して 29.1 を得るため、この数は 30 に切り上げられることになり、スケーリングアクションは実行されません。別の例として、選択したインスタンスの重みが 2、4、8 であり、ターゲット容量が 10 であるとします。重み 2 のインスタンスが利用できなかつたために、スポットフリート は重み 4 と 8 のインスタンスをプロビジョニングして取得済みの容量が 12 になったとします。スケーリングポリシーがターゲット容量を 20% 減らしてアラームがトリガーされた場合、自動スケーリングプロセスは 12 から  $12 \times 0.2$  を減算して 9.6 を得るため、この数は 10 に切り上げられることになり、スケーリングアクションは実行されません。

スポットフリート 用に作成したスケーリングポリシーはクールダウン期間をサポートしています。クールダウン期間は、以前のトリガー関連のスケーリングアクティビティが以後のスケーリングイベントに影響を及ぼすことができる期限であり、スケーリングアクティビティが終了した時点からの秒数として指定します。スケールアウトポリシーにクールダウン期間を設定すると、その期間中にクールダウンを開始したスケールアウトイベントによって追加された容量は、次のスケールアウトに予定される容量の一部として繰り入れられます。これにより、スケールアウトが継続的に(ただし過剰になることなく)行われます。スケールインポリシーにクールダウン期間を設定すると、その期間が過ぎるまでは以後のスケールインリクエストがブロックされます。これにより、スケールインが抑制されてアプリケーションの可用性が確保されます。ただし、スケールイン後のクールダウン期間中に別のアラームによってスケールアウトポリシーがトリガーされると、自動スケーリングによってスケーラブルなターゲットが即座にスケールアウトされます。

使用率の変化に迅速に対応できるように、1 分間隔でインスタンスのメトリクスをスケーリングすることをお勧めします。5 分間隔でメトリクスをスケールすると、応答時間が低速になり、古いメトリクスデータ

タに基づいてスケールすることになる可能性があります。1分ごとにインスタンスのメトリクスデータを CloudWatch に送信するには、インスタンスで詳細モニタリングを有効にできます。詳細については、「[インスタンスの詳細モニタリングの有効化または無効化 \(p. 642\)](#)」および「[定義済みパラメータを使用してスポットフリートを作成する \(コンソール\) \(p. 351\)](#)」を参照してください。

スポットフリートのスケーリングの設定の詳細については、次のリソースを参照してください。

- AWS CLI Command Reference の [application-autoscaling](#) セクション
- Application Auto Scaling API リファレンス
- Application Auto Scaling ユーザーガイド

## スポットフリート Auto Scaling に必要な IAM のアクセス権限

スポットフリートの Auto Scaling は Amazon EC2、Amazon CloudWatch、および Application Auto Scaling API と組み合わせることで機能します。スポットフリート リクエストは Amazon EC2 で作成され、アラームは CloudWatch で作成され、スケーリングポリシーは Application Auto Scaling で作成されます。

スポットフリート および Amazon EC2 への IAM アクセス許可 (p. 347) に加えて、フリートのスケーリング設定にアクセスする IAM ユーザーには、動的スケーリングをサポートするサービスへの適切なアクセス許可も必要です。IAM ユーザーには、次のポリシー例に示されているアクションを使用するためのアクセス許可が必要です。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "application-autoscaling:*",  
                "ec2:DescribeSpotFleetRequests",  
                "ec2:ModifySpotFleetRequest",  
                "cloudwatch:DeleteAlarms",  
                "cloudwatch:DescribeAlarmHistory",  
                "cloudwatch:DescribeAlarms",  
                "cloudwatch:DescribeAlarmsForMetric",  
                "cloudwatch:GetMetricStatistics",  
                "cloudwatch>ListMetrics",  
                "cloudwatch:PutMetricAlarm",  
                "cloudwatch:DisableAlarmActions",  
                "cloudwatch:EnableAlarmActions",  
                "iam>CreateServiceLinkedRole",  
                "sns>CreateTopic",  
                "sns:Subscribe",  
                "sns:Get*",  
                "sns>List*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

独自の IAM ポリシーを作成し、Application Auto Scaling API を呼び出すためのよりきめ細かなアクセス許可を付与することもできます。詳細については、Application Auto Scaling ユーザーガイドの「[認証とアクセスコントロール](#)」を参照してください。

Application Auto Scaling サービスには、スポットフリートと CloudWatch のアラームを記述するためのアクセス許可と、ユーザーに代わってスポットフリートのターゲット容量を変更するためのアクセス許可も必要です。スポットフリートの自動スケーリングを有効にすると、サービスにリンクされた

ロールが `AWSServiceRoleForApplicationAutoScaling_EC2SpotFleetRequest` という名前で作成されます。このサービスにリンクされたロールは、Application Auto Scaling に対して、ポリシーのアラームの記述、フリートの現容量のモニタリング、およびフリートの容量の変更を行うためのアクセス許可を付与します。Application Auto Scaling の元のマネージド型のスポットフリートロールは `aws-ec2-spot-fleet-autoscale-role` ですが、これは不要になりました。サービスにリンクされたロールは、Application Auto Scaling のデフォルトロールです。詳細については、Application Auto Scaling ユーザーガイドの「[サービスにリンクされたロール](#)」を参照してください。

## ターゲット追跡ポリシーを使用した スpot フリート のスケーリング

ターゲット追跡スケーリングポリシーでは、メトリクスを選択してターゲット値を設定します。スポットフリートはスケーリングポリシーをトリガーする CloudWatch アラームを作成および管理し、メトリクスとターゲット値に基づいてスケーリング調整値を計算します。スケーリングポリシーは、指定されたターゲット値、またはそれに近い値にメトリクスを維持するため、必要に応じて容量を追加または削除します。ターゲットの追跡スケーリングポリシーは、メトリクスをターゲット値近くに維持することに加えて、負荷パターンの変動によるメトリクスの変動に合わせて調整し、フリートの容量の急速な変動を最小化します。

それぞれが異なるメトリクスを使用していれば、スポットフリートに対して複数のターゲットの追跡スケーリングポリシーを設定できます。フリートは、最大のフリート容量を提供する方針に基づいてスケーリングされます。これにより、複数のシナリオに対応して、アプリケーションワークロードを処理するのに十分な容量が常に確保されます。

アプリケーションの可用性を高めるために、フリートのスケールアウトはメトリクスに比例して可能な限り高速に行われますが、スケールインはより緩やかです。

ターゲット容量が減ったためにスポットフリートによってインスタンスが削除される場合、インスタンスはスポットインスタンスの中断通知を受け取ります。

ターゲットの追跡スケーリングポリシー用にスポットフリートが管理する CloudWatch アラームを編集または削除しないでください。ターゲット追跡スケーリングポリシーを削除すると、スポットフリートが自動的にアラームを削除します。

### 制限

- スpot フリート リクエストには、タイプが `maintain` のリクエストが必要です。自動スケーリングは 1 回限りのリクエストまたはスポットブロックではサポートされません。

### ターゲットの追跡スケーリングポリシーを設定するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。
3. スpot フリート リクエストを選択し、[Auto Scaling] を選択します。
4. 自動スケーリングが設定されていない場合は、[Configure] を選択します。
5. スpot フリートの最小容量および最大容量を設定するには、[Scale capacity between] を使用します。自動スケーリングにより、最小容量以下または最大容量以上にスポットフリートがスケールされることはありません。
6. [Policy name] に、ポリシーの名前を入力します。
7. [Target metric] を選択します。
8. メトリクスの [Target value] を入力します。
9. (オプション) [クールダウン期間] を設定して、デフォルトのクールダウン期間を変更します。
10. (オプション) 現在の構成に基づいてスケールインポリシーの作成を省略するには、[スケールインの無効化] を選択します。別の構成を使用してスケールインポリシーを作成できます。
11. [Save (保存)] を選択します。

AWS CLI を使用して、ターゲットの追跡スケーリングポリシーを設定します。

1. [register-scalable-target](#) コマンドを使用して、スケーラブルなターゲットとして スポットフリート リクエストを登録します。
2. [put-scaling-policy](#) コマンドを使用してスケーリングポリシーを作成します。

## ステップスケーリングポリシーを使用した スポットフリート のスケーリング

ステップスケーリングポリシーでは、CloudWatch アラームを指定してスケーリングプロセスをトリガります。たとえば、CPU 利用率が一定のレベルに達したときにスケールアウトする場合、Amazon EC2 によって提供される CPUUtilization メトリクスを使用してアラームを作成します。

ステップスケーリングポリシーを作成したら、次のいずれかのスケーリング調整タイプを指定する必要があります。

- [Add] – 指定した数の容量ユニットまたは指定した割合の現在の容量で、スポット群のターゲット容量を増やします。
- [Remove] – 指定した数の容量ユニットまたは指定した割合の現在の容量で、スポット群のターゲット容量を減らします。
- [設定] – スpot群のターゲット容量を、指定した数の容量ユニットに設定します。

アラームがトリガーされると、自動スケーリングプロセスは、取得済み容量およびスケーリングポリシーを使用して新しいターゲット容量を計算し、必要に応じてターゲット容量を更新します。たとえば、ターゲット容量と取得済み容量がそれぞれ 10 で、スケーリングポリシーが 1 を加算するとします。アラームがトリガーされると、自動スケーリングプロセスは 10 に 1 を加えて 11 を得るため、スポットフリートは 1 つのインスタンスを起動します。

ターゲット容量が減ったために スpotフリート によってインスタンスが削除される場合、インスタンスは スpotインスタンス の中断通知を受け取ります。

### 制限

- スpotフリート リクエストには、タイプが `maintain` のリクエストが必要です。自動スケーリングは 1 回限りのリクエストまたはスspotブロックではサポートされません。

### 前提条件

- アプリケーションにとって重要な CloudWatch メトリクスを検討します。AWS または独自のカスタムメトリクスによって提供されるメトリクスに基づいて CloudWatch アラームを作成できます。
- スケーリングポリシーで使用する AWS メトリクスについて、メトリクスを提供するサービスでデフォルトで有効にならない場合、CloudWatch メトリクスの収集を有効にします。

### CloudWatch アラームを作成するには

1. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
2. ナビゲーションペインで、[Alarms] を選択します。
3. [アラームの作成] を選択します。
4. [Specify metric and conditions (メトリクスと条件を指定)] ページで、[メトリクスの選択] を選択します。
5. [EC2 スpot]、[フリートリクエストのメトリクス] の順に選択し、メトリクス (CPUUtilization など) を選択して [メトリクスの選択] を選択します。

[Specify metric and conditions (メトリクスと条件の指定)] ページに、選択したメトリクスに関するグラフや他の情報が表示されます。

- [期間] でアラームの評価期間 (1 分など) を選択します。アラームを評価する場合、各期間は 1 つのデータポイントに集約されます。

Note

期間が短いほど、作成されるアラームの感度が高くなります。

- [条件] で、しきい値条件を定義してアラームを定義します。たとえば、メトリクスの値が 80% 以上になるたびにアラームをトリガーするように、しきい値を定義できます。
- [Additional configuration (追加設定)] の [Datapoints to alarm (アラームするデータポイント)] で、アラームをトリガーするために ALARM 状態になる必要があるデータポイント (評価期間) の数を指定します (3 個の評価期間のうち 1 個または 2 個の評価期間など)。これでアラームが作成されます。このアラームは、指定した数の期間で連続してしきい値を超過すると、ALARM 状態に移行します。詳細については、『Amazon CloudWatch ユーザーガイド』の「[アラームを評価する](#)」を参照してください。
- [Missing data treatment (不足しているデータの扱い)] で、いずれかのオプションを選択します (または、デフォルトの [Treat missing data as missing (不足しているデータを不足として扱う)] のままにします)。詳細については、『Amazon CloudWatch ユーザーガイド』の「[CloudWatch アラームが欠落データを処理する方法の設定](#)」を参照してください。
- [Next] を選択します。
- (オプション) スケーリングイベントの通知を受け取る場合は、[通知] で、通知を受け取るために使用する Amazon SNS トピックを選択または作成できます。それ以外の場合は、通知を削除し、必要に応じて後で追加できます。
- [Next] を選択します。
- [Add a description (説明の追加)] にアラームの名前と説明を入力し、[次へ] を選択します。
- [アラームの作成] を選択します。

スポットフリート のステップスケーリングポリシーを設定するには (コンソール)

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Spot Requests] を選択します。
- スポットフリート リクエストを選択し、[Auto Scaling] を選択します。
- 自動スケーリングが設定されていない場合は、[Configure] を選択します。
- スポットフリートの最小容量および最大容量を設定するには、[Scale capacity between] を使用します。自動スケーリングにより、最小容量以下または最大容量以上にスポットフリートがスケールされることはありません。
- 初期状態では、[Scaling policies] には ScaleUp と ScaleDown という名前のポリシーが含まれています。これらのポリシーは、完了するか、[Remove policy] を選択して削除できます。[Add policy] を選択することもできます。
- ポリシーを定義するには、以下の作業を行います。
  - [Policy name] に、ポリシーの名前を入力します。
  - [ポリシートリガー] で、既存のアラームを選択するか、[新しいアラームの作成] を選択して Amazon CloudWatch コンソールを開き、アラームを作成します。
  - [Modify capacity] でスケーリングの調整タイプ、数、単位を選択します。
  - (オプション) ステップスケーリングを実行するには、[Define steps] を選択します。デフォルトでは、追加ポリシーには負の無限の下限値とアラームしきい値の上限値があります。デフォルトでは、削除ポリシーにはアラームしきい値の下限値と正の無限大の上限値があります。別のステップを追加するには、[Add step] を選択します。
  - (オプション) クールダウン期間のデフォルト値を変更するには、[Cooldown period] から数値を選択します。
- [Save (保存)] を選択します。

AWS CLI を使用して スポットフリート のステップスケーリングポリシーを設定するには

1. [register-scalable-target](#) コマンドを使用して、スケーラブルなターゲットとして スポットフリート リクエストを登録します。
2. [put-scaling-policy](#) コマンドを使用して スケーリングポリシーを作成します。
3. [put-metric-alarm](#) コマンドを使用して スケーリングポリシーをトリガーするアラームを作成します。

## スケーリングのスケジュールを使用した スポットフリート のスケール

スケジュールに基づくスケーリングにより、予想可能な需要の変化に応じてアプリケーションを拡張することができます。スケジュールに基づくスケーリングを使用するには、スケジュールされたアクションを作成します。それにより、指定された時間に規模の拡大や縮小を行うように スポットフリート に伝えます。スケジュールされたアクションを作成する際、スポットフリート のスケーリングアクティビティーが起こる時刻、最小容量、最大容量を指定できます。スケジュールされたアクションは1回だけ、または反復して行われるように作成できます。

### 制限

- スpotフリート リクエストには、タイプが `maintain` のリクエストが必要です。自動スケーリングは 1 回限りのリクエストまたはスポットブロックではサポートされません。

### 1 回のアクションを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。
3. スpotフリート リクエストを選択し、[スケジュールに基づくスケーリング] を選択します。
4. [予定アクションの作成] を選択します。
5. [名前] に、予定アクションの名前を指定します。
6. [最小容量]、[最大容量]、または両方の値を入力します。
7. [繰り返し] で、[1 回] を選択します。
8. (オプション) [開始時刻]、[終了時刻]、またはその両方の日付と時刻を選択します。
9. [Submit] を選択します。

### 定期的なスケジュールでスケールするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。
3. スpotフリート リクエストを選択し、[スケジュールに基づくスケーリング] を選択します。
4. [繰り返し] で、事前定義済みのスケジュール (たとえば、[毎日]) のいずれかを選択するか、[カスタム] を選択して cron 式を入力します。スケジュールに基づくスケーリングがサポートする cron 式の詳細については、『Amazon CloudWatch Events ユーザーガイド』の「[cron 式](#)」を参照してください。
5. (オプション) [開始時刻]、[終了時刻]、またはその両方の日付と時刻を選択します。
6. [Submit] を選択します。

### スケジュールされたアクションを編集するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択します。
3. スpotフリート リクエストを選択し、[スケジュールに基づくスケーリング] を選択します。
4. スケジュールされたアクション を選択して、[Actions]、[Edit] の順に選択します。

- 必要な変更を加えて、[Submit] を選択します。

スケジュールされたアクションを削除するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Spot Requests] を選択します。
- スポットフリート リクエストを選択し、[スケジュールに基づくスケーリング] を選択します。
- スケジュールされたアクションを選択して、[アクション]、[削除] の順に選択します。
- 確認を求めるメッセージが表示されたら、[削除] を選択します。

AWS CLI を使用してスケジュールされたスケーリングを管理するには

次のコマンドを使用します。

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

## スポットリクエストステータス

スポットインスタンス リクエストを追跡し、スポットインスタンス の使用計画を策定するには、Amazon EC2 によって提供されるリクエストステータスを使用します。たとえば、リクエストステータスによって、スポットリクエストがまだ受理されていない理由や、スポットリクエストの受理を妨げている制約の一覧を確認できます。

このプロセスの各ステップ（スポットリクエストのライフサイクルとも呼ばれる）では、特定のイベントによって後続のリクエスト状態が決まります。

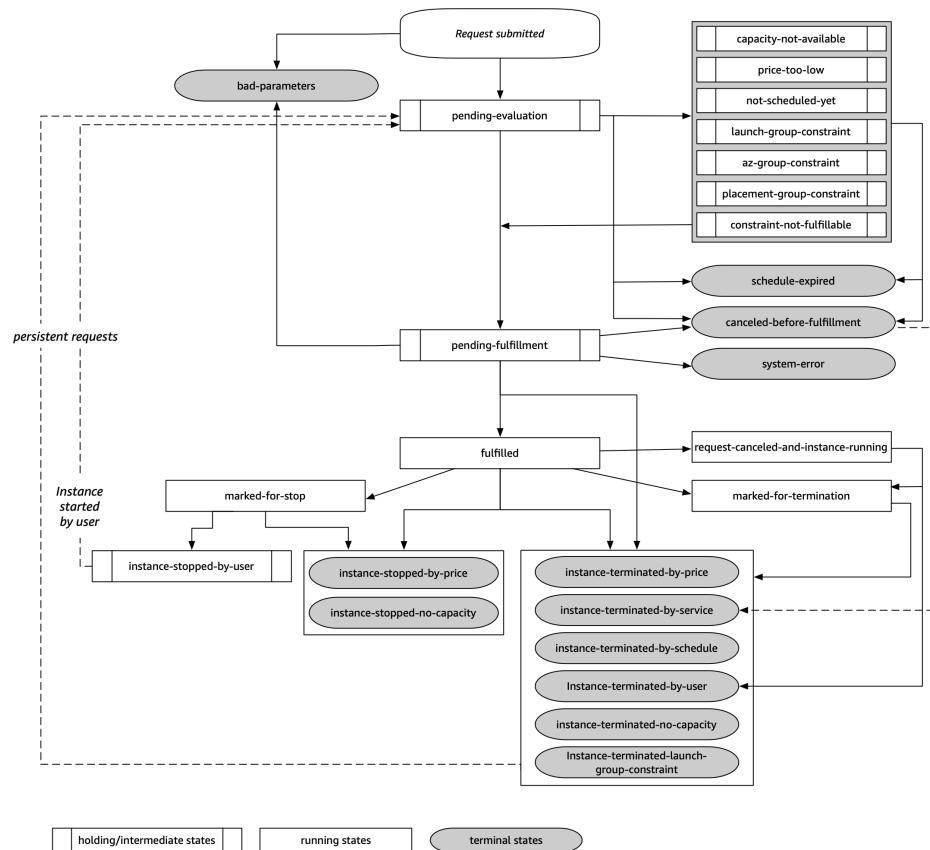
### 目次

- [スポットリクエストのライフサイクル \(p. 379\)](#)
- [リクエストステータス情報の取得 \(p. 383\)](#)
- [スポットリクエストコード \(p. 383\)](#)

## スポットリクエストのライフサイクル

次の図は、申請から終了まで、スポットリクエストがライフサイクル全体を通してたどり得る経路を示しています。各ステップはノードとして表現され、各ノードのステータスコードはスポットリクエストおよびスポットインスタンスのステータスを示します。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
スポットインスタンス



### 評価保留

スポットインスタンスリクエストを作成すると、1つ以上のリクエストパラメータが有効ではない場合 (**pending-evaluation**) を除き、リクエストは **bad-parameters** 状態になります。

ステータスコード	リクエストの状態	インスタンスの状態
<b>pending-evaluation</b>	<b>open</b>	該当なし
<b>bad-parameters</b>	<b>closed</b>	該当なし

### 保持

1つ以上のリクエストによる制約が有効であるが、まだ満足することができない場合や、容量が十分ではない場合、リクエストは制約が満たされるまで待機する保持状態になります。リクエストのオプションは、リクエストが受理される可能性に影響します。たとえば、現在のスポット料金の下で上限料金を指定する場合、リクエストはスポット料金が上限料金を下回るまで保持状態になります。アベイラビリティーゾーングループを指定する場合、アベイラビリティーゾーンの制約が満たされるまで、リクエストは保持状態になります。

いずれかのアベイラビリティーゾーンが停止した場合、他のアベイラビリティーゾーンの スポットインスタンス 要求に使用可能な予備の EC2 容量が影響を受ける可能性があります。

ステータスコード	リクエストの状態	インスタンスの状態
<b>capacity-not-available</b>	<b>open</b>	該当なし

ステータスコード	リクエストの状態	インスタンスの状態
price-too-low	open	該当なし
not-scheduled-yet	open	該当なし
launch-group-constraint	open	該当なし
az-group-constraint	open	該当なし
placement-group-constraint	open	該当なし
constraint-not-fulfillable	open	該当なし

#### 評価保留/受理終了

特定の期間のみ有効なリクエストを作成し、リクエストが受理保留段階に到達する前にこの期間の期限が切れた場合、スポットインスタンス リクエストは terminal 状態になることがあります。これは、お客様がリクエストをキャンセルした場合、またはシステムエラーが発生した場合にも発生する場合があります。

ステータスコード	リクエストの状態	インスタンスの状態
schedule-expired	cancelled	該当なし
canceled-before-fulfillment*	cancelled	該当なし
bad-parameters	failed	該当なし
system-error	closed	該当なし

\* リクエストをキャンセルする場合。

#### 受理保留

指定した制約条件(存在する場合)が満たされ、上限料金が現在のスポット料金以上である場合、スポットリクエストは pending-fulfillment 状態になります。

この時点で、Amazon EC2 は要求されたインスタンスを提供するよう準備します。この時点でプロセスが停止した場合、これは スpotトインスタンス が起動される前にユーザーがリクエストをキャンセルしたことによる可能性があります。または、予期しないシステムエラーが発生したことが原因である可能性もあります。

ステータスコード	リクエストの状態	インスタンスの状態
pending-fulfillment	open	該当なし

#### 受理済み

スポットインスタンスに対するすべての指定が満たされると、スポットリクエストが受理されます。Amazon EC2 によって スpotトインスタンス が起動されます。起動には数分かかる場合があります。中断時に スpotトインスタンス が休止または停止した場合、リクエストが再度受理されるかキャンセルできるまで、この状態のままになります。

ステータスコード	リクエストの状態	インスタンスの状態
fulfilled	active	pending → running
fulfilled	active	stopped → running

スポットインスタンスを停止すると、スポットインスタンスを再起動できるようになるか、リクエストがキャンセルされるまで、スポットリクエストは `marked-for-stop` または `instance-stopped-by-user` 状態になります。

ステータスコード	リクエストの状態	インスタンスの状態
<code>marked-for-stop</code>	active	stopping
<code>instance-stopped-by-user</code> *	<code>disabled</code> または <code>cancelled</code> **	stopped

\* インスタンスを停止するか、インスタンスからシャットダウンコマンドを実行すると、スポットインスタンスが `instance-stopped-by-user` 状態になります。インスタンスを停止した後は、インスタンスを再起動できるようになります。再起動時に、スポットインスタンスリクエストは `pending-evaluation` 状態に戻り、制約が満たされると Amazon EC2 によって新しいスポットインスタンスが起動されます。

\*\* スpotトインスタンスを停止したが、リクエストをキャンセルしていない場合、スポットリクエストの状態は `disabled` になります。スポットインスタンスが停止しており、リクエストの有効期限が切れている場合、リクエストの状態は `cancelled` になります。

#### 受理済み終了

上限価格がスポット料金以上であり、インスタンスタイプで使用できる容量があり、お客様がインスタンスを終了しない限り、スポットインスタンスの実行は続行されます。スポット料金や利用可能な容量の変化により、Amazon EC2 でスポットインスタンスを終了する必要がある場合、スポットリクエストは終了状態になります。リクエストは、お客様がスポットリクエストをキャンセルした場合や、スポットインスタンスを終了した場合も、終了状態になります。

ステータスコード	リクエストの状態	インスタンスの状態
<code>request-canceled-and-instance-running</code>	<code>cancelled</code>	<code>running</code>
<code>marked-for-stop</code>	<code>active</code>	<code>running</code>
<code>marked-for-termination</code>	<code>closed</code>	<code>running</code>
<code>instance-stopped-by-price</code>	<code>disabled</code>	<code>stopped</code>
<code>instance-stopped-by-user</code>	<code>disabled</code>	<code>stopped</code>
<code>instance-stopped-no-capacity</code>	<code>disabled</code>	<code>stopped</code>
<code>instance-terminated-by-price</code>	<code>closed</code> (ワンタイム)、 <code>open</code> (永続)	<code>terminated</code>
<code>instance-terminated-by-schedule</code>	<code>closed</code>	<code>terminated</code>

ステータスコード	リクエストの状態	インスタンスの状態
instance-terminated-by-service	cancelled	terminated
instance-terminated-by-user	closed または cancelled *	terminated
instance-terminated-no-capacity	closed (ワンタイム)、open (永続)	terminated
instance-terminated-launch-group-constraint	closed (ワンタイム)、open (永続)	terminated

\* インスタンスを終了したが、リクエストをキャンセルしていない場合、リクエストの状態は `closed` になります。インスタンスを終了し、リクエストをキャンセルする場合、リクエストの状態は `cancelled` になります。スポットリクエストをキャンセルする前にスポットインスタンスを終了した場合でも、スポットインスタンスが終了したと Amazon EC2 によって検出されるまでに遅延が生じる可能性があります。この場合、リクエストの状態は `closed` または `cancelled` となります。

#### 永続リクエスト

スポットインスタンスが(お客様または Amazon EC2 によって)終了するときに、スポットリクエストが永続リクエストである場合、リクエストは `pending-evaluation` 状態に戻り、そして制約が満たされたときに Amazon EC2 は新しいスポットインスタンスを起動できます。

### リクエストステータス情報の取得

AWS マネジメントコンソールまたはコマンドラインツールを使用して、リクエストステータス情報を取得できます。

#### リクエストステータス情報を取得するには(コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Spot Requests] を選択し、スポットリクエストを選択します。
3. ステータスを確認するには、[Description]、[Status] の順に選択します。

#### コマンドラインを使用してリクエストステータス情報を取得する

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `describe-spot-instance-requests` (AWS CLI)
- `Get-EC2SpotInstanceRequest` (AWS Tools for Windows PowerShell)

### スポットリクエストコード

スポットリクエストステータス情報は、ステータスコード、更新時刻、およびステータスマッセージで構成されます。同時に、リクエスト入札ステータス情報は、スポットリクエストの処理を決定する場合にも役に立ちます。

スポットリクエストステータスコードは、次のとおりです。

`az-group-constraint`

Amazon EC2 は、同じアベイラビリティーゾーンでお客様が要求したインスタンスをすべて起動できるとは限りません。

`bad-parameters`

スポットリクエストの 1 つ以上のパラメータが有効ではありません（たとえば、指定した AMI が存在していません）。ステータスマッセージによって、どのパラメータが無効かを確認できます。

`canceled-before-fulfillment`

スポットリクエストが受理される前にユーザーがスポットリクエストをキャンセルしました。

`capacity-not-available`

要求したインスタンスに使用できる十分な容量が存在しません。

`constraint-not-fulfillable`

1 つ以上の制約条件が有効ではないため、スポットリクエストを受理できません（たとえば、アベイラビリティーゾーンが存在していません）。ステータスマッセージによって、どの制約条件が無効かを確認できます。

`fulfilled`

スポットリクエストは `active` であり、Amazon EC2 はスポットインスタンスを起動しています。

`instance-stopped-by-price`

スポット料金が上限価格を超えたため、インスタンスは停止しました。

`instance-stopped-by-user`

ユーザーがインスタンスを停止したか、インスタンスからシャットダウンコマンドを実行したために、インスタンスが停止されました。

`instance-stopped-no-capacity`

インスタンスで使用できる十分な Spot キャパシティーがなくなったため、インスタンスは停止しました。

`instance-terminated-by-price`

スポット料金が上限価格を超えたため、インスタンスは削除されました。リクエストが永続入札の場合、プロセスが再開され、リクエストが評価保留となります。

`instance-terminated-by-schedule`

スポットインスタンスは、スケジュールされた所要時間の最後に終了されました。

`instance-terminated-by-service`

インスタンスが停止状態から削除されました。

`instance-terminated-by-user` または `spot-instance-terminated-by-user`

受理済みの スpot インスタンスを終了させたので、リクエストステータスは `closed` 状態になり（永続リクエストでない場合）、インスタンスは `terminated` 状態になりました。

`instance-terminated-launch-group-constraint`

起動グループ内のインスタンスの 1 つ以上が終了したため、起動グループの制約条件が満たされなくなりました。

`instance-terminated-no-capacity`

インスタンスで使用できる十分な Spot キャパシティーがないため、インスタンスは削除されました。

`launch-group-constraint`

Amazon EC2 は、お客様が同時に要求したインスタンスをすべて起動できるわけではありません。同じ起動グループ内のインスタンスはすべて、同時に起動されて同時に終了します。

`limit-exceeded`

EBS ボリューム数または合計ボリュームストレージの上限を超えた。これらの制限および増加を要求する方法の詳細については、『アマゾン ウェブ サービス全般のリファレンス』の「[Amazon EBS の制限](#)」を参照してください。

`marked-for-stop`

スポットインスタンス に停止のためのマークが付けられます。

`marked-for-termination`

スポットインスタンス に終了のためのマークが付けられます。

`not-scheduled-yet`

スポットリクエストは、スケジュール設定された日付になるまで評価されません。

`pending-evaluation`

スポットインスタンス リクエストを作成すると、そのリクエストは `pending-evaluation` 状態となり、システムはリクエストのパラメータを評価します。

`pending-fulfillment`

Amazon EC2 は スポットインスタンス をプロビジョニングしようとしています。

`placement-group-constraint`

現時点で スポットインスタンス をプレイスメントグループに追加できないため、スポットリクエストをまだ受理することができません。

`price-too-low`

上限料金がスポット料金を下回っているため、リクエストを受理できません。この場合、インスタンスは起動されず、リクエストは `open` のままになります。

`request-canceled-and-instance-running`

スポットインスタンス がまだ実行されている間に、リクエストをキャンセルしました。リクエストは `cancelled` ですが、インスタンスは `running` のままでです。

`schedule-expired`

スポットリクエストは、指定された日付までに受理されなかつたため、有効期限切れとなりました。

`system-error`

予期しないシステムエラーが発生しました。これが反復性の問題である場合は、AWS サポートに連絡してください。

## スポットインスタンス の中断

スポットインスタンス に対する需要は刻一刻と大幅に変化する可能性があります。また、スポットインスタンス の可用性も利用可能な未使用の EC2 インスタンスの数に応じて大きく変化する可能性があります。スポットインスタンス が中止される可能性は常にあります。したがって、アプリケーションでスポットインスタンス の中止に対して準備する必要があります。

EC2 フリート または スポットフリート で指定した オンデマンドインスタンス は中止されません。

### 目次

- [中止の理由 \(p. 386\)](#)
- [中止動作 \(p. 386\)](#)
- [中止に対する準備 \(p. 388\)](#)
- [インスタンス休止の準備 \(p. 389\)](#)
- [スポットインスタンス 中止の通知 \(p. 389\)](#)

- 中止された スポットインスタンス の請求 (p. 391)

## 中止の理由

Amazon EC2 が スポットインスタンス を中止する場合、次のような理由が考えられます。

- 価格 – スポット料金が上限料金を上回っています。
- 容量 – スpotインスタンス の需要を満たすのに十分な未使用の EC2 インスタンスがない場合、Amazon EC2 は スpotインスタンス を中止します。インスタンスが中止される順序は、Amazon EC2 によって決定されます。
- 制約 – リクエストに起動グループやアベイラビリティゾーングループなど制約が含まれている場合、制約条件が満たされなくなったときに、その スpotインスタンス はグループとして終了されます。

## 中止動作

中止時に Amazon EC2 が スpotインスタンス を休止、停止、または終了するかを指定できます。お客様のニーズに合う中止時の動作を選択できます。デフォルトでは、スspotインスタンス は中止されると終了されます。中止動作を変更するには、Spot リクエストの作成中に、コンソールで [Interruption behavior] のオプションを選択するか、起動設定または起動テンプレートで Instance Interruption Behavior を指定します。Spot リクエストの作成中に、コンソールで中止動作を選択するには、[ターゲット容量を維持する] を選択します。このオプションを選択すると、[中止動作] が表示され、Spot サービスが中止されたときに Spot サービスを終了、停止、または休止状態にするよう指定することができます。

## 中止した スpotインスタンス の停止

この動作を変更し、次の要件が満たされたときに、Amazon EC2 が スpotインスタンス を停止するようにできます。

### 要件

- スspotインスタンス リクエストの場合、タイプは persistent にする必要があります。スspotインスタンス リクエストで起動グループを指定することはできません。
- EC2 フリート または スpot フリート リクエストの場合、タイプは maintain にする必要があります。
- ルートボリュームは、インスタンスストアボリュームではなく EBS ボリュームにする必要があります。

スspotインスタンス がスspotサービスによって停止されたら、スspotインスタンス はスspotサービスでのみ再起動できるため、同じ起動仕様を使用する必要があります。

persistent スspotインスタンス リクエストで起動された スspotインスタンス の場合、キャパシティーが、停止したインスタンスと同じアベイラビリティゾーンで利用可能、かつ同じインスタンスタイプの場合に、スspotサービスによって、その停止したインスタンスは再起動されます。

EC2 フリート または スpot フリート インスタンスが停止し、フリートのタイプが maintain の場合、スspotサービスは代替インスタンスを起動してターゲット容量を維持します。スspotサービスでは、指定された配分戦略 (lowestPrice, diversified, InstancePoolsToUseCount) に基づき、最適なプールを検索します。以前に停止したインスタンスでプールに優先順位が付けられることはできません。後に、配分戦略で、以前に停止したインスタンスを含むプールに導かれる場合、スspotサービスは、ターゲット容量に合うように停止したインスタンスを再起動します。

たとえば、配分戦略が lowestPrice の場合の スpot フリート を検討します。初回起動時、c3.large プールは、起動仕様の lowestPrice 条件を満たしています。後に、c3.large インスタンスが中止されると、スspotサービスはそのインスタンスを停止し、lowestPrice 戰略に合う別のプールからキャパシティーを補充します。今回の場合は、プールは c4.large プールになり、スspotサービスはターゲット容量を満たすように c4.large インスタンスを起動します。同様に、次回は スpot フリート によって、c5.large プールに移動されます。これらの各遷移では、スspotサービスは、以前に停止し

たインスタンスを含むプールを優先せずに、指定された配分戦略を純粋に優先します。lowestPrice 戰略では、以前に停止したインスタンスを含むプールに戻る場合があります。たとえば、インスタンスが c5.large プールで中断され、lowestPrice 戰略によって c3.large または c4.large プールに戻った場合、以前に停止したインスタンスはターゲット容量を満たすために再起動されます。

スポットインスタンスが停止している間、インスタンスの属性の一部は変更できますが、インスタンスタイプを変更することはできません。EBS ボリュームをデタッチまたは削除すると、スポットインスタンスが起動したときにアタッチされません。ルートボリュームをデタッチし、スポットサービスがスポットインスタンスを起動しようとすると、インスタンスの起動は失敗し、スポットサービスは停止されたインスタンスを削除します。

停止中に、スポットインスタンスを終了できます。スポットリクエスト、EC2 フリート、またはスポットフリートをキャンセルすると、スポットサービスは停止中の関連スポットインスタンスを終了します。

スポットインスタンスの停止中は、維持されている EBS ボリュームに対してのみ課金されます。EC2 フリートおよびスポットフリートでは、停止中のインスタンスの数が多い場合、アカウントの EBS ボリューム数の制限を超えることがあります。

### 中断したスポットインスタンスの休止

この動作を変更し、次の要件が満たされたときに、Amazon EC2 がスポットインスタンスを休止するようにできます。

#### 要件

- スpotトインスタンスリクエストの場合、タイプは `persistent` にする必要があります。スポットインスタンスリクエストで起動グループを指定することはできません。
- EC2 フリートまたはスポットフリートリクエストの場合、タイプは `maintain` にする必要があります。
- ルートボリュームはインスタンストアボリュームではなく、EBS ボリュームでなければならず、休止中にインスタンスマモリ (RAM) を格納するのに十分な大きさでなければなりません。
- C3、C4、C5、M4、M5、R3、R4 (100 GB 未満のメモリ) のインスタンスがサポートされています。
- 以下のオペレーティングシステムがサポートされています: Amazon Linux 2、Amazon Linux AMI、AWS チューニングされた Ubuntu カーネル (`linux-aws`) 4.4.0-1041 以降の Ubuntu、および Windows Server 2008 R2 以降。
- サポートされているオペレーティングシステムに休止エージェントをインストールするか、すでにエージェントが含まれている次の AMI のいずれかを使用します。
  - Amazon Linux 2
  - Amazon Linux AMI 2017.09.1 以降
  - Ubuntu Xenial 16.04 2017.11.21 以降
  - Windows Server 2008 R2 AMI 2017.11.19 以降
  - Windows Server 2012 または Windows Server 2012 R2 AMI 2017.11.19 以降
  - Windows Server 2016 AMI 2017.11.19 以降
  - Windows Server 2019
- エージェントを開始します。インスタンスの起動時にユーザーデータを使用してエージェントを起動することをお勧めします。または、エージェントを手動で開始できます。

#### 推奨事項

- 休止中はインスタンスマモリがルートボリュームに格納されるため、暗号化された Amazon EBS ボリュームをルートボリュームとして使用することを強くお勧めします。これにより、データがボリューム上にあるときや、インスタンスとボリューム間でデータが移動しているときに、メモリ (RAM) の内容が暗号化されます。ルートボリュームが暗号化された Amazon EBS ボリュームであることを確認するには、次の 3 つのオプションのいずれかを使用します。

- EBS の「シングルステップ」暗号化: 1 回の run-instances API 呼び出しで、暗号化されていない AMI から暗号化された EBS-Backed EC2 インスタンスを起動できます。詳細については、「[EBS-Backed AMI での暗号化の利用 \(p. 151\)](#)」を参照してください。
- デフォルトでの EBS 暗号化: EBS 暗号化をデフォルトで有効にして、AWS アカウントで作成されたすべての新しい EBS ボリュームを暗号化できます。詳細については、「[デフォルトでの暗号化 \(p. 1017\)](#)」を参照してください。
- 暗号化された AMI: 暗号化された AMI を使用してインスタンスを起動することで、EBS 暗号化を有効にすることができます。暗号化されたルートスナップショットが AMI にない場合は、それを新しい AMI にコピーして暗号化をリクエストできます。詳細については、「[コピー時に暗号化されていないイメージを暗号化する \(p. 155\)](#)」および「[AMI のコピー \(p. 159\)](#)」を参照してください。

スポットインスタンス ガスポットサービスによって休止状態になると、EBS ボリュームは保存され、インスタンスマモリ (RAM) はルートボリュームに保存されます。インスタンスのプライベート IP アドレスも保存されます。Elastic IP アドレス以外のインスタンスストレージボリュームとパブリック IP アドレスは保持されません。インスタンスが休止状態の間は、EBS ボリュームに対してのみ課金されます。EC2 フリート および スポットフリート では、休止中のインスタンスの数が多い場合、アカウントの EBS ボリューム数の制限を超えることがあります。

エージェントは、インスタンスがスポットサービスから信号を受信すると、オペレーティングシステムが休止状態になるように要求します。エージェントがインストールされていない場合、基盤となるオペレーティングシステムは休止状態をサポートしていないか、インスタンスマモリを保存するのに十分なボリュームスペースがなく、休止状態が失敗し、代わりにスポットサービスがインスタンスを停止します。

スポットサービスがスポットインスタンスを休止すると、中断通知が届きますが、スポットインスタンスが中断されるまでに 2 分かかりません。休止は直ちに開始されます。インスタンスが休止状態にある間は、インスタンスのヘルスチェックが失敗する場合があります。休止プロセスが完了すると、インスタンスの状態は stopped になります。

スポットインスタンス ガスポットサービスによって休止状態になった後は、スポットサービスによってのみ再開できます。スポットサービスは、スポット料金が指定された上限料金を下回り、容量が利用可能になったときにインスタンスを再開します。

詳細については、「[インスタンス休止の準備 \(p. 389\)](#)」を参照してください。

オンデマンドインスタンス の休止の詳細については、「[Linux インスタンスの休止 \(p. 532\)](#)」を参照してください。

## 中断に対する準備

スポットインスタンス を使用する場合のベストプラクティスを以下に示します。

- オンデマンド価格であるデフォルトの上限料金を使用します。
- 必要なソフトウェア設定を含む Amazon Machine Image (AMI) を使用することにより、リクエストが受理されたらすぐにインスタンスを実行できるように、準備が完了していることを確認します。また、ユーザーデータを使用して起動時にコマンドを実行することもできます。
- スpotトインスタンス の終了の影響を受けない場所に、定期的に重要なデータを保存します。たとえば、Amazon S3、Amazon EBS、または DynamoDB を使用できます。
- 作業を頻繁に保存できるように、作業を (Grid、Hadoop、キューベースのアーキテクチャを使用して) 細かいタスクに分割するか、チェックポイントを使用します。
- スpotトインスタンス の中断の通知を使用して、スポットインスタンスのステータスをモニタリングします。
- AWS では、この警告はできるだけ早く提供するよう努めていますが、警告が提供される前に スpotトインスタンス が削除される可能性があります。アプリケーションをテストして、中断通知のテストを行っている場合であっても、予期しないインスタンスの終了をアプリケーションが適切に処理できることを確認します。オンデマンドインスタンス を使用してアプリケーションを実行し、オンデマンドインスタンス を自分で終了することでこれを確認できます。

## インスタンス休止の準備

すでにエージェントが含まれている AMI を使用していない限り、インスタンスに休止エージェントをインストールする必要があります。エージェントが AMI に含まれているかどうか、または自分でインストールしたかどうかに關係なく、インスタンスの起動時にエージェントを実行する必要があります。

次の手順は、Linux インスタンスの準備に役立ちます。Windows インスタンスを準備する手順については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[インスタンス休止の準備](#)」を参照してください。

Amazon Linux インスタンスを準備するには

1. カーネルが休止をサポートしていることを確認し、必要に応じてカーネルを更新します。
2. AMI にエージェントが含まれていない場合は、次のコマンドを使用してエージェントをインストールします。

```
sudo yum update; sudo yum install hibagent
```

3. 以下をユーザーデータに追加します。

```
#!/bin/bash
/usr/bin/enable-ec2-spot-hibernation
```

Ubuntu インスタンスを準備するには

1. AMI にエージェントが含まれていない場合は、次のコマンドを使用してエージェントをインストールします。休止状態のエージェントは、Ubuntu 16.04 以降でのみ使用できます。

```
sudo apt-get install hibagent
```

2. 以下をユーザーデータに追加します。

```
#!/bin/bash
/usr/bin/enable-ec2-spot-hibernation
```

## スポットインスタンス 中断の通知

スポットインスタンス の中断から保護する最善の方法は、アプリケーションを耐障害性のある設計にすることです。さらに、スポットインスタンス の中断の通知を活用できます。これによって、Amazon EC2 がスポットインスタンス を停止または削除する 2 分前に警告が提供されます。5 秒ごとにこれらの警告を確認することをお勧めします。

この警告は、CloudWatch イベントとして、および スポットインスタンス 上の [インスタンスマタデータ \(p. 593\)](#) の項目として使用可能です。

中断動作として休止状態を指定した場合は、中断通知が表示されますが、休止状態プロセスはすぐに開始されるため、2 分間の警告は表示されません。

### EC2 スpot インスタンス Interruption Notice

Amazon EC2 は、スポットインスタンス を中断する場合、実際の中斷に先立つ 2 分前にイベントを発生させます。このイベントは Amazon CloudWatch Events で検出できます。詳細については、[Amazon CloudWatch Events ユーザーガイド](#)を参照してください。

以下に示しているのは、スポットインスタンス 割り込みのイベントの例です。instance-action の可能な値は hibernate、stop、terminate です。

```
{  
    "version": "0",  
    "id": "12345678-1234-1234-1234-123456789012",  
    "detail-type": "EC2 Spot Instance Interruption Warning",  
    "source": "aws.ec2",  
    "account": "123456789012",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-2",  
    "resources": ["arn:aws:ec2:us-east-2:123456789012:instance/i-1234567890abcdef0"],  
    "detail": {  
        "instance-id": "i-1234567890abcdef0",  
        "instance-action": "action"  
    }  
}
```

### instance-action

スポットインスタンスがスポットサービスによって、停止または終了されるとマークされた場合、[instance-action 項目がインスタンスマタデータ \(p. 593\)](#)に存在します。そうでない場合、これは存在しません。instance-action は以下のように取得できます。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/spot/instance-action
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/spot/instance-action
```

instance-action 項目は、アクションおよびアクションのおよその発生時刻 (UTC) を指定します。

次の例では、このインスタンスの停止時刻を示します。

```
{"action": "stop", "time": "2017-09-18T08:22:00Z"}
```

次の例では、このインスタンスの終了時刻を示します。

```
{"action": "terminate", "time": "2017-09-18T08:22:00Z"}
```

Amazon EC2 がインスタンスを停止または終了する準備をしていない場合や、お客様が自分でインスタンスを終了した場合、instance-action は存在せず、HTTP 404 エラーが出力されます。

#### termination-time

この項目は下位互換性のために維持されています。代わりに instance-action を使用してください。

スポットインスタンスがスポットサービスによって削除されるとマークされた場合、termination-time 項目がインスタンスマタデータに存在します。そうでない場合、これは存在しません。termination-time は以下のように取得できます。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
```

```
[ec2-user ~]$ if curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo terminated; fi
```

IMDSv1

```
[ec2-user ~]$ if curl -s http://169.254.169.254/latest/meta-data/spot/termination-time | grep -q .*T.*Z; then echo terminated; fi
```

`termination-time` 項目は、インスタンスがシャットダウン信号を受信するおよその時刻 (UTC) を指定します。例:

```
2015-01-05T18:02:00Z
```

Amazon EC2 がインスタンスを終了する準備をしていない場合や、お客様が自分でスポットインスタンスを終了した場合、`termination-time` 項目は存在しない (この場合、HTTP 404 エラーが output されます) か、時刻値以外の値が含まれます。

Amazon EC2 がインスタンスの終了に失敗した場合は、リクエストステータスが `fulfilled` に設定されます。`termination-time` 値は、元のおよその時刻のまま (過去の時刻になっていますが)、インスタンスのメタデータに残ります。

## 中断された スポットインスタンス の請求

スポットインスタンス (Spot ブロックではない) が中断された場合は、次のように請求されます。

スポットインスタンスを中断するユーザー	オペレーティングシステム	最初の 1 時間で中断	最初の 1 時間後の任意の時間に中断
お客様がスポットインスタンスを停止または終了した場合	Linux (RHEL および SUSE は除く) Windows、RHEL、SUSE	使用された時間 (秒) の請求 使用時間が 1 時間未満の場合でも、1 時間分の料金を請求	使用された時間 (秒) の請求 使用された 1 時間分 (中断された時間が 1 時間未満の場合も 1 時間分) を請求
Amazon EC2 でスポットインスタンスが中断された場合	Linux (RHEL および SUSE は除く) Windows、RHEL、SUSE	料金は発生しない 料金は発生しない	使用された時間 (秒) の請求 使用された 1 時間分は請求されるが、中断された時間が 1 時間未満の場合は請求されない

スポットインスタンス (Spot ブロック内) が中断された場合は、次のように請求されます。

スポットインスタンスを中断するユーザー	オペレーティングシステム	最初の 1 時間で中断	最初の 1 時間後の任意の時間に中断
お客様がスポットインスタンスを停止または終了した場合	Linux (RHEL および SUSE は除く) Windows、RHEL、SUSE	使用された時間 (秒) の請求 使用時間が 1 時間未満の場合でも、1 時間分の料金を請求	使用された時間 (秒) の請求 使用された 1 時間分 (中断された時間が 1 時間未満の場合も 1 時間分) を請求

スポットインスタンスを中断するユーザー	オペレーティングシステム	最初の 1 時間で中断	最初の 1 時間後の任意の時間に中断
Amazon EC2 で スポットインスタンスが中断された場合	Linux (RHEL および SUSE は除く) Windows、RHEL、SUSE	料金は発生しない 料金は発生しない	料金は発生しない
			料金は発生しない

## スポットインスタンスデータフィード

スポットインスタンスの料金について理解しやすくするため、Amazon EC2 では、スポットインスタンスの使用状況と料金を示すデータフィードを提供しています。このデータフィードは、データフィードを購読するときに指定する Amazon S3 バケットに送信されます。

データフィードファイルは、通常、1 時間に 1 回バケットに届き、各使用時は、通常、単一のデータファイルでカバーされます。このファイルは、バケットに配信される前に圧縮 (gzip) されます。ファイルが大きい場合は、Amazon EC2 は指定した時間の使用状況に関するファイルを複数書き込むことができます (ある時間のファイル内容が圧縮前に 50 MB を超える場合など)。

### Note

ある時間に対してスポットインスタンスが実行されていない場合、その時間のデータフィードファイルは届きません。

### 目次

- データフィードのファイル名と形式 (p. 392)
- Amazon S3 バケットの要件 (p. 393)
- スポットインスタンスのデータフィードの購読 (p. 393)
- スポットインスタンスのデータフィードの削除 (p. 394)

## データフィードのファイル名と形式

スポットインスタンスのデータフィードのファイル名には次の形式を使用します (UTC の日付と時刻を使用)。

```
bucket-name.s3.amazonaws.com/{optional prefix}/aws-account-id.YYYY-MM-DD-HH.n.unique-id.gz
```

たとえば、バケット名が **aws-s3-bucket1** で、プレフィックスが **myprefix** である場合、ファイル名は次のようにになります。

```
aws-s3-bucket1.s3.amazonaws.com/myprefix/111122223333.2014-03-17-20.001.pwBdGTJG.gz
```

スポットインスタンスのデータフィードファイルはタブで区切られています。データファイルの各行は、1 個のインスタンス時間に対応し、次の表に示すフィールドが含まれています。

フィールド	説明
Timestamp	そのインスタンス使用量に対して請求される価格を決定するために使用されるタイムスタンプ。
UsageType	請求の対象となっている使用タイプおよびインスタンスタイプ。 <code>m1.small</code> スポットインスタンスの場合、このフィールドは <code>SpotUsage</code> に設定されます。他のすべてのインスタンスタイプでは、このフィールドは <code>SpotUsage:{instance-type}</code> に設定されます。たとえば、 <code>SpotUsage:c1.medium</code> と指定します。

フィールド	説明
Operation	請求の対象となっている製品。Linux スポットインスタンスの場合、このフィールドは RunInstances に設定されます。Windows スポットインスタンスの場合、このフィールドは RunInstances:0002 に設定されます。スポット使用状況は、利用可能ゾーンに従ってグループ化されます。
InstanceId	このインスタンス使用量を生成した スポットインスタンス のインスタンス ID。
MyBidID	このインスタンス使用量を生成した スポットインスタンス リクエストの ID。
MyMaxPrice	この スポットインスタンス リクエストに指定された上限価格。
MarketPrice	Timestamp フィールドに指定された時刻のスポット料金。
Charge	このインスタンス使用量に請求される価格。
Version	このレコードのデータフィードファイル名に含まれるバージョン。

## Amazon S3 バケットの要件

データフィードの購読時に、データフィードファイルを格納する Amazon S3 バケットを指定する必要があります。データフィード用の Amazon S3 バケットを選択する前に、以下の点を考慮します。

- FULL\_CONTROL および s3:GetBucketAcl アクションのアクセス許可を含むバケットへの s3:PutBucketAcl アクセス許可が必要です。  
バケット所有者には、デフォルトでこの権限があります。それ以外の場合、バケット所有者は AWS アカウントにこのアクセス許可を付与する必要があります。
- データフィードを購読すると、これらのアクセス許可を使用してバケット ACL が更新され、AWS データフィードアカウントに FULL\_CONTROL アクセス許可が付与されます。AWS データフィードアカウントは、データフィードファイルをバケットに書き込みます。アカウントに必要なアクセス許可がない場合、データフィードファイルをバケットに書き込むことはできません。

### Note

ACL を更新して AWS データフィードアカウントのアクセス許可を削除すると、データフィードファイルをバケットに書き込むことができなくなります。データフィードファイルを受け取るには、データフィードを再購読する必要があります。

- 各データフィードファイルには、独自の ACL があります(バケットの ACL とは別です)。バケット所有者には、データファイルに対して FULL\_CONTROL のアクセス許可があります。AWS データフィードアカウントには読み書きのアクセス許可があります。
- AWS データフィードの購読を削除しても、Amazon EC2 ではバケットまたはデータファイルのデータフィードアカウントの読み書きのアクセス許可は削除されません。これらのアクセス許可は自分で削除する必要があります。

## スポットインスタンス のデータフィードの購読

データフィードを購読するには、次の `create-spot-datafeed-subscription` コマンドを使用します。

```
aws ec2 create-spot-datafeed-subscription --bucket aws-s3-bucket1 [--prefix myprefix]
```

出力例を次に示します。

```
{
  "SpotDatafeedSubscription": {
    "OwnerId": "111122223333",
```

```
        "Prefix": "myprefix",
        "Bucket": "aws-s3-bucket1",
        "State": "Active"
    }
}
```

## スポットインスタンス のデータフィードの削除

データフィードを削除するには、次の `delete-spot-datafeed-subscription` コマンドを使用します。

```
aws ec2 delete-spot-datafeed-subscription
```

## スポットインスタンス の制限

スポットインスタンス リクエストには、次の制限が適用されます。

### 制限

- [スポットリクエスト制限 \(p. 394\)](#)
- [スポットフリート の制限 \(p. 394\)](#)
- [T3スポットインスタンス \(p. 395\)](#)
- [T2スポットインスタンス \(p. 395\)](#)

## スポットリクエスト制限

デフォルトでは、アカウントの制限はリージョンごとに 20 個の スポットインスタンス です。スポットインスタンス を終了してもリクエストはキャンセルしない場合、Amazon EC2 がその終了を検出してリクエストを終了するまで、リクエストはこの制限に対してカウントされます。

スポットインスタンス の制限は動的です。新規のアカウントの場合、制限が 20 個未満で開始することがありますが、時間とともに増加する場合があります。また、アカウントに特定の スポットインスタンス タイプの制限がある場合があります。スポットインスタンス リクエストを送信したときに `Max spot instance count exceeded` エラーが表示された場合は、AWS サポートセンターで、スポットインスタンス 制限の引き上げをリクエストする [[ケースの作成](#)] フォームを送信できます。[Limit Type (制限のタイプ)] で [EC2 スpotトインスタンス] を選択します。詳細については、「[Amazon EC2 サービスの制限 \(p. 1130\)](#)」を参照してください。

## スポットフリート の制限

スポットフリート または EC2 フリート によって起動されるインスタンスには、Amazon EC2 の通常の制限 (スポットリクエスト価格制限、インスタンス制限、容量制限など) が適用されます。また、以下の制限も適用されます。

- リージョンあたりのアクティブな スpotトフリート および EC2 フリート の数: 1,000\*
- フリートあたりの起動仕様の数: 50\*
- 起動仕様内のユーザーデータのサイズ: 16 KB\*
- スpotトフリート または EC2 フリート あたりのターゲット容量: 10,000
- リージョン内のすべての スpotトフリート および EC2 フリート におけるターゲット容量: 100,000
- スpotトフリート リクエストまたは EC2 フリート リクエストは、リージョンにまたがることはできません。
- スpotトフリート リクエストまたは EC2 フリート リクエストは、同じアベイラビリティーゾーンから複数の異なるサブネットにまたがることはできません。

ターゲット容量のデフォルトの制限を超える容量が必要な場合は、AWS サポートセンターの [ケースの作成](#) フォームから制限の引き上げをリクエストできます。[制限のタイプ] で、[EC2 フリート] を選択して、

リージョンを選択し、[フリートごとのターゲットフリート容量(単位)]か、[リージョンごとのターゲットフリート容量(単位)]、または両方を選択します。

\* これらはハード制限です。この制限の引き上げをリクエストすることはできません。

## T3スポットインスタンス

すぐに T3 スポットインスタンス を短期間使用する予定で、CPU クレジットを獲得するためのアイドル時間がない場合は、より高い費用を支払うことを避けるために、T3 スポットインスタンスを [standard \(p. 211\)](#) モードを起動することをお勧めします。

T3 スポットインスタンス を [unlimited \(p. 203\)](#) モードで起動し、すぐに CPU を破棄すると、余分なクレジットが破棄されます。インスタンスを短期間使用すると、インスタンスは CPU クレジットが発生して余剰クレジットを払う時間がなくなり、インスタンスを終了するときに余剰クレジットが課金されます。

T3 スポットインスタンス の Unlimited モードは、CPU クレジットをバーストするためにインスタンスが十分長く実行される場合にのみ適しています。それ以外の場合、余剰クレジットを支払うことで、T3 スポットインスタンス は M5 または C5 インスタンスよりも高価になります。詳細については、「[Unlimited モードと固定 CPU を使用する場合 \(p. 205\)](#)」を参照してください。

## T2スポットインスタンス

起動クレジットは、インスタンスを構成するために十分なコンピューティングリソースを提供し、T2 インスタンスの初期起動を効率的に実現することを意図しています。T2 インスタンスの起動を繰り返して新しい起動クレジットを利用することは許可されていません。CPU が持続的に必要な場合、(一定期間のアイドリングにより) クレジットを獲得して [T2 無制限 \(p. 203\)](#) を使用するか、専用 CPU (c4.largeなど) があるインスタンスタイプを使用します。

## Dedicated Hosts

Amazon EC2 Dedicated Host は、EC2 インスタンスを起動できるお客様専用の物理サーバーです。Dedicated Hosts を使うと、Windows Server、Microsoft SQL Server、SUSE および Linux Enterprise Server を含むソフトウェアのライセンスを、既存のソケット単位、コア単位または VM 単位で使用できます。

### コンテンツ

- [Dedicated Hosts と ハードウェア専有インスタンス の違い \(p. 395\)](#)
- [自分のライセンスを使用する \(p. 396\)](#)
- [Dedicated Host インスタンスの容量 \(p. 396\)](#)
- [Dedicated Hosts に関する制限事項 \(p. 397\)](#)
- [価格と請求 \(p. 397\)](#)
- [Dedicated Hosts の使用 \(p. 398\)](#)
- [共有 Dedicated Hosts の使用 \(p. 416\)](#)
- [ホスト復旧 \(p. 420\)](#)
- [設定変更の追跡 \(p. 424\)](#)

## Dedicated Hosts と ハードウェア専有インスタンス の違い

Dedicated Hosts と ハードウェア専有インスタンス のどちらを使用しても、お客様専用の物理サーバーに Amazon EC2 インスタンスを起動することができます。

ハードウェア専有インスタンス と Dedicated Hosts のインスタンスの間に、パフォーマンス、セキュリティ、または物理的な違いはありません。ただし、この 2 つの間にはいくつかの違いがあります。次のテーブルでは、Dedicated Hosts と ハードウェア専有インスタンス の主な違いをいくつか紹介します。

	Dedicated Host	ハードウェア専有インスタンス
請求	ホストごとの請求	インスタンスごとの請求
ソケット、コア、ホスト ID の可視性	ソケットと物理コアの数が見える	可視性なし
ホストおよびインスタンスアフィニティ	インスタンスを同じ物理サーバーに徐々にデプロイし続けることができる	サポート外
ターゲットを絞ったインスタンスの配置	インスタンスを物理サーバーに配置する方法についての可視性と制御が高い	サポート外
インスタンスの自動復旧	サポート対象。詳細については、「 <a href="#">ホスト復旧 (p. 420)</a> 」を参照してください。	サポート対象
自分のライセンス使用 (BYOL)	サポート対象	サポート外

## 自分のライセンスを使用する

Dedicated Hosts を利用すると、ソフトウェアライセンスを、既存のソケット単位、コア単位、または VM 単位で使用できます。自分のライセンスを使用する場合、お客様は自分のライセンスを管理する責任があります。ただし、Amazon EC2 ではインスタンスアフィニティやターゲットを絞ったプレイスメントなど、ライセンスのコンプライアンスを維持するための機能を利用できます。

自分のボリュームライセンスマシンのイメージを Amazon EC2 で使用するための一般的な手順を以下に示します。

- マシンイメージの使用を制御するライセンス条件が、仮想化クラウド環境での使用を許可していることを確認します。
- マシンイメージを Amazon EC2 内で使用できることを確認したら、VM Import/Export を使用してインポートします。マシンイメージをインポートする方法については、「[VM Import/Export ユーザーガイド](#)」を参照してください。
- マシンイメージをインポートしたら、自分のアカウント内のアクティブな Dedicated Hosts で、そのイメージからインスタンスを起動できます。
- オペレーティングシステムによっては、これらのインスタンスを実行するときに、独自の KMS サーバー

### Note

イメージを AWS で使用する方法を追跡するには、AWS Config でホストの記録を有効にします。AWS Config を使用すると、Dedicated Host への設定の変更を記録し、出力をライセンスレポートのデータソースとして使用することができます。詳細については、「[設定変更の追跡 \(p. 424\)](#)」を参照してください。

## Dedicated Host インスタンスの容量

AWSNitro システムを基盤とする Dedicated Hosts は、同じインスタンスファミリーに属する複数のインスタンスタイプをサポートしています。たとえば、r5 Dedicated Host を割り当てると、2 つのソケットと 48 の物理コアを持つホストを使用でき、このホストにおいて、r5.2xlarge や r5.4xlarge といった異

なるインスタンスタイプを実行できます。インスタンスは、該当ホストと関係のあるコアの容量に応じた数まで実行できます。以下の表は、Dedicated Hostにおいて実行できる異なるインスタンスタイプの組合せ例です。

インスタンスマリーファミリー	インスタンスタイプの組合せ例
R5	<ul style="list-style-type: none"><li>具体例 1: 4 x r5.4xlarge + 4 x r5.2xlarge</li><li>具体例 2: 1 x r5.12xlarge + 1 x r5.4xlarge + 1 x r5.2xlarge + 5 x r5.xlarge + 2 x r5.large</li></ul>
C5	<ul style="list-style-type: none"><li>具体例 1: 1 x c5.9xlarge + 2 x c5.4xlarge + 1 x c5.xlarge</li><li>具体例 2: 4 x c5.4xlarge + 1 x c5.xlarge + 2 x c5.large</li></ul>
M5	<ul style="list-style-type: none"><li>具体例 1: 4 x m5.4xlarge + 4 x m5.2xlarge</li><li>具体例 2: 1 x m5.12xlarge + 1 x m5.4xlarge + 1 x m5.2xlarge + 5 x m5.xlarge + 2 x m5.large</li></ul>

AWS Nitro システムを基盤としていないインスタンスマリーファミリーについては、特定のインスタンスタイプに Dedicated Host を設定することのみ行えます。Dedicated Hostsにおいてサポートされているすべてのインスタンスマリーファミリーおよびインスタンスタイプの一覧については、[Amazon EC2 Dedicated Hosts の料金](#)を参照してください。

## Dedicated Hosts に関する制限事項

Dedicated Hosts を割り当てる際は、次の制限と制約に注意してください。

- RHEL、SUSE Linux および Windows AMI (AWS が提供するものであるか、AWS マーケットプレイスにおいて提供されるものであるかは問いません) を Dedicated Hosts と併用することはできません。
- リージョンごとに、インスタンスマリーファミリーあたり最大 2 つのオンデマンド Dedicated Hosts を割り当てることができます。制限値を引き上げることもできます。[Amazon EC2 Dedicated Hosts の割り当て制限の引き上げを申請してください](#)。
- Dedicated Host で実行されるインスタンスは、VPC でのみ起動できます。
- Auto Scaling グループは、ホストリソースグループを指定する起動テンプレートを使用する場合にサポートされます。詳細については、Amazon EC2 Auto Scaling ユーザーガイドの「[Auto Scaling グループの起動テンプレートの作成](#)」を参照してください。
- Amazon RDS インスタンスはサポートされません。
- AWS 無料利用枠は Dedicated Hosts には使用できません。
- インスタンスのplacement 制御は、Dedicated Hosts でのインスタンスの起動管理を表します。placement グループは Dedicated Hosts ではサポートされません。

## 価格と請求

Dedicated Host の料金は支払いオプションによって異なります。

### 支払いオプション

- [オンデマンド Dedicated Hosts \(p. 398\)](#)
- [Dedicated Host の予約 \(p. 398\)](#)
- [Savings Plans \(p. 398\)](#)

## オンデマンド Dedicated Hosts

アカウントに Dedicated Host を割り当てるとき、自動的にオンデマンド請求がアクティブになります。

Dedicated Host のオンデマンド価格は、インスタンスファミリーとリージョンによって異なります。起動対象として選択したインスタンス数量とサイズに関係なく、Dedicated Host の時間料金が課金されます。つまり、実行対象として選択した個別のインスタンスではなく、Dedicated Host 全体の料金が課金されます。オンデマンド価格の詳細については、「[Amazon EC2 Dedicated Hosts のオンデマンド価格](#)」を参照してください。

オンデマンド Dedicated Host はいつでも解放して、料金の発生を止めることができます。Dedicated Host の解放の詳細については、「[Dedicated Host の解放 \(p. 412\)](#)」を参照してください。

## Dedicated Host の予約

Dedicated Host の予約では、オンデマンド Dedicated Hosts の実行と比べて請求の割引が得られます。予約は、3 つの支払いオプションで利用できます。

- 前払いなし — 前払いなしの予約では、期間内の Dedicated Host の使用に対して割引があり、前払い料金は必要ありません。1 年契約でのみ利用できます。
- 一部前払い — 予約の一部を前払いする必要があります。期間内の残りの時間は割引された時間料金で請求されます。1 年および 3 年契約で利用できます。
- 全額前払い — 実質的に最低価格で利用できます。1 年および 3 年契約で利用でき、期間中のすべてのコストが含まれます。それ以外の料金は発生しません。

予約を購入するには、アカウントでアクティブな Dedicated Hosts が必要です。各予約は、アカウントの単一で特定の Dedicated Host が対象です。予約は、インスタンスサイズではなくホストのインスタンスファミリーに適用されます。インスタンスサイズが異なる 3 つの Dedicated Hosts (m4.xlarge、m4.medium、および m4.large) がある場合、1 つの m4 予約をこれらすべての Dedicated Hosts に関連付けることができます。予約のインスタンスファミリーとリージョンは、関連付ける Dedicated Host のインスタンスファミリーとリージョンに一致する必要があります。

予約が Dedicated Host に関連付けられている場合、Dedicated Host は予約期間が終了するまで解放できません。

予約の料金の詳細については、「[Amazon EC2 Dedicated Hosts 料金表](#)」を参照してください。

## Savings Plans

Savings Plans は、オンデマンドインスタンスと比べて大幅に料金を節約できる柔軟な料金モデルです。Savings Plans は、1~3 年間は時間あたりの USD 建て料金で一定量の使用を継続するという確約を条件とする割引プランです。これによって、特定の Dedicated Host を使用することをコミットせずに、ニーズに最も適した Dedicated Hosts を使用して、コストを削減し続ける柔軟性が得られます。詳細については、[AWS Savings Plans ユーザーガイド](#) を参照してください。

## Dedicated Hosts の使用

Dedicated Host を使用するには、まずアカウントで使用するホストを割り当てます。次にインスタンスのホストテナントを指定して、ホストにインスタンスを起動します。インスタンスを起動する特定のホストを選択する必要があります。または、自動配置が有効になっており、そのインスタンスタイプが一致する他のホストでも起動できるようにすることもできます。インスタンスを停止して再起動する場合、同じホストで再起動されるか別のホストで再起動されるかは、ホストのアフィニティ設定によって決まります。

あるオンデマンドホストが不要になった場合は、そのホストで実行されているインスタンスを停止し、別のホストで起動するように指定してから、ホストを解放することができます。

Dedicated Hosts は AWS License Manager にも統合されています。License Manager では、ホストリソースグループを作成できます。ホストリソースグループは、1 つのエンティティとして管理される

Dedicated Hosts コレクションです。ホストリソースグループを作成する場合は、Dedicated Hosts のホスト管理設定（自動割り当てや自動解放など）を指定します。これにより、これらのホストを手動で割り当てて管理することなく、Dedicated Hosts にインスタンスを作成できます。詳細については、AWS License Manager ユーザーガイドの「[ホストリソースグループ](#)」を参照してください。

## 目次

- [Dedicated Hosts の割り当て \(p. 399\)](#)
- [Dedicated Host にインスタンスを作成する \(p. 401\)](#)
- [ホストリソースグループにインスタンスを作成する \(p. 403\)](#)
- [自動配置とアフィニティについて \(p. 404\)](#)
- [Dedicated Host 自動配置の変更 \(p. 405\)](#)
- [サポート対象インスタンスタイプの修正 \(p. 406\)](#)
- [インスタンスのテナンシーおよびアフィニティの変更 \(p. 408\)](#)
- [Dedicated Hosts の表示 \(p. 409\)](#)
- [Dedicated Host のタグ付け \(p. 410\)](#)
- [Dedicated Hosts のモニタリング \(p. 411\)](#)
- [Dedicated Host の解放 \(p. 412\)](#)
- [Dedicated Host の予約 の購入 \(p. 413\)](#)
- [Dedicated Host の予約の表示 \(p. 415\)](#)
- [Dedicated Host の予約 のタグ付け \(p. 415\)](#)

## Dedicated Hosts の割り当て

Dedicated Hosts の使用を始めるには、Amazon EC2 コンソールまたはコマンドラインツールを使い、ご自身のアカウントにおいて Dedicated Hosts を割り当てる必要があります。

Dedicated Host を割り当てるとき、ご自身のアカウントにおいて Dedicated Host の容量がすぐに使用できるようになり、Dedicated Host におけるインスタンスの起動を開始できます。

Dedicated Host は、次の方法で割り当てることができます。

### 新しいコンソール

#### Dedicated Host を割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts]、[Dedicated Host の割り当て] の順に選択します。
3. [インスタンスファミリー] については、Dedicated Host 向けのインスタンスファミリーを選択します。
4. Dedicated Host について、選択したインスタンスファミリー内にある複数のインスタンスをサポートするか、特定のインスタンスタイプのみサポートするかを指定します。次のいずれかを行ってください。
  - 選択したインスタンスファミリー内の複数のインスタンスタイプをサポートするように Dedicated Host を設定するには、[複数のインスタンスタイプをサポートする]、[有効化] の順に選択します。この選択を行うと、Dedicated Host において、同一インスタンスファミリー内の異なるインスタンスタイプを起動できるようになります。たとえば、m5 インスタンスファミリーとこのオプションを選択すると、Dedicated Host において m5.xlarge インスタンスと m5.4xlarge インスタンスを起動できます。複数のインスタンスタイプをサポートするように設定できるインスタンスファミリーは、A1, C5, C5n, M5, M5n, R5, and R5n です。
  - 選択したインスタンスファミリー内にある特定のインスタンスのみサポートするように Dedicated Host を設定するには、[複数のインスタンスをサポートする] をクリアしてから、

[インスタンスタイプ]について、サポートするインスタンスタイプを選択します。これにより、Dedicated Hostで1つのインスタンスタイプを起動できます。たとえば、このオプションを選択し、m5.4xlargeをサポート対象インスタンスタイプとして指定すると、Dedicated Hostにおいてはm5.4xlargeインスタンスに限り起動できます。

5. [アベイラビリティーゾーン]については、Dedicated Hostを割り当てるアベイラビリティーゾーンを選択します。
6. インスタンスタイプが一致する、ターゲットを絞らないインスタンスの起動を受け入れることをDedicated Hostに許可するには、[Instance auto-placement (インスタンスの自動プレイスメント)]で、[Enable (有効)]を選択します。自動配置の詳細については、「[自動配置とアフィニティについて \(p. 404\)](#)」を参照してください。
7. Dedicated Hostのホスト復旧を有効にするには、[Host recovery (ホスト復旧)]、[有効化]の順に選択します。詳細については、「[ホスト復旧 \(p. 420\)](#)」を参照してください。
8. [数量]については、割り当てるDedicated Hostsの数を入力します。
9. (オプション) [Add Tag (タグの追加)]を選択し、タグキーとタグ値を入力します。
10. [Allocate]を選択します。

## 古いコンソール

### Dedicated Hostを割り当てるには

1. <https://console.aws.amazon.com/ec2/>でAmazon EC2コンソールを開きます。
2. ナビゲーションペインで、[Dedicated Hosts]、[Dedicated Hostの割り当て]の順に選択します。
3. [インスタンスマルチアーフィールド]については、Dedicated Host向けのインスタンスマルチアーフィールドを選びます。
4. Dedicated Hostについて、選択したインスタンスマルチアーフィールド内にある複数のインスタンスをサポートするか、特定のインスタンスタイプのみサポートするかを指定します。次のいずれかを行ってください。
  - 選択したインスタンスマルチアーフィールド内にある複数のインスタンスをサポートするようにDedicated Hostを設定するには、[複数のインスタンスをサポートする]を選択します。この選択を行うと、Dedicated Hostにおいて、同一インスタンスマルチアーフィールド内の異なるインスタンスタイプを起動できるようになります。たとえば、m5インスタンスマルチアーフィールドとこのオプションを選択すると、Dedicated Hostにおいてm5.xlargeインスタンスとm5.4xlargeインスタンスを起動できます。複数のインスタンスタイプをサポートするように設定できるインスタンスマルチアーフィールドは、A1, C5, C5n, M5, M5n, R5, and R5nです。
  - 選択したインスタンスマルチアーフィールド内にある特定のインスタンスのみサポートするようにDedicated Hostを設定するには、[複数のインスタンスをサポートする]をクリアしてから、[インスタンスタイプ]について、サポートするインスタンスタイプを選択します。この選択を行うと、Dedicated Hostにおいて単一のインスタンスタイプを起動できるようになります。たとえば、このオプションを選択し、m5.4xlargeをサポート対象インスタンスタイプとして指定すると、Dedicated Hostにおいてはm5.4xlargeインスタンスに限り起動できます。
5. [アベイラビリティーゾーン]については、Dedicated Hostを割り当てるアベイラビリティーゾーンを選択します。
6. インスタンスタイプが一致する、ターゲットを絞らないインスタンスの起動を受け入れることをDedicated Hostに許可するには、[Instance auto-placement (インスタンスの自動プレイスメント)]で、[Enable (有効)]を選択します。自動配置の詳細については、「[自動配置とアフィニティについて \(p. 404\)](#)」を参照してください。
7. Dedicated Hostのホスト復旧を有効にするには、[Host recovery (ホスト復旧)]、[Enable (有効)]の順に選択します。詳細については、「[ホスト復旧 \(p. 420\)](#)」を参照してください。
8. [数量]については、割り当てるDedicated Hostsの数を入力します。
9. (オプション) [Add Tag (タグの追加)]を選択し、タグキーとタグ値を入力します。
10. [ホストの割り当て]を選択します。

## AWS CLI

Dedicated Host を割り当てるには

`allocate-hosts` AWS CLI コマンドを使用します。次のコマンドでは、`us-east-1a` アベイラビリティゾーンで `m5` インスタンスファミリー内の複数のインスタンスタイプをサポートしている Dedicated Host を割り当てます。ホストでもホスト復旧が有効になっており、自動配置は無効になっています。

```
aws ec2 allocate-hosts --instance-family "m5" --availability-zone "us-east-1a" --auto-placement "off" --host-recovery "on" --quantity 1
```

次のコマンドでは、`eu-west-1a` アベイラビリティゾーンでターゲット未指定の `m4.large` インスタンス起動をサポートする Dedicated Host を割り当て、ホスト復旧を有効にして、`purpose` のキーと `production` の値を使用してタグを適用します。

```
aws ec2 allocate-hosts --instance-type "m4.large" --availability-zone "eu-west-1a" --auto-placement "on" --host-recovery "on" --quantity 1 --tag-specifications 'ResourceType=dedicated-host,Tags=[{Key=purpose,Value=production}]'
```

## PowerShell

Dedicated Host を割り当てるには

`New-EC2Host` AWS Tools for Windows PowerShell コマンドを使用します。次のコマンドでは、`us-east-1a` アベイラビリティゾーンで `m5` インスタンスファミリー内の複数のインスタンスタイプをサポートしている Dedicated Host を割り当てます。ホストでもホスト復旧が有効になっており、自動配置は無効になっています。

```
PS C:\> New-EC2Host -InstanceFamily m5 -AvailabilityZone us-east-1a -AutoPlacement Off -HostRecovery On -Quantity 1
```

次のコマンドでは、`eu-west-1a` アベイラビリティゾーンで ターゲットを絞らない `m4.large` インスタンスの起動をサポートする Dedicated Host を割り当て、ホスト復旧を有効にして、`purpose` のキーと `production` の値を使用してタグを適用します。

作成時に Dedicated Host にタグを付けるために使用される `TagSpecification` パラメータには、タグ付けされるリソースのタイプ、タグキー、タグ値を指定するオブジェクトが必要です。次のコマンドは必要なオブジェクトを作成します。

```
PS C:\> $tag = @{ Key="purpose"; Value="production" }
PS C:\> $tagspec = new-object Amazon.EC2.Model.TagSpecification
PS C:\> $tagspec.ResourceType = "dedicated-host"
PS C:\> $tagspec.Tags.Add($tag)
```

次のコマンドは Dedicated Host を割り当て、`$tagspec` オブジェクトで指定されたタグを適用します。

```
PS C:\> New-EC2Host -InstanceType m4.large -AvailabilityZone eu-west-1a -AutoPlacement On -HostRecovery On -Quantity 1 -TagSpecification $tagspec
```

## Dedicated Host にインスタンスを作成する

Dedicated Host を割り当てたら、そのホストにインスタンスを起動できます。起動するインスタンスタイプに使用できる十分な容量を持つアクティブな Dedicated Hostsがない場合には、host テナントでインスタンスを起動できません。

#### Note

Dedicated Hosts に起動されるインスタンスは、VPC でのみ起動できます。詳細については、「[Amazon VPC とは](#)」を参照してください。

インスタンスを起動する前に、制限事項を確認してください。詳細については、「[Dedicated Hosts に関する制限事項 \(p. 397\)](#)」を参照してください。

次の方法を使用して Dedicated Host でインスタンスを起動できます。

#### Console

Dedicated Hosts ページから特定の Dedicated Host でインスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Dedicated Hosts] を選択します。
3. [Dedicated Hosts] ページでホストを選択し、[アクション]、[ホストでインスタンスを起動] の順に選択します。
4. リストから AMI を選択します。Amazon EC2 によって提供されている Windows、SUSE、RHEL AMI を Dedicated Hosts で使用することはできません。
5. [インスタンスタイプの選択] ページで、起動するインスタンスタイプを選択し、[次の手順: インスタンス詳細の設定] を選択します。

Dedicated Hostが単一のインスタンスタイプのみサポートしている場合、デフォルトでは、サポートされているインスタンスタイプが選択され、変更できません。

Dedicated Host が複数のインスタンスタイプをサポートしている場合は、Dedicated Host の使用可能なインスタンスキャパシティに基づいて、サポートされているインスタンスマトリクス内のインスタンスタイプを選択する必要があります。大きいインスタンスサイズから始め、必要に応じて残りのインスタンス容量を小さいインスタンスサイズで埋めることをお勧めします。

6. [Configure Instance Details (インスタンス詳細の設定)] ページで、必要に合うインスタンス設定を設定し、[アフィニティ] で以下のいずれかのオプションを選択します。
  - Off — インスタンスは指定されたホストで起動されますが、停止されると同じ Dedicated Host で再開されない場合があります。
  - Host — 停止したインスタンスは、常にその特定のホストで再開されます。

アフィニティの詳細については、「[自動配置とアフィニティについて \(p. 404\)](#)」を参照してください。

[Tenancy (テナント)] オプションと [Host (ホスト)] オプションは、選択したホストに基づき、事前に設定されています。

7. [Review and Launch (確認と起動)] を選択します。
8. [Review Instance Launch] ページで、[Launch] を選択します。
9. プロンプトが表示されたら、既存のキーペアを選択するか、新しいキーペアを作成し、[Launch Instances (インスタンスの起動)] を選択します。

インスタンス起動ウィザードを使用してインスタンスを Dedicated Host で起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス]、[インスタンスの起動] の順に選択します。
3. リストから AMI を選択します。Amazon EC2 によって提供されている Windows、SUSE、RHEL AMI を Dedicated Hosts で使用することはできません。
4. 起動するインスタンスのタイプを選択し、[Next: Configure Instance Details (次へ: インスタンス詳細の設定)] を選択します。

5. [インスタンスの詳細の設定] ページで、ニーズに合うインスタンス設定を構成し、さらに Dedicated Host 固有の以下の設定を構成します。
  - [Tenancy (テナンシー)] — [Dedicated Host - Launch this instance on a Dedicated Host] を選択します。
  - [Host (ホスト)] — [Use auto-placement (自動配置の使用)] を選択して自動配置が有効な Dedicated Host でインスタンスを起動するか、リストで特定の Dedicated Host を選択します。一覧には、選択されたインスタンスタイプをサポートしている Dedicated Hostsだけが表示されています。
  - [Affinity (アフィニティ)] — 次のオプションのいずれかを選択します。
    - Off — インスタンスは指定されたホストで起動されますが、停止されると同じホストで再開されない場合があります。
    - Host — 停止したインスタンスは、常に指定されたホストで再開されます。

詳細については、「[自動配置とアフィニティについて \(p. 404\)](#)」を参照してください。

これらの設定が表示されない場合は、[Network] メニューで VPC を選択したことを確認してください。

6. [Review and Launch (確認と起動)] を選択します。
7. [Review Instance Launch] ページで、[Launch] を選択します。
8. プロンプトが表示されたら、既存のキーペアを選択するか、新しいキーペアを作成し、[Launch Instances (インスタンスの起動)] を選択します。

#### AWS CLI

Dedicated Host でインスタンスを起動するには

`run-instances` AWS CLI コマンドを使用し、Placement リクエストパラメータでインスタンスのアフィニティ、テナンシー、およびホストを指定します。

#### PowerShell

Dedicated Host でインスタンスを起動するには

`New-EC2Instance` AWS Tools for Windows PowerShell コマンドを使用し、Placement リクエストパラメータでインスタンスのアフィニティ、テナンシー、およびホストを指定します。

## ホストリソースグループにインスタンスを作成する

インスタンスの作成先のホストリソースグループ内のいずれかの Dedicated Host にインスタンス用の空き容量がある場合、Amazon EC2 はそのホストにインスタンスを作成します。ホストリソースグループ内のいずれのホストにもインスタンス用の空き容量がない場合、Amazon EC2 はホストリソースグループ内に新しいホストを自動的に割り当て、そのホストにインスタンスを作成します。詳細については、AWS License Manager ユーザーガイドの「[ホストリソースグループ](#)」を参照してください。

次の方法を使用して、ホストリソースグループにインスタンスを起動できます。

#### Console

ホストリソースグループにインスタンスを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス]、[インスタンスの起動] の順に選択します。
3. リストから AMI を選択します。Amazon EC2 によって提供されている Windows、SUSE、RHEL AMI を Dedicated Hosts で使用することはできません。

4. 起動するインスタンスのタイプを選択し、[Next: Configure Instance Details (次へ: インスタンス詳細の設定)] を選択します。
5. [インスタンスの詳細の設定] ページで、必要に応じたインスタンス設定を行い、次の操作を実行します。
  - a. [テナント] で、[Dedicated Host] を選択します。
  - b. [Host resource group (ホストリソースグループ)] で、[Launch instance into a host resource group (ホストリソースグループにインスタンスを作成する)] を選択します。
  - c. [Host resource group name (ホストリソースグループ名)] で、インスタンスの作成先のホストリソースグループを選択します。
6. ホスト ID を選択して特定のホストをターゲットにすることはできません。また、ホストリソースグループにインスタンスを作成するときに、インスタンスのアフィニティを有効にすることはできません。
7. [Review and Launch (確認と起動)] を選択します。
8. [Review Instance Launch] ページで、[Launch] を選択します。
9. プロンプトが表示されたら、既存のキーペアを選択するか、新しいキーペアを作成し、[Launch Instances (インスタンスの起動)] を選択します。

#### AWS CLI

ホストリソースグループにインスタンスを作成するには

`run-instances` AWS CLI コマンドを使用し、`Placement` リクエストパラメータでテナントオプションを省略してホストリソースグループ ARN を指定します。

#### PowerShell

ホストリソースグループにインスタンスを作成するには

`New-EC2Instance` AWS Tools for Windows PowerShell コマンドを使用し、`Placement` リクエストパラメータでテナントオプションを省略してホストリソースグループ ARN を指定します。

## 自動配置とアフィニティについて

Dedicated Hosts の配置制御は、インスタンスおよびホストの両レベルで行われます。

### 自動配置

自動配置はホストレベルで設定されます。自動配置を使用すると、起動するインスタンスについて、特定のホストで起動されるようにするか、設定が合致する任意のホストで起動されるようにするかを管理できます。

Dedicated Host の自動配置が無効になっている場合は、一意のホスト ID を指定するホストテナントインスタンス起動のみが受け入れられます。これは、新しい Dedicated Hosts に対する既定の設定です。

Dedicated Host の自動配置が有効になっている場合は、インスタンスタイプ設定が一致するすべてのターゲット未指定のインスタンス起動が受け入れられます。

インスタンスの起動時に、テナントを設定する必要があります。特定の `HostId` を指定せずに Dedicated Host でインスタンスを起動すると、自動配置が有効で、インスタンスタイプが一致するすべての Dedicated Host でインスタンスを起動できます。

### ホストのアフィニティ

ホストアフィニティは、インスタンスレベルで設定されます。また、インスタンスと Dedicated Host の間に関係を作成します。

アフィニティが Host に設定されている場合は、特定のホストで起動したインスタンスが停止しても、常に同じホストで再開されます。これは、ターゲットを絞った起動にもターゲットを絞らない起動にも適用されます。

アフィニティが off に設定されているときにインスタンスを停止して再起動する場合は、使用可能な任意のホスト上で再起動できます。ただし、最後に実行した Dedicated Host 上でベストエフォートベースでの再起動を試みます。

## Dedicated Host 自動配置の変更

Dedicated Host の自動配置設定は、AWS アカウントへの割り当て後に、次のいずれかの方法を使用して変更できます。

### 新しいコンソール

Dedicated Host の自動配置を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. ホストを選択し、[アクション]、[ホストの変更] の順に選択します。
4. [インスタンスの自動配置] で、[有効化] を選択して自動配置を有効にするか、[有効化] をオフにして自動配置を無効にします。詳細については、「[自動配置とアフィニティについて \(p. 404\)](#)」を参照してください。
5. [Save] を選択します。

### 古いコンソール

Dedicated Host の自動配置を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Dedicated Hosts] を選択します。
3. [Dedicated Hosts] ページで、ホストを選択し、[Actions (アクション)] を選択して、[Modify Auto-Placement (自動配置の変更)] を選択します。
4. [Modify Auto-placement] ウィンドウの [Allow instance auto-placement] で、[Yes] を選択して自動配置を有効にするか、[No] を選択して自動配置を無効にします。詳細については、「[自動配置とアフィニティについて \(p. 404\)](#)」を参照してください。
5. [Save] を選択します。

### AWS CLI

Dedicated Host の自動配置を変更するには

`modify-hosts` AWS CLI コマンドを使用します。次の例では、指定した Dedicated Host の自動配置を有効にします。

```
aws ec2 modify-hosts --auto-placement on --host-ids h-012a3456b7890cdef
```

### PowerShell

Dedicated Host の自動配置を変更するには

`Edit-EC2Host` AWS Tools for Windows PowerShell コマンドを使用します。次の例では、指定した Dedicated Host の自動配置を有効にします。

```
PS C:\> Edit-EC2Host --AutoPlacement 1 --HostId h-012a3456b7890cdef
```

## サポート対象インスタンスタイプの修正

お客様は、Dedicated Hostを修正することで、このホストがサポートするインスタンスタイプを変更できます。単一のインスタンスタイプのみサポートしているホストの場合には、インスタンスマリーニー内にある複数のインスタンスタイプをサポートするように修正できます。同様に、複数のインスタンスタイプをサポートしているホストの場合には、単一のインスタンスタイプのみをサポートするように修正できます。

複数のインスタンスタイプをサポートするようにDedicated Hostを修正するには、初めに、該当ホスト上で実行中のインスタンスをすべて停止する必要があります。修正は約 10 分で完了します。修正の実行中には、Dedicated Hostがpending状態に移行します。pending状態にあるDedicated Hostにおいて停止中のインスタンスを開始したり、新たなインスタンスを起動したりすることはできません。複数のインスタンスタイプをサポートするように修正できるインスタンスマリーニーは、A1, C5, C5n, M5, M5n, R5, and R5nです。

複数のインスタンスタイプをサポートしているDedicated Hostを、特定のインスタンスタイプのみをサポートするように修正するには、当該ホストがいかなるインスタンスも実行していない状態であるか、実行中のインスタンスが当該ホストにサポートさせたいインスタンスタイプである状態でなければなりません。具体例を挙げると、m5インスタンスマリーニー内にある複数のインスタンスタイプをサポートしているホストを、m5.largeインスタンスのみをサポートするように修正するには、Dedicated Hostが、いかなるインスタンスも実行していない状態であるか、m5.large実行中のインスタンスのみの状態でなければなりません。

サポートされているインスタンスタイプは、次のいずれかの方法で変更できます。

### 新しいコンソール

Dedicated Host のサポートされているインスタンスタイプを変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Host] を選択します。
3. 変更する Dedicated Host を選択し、[アクション]、[ホストの変更] の順に選択します。
4. Dedicated Host の現在の設定に応じて、次のいずれかの操作を実行します。
  - Dedicated Host が特定のインスタンスタイプを現在サポートしている場合は、[複数のインスタンスタイプをサポートする] は有効にならず、現在サポートされているインスタンスタイプが [インスタンスタイプ] に表示されます。現在のインスタンスマリーニー内にある複数のタイプをサポートするようにホストを変更するには、[複数のインスタンスタイプをサポートする] の [有効化] を選択します。

複数のインスタンスタイプをサポートするようにホストを修正するには、初めに、該当ホスト上で実行されているすべてのインスタンスを停止する必要があります。

- Dedicated Host がインスタンスマリーニー内の複数のインスタンスタイプを現在サポートしている場合は、[複数のインスタンスタイプをサポートする] の [有効] が選択されています。特定のインスタンスタイプをサポートするようにホストを変更するには、[複数のインスタンスタイプをサポートする] で、[有効化] をオフにし、[インスタンスタイプ] で、サポートする特定のインスタンスタイプを選択します。

Dedicated Host がサポートするインスタンスマリーニーを変更することはできません。

5. [Save] を選択します。

### 古いコンソール

Dedicated Host のサポートされているインスタンスタイプを変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで [Dedicated Host] を選択します。
  3. 修正する Dedicated Host を選択して、[アクション]、[サポート対象インスタンスタイプの修正] の順に選択します。.
  4. Dedicated Host の現在の設定に応じて、次のいずれかの操作を実行します。
    - Dedicated Host が特定のインスタンスタイプのみサポートしている場合には、[複数インスタンスタイプのサポート] の選択が [いいえ] になっています。インスタンスマリ内にある複数のインスタンスタイプをサポートするようにホストを修正するには、[複数インスタンスタイプのサポート] の選択を [はい] に変更します。
- 複数のインスタンスタイプをサポートするようにホストを修正するには、初めに、該当ホスト上で実行されているすべてのインスタンスを停止する必要があります。
- Dedicated Host が、インスタンスマリ内にある複数のインスタンスタイプをサポートしている場合には、[複数インスタンスタイプのサポート] の選択が [はい] になっており、[インスタンスマリ] に、サポート対象インスタンスマリが表示されています。特定のインスタンスタイプをサポートするようにホストを修正するには、[複数インスタンスタイプのサポート] の選択を [いいえ] に変更し、[インスタンスタイプ] において、サポートする特定のインスタンスタイプを選択します。
- Dedicated Host がサポートするインスタンスマリを変更することはできません。
5. [Save] を選択します。

#### AWS CLI

Dedicated Host のサポートされているインスタンスタイプを変更するには

[modify-hosts](#) AWS CLI コマンドを使用します。

以下のコマンドは、m5 インスタンスマリ内にある複数のインスタンスタイプをサポートするように Dedicated Host を修正します。

```
aws ec2 modify-hosts --instance-family m5 --host-ids h-012a3456b7890cdef
```

以下のコマンドは、m5.xlarge インスタンスのみをサポートするように Dedicated Host を修正します。

```
aws ec2 modify-hosts --instance-type m5.xlarge --instance-family --host-ids h-012a3456b7890cdef
```

#### PowerShell

Dedicated Host のサポートされているインスタンスタイプを変更するには

[Edit-EC2Host](#) AWS Tools for Windows PowerShell コマンドを使用します。

以下のコマンドは、m5 インスタンスマリ内にある複数のインスタンスタイプをサポートするように Dedicated Host を修正します。

```
PS C:\> Edit-EC2Host --InstanceFamily m5 --HostId h-012a3456b7890cdef
```

以下のコマンドは、m5.xlarge インスタンスのみをサポートするように Dedicated Host を修正します。

```
PS C:\> Edit-EC2Host --InstanceType m5.xlarge --HostId h-012a3456b7890cdef
```

## インスタンスのテナンシーおよびアフィニティの変更

インスタンスのテナンシーは、インスタンスの起動後に `dedicated` から `host` に変更したり、`host` から `dedicated` に変更したりできます。インスタンスとホストの間でアフィニティを修正することもできます。インスタンスのテナンシーまたはアフィニティを修正するには、そのインスタンスを `stopped` 状態にする必要があります。

インスタンスのテナンシーとアフィニティは、次の方法を使用して変更できます。

### Console

インスタンスのテナンシーまたはアフィニティを変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [Instances (インスタンス)] を選択し、変更するインスタンスを選択します。
3. [Actions]、[Instance State]、[Stop] の順に選択します。
4. インスタンスのコンテキスト(右クリック)メニューを開き、[Instance Settings]、[Modify Instance Placement] の順に選択します。
5. [Modify Instance Placement] ページで、次の設定を行います。
  - [Tenancy] — 次のいずれかを選択します。
    - [専用ハードウェアインスタンスの実行] — インスタンスを ハードウェア専有インスタンスとして起動します。詳細については、「[ハードウェア専有インスタンス \(p. 425\)](#)」を参照してください。
    - [Launch the instance on a Dedicated Host] — 設定可能なアフィニティを使用してインスタンスを Dedicated Host で起動します。
  - [Affinity] — 次のいずれかを選択します。
    - [This instance can run on any one of my hosts] — インスタンスは、そのインスタンスタイプをサポートするアカウントの利用可能な Dedicated Host で起動されます。
    - [This instance can only run on the selected host] — インスタンスは、[Target Host (ターゲットホスト)] として選択された Dedicated Host でのみ実行できます。
  - [Target Host] — インスタンスを実行する必要がある Dedicated Host を選択します。ターゲットホストが表示されない場合は、アカウントに利用可能な、互換性のある Dedicated Hosts がない可能性があります。

詳細については、「[自動配置とアフィニティについて \(p. 404\)](#)」を参照してください。

6. [Save] を選択します。

### AWS CLI

インスタンスのテナンシーまたはアフィニティを変更するには

`modify-instance-placement` AWS CLI コマンドを使用します。次の例では、指定したインスタンスのアフィニティを `default` から `host` に変更し、インスタンスがアフィニティを持つ対象の Dedicated Host を指定します。

```
aws ec2 modify-instance-placement --instance-id i-1234567890abcdef0 --affinity host --host-id h-012a3456b7890cdef
```

### PowerShell

インスタンスのテナンシーまたはアフィニティを変更するには

**Edit-EC2InstancePlacement** AWS Tools for Windows PowerShell コマンドを使用します。次の例では、指定したインスタンスのアフィニティを `default` から `host` に変更し、インスタンスがアフィニティを持つ対象の Dedicated Host を指定します。

```
PS C:\> Edit-EC2InstancePlacement -InstanceId i-1234567890abcdef0 -Affinity host -  
HostId h-012a3456b7890cdef
```

## Dedicated Hosts の表示

Dedicated Host およびその各インスタンスの詳細を表示するには、次の方法を使用できます。

### 新しいコンソール

Dedicated Host の詳細を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. [Dedicated Hosts] ページでホストを選択します。
4. ホストの情報を表示するには、[詳細] を選択します。

[使用可能な vCPU] は、Dedicated Host における新たなインスタンスの起動に使用できる vCPU を示します。具体例を挙げると、c5 インスタンスファミリー内にある複数のインスタンスタイプをサポートしている Dedicated Host の場合、いかなるインスタンスも実行されなければ 72 の vCPU を使用できます。これは、Dedicated Hostにおいては 72 の使用可能な vCPU を使って異なるインスタンスタイプの組合せを起動できることを意味します。

ホストで実行中のインスタンスの情報を表示するには、[実行中のインスタンス] を選択します。

### 古いコンソール

Dedicated Host の詳細を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. [Dedicated Hosts] ページでホストを選択します。
4. ホストの情報を表示するには、[Description] を選択します。[使用可能な vCPU] は、Dedicated Host における新たなインスタンスの起動に使用できる vCPU を示します。具体例を挙げると、c5 インスタンスファミリー内にある複数のインスタンスタイプをサポートしている Dedicated Host の場合、いかなるインスタンスも実行されなければ 72 の vCPU を使用できます。これは、Dedicated Hostにおいては 72 の使用可能な vCPU を使って異なるインスタンスタイプの組合せを起動できることを意味します。

ホストで実行中のインスタンスの情報を表示するには、[Instances] を選択します。

### AWS CLI

Dedicated Host の容量を表示するには

**describe-hosts** AWS CLI コマンドを使用します。

次の例では、c5 インスタンスファミリー内の複数のインスタンスタイプをサポートしている Dedicated Host の使用可能なインスタンス容量を表示するために、**describe-hosts** (AWS CLI) コマンドを使用しています。この Dedicated Hostにおいては、すでに 2 つの c5.4xlarge インスタンスと 4 つの c5.2xlarge インスタンスが実行されています。

```
$ aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

```
"AvailableInstanceCapacity": [  
    { "AvailableCapacity": 2,  
      "InstanceType": "c5.xlarge",  
      "TotalCapacity": 18 },  
    { "AvailableCapacity": 4,  
      "InstanceType": "c5.large",  
      "TotalCapacity": 36 }  
],  
"AvailableVCpus": 8
```

#### PowerShell

Dedicated Host のインスタンス容量を表示するには

[Get-EC2Host](#) AWS Tools for Windows PowerShell コマンドを使用します。

```
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

## Dedicated Host のタグ付け

既存の Dedicated Host にカスタムタグを割り当てて、目的、所有者、環境など、さまざまな方法で分類できます。これにより、割り当てたカスタムタグに基づいて特定の Dedicated Host をすばやく見つけることができます。Dedicated Host タグは、コスト配分の追跡にも使用できます。

作成時に Dedicated Hosts にタグを適用することもできます。詳細については、「[Dedicated Hosts の割り当て \(p. 399\)](#)」を参照してください。

Dedicated Host にタグを付けるには、次の方法を使用できます。

### 新しいコンソール

Dedicated Host にタグを付けるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. タグを付ける対象の Dedicated Host を選択し、[アクション]、[タグの管理] の順に選択します。
4. [タグの管理] 画面で、[タグの追加] を選択し、タグのキーと値を指定します。
5. (オプション) [タグの追加] を選択して、Dedicated Host に付けるタグを追加します。
6. [Save changes] を選択します。

### 古いコンソール

Dedicated Host にタグを付けるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. タグを追加する Dedicated Host を選択して、[Tags (タグ)] を選択します。
4. [Add/Edit Tags (タグの追加/編集)] を選択します。
5. [Add/Edit Tags (タグの追加/編集)] ダイアログボックスで、[Create Tag (タグの作成)] を選択してから、各タグのキーと値を指定します。
6. (オプション) [Create Tag (タグの作成)] を選択して Dedicated Host に追加のタグを追加します。
7. [Save] を選択します。

## AWS CLI

Dedicated Host にタグを付けるには

[create-tags](#) AWS CLI コマンドを使用します。

次のコマンドでは、指定した Dedicated Host に Owner=TeamA のタグを付けます。

```
aws ec2 create-tags --resources h-abc12345678909876 --tags Key=Owner,Value=TeamA
```

## PowerShell

Dedicated Host にタグを付けるには

[New-EC2Tag](#) AWS Tools for Windows PowerShell コマンドを使用します。

New-EC2Tag コマンドには、Dedicated Host のタグに使用するキーと値のペアを指定する Tag オブジェクトが必要です。以下のコマンドでは、キーと値のペアとして Owner と TeamA を使用し、\$tag という名前の Tag オブジェクトを作成します。

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

次のコマンドでは、指定した Dedicated Host に \$tag オブジェクトをタグ付けします。

```
PS C:\> New-EC2Tag -Resource h-abc12345678909876 -Tag $tag
```

## Dedicated Hosts のモニタリング

Amazon EC2 は、Dedicated Hosts の状態を絶えずモニタリングします。更新は Amazon EC2 コンソールで伝達されます。Dedicated Host に関する情報は、次の方法を使用して表示できます。

### Console

Dedicated Host の状態を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. リストで Dedicated Host を見つけ、[State] 列で値を確認します。

## AWS CLI

Dedicated Host の状態を表示するには

[describe-hosts](#) AWS CLI コマンドを使用して、hostSet レスポンス要素の state プロパティを確認します。

```
aws ec2 describe-hosts --host-id h-012a3456b7890cdef
```

## PowerShell

Dedicated Host の状態を表示するには

[Get-EC2Host](#) AWS Tools for Windows PowerShell コマンドを使用し、hostSet レスポンス要素の state プロパティを確認します。

```
PS C:\> Get-EC2Host -HostId h-012a3456b7890cdef
```

以下の表では、表示される可能性のある Dedicated Host の状態について説明します。

状態	説明
available	AWS は Dedicated Host で問題を検出しませんでした。予定されているメンテナンスまたは修復はありません。この専有ホストでインスタンスを起動できます。
released	Dedicated Host が解放されました。ホスト ID は使用中ではありません。解放済みのホストは再使用できません。
under-assessment	AWS は Dedicated Host の潜在的な問題を調査しています。アクションを実行する必要がある場合は、AWS マネジメントコンソールまたは E メールで通知されます。この状態の Dedicated Host ではインスタンスを起動できません。
pending	この状態の Dedicated Host は新たなインスタンスの起動に使用できません。この状態は、複数のインスタンスタイプをサポートするように修正されている (p. 406) 状態か、ホスト復旧 (p. 420) 実行中の状態です。
permanent-failure	回復不可能な障害が検出されました。インスタンスおよび E メールで削除通知が届きます。インスタンスは引き続き実行する場合があります。この状態の Dedicated Host のすべてのインスタンスを停止または終了すると、AWS はそのホストを廃止します。AWS ではこの状態のインスタンスは再起動されません。この状態の Dedicated Hosts ではインスタンスを起動できません。
released-permanent-failure	AWS は、失敗してインスタンスが実行されていない Dedicated Hosts を完全に解放します。Dedicated Host ID も使用できなくなります。

## Dedicated Host の解放

ホストを解放する前に、Dedicated Host で実行中のインスタンスを停止する必要があります。これらのインスタンスはアカウントの他の Dedicated Hosts に移行し、引き続き使用することができます。これらのステップは、オンデマンド Dedicated Hosts にのみ適用されます。

Dedicated Host を解放するには、次の方法を使用できます。

### 新しいコンソール

#### Dedicated Host を解放するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. [Dedicated Hosts] ページで解放する Dedicated Host を選択します。
4. [アクション]、[ホストの解放] の順に選択します。
5. [解放] を選択して確定します。

### 古いコンソール

#### Dedicated Host を解放するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで、[Dedicated Hosts] を選択します。
3. [Dedicated Hosts] ページで解放する Dedicated Host を選択します。
4. [Actions]、[Release Hosts] の順に選択します。
5. [Release] を選択して確定します。

#### AWS CLI

Dedicated Host を解放するには

[release-hosts](#) AWS CLI コマンドを使用します。

```
aws ec2 release-hosts --host-ids h-012a3456b7890cdef
```

#### PowerShell

Dedicated Host を解放するには

[Remove-EC2Hosts](#) AWS Tools for Windows PowerShell コマンドを使用します。

```
PS C:\> Remove-EC2Hosts -HostId h-012a3456b7890cdef
```

Dedicated Hostを解放すると、同じホストまたはホスト ID を再使用できなくなり、該当ホストのオンデマンド料金請求が停止します。Dedicated Host の状態は `released` に変わり、このホストではインスタンスを起動できなくなります。

#### Note

最近 Dedicated Hosts を解放した場合、制限に加算されなくなるまでに少し時間がかかることがあります。それまでは、新しい Dedicated Hosts を割り当てようすると `LimitExceeded` エラーが発生する場合があります。このエラーが発生した場合は、数分後に新しいホストを再び割り当ててみてください。

停止したインスタンスはまだ使用可能であり、[Instances] ページに表示されます。その [host] テナント設定も維持されています。

## Dedicated Host の予約 の購入

予約は、次の方法で購入できます。

#### Console

予約を購入するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [Dedicated Hosts]、[Dedicated Host の予約]、[Dedicated Host の予約 の購入] の順に選択します。
3. [Dedicated Host の予約 の購入] 画面で、デフォルト設定を使用して利用可能なオファーリングを検索するか、以下のカスタム値を指定することができます。
  - [Host instance family] — 表示されるオプションは、アカウントで予約に割り当てられていない Dedicated Hosts に対応します。
  - [Availability Zone] — アカウントで予約に割り当てられていない Dedicated Hosts のアベイラビリティーゾーン。

- ・ [Payment Option] — オファーの支払いオプション。
  - ・ [Term] — 予約の期間。1 年または 3 年とすることができます。
4. [Find offering (提供タイプの検索)] を選択し、要件に一致するオファーを選択します。
  5. 予約と関連付ける Dedicated Hosts を選択し、[Review (確認)] を選択します。
  6. 注文を確認し、[Order (注文)] を選択します。

## AWS CLI

### 予約を購入するには

1. [describe-host-reservation-offerings](#) AWS CLI コマンドを使用して、ニーズに合った利用可能なオファリングを一覧表示します。次の例では、m4 インスタンスファミリー内のインスタンスをサポートし、契約期間が 1 年のオファリングを一覧表示します。

#### Note

期間は秒単位で指定されます。1 年契約は 31,536,000 秒で、3 年契約は 94,608,000 秒です。

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4  
--max-duration 31536000
```

コマンドは、条件に合ったオファリングのリストを返します。購入するオファリングの offeringId を書き留めます。

2. [purchase-host-reservation](#) AWS CLI コマンドを使用してオファリングを購入し、前のステップで書き留めた offeringId を指定します。次の例では、指定した予約を購入し、それを AWS アカウントに割り当て済みの特定の Dedicated Host と関連付けます。

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-  
set h-013abcd2a00cbd123
```

## PowerShell

### 予約を購入するには

1. [Get-EC2HostReservationOffering](#) AWS Tools for Windows PowerShell コマンドを使用して、ニーズに合った利用可能なオファリングを一覧表示します。以下の例では、m4 インスタンスファミリーでインスタンスをサポートし、1 年契約を持っているオファーをリストします。

#### Note

期間は秒単位で指定されます。1 年契約は 31,536,000 秒で、3 年契約は 94,608,000 秒です。

```
PS C:\> $filter = @{Name="instance-family"; Value="m4"}
```

```
PS C:\> Get-EC2HostReservationOffering -filter $filter -MaxDuration 31536000
```

コマンドは、条件に合ったオファリングのリストを返します。購入するオファーの offeringId を書き留めます。

2. [New-EC2HostReservation](#) AWS Tools for Windows PowerShell コマンドを使用してオファリングを購入し、前のステップで書き留めた offeringId を指定します。次の例では、指定した予約を購入し、それを AWS アカウントに割り当て済みの特定の Dedicated Host と関連付けます。

```
PS C:\> New-EC2HostReservation -OfferingId hro-03f707bf363b6b324 -  
HostIdSet h-013abcd2a00cbd123
```

## Dedicated Host の予約の表示

予約に関連する Dedicated Hosts の情報として以下を表示できます。

- 予約の期間
- 支払いオプション
- 開始日と終了日

Dedicated Host 予約の詳細は、次の方法で表示できます。

Console

Dedicated Host 予約の詳細を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Dedicated Hosts] を選択します。
3. [Dedicated Hosts] ページで、[Dedicated Host の予約] を選択し、表示されるリストから予約を選択します。
4. 予約の詳細については、[詳細] を選択します。
5. 予約が関連付けられている Dedicated Hosts に関する情報については、[Hosts (ホスト)] を選択します。

AWS CLI

Dedicated Host 予約の詳細を表示するには

[describe-host-reservations](#) AWS CLI コマンドを使用します。

```
aws ec2 describe-host-reservations
```

PowerShell

Dedicated Host 予約の詳細を表示するには

[Get-EC2HostReservation](#) AWS Tools for Windows PowerShell コマンドを使用します。

```
PS C:\> Get-EC2HostReservation
```

## Dedicated Host の予約 のタグ付け

Dedicated Host の予約にカスタムタグを割り当てて、目的、所有者、環境など、さまざまな方法で分類できます。これにより、割り当てたカスタムタグに基づいて特定の Dedicated Host の予約をすばやく見つけることができます。

Dedicated Host の予約にタグを付けるには、コマンドラインツールのみを使用できます。

AWS CLI

Dedicated Host の予約にタグを付けるには

[create-tags](#) AWS CLI コマンドを使用します。

```
aws ec2 create-tags --resources hr-1234563a4ffc669ae --tags Key=Owner,Value=TeamA
```

PowerShell

Dedicated Host の予約にタグを付けるには

[New-EC2Tag](#) AWS Tools for Windows PowerShell コマンドを使用します。

New-EC2Tag コマンドには、Dedicated Host の予約のタグに使用するキーと値のペアを指定する Tag パラメータが必要です。以下のコマンドでは、Tag パラメータを作成します。

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag  
PS C:\> $tag.Key = "Owner"  
PS C:\> $tag.Value = "TeamA"
```

```
PS C:\> New-EC2Tag -Resource hr-1234563a4ffc669ae -Tag $tag
```

## 共有 Dedicated Hosts の使用

Dedicated Host 共有を使用すると、Dedicated Host 所有者は Dedicated Hosts を他の AWS アカウントと共有したり、AWS 組織内で共有したりできます。これにより、Dedicated Hosts の作成と管理を一元的に行い、Dedicated Host を複数の AWS アカウント間で共有したり、AWS 組織内で共有したりできます。

このモデルでは、Dedicated Host を所有する AWS アカウント(所有者)が、他の AWS アカウント(コンシューマー)との共有を行います。コンシューマーは、各自のアカウントに割り当てた Dedicated Hosts にインスタンスを作成する場合と同じように、共有している Dedicated Hosts にインスタンスを作成できます。所有者は、Dedicated Host およびそこに作成したインスタンスの管理に責任を負います。所有者は、コンシューマーが共有 Dedicated Hosts に作成したインスタンスを変更することはできません。コンシューマーは、自己が共有している Dedicated Hosts に作成したインスタンスの管理に責任を負います。コンシューマーは、他のコンシューマーまたは Dedicated Host 所有者が所有するインスタンスを表示または変更することはできません。また、自己が共有している Dedicated Hosts を変更することはできません。

Dedicated Host 所有者が Dedicated Host を共有できる相手は次のとおりです。

- AWS 組織内または組織外の特定の AWS アカウント
- AWS 組織内の組織単位
- AWS 組織全体

### コンテンツ

- [Dedicated Hosts を共有するための前提条件 \(p. 417\)](#)
- [関連サービス \(p. 417\)](#)
- [アベイラビリティーゾーン間の共有 \(p. 417\)](#)
- [Dedicated Host の共有 \(p. 417\)](#)
- [共有 Dedicated Host の共有解除 \(p. 418\)](#)
- [共有 Dedicated Host の特定 \(p. 418\)](#)
- [共有専有ホストで実行されているインスタンスの表示 \(p. 419\)](#)
- [共有 Dedicated Host のアクセス許可 \(p. 419\)](#)
- [請求と使用量測定 \(p. 419\)](#)

- Dedicated Host の制限 (p. 419)
- ホストの復旧と Dedicated Host の共有 (p. 420)

## Dedicated Hosts を共有するための前提条件

- Dedicated Host を共有するには、それを AWS アカウント内で所有している必要があります。既に共有している Dedicated Host を共有することはできません。
- AWS 組織や AWS 組織内の組織単位と Dedicated Host を共有するには、AWS Organizations との共有を有効にする必要があります。詳細については、AWS RAM ユーザーガイドの「[Enable Sharing with AWS Organizations](#)」を参照してください。

## 関連サービス

### AWS Resource Access Manager

Dedicated Host の共有は AWS Resource Access Manager (AWS RAM) と連携します。AWS RAM は、AWS リソースを任意の AWS アカウントと共有したり、AWS Organizations を介して共有したりするためのサービスです。AWS RAM を使用すると、リソース共有を作成することで、自身が所有するリソースを共有できます。リソース共有では、共有対象のリソースと、共有先となるコンシューマーを指定します。コンシューマーには、個人の AWS アカウントや、AWS Organizations 内の組織単位または組織全体を指定できます。

AWS RAM の詳細については、「[AWS RAM ユーザーガイド](#)」を参照してください。

### アベイラビリティーゾーン間の共有

リソースがリージョンの複数のアベイラビリティーゾーンに分散されるようにするために、アベイラビリティーゾーンは各アカウントの名前に個別にマッピングされます。このため、アカウントが異なると、アベイラビリティーゾーンの命名方法が異なる場合があります。たとえば、AWS アカウントのアベイラビリティーゾーン `us-east-1a` の場所は、別の AWS アカウントのアベイラビリティーゾーン `us-east-1a` の場所と異なる可能性があります。

自己のアカウントを基準にして Dedicated Hosts の場所を特定するには、アベイラビリティーゾーン ID (AZ ID) を使用する必要があります。アベイラビリティーゾーン ID は、すべての AWS アカウントにわたって各アベイラビリティーゾーンを一意に識別する ID です。たとえば、`use1-az1` は `us-east-1` リージョンのアベイラビリティーゾーン ID であり、すべての AWS アカウントで同じ場所を示します。

アカウントのアベイラビリティーゾーンのアベイラビリティーゾーン ID を表示するには

1. AWS RAM コンソール (<https://console.aws.amazon.com/ram>) を開きます。
2. 現在のリージョンのアベイラビリティーゾーン ID は、画面の右側にある [お客様の AZ ID] パネルに表示されます。

## Dedicated Host の共有

所有者が Dedicated Host を共有すると、コンシューマーはそのホストにインスタンスを作成できます。コンシューマーは、共有ホストに空き容量がある限り、そこに必要なだけのインスタンスを作成できます。

自動配置を有効にして Dedicated Host を共有する場合は、意図しない形で Dedicated Host が使用されないよう、次の点に注意してください。

- コンシューマーが Dedicated Host テナントを使用してインスタンスを作成する場合、自己のアカウントで所有している Dedicated Host に空き容量がないと、インスタンスは自動的に共有 Dedicated Host に作成されます。

Dedicated Host を共有するには、それをリソース共有に追加する必要があります。リソース共有とは、自身のリソースを AWS アカウント間で共有するための AWS RAM リソースです。リソース共有では、共有対象のリソースと、共有先のコンシューマーを指定します。Dedicated Host は、既存のリソースに追加することも、新しいリソース共有に追加することもできます。

AWS Organizations の組織に属している場合、組織内での共有が有効になっていると、組織内のコンシューマーには共有 Dedicated Host へのアクセス許可が自動的に付与されます。それ以外の場合、コンシューマーはリソース共有への参加の招待を受け取り、その招待を受け入れた後で、共有 Dedicated Host へのアクセス許可が付与されます。

#### Note

Dedicated Host を共有した場合、コンシューマーがそれにアクセスできるまでに数分かかることがあります。

自己所有の Dedicated Host は、AWS RAM コンソールまたは AWS CLI を使用して共有できます。

AWS RAM コンソールを使用して、自己所有の Dedicated Host を共有するには

AWS RAM ユーザーガイド の「[リソース共有の作成](#)」を参照してください。

AWS CLI を使用して、自己所有の Dedicated Host を共有するには

[create-resource-share](#) コマンドを使用します。

## 共有 Dedicated Host の共有解除

Dedicated Host 所有者は、共有 Dedicated Host をいつでも共有解除できます。共有 Dedicated Host を共有解除する場合、以下のルールが適用されます。

- Dedicated Host を共有しているコンシューマーは、そこに新しいインスタンスを作成できなくなります。
- 共有解除時に Dedicated Host で実行されていたコンシューマー所有のインスタンスは、引き続き実行されますが、[リタイア](#)が予定されます。コンシューマーは、インスタンスのリタイア通知を受け取り、2週間以内に通知に対処します。ただし、リタイア通知期間内に Dedicated Host がコンシューマーに再共有されると、インスタンスのリタイアはキャンセルされます。

自己所有の共有 Dedicated Host を共有解除するには、それをリソース共有から削除する必要があります。これを行うには、AWS RAM コンソールまたは AWS CLI を使用します。

AWS RAM コンソールを使用して、自己所有の共有 Dedicated Host を共有解除するには

AWS RAM ユーザーガイド の「[リソース共有の更新](#)」を参照してください。

AWS CLI を使用して、自己所有の共有 Dedicated Host を共有解除するには

[disassociate-resource-share](#) コマンドを使用します。

## 共有 Dedicated Host の特定

所有者とコンシューマーは、Amazon EC2 コンソールまたは AWS CLI を使用して、共有 Dedicated Hosts を特定できます。

Amazon EC2 コンソールを使用して共有 Dedicated Host を特定するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。この画面には、自分が所有している Dedicated Hosts と共有している Dedicated Hosts が一覧表示されます。[所有者] 列には、Dedicated Host 所有者の AWS アカウント ID が表示されます。

AWS CLI を使用して共有 Dedicated Host を特定するには

`describe-hosts` コマンドを使用します。このコマンドは、自己が所有している Dedicated Hosts と共有している Dedicated Hosts を返します。

## 共有専有ホストで実行されているインスタンスの表示

所有者とコンシューマーは、Amazon EC2 コンソールや AWS CLI を使用して、共有 Dedicated Host で実行されているインスタンスをいつでも表示できます。

Amazon EC2 コンソールを使用して共有 Dedicated Host で実行されているインスタンスを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. インスタンスを表示する対象の Dedicated Host を選択し、[インスタンス] を選択します。ホストで実行されているインスタンスがタブに一覧表示されます。所有者は、コンシューマーによって作成されたインスタンスも含めて、ホストで実行されているすべてのインスタンスを表示できます。コンシューマーは、ホストで実行されている自己作成のインスタンスのみを表示できます。[所有者] 列には、インスタンスを作成した AWS アカウントのアカウント ID が表示されます。

AWS CLI を使用して共有 Dedicated Host で実行されているインスタンスを表示するには

`describe-hosts` コマンドを使用します。このコマンドは、各 Dedicated Host で実行されているインスタンスを返します。所有者は、ホストで実行されているすべてのインスタンスを表示できます。コンシューマーは、共有ホストで実行されている自己作成のインスタンスのみを表示できます。`InstanceOwnerId` は、インスタンス所有者の AWS アカウント ID を示します。

## 共有 Dedicated Host のアクセス許可

### 所有者のアクセス許可

所有者は、共有 Dedicated Hosts およびそこに作成したインスタンスの管理に責任を負います。所有者は、コンシューマーによって作成されたインスタンスも含めて、共有 Dedicated Host で実行されているすべてのインスタンスを表示できます。ただし、所有者は、コンシューマーによって作成された実行中のインスタンスに対してアクションを実行することはできません。

### コンシューマーのアクセス許可

コンシューマーは、共有 Dedicated Host に作成したインスタンスの管理に責任を負います。コンシューマーは、共有 Dedicated Host を一切変更できません。また、他のコンシューマーや Dedicated Host 所有者が作成したインスタンスを表示または変更することもできません。

## 請求と使用量測定

Dedicated Hosts の共有に追加料金はかかりません。

所有者は、自己が共有する Dedicated Hosts に対して課金されます。コンシューマーは、共有 Dedicated Hosts に作成したインスタンスに対して課金されません。

Dedicated Host の予約は、共有 Dedicated Hosts に対して引き続き請求割引を提供します。Dedicated Host 所有者のみが、自己が所有する共有 Dedicated Hosts 用の Dedicated Host の予約を購入できます。

## Dedicated Host の制限

共有 Dedicated Hosts は、所有者の Dedicated Hosts 制限に対してのみカウントされます。共有 Dedicated Hosts は、コンシューマーの Dedicated Hosts 制限に対してはカウントされません。同様に、コンシュー

マーが共有 Dedicated Hosts に作成するインスタンスは、コンシューマーのインスタンス制限に対してカウントされません。

## ホストの復旧と Dedicated Host の共有

ホストの復旧は、Dedicated Host の 所有者とその共有相手のコンシューマーによって作成されたインスタンスを復旧します。代替 Dedicated Host は所有者のアカウントに割り当てられます。元の Dedicated Host と同じリソース共有に追加され、同じコンシューマと共有されます。

詳細については、「[ホスト復旧 \(p. 420\)](#)」を参照してください。

## ホスト復旧

ホスト復旧では、Dedicated Host で障害が検出されると、インスタンスが新しい代替ホストで自動的に再起動されます。ホスト復旧は、Dedicated Host の予期しない障害が発生した場合に、手動の介入の必要性を減らし、オペレーションの負担を軽減します。

また、ホスト復旧が発生した場合、組み込まれている AWS License Manager との統合により、ライセンスの追跡と管理が自動化されます。

### Note

AWS License Manager との統合は、AWS License Manager を利用できるリージョンでのみサポートされます。

### 目次

- [ホスト復旧の基本 \(p. 420\)](#)
- [ホスト復旧の設定 \(p. 421\)](#)
- [ホスト復旧の状態 \(p. 422\)](#)
- [サポートされるインスタンスタイプ \(p. 423\)](#)
- [サポートされていないインスタンスの手動復旧 \(p. 423\)](#)
- [関連サービス \(p. 423\)](#)
- [料金 \(p. 424\)](#)

## ホスト復旧の基本

ホスト復旧では、ホストレベルのヘルスチェックを使用して専有ホストの可用性を評価し、基盤システムの障害を検出します。ホストレベルのヘルスチェックが失敗する場合、原因として以下のような問題が考えられます。

- ネットワーク接続の喪失
- システム電源の喪失
- 物理ホストのハードウェアまたはソフトウェアの問題

システム障害が Dedicated Host で検出されると、ホスト復旧が開始され、Amazon EC2 代替 Dedicated Host が自動的に割り当てられます。代替 Dedicated Host は新しいホスト ID を受け取りますが、元の Dedicated Host と同じ以下の属性を保持します。

- アベイラビリティゾーン
- インスタンスタイプ
- タグ
- 自動プレイスメントの設定

代替 Dedicated Host が割り当てられると、インスタンスは代替 Dedicated Host に復旧されます。復旧されたインスタンスは、元のインスタンスと同じ以下の属性を保持します。

- インスタンス ID
- プライベート IP アドレス
- Elastic IP アドレス
- EBS ボリュームアタッチメント
- すべてのインスタンスマタデータ

インスタンスと障害が発生した Dedicated Host との間にホストのアフィニティがある場合、復旧したインスタンスは代替 Dedicated Host との間にホストのアフィニティを確立します。

すべてのインスタンスが代替 Dedicated Host に復旧されると、障害が発生した Dedicated Host が解放されて、代替 Dedicated Host が使用可能になります。

ホスト復旧が開始されると、AWS アカウントの所有者に対してメールまたは AWS Personal Health Dashboard イベントで通知されます。ホスト復旧が正常に完了すると、2 つ目の通知が送信されます。

停止したインスタンスは代替 Dedicated Host に復旧されません。障害が発生した Dedicated Host を対象とする、停止したインスタンスを起動しようとすると、インスタンスの起動は失敗します。停止したインスタンスを変更して別の Dedicated Host を対象とするか、停止したインスタンスと一致する設定を持ち、自動配置が有効になっている Dedicated Host で起動することをお勧めします。

インスタンスマルチストレージのあるインスタンスは代替 Dedicated Host に復旧されません。是正措置として、障害が発生した Dedicated Host はリタイアとしてマークされ、ホスト復旧の完了後にリタイア通知が送信されます。リタイア通知に記載されている是正措置に従い、指定された期間内に障害が発生した Dedicated Host の残りのインスタンスを手動で復旧します。

AWS License Manager を使用してライセンスを追跡している場合は、ライセンス設定の制限に応じて AWS License Manager から新しいライセンスが代替 Dedicated Host に割り当てられます。ホスト復旧の結果としてライセンス設定のハード制限を超える場合、復旧プロセスは許可されず、Amazon SNS 通知を介してホスト復旧の失敗が通知されます。ホスト復旧の結果としてライセンス設定のソフト制限を超える場合は、復旧の続行が許可され、Amazon SNS 通知を介して制限の超過が通知されます。詳細については、AWS License Manager ユーザーガイドの「[ライセンス設定の使用](#)」を参照してください。

## ホスト復旧の設定

ホスト復旧の設定は、Dedicated Host を割り当てるときに行うか、Amazon EC2 コンソールまたは AWS Command Line Interface (CLI) を用いて割り当てた後に行うことができます。

### 目次

- [ホスト復旧の有効化 \(p. 421\)](#)
- [ホスト復旧の無効化 \(p. 422\)](#)
- [ホスト復旧の設定の表示 \(p. 422\)](#)

### ホスト復旧の有効化

ホスト復旧は、Dedicated Host の割り当て時または割り当て後に有効にすることができます。

ホスト復旧を Dedicated Host の割り当て時に有効にする方法の詳細については、「[Dedicated Hosts の割り当て \(p. 399\)](#)」を参照してください。

### ホスト復旧を割り当て後に有効にするには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. ホスト復旧を有効にする Dedicated Host を選択し、[Actions (アクション)]、[Modify Host Recovery (ホスト復旧の変更)] の順に選択します。
4. [Host recovery (ホスト復旧)] で、[Enable (有効化)]、」[Save (保存)] の順に選択します。

ホスト復旧を割り当て後に有効にするには (AWS CLI)

`modify-hosts` コマンドを使用して `host-recovery` パラメータを指定します。

```
$ aws ec2 modify-hosts --host-recovery on --host-ids h-012a3456b7890cdef
```

### ホスト復旧の無効化

ホスト復旧は Dedicated Host の割り当て後にいつでも無効にすることができます。

ホスト復旧を割り当て後に無効にするには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. ホスト復旧を無効にする Dedicated Host を選択し、[Actions (アクション)]、[Modify Host Recovery (ホスト復旧の変更)] の順に選択します。
4. [Host recovery (ホスト復旧)] で、[Disable (無効化)]、[Save (保存)] の順に選択します。

ホスト復旧を割り当て後に無効にするには (AWS CLI)

`modify-hosts` コマンドを使用して `host-recovery` パラメータを指定します。

```
$ aws ec2 modify-hosts --host-recovery off --host-ids h-012a3456b7890cdef
```

### ホスト復旧の設定の表示

Dedicated Host のホスト復旧の設定はいつでも表示できます。

Dedicated Host のホスト復旧の設定を表示するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Dedicated Hosts] を選択します。
3. Dedicated Host を選択し、[Description (説明)] タブの [Host Recovery (ホスト復旧)] フィールドを確認します。

Dedicated Host のホスト復旧の設定を表示するには (AWS CLI)

`describe-hosts` コマンドを使用します。

```
$ aws ec2 describe-hosts --host-ids h-012a3456b7890cdef
```

`HostRecovery` レスポンス要素に、ホスト復旧が有効であるか無効であるかが示されます。

### ホスト復旧の状態

Dedicated Host の障害が検出されると、障害が発生した Dedicated Host は `under-assessment` 状態になり、すべてのインスタンスは `impaired` 状態になります。障害が起きている Dedicated Host が `under-assessment` 状態の間は、このホストでインスタンスを起動できません。

代替 Dedicated Host が割り当てられると、この代替ホストは pending 状態になります。ホスト復旧プロセスが完了するまでは、この状態に留まります。代替 Dedicated Host が pending 状態の間は、このホストでインスタンスを起動できません。代替 Dedicated Host に復旧されたインスタンスは、復旧プロセス中、impaired 状態に留まります。

ホスト復旧が完了すると、代替 Dedicated Host は available 状態になり、復旧されたインスタンスは running 状態に戻ります。代替 Dedicated Host が available 状態になると、このホストでインスタンスを起動できます。障害が発生した元の Dedicated Host は完全に解放され、released-permanent-failure 状態になります。

障害が発生した Dedicated Host にホスト復旧をサポートしていないインスタンス (instance store-backed ボリュームのインスタンスなど) がある場合、Dedicated Host は解放されません。代わりに、そのホストはリタイアとしてマークされ、permanent-failure 状態になります。

## サポートされるインスタンスタイプ

ホスト復旧をサポートしているインスタンスファミリーは、A1、C3、C4、C5、C5n、Inf1、M3、M4、M5、M5n、P3、R3、R4、R5、R5n、X1、X1e、u-6tb1、u-9tb1、u-12tb1 および u-24tb1 です。

サポートされていないインスタンスを復旧するには、「[サポートされていないインスタンスの手動復旧 \(p. 423\)](#)」を参照してください。

## サポートされていないインスタンスの手動復旧

ホスト復旧は、インスタンストアボリュームを使用するインスタンスの復旧をサポートしていません。自動的に復旧されないインスタンスがある場合は、以下の手順に従って、これらのインスタンスを手動で復旧します。

### Warning

インスタンストアボリュームのデータは、インスタンスの停止または終了に伴って失われます。これには、EBS ボリュームをルートデバイスとするインスタンスにアタッチされたインスタンストアボリュームも含まれます。インスタンストアボリュームのデータを保護するには、インスタンスが停止または終了する前に、データを永続的ストレージにバックアップします。

### EBS-backed インスタンスの手動復旧

自動的に復旧されない EBS-backed インスタンスの場合は、インスタンスを手動で停止または終了させて、新しい Dedicated Host に復旧することをお勧めします。インスタンスの停止や、インスタンスの停止に伴うインスタンス設定の変更の詳細については、「[インスタンスの停止と起動 \(p. 529\)](#)」を参照してください。

### instance store-backed インスタンスの手動復旧

自動的に復旧されない instance store-backed インスタンスの場合は、以下の操作を行うことをお勧めします。

1. 新しい Dedicated Host で、最新の AMI から代替インスタンスを起動します。
2. すべての必要なデータを代替インスタンスに移行させます。
3. 障害が発生した Dedicated Host で元のインスタンスを終了します。

## 関連サービス

Dedicated Host は以下の AWS のサービスと統合されます。

- AWS License Manager — Amazon EC2 Dedicated Hosts 全体でライセンスを追跡します (AWS License Manager が利用可能なリージョンでのみサポートされます)。詳細については、[AWS License Manager ユーザーガイド](#) を参照してください。

## 料金

ホスト復旧の使用に伴う追加の料金はありません。通常の Dedicated Host 料金が適用されます。詳細については、「[Amazon EC2 Dedicated Hosts 料金](#)」を参照してください。

ホスト復旧が開始されると同時に、障害が発生した Dedicated Host には課金されなくなります。代替の専有ホストに対する課金は、専有ホストが available 状態になった後でのみ開始されます。

障害が発生した Dedicated Host の課金にオンデマンド料金が使用されていた場合は、代替の Dedicated Host の課金にもオンデマンド料金が使用されます。障害が発生した Dedicated Host でアクティブになっていた Dedicated Host の予約は、代替の Dedicated Host に転送されます。

## 設定変更の追跡

AWS Configを使用すると、Dedicated Hostsや、そのホストにおいて起動、停止または終了されるインスタンスに関する設定の変更を記録できます。そして、AWS Config でキャプチャされた情報をライセンスレポートのデータソースとして使用することができます。

AWS Configは、Dedicated Hostsやインスタンスの設定情報を個別に記録し、関係のある情報を組み合われます。3つのレポート条件があります。

- AWS Config の記録ステータス — [On (オン)] のとき、AWS Config は 1 つ以上の AWS リソースタイプを記録中です。これには、Dedicated Hosts や ハードウェア専有インスタンス を含めることができます。ライセンスレポートに必要な情報をキャプチャするには、次のフィールドによって Host とインスタンスが記録されていることを確認します。
- Host recording status — [Enabled] の場合は、Dedicated Hosts の設定情報が記録されます。
- インスタンスの記録ステータス — [Enabled (有効)] の場合は、ハードウェア専有インスタンス の設定情報が記録されます。

これら 3 つの条件のいずれかが無効になっている場合、[設定記録の編集] ボタン内のアイコンは赤です。このツールのメリットをすべて引き出すために、3つの記録方法すべてを有効にしてください。3つすべてが有効なとき、アイコンは緑です。設定を編集するには、[設定記録の編集] を選択します。AWS Config コンソールに [Set up AWS Config] ページが表示され、そこで AWS Config を設定し、ホスト、インスタンス、およびその他のサポートされるリソースタイプの記録を開始できます。詳細については、『AWS Config 開発者ガイド』の「[コンソールを使用した AWS Config のセットアップ](#)」を参照してください。

### Note

AWS Config はリソースを検出(数分かかる場合があります)して、記録します。

AWS Config がホストおよびインスタンスへの設定変更の記録を開始した後、ユーザーが割り当てたが解放したホストと、起動、停止、または終了したインスタンスの設定履歴を取得できます。たとえば、Dedicated Host の設定履歴の任意の時点で、そのホストのソケット数とコア数と共に、そのホストで起動されているインスタンスの数を調べることができます。それらのインスタンスについても、その Amazon マシンイメージ (AMI) の ID を調べることができます。これらの情報を使用して、ソケット単位またはコア単位でライセンスが与えられているサーバーバインドソフトウェアのライセンスに関するレポートを作成できます。

設定履歴は以下のいずれかの方法で閲覧できます。

- AWS Config コンソールを使用する。記録されたリソースごとに、設定の詳細の履歴を提供するタイムラインページを表示することができます。このページを表示するには、[Dedicated Hosts] ページの [設定タイムライン] 列にあるグレーのアイコンを選択します。詳細については、『AWS Config 開発者ガイド』の「[AWS Config コンソールでの設定詳細の表示](#)」を参照してください。
- AWS CLI コマンドを実行する。まず、`list-discovered-resources` コマンドを使用して、すべてのホストとインスタンスのリストを取得できます。次に、`get-resource-config-history` コマンドを使用して、特定の時間間隔でホストまたはインスタンスの設定の詳細を取得できます。詳細については、『AWS Config 開発者ガイド』の「[CLI による設定詳細の表示](#)」を参照してください。

- ・ アプリケーションで AWS Config API を使用する。まず、[ListDiscoveredResources](#) アクションを使用して、すべてのホストとインスタンスのリストを取得できます。次に、[GetResourceConfigHistory](#) アクションを使用して、特定の時間間隔でホストまたはインスタンスの設定の詳細を取得できます。

たとえば、AWS Config からすべての Dedicated Hosts のリストを取得するには、次のような CLI コマンドを実行します。

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Host
```

AWS Config から Dedicated Host の設定履歴を取得するには、次のような CLI コマンドを実行します。

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --resource-id i-1234567890abcdef0
```

コンソールを使用して AWS Config の設定を管理するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [Dedicated Hosts] ページで、[設定記録の編集] を選択します。
3. AWS Config コンソールで、次の手順に従って記録をオンにします。 詳細については、「[コンソールを使用した AWS Config のセットアップ](#)」を参照してください。

詳細については、「[AWS Config コンソールでの設定詳細の表示](#)」を参照してください。

コマンドラインまたは API を使用して AWS Config をアクティブ化するには

- ・ AWS CLI の使用については、AWS Config 開発者ガイドの「[設定詳細の表示 \(AWS CLI\)](#)」を参照してください。
- ・ Amazon EC2 API の使用については、「[GetResourceConfigHistory](#)」を参照してください。

## ハードウェア専有インスタンス

ハードウェア専有インスタンスは、単一のカスタマー専用のハードウェアの Virtual Private Cloud (VPC) で実行される Amazon EC2 インスタンスです。異なる AWS アカウントに属する ハードウェア専有インスタンスは、ハードウェアレベルで物理的に分離されています。さらに、単一の支払いアカウントにリンクされている AWS アカウントに属する ハードウェア専有インスタンスは、ハードウェアレベルで物理的に分離されています。ただし、ハードウェア専有インスタンスは、同じ AWS アカウントからの他のインスタンスで、ハードウェア専有インスタンスではないインスタンスと、ハードウェアをシェアする可能性があります。

### Note

また、Dedicated Host はお客様専用の物理サーバーです。 Dedicated Host では、インスタンスをサーバーに配置する方法について可視性と制御を高めることができます。 詳細については、「[Dedicated Hosts \(p. 395\)](#)」を参照してください。

## ハードウェア専有インスタンスの基礎

VPC 内に起動する各インスタンスにはテナント属性があります。この属性の値を次に示します。

テナント属性値	説明
default	インスタンスは共有するハードウェアで実行されます。
dedicated	インスタンスはシングルテナントのハードウェアで実行されます。

テナント属性値	説明
host	インスタンスは Dedicated Host で実行します。Dedicated Host はユーザーが設定を制御できる隔離サーバーです。

インスタンスを起動した後、そのテナント属性を変更する場合はいくつかの制限があります。

- ・インスタンスのテナント属性は、インスタンスの起動後は `default` から `dedicated` または `host` に変更できません。
- ・インスタンスのテナント属性は、インスタンスの起動後は `dedicated` または `host` から `default` に変更できません。

インスタンスのテナント属性は、インスタンスの起動後に `dedicated` から `host` に変更したり、`host` から `dedicated` に変更したりできます。詳細については、「[インスタンスのテナント属性の変更 \(p. 430\)](#)」を参照してください。

各 VPC には関連したインスタンスのテナント属性があります。この属性の値を次に示します。

テナント属性値	説明
<code>default</code>	VPC で起動されたインスタンスはデフォルトでは共有ハードウェアで実行されます。ただし、これはインスタンスの起動時に別のテナントを明示的に指定しない場合に限ります。
<code>dedicated</code>	VPC で起動されたインスタンスはデフォルトでは ハードウェア専有インスタンスです。ただし、これはインスタンスの起動時に <code>host</code> のテナントを明示的に指定しない場合に限ります。インスタンスの起動時に <code>default</code> のテナントを指定することはできません。

VPC の作成後に VPC インスタンスのテナント属性を `dedicated` から `default` に変更することができます。VPC のインスタンスのテナント属性を `dedicated` に変更することはできません。

ハードウェア専有インスタンスは以下の方法で作成できます。

- ・インスタンスのテナント属性を `dedicated` (この VPC 内に起動されたすべてのインスタンスは ハードウェア専有インスタンス) に設定して VPC を作成します。
- ・インスタンスのテナント属性を `default` に設定して VPC を作成し、インスタンスの起動時にテナント属性として `dedicated` を指定します。

## ハードウェア専有インスタンス の制約事項

AWS の一部のサービスまたは機能は、インスタンスのテナント属性が `dedicated` に設定されている VPC では動作しません。そのほかにも制限事項があるかどうかを確認するには、サービスのドキュメントを参照してください。

一部の種類のインスタンスは、インスタンスのテナント属性が `dedicated` に設定されている VPC では起動できません。サポートされているインスタンスの種類の詳細については、「[Amazon EC2 ハードウェア専有インスタンス](#)」を参照してください。

## Amazon EBS と ハードウェア専有インスタンス

Amazon EBS バックトラック/ハードウェア専有インスタンスを起動した場合、シングルテナントのハードウェアで EBS ボリュームは実行できません。

## 専有テナント属性がある リザーブドインスタンス

ハードウェア専有インスタンスを起動できるだけの十分な空き容量を確保するために、ハードウェア専有リザーブドインスタンスを購入できます。詳細については、「[リザーブドインスタンス \(p. 279\)](#)」を参照してください。

ハードウェア専有リザーブドインスタンスを購入すると、VPC 内にハードウェア専有インスタンスを起動するための容量を格安の料金で利用できます。使用料金引き下げは、専有テナントでインスタンスを起動した場合にのみ適用されます。デフォルトテナシードリザーブドインスタンスを購入する場合、これは default テナシードリザーブドインスタンスにのみ適用され、dedicated テナシードリザーブドインスタンスには適用されません。

さらに、リザーブドインスタンスの購入後に変更プロセスを使用してそのテナシードリザーブドインスタンスを別のテナシードリザーブドインスタンスに交換することはできます。

## ハードウェア専有インスタンスの自動スケーリング

Amazon EC2 Auto Scaling を使用してハードウェア専有インスタンスを起動できます。詳細については「[VPC での Auto Scaling インスタンスの起動](#)」(Amazon EC2 Auto Scaling ユーザーガイド)を参照してください。

## ハードウェア専有インスタンスの自動復旧

内在のハードウェア障害または修復に AWS を必要とする問題によって正常に機能しなくなった場合は、ハードウェア専有インスタンスの自動復元を設定できます。詳細については、「[インスタンスの復旧 \(p. 551\)](#)」を参照してください。

## ハードウェア専有スポットインスタンス

スポットインスタンスのリクエストを作成するとき、dedicated のテナントを指定することにより、ハードウェア専有スポットインスタンスを実行できます。詳細については、「[スポットインスタンスのテナシードリザーブドインスタンス \(p. 337\)](#)」を参照してください。

## ハードウェア専有インスタンスの料金表

ハードウェア専有インスタンスの料金表は、オンデマンドインスタンスの料金表と異なります。詳細については、「[Amazon EC2 ハードウェア専有インスタンス 製品ページ](#)」を参照してください。

## ハードウェア専有インスタンスでのバーストパフォーマンスインスタンス

the section called “バースト可能パフォーマンスインスタンス”(p. 199) では、専有テナントハードウェアで実行することの利点を活用できます。T3 ハードウェア専有インスタンスは、デフォルトで Unlimited モードで起動します。また、ベースラインレベルの CPU パフォーマンスを提供し、ワークロードの必要に応じてより高い CPU レベルにバーストできます。T3 ベースラインパフォーマンスとバースト機能は、CPU クレジットによって管理されます。T3 インスタンスタイプはバーストであるため、最適なパフォーマンスを得るために T3 インスタンスで専用ハードウェアの CPU リソースをどのように使用しているかをモニタリングすることをお勧めします。T3 ハードウェア専有インスタンスは、お客様のワークロードが多様で CPU がランダムな動作を示すが、平均的な CPU 使用量が適切なベースライン使用量以下である場合に向いています。詳細については、「[the section called “CPU クレジットおよびベースラインパフォーマンス” \(p. 200\)](#)」を参照してください。

Amazon EC2 には、パフォーマンスの変動を特定して修正するためのシステムが用意されています。ただし、CPU 使用パターンが相關する複数の T3 ハードウェア専有インスタンスを起動すると、依然として短期的な変動が発生する可能性があります。これらのより要求の厳しいワークロードや相関関係のあるワークロードについては、T3 ハードウェア専有インスタンスではなく、M5 または M5a ハードウェア専有インスタンスを使用することをお勧めします。

## ハードウェア専有インスタンス の操作

VPC の作成時にインスタンスのテナント属性として `dedicated` を指定すると、VPC 内に起動されるすべてのインスタンスを ハードウェア専有インスタンス にすることができます。インスタンスのテナント属性は起動時に指定することもできます。

### トピック

- [インスタンスのテナント属性が専有である VPC を作成する \(p. 428\)](#)
- [VPC 内で ハードウェア専有インスタンス を起動する \(p. 428\)](#)
- [テナント属性情報を表示する \(p. 429\)](#)
- [インスタンスのテナント属性の変更 \(p. 430\)](#)
- [VPC のテナント属性の変更 \(p. 430\)](#)

### インスタンスのテナント属性が専有である VPC を作成する

VPC を作成するときにインスタンスのテナント属性を指定できます。Amazon VPC コンソールを使用している場合、VPC ウィザードまたは [Your VPCs] ページを使用して VPC を作成できます。

インスタンスのテナント属性がハードウェア専有である VPC を作成するには (VPC ウィザード)

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. ダッシュボードで、[VPC ウィザードの開始] を選択します。
3. VPC 設定を選択し、[選択] を選択します。
4. ウィザードの次のページで、[ハードウェアのテナント] のリストから [ハードウェア専有] を選択します。
5. [Create VPC] を選択します。

インスタンスのテナント属性がハードウェア専有である VPC を作成するには (VPC ダイアログ ボックスの作成)

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. ナビゲーションペインで [VPC]、[VPC の作成] の順に選択します。
3. [Tenancy] で、[Dedicated] を選択します。CIDR ブロックを指定し、[Yes, Create] を選択します。

コマンドラインを使用して VPC を作成するときにテナント属性オプションを設定するには

- [create-vpc \( AWS CLI \)](#)
- [New-EC2Vpc \( AWS Tools for Windows PowerShell \)](#)

インスタンスのテナント属性が `dedicated` である VPC 内にインスタンスを起動すると、インスタンスのテナント属性とは関係なく、インスタンスは自動的に ハードウェア専有インスタンス となります。

### VPC 内で ハードウェア専有インスタンス を起動する

ハードウェア専有インスタンス は、Amazon EC2 インスタンス起動ウィザードを使用して起動できます。

コンソールを使用してデフォルトテナント VPC にハードウェア専有インスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。

3. [Amazon マシンイメージ (AMI)] ページで、AMI を選択し、[選択] を選択します。
4. [Choose an Instance Type] ページで、インスタンスタイプを選択し、[Next: Configure Instance Details] を選択します。

Note

ハードウェア専有インスタンスとしてサポートされているインスタンスタイプを必ず選択します。詳細については、「[Amazon EC2 ハードウェア専有インスタンス](#)」を参照してください。

5. [Configure Instance Details] ページで、VPC とサブネットを選択します。[Tenancy] のリストから [Dedicated - Run a dedicated instance] を選択し、[Next: Add Storage] を選択します。
6. ウィザードに従って続行します。[Review Instance Launch] ページでオプションの確認が終了したら、[Launch] を選択し、キーペアを選択して ハードウェア専有インスタンスを作成します。

テナント属性として host を使用したインスタンスの作成の詳細については、「[Dedicated Host にインスタンスを作成する \(p. 401\)](#)」を参照してください。

コマンドラインを使用して起動中にインスタンスのテナント属性オプションを設定するには

- [run-instances](#) (AWS CLI)
- [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

## テナント属性情報を表示する

コンソールを使用して VPC のテナント属性情報を表示するには

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. [テナンシー] 列で、VPC のインスタンスのテナント属性を確認します。
4. [テナンシー] 列が表示されていない場合は、[テーブル列の編集] (歯車型のアイコン) を選択します。  
[列の表示/非表示] ダイアログボックスで [テナンシー] を選択し、[閉じる] を選択します。

コンソールを使用してインスタンスのテナント属性情報を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. [テナンシー] 列でインスタンスのテナント属性を確認します。
4. [テナンシー] 列が表示されていない場合は、次のいずれかを行います。
  - [列の表示/非表示] (歯車型のアイコン) を選択し、[列の表示/非表示] ダイアログボックスで [テナンシー] を選択して [閉じる] を選択します。
  - インスタンスを選択します。詳細ペインの [説明] タブに、テナント属性を含めてインスタンスに関する情報が表示されます。

コマンドラインを使用して VPC のテナンシーを記述するには

- [describe-vpcs](#) ( AWS CLI )
- [Get-EC2Vpc](#) ( AWS Tools for Windows PowerShell )

コマンドラインを使用してインスタンスのテナント属性を記述するには

- [describe-instances](#) ( AWS CLI )

- [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)

コマンドラインを使用してリザーブドインスタンスのテナント属性値を記述するには

- [describe-reserved-instances](#) (AWS CLI)
- [Get-EC2ReservedInstance](#) (AWS Tools for Windows PowerShell)

コマンドラインを使用してリザーブドインスタンス製品のテナント属性値を記述するには

- [describe-reserved-instances-offerings](#) (AWS CLI)
- [Get-EC2ReservedInstancesOffering](#) (AWS Tools for Windows PowerShell)

## インスタンスのテナント属性の変更

インスタンスタイプおよびプラットフォームによっては、インスタンスの起動後に、停止された ハードウェア専有インスタンス のテナント属性を `host` に変更できます。次回のインスタンスの起動時に、インスタンスはアカウントに割り当てられた Dedicated Host で実行されます。Dedicated Hosts の割り当てと使用、および Dedicated Hosts で使用できるインスタンスタイプの詳細については、「[Dedicated Hosts の使用 \(p. 398\)](#)」を参照してください。同様に、インスタンスの起動後に、停止された Dedicated Host インスタンスのテナント属性を `dedicated` に変更できます。次回のインスタンスの起動時に、インスタンスは Amazon が管理するシングルテナントのハードウェアで実行されます。

コンソールを使用してインスタンスのテナント属性を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. [Actions]、[Instance State]、[Stop] の順に選択します。
4. [Actions]、[Instance Settings]、[Modify Instance Placement] の順に選択します。
5. [Tenancy] のリストで、インスタンスを専有ハードウェアで実行するか、Dedicated Host で実行するかを選択します。[Save] を選択します。

コマンドラインを使用してインスタンスのテナント属性値を変更するには

- [modify-instance-placement](#) (AWS CLI)
- [Edit-EC2InstancePlacement](#) (AWS Tools for Windows PowerShell)

## VPC のテナント属性の変更

VPC インスタンスのテナント属性を `dedicated` から `default` に変更することができます。VPC インスタンスのテナント属性を変更しても、VPC 内の既存のインスタンスのテナント属性には影響が及ぼしません。次回 VPC でインスタンスを起動すると、起動時に指定していなければ、テナント属性が `default` になります。

VPC のインスタンスのテナント属性を `dedicated` に変更することはできません。

VPC インスタンスのテナント属性は、AWS CLI、AWS SDK、Amazon EC2 API のみを使用して変更できます。

AWS CLI を使用して VPC インスタンスのテナント属性を変更するには

- VPC の ID とインスタンスのテナント属性値を指定するには、[modify-vpc-tenancy](#) コマンドを使用します。`default` はサポートされる唯一の値です。

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

## オンデマンドキャパシティー予約

オンデマンドキャパシティー予約 を使用すると、任意の所要時間の特定のアベイラビリティゾーンで Amazon EC2 インスタンスのキャパシティーを予約できます。これにより、Savings Plans またはリージョンリザーブドインスタンスが提供する請求割引とは独立して、キャパシティー予約を登録および管理することができます。キャパシティーの予約を作成することで、必要なときに、必要な期間中、EC2 キャパシティーに常にアクセスできるようになります。キャパシティーの予約は、1 年間または 3 年間のコミットメント期間なしにいつでも作成でき、キャパシティーはすぐに利用可能になります。予約が必要なくなった場合は、キャパシティーの予約をキャンセルして料金の発生を停止できます。

キャパシティーの予約を作成するときは、以下を指定します。

- キャパシティーが予約されているアベイラビリティゾーン
- キャパシティーを予約するインスタンスの数
- インスタンスタイプ、テナント、プラットフォーム/OS を含む、インスタンスの属性

キャパシティーの予約を使用できるのは、属性が一致するインスタンスのみです。デフォルトでは、属性に一致する実行中のインスタンスによって自動的に使用されます。キャパシティーの予約の属性と一致する実行中のインスタンスがない場合は、一致する属性を持つインスタンスを起動するまでは使用されません。

また、Savings Plans とリージョンリザーブドインスタンスをキャパシティーの予約と組み合わせて使用して、請求割引を受けることもできます。キャパシティーの予約の属性が Savings Plan またはリージョンリザーブドインスタンスの属性と一致する場合、自動的に割引が適用されます。詳細については、「[請求割引 \(p. 433\)](#)」を参照してください。

### コンテンツ

- [キャパシティーの予約、リザーブドインスタンス、および Savings Plans の違い \(p. 431\)](#)
- [キャパシティーの予約の制限 \(p. 432\)](#)
- [キャパシティーの予約の制約と制限 \(p. 432\)](#)
- [キャパシティーの予約 料金と請求 \(p. 433\)](#)
- [キャパシティーの予約を使用する \(p. 434\)](#)
- [共有キャパシティーの予約の使用 \(p. 439\)](#)

## キャパシティーの予約、リザーブドインスタンス、および Savings Plans の違い

次のテーブルでは、キャパシティーの予約、リザーブドインスタンス、および Savings Plans の主な違いを紹介します。

	キャパシティーの予約	ゾーン リザーブドインスタンス	リージョン リザーブドインスタンス	Savings Plans
期間	コミットメントは不要です。必要に応じて作成およびキャンセルすることができます。		固定の 1 年または 3 年のコミットメントが必要です。	

	キャパシティーの予約	ゾーンリザーブドインスタンス	リージョンリザーブドインスタンス	Savings Plans
キャパシティーの利点	特定のアベイラビリティーゾーンで予約されるキャパシティー。		アベイラビリティーゾーンでキャパシティーを予約しません	
請求割引	請求割引がありません。キャパシティーの予約に起動されたインスタンスは、標準のオンデマンドレートで課金されます。ただし、Savings Plans またはリージョンリザーブドインスタンスとキャパシティーの予約を併用することで、請求割引を受けることができます。ゾーンリザーブドインスタンスはキャパシティーの予約には適用されません。	請求割引を提供します。		
インスタンスの制限	リージョンごとの オンデマンドインスタンス 制限に制限されています。	アベイラビリティーゾーンあたり 20 に制限されています。制限の引き上げをリクエストできます。	リージョンあたり 20 に制限されています。制限の引き上げをリクエストできます。	制限なし。

詳細については、以下を参照してください。

- [リザーブドインスタンス \(p. 279\)](#)
- [AWS Savings Plans ユーザーガイド](#)

## キャパシティーの予約 の制限

キャパシティーの予約が許可されているインスタンスの数は、アカウントの オンデマンドインスタンスの制限に基づいています。上限に到達しない限り、すでに実行されているインスタンスの数を差し引いた任意の数のインスタンスのキャパシティーを予約できます。

## キャパシティーの予約 の制約と制限

キャパシティーの予約を作成する前に、次の制限と制約に注意してください。

- オンデマンドインスタンス制限に対するアクティブで未使用的のキャパシティーの予約カウント
- キャパシティーの予約は 1 つの AWS アカウントから別のアカウントに譲渡できません
- ゾーンリザーブドインスタンス請求割引はキャパシティーの予約には適用されません
- プレイスマントグループに キャパシティーの予約を作成することはできません
- キャパシティーの予約を Dedicated Hosts と一緒に使用することはできません
- キャパシティーの予約を持ち込みライセンス (BYOL、Bring Your Own License) と一緒に使用することはできません

## キャパシティーの予約 料金と請求

キャパシティーの予約の料金は支払いオプションによって異なります。

### 料金表

キャパシティーの予約がアクティブな場合、インスタンスを実行するかどうかにかかわらず、同等のオンデマンド料金が請求されます。予約を使用しない場合、この予約は EC2 請求書に未使用予約として記載されます。予約の属性に一致するインスタンスを実行するときは、そのインスタンスの料金のみを支払い、予約に料金はかかりません。前払い、または追加の料金はありません。

たとえば、20 個の m4.large Linux インスタンスに対してキャパシティーの予約を作成し、同じアベイラビリティーゾーンで 15 個の m4.large Linux インスタンスを実行すると、15 個のアクティブインスタンスと予約されている 5 個の未使用のインスタンス分が課金されます。

Savings Plans とリージョンリザーブドインスタンスの請求割引がキャパシティーの予約に適用されます。詳細については、「[請求割引 \(p. 433\)](#)」を参照してください。

Amazon EC2 の料金の詳細については、「[Amazon EC2 料金表](#)」を参照してください。

### 請求

キャパシティーの予約は、秒単位で課金されます。つまり、1 時間に満たない分に対して課金されます。たとえば、24 時間 15 分の間、アカウント内で予約が有効な場合は、24.25 予約時間が課金されます。

次の例は、キャパシティーの予約の請求方法を示しています。キャパシティーの予約は 1 つの m4.large Linux インスタンスに対して作成され、オンデマンド料金は 1 時間あたり 0.10 USD です。この例では、キャパシティーの予約はアカウントで 5 時間有効です。キャパシティーの予約は最初の 1 時間は使用されないため、m4.large インスタンスタイプの標準オンデマンド料金で未使用の 1 時間分の料金が請求されます。2~5 時間目は、キャパシティーの予約は m4.large インスタンスによって占有されます。この間、キャパシティーの予約に料金は発生せず、代わりにそれを占有している m4.large インスタンスに対してアカウントが請求されます。6 時間目にはキャパシティーの予約がキャンセルされ、m4.large インスタンスはリザーブドキャパシティー外で通常どおりに実行されます。その時間は、m4.large インスタンスタイプのオンデマンド料金で請求されます。

Hour	1	2	3	4	5	
<b>Unused Capacity Reservation</b>	\$0.10	\$0.00	\$0.00	\$0.00	\$0.00	\$
<b>On-demand Instance Usage</b>	\$0.00	\$0.10	\$0.10	\$0.10	\$0.10	\$
<b>Hourly cost</b>	<b>\$0.10</b>	<b>\$0.10</b>	<b>\$0.10</b>	<b>\$0.10</b>	<b>\$0.10</b>	<b>\$</b>

### 請求割引

Savings Plans とリージョンリザーブドインスタンスの請求割引がキャパシティーの予約に適用されます。AWS では、一致している属性を持つキャパシティーの予約にこれらの割引を自動的に適用します。キャパシティーの予約がインスタンスによって使用されると、割引がインスタンスに適用されます。割引は、未使用的キャパシティーの予約を対象とする前に、インスタンスの使用に優先的に適用されます。

ゾーンリザーブドインスタンスの請求割引はキャパシティーの予約には適用されません。

詳細については、以下を参照してください。

- リザーブドインスタンス (p. 279)
- AWS Savings Plans ユーザーガイド

## 請求の表示

アカウントの請求と料金は、AWS Billing and Cost Management コンソールで確認できます。

- ・[ダッシュボード] には、アカウント利用料の概要が表示されます。
- ・[請求書] ページの [明細] で、[Elastic Compute Cloud] セクションとリージョンを展開して、キャパシティーの予約 の請求情報を取得します。

請求額をオンラインで表示することも、CSV ファイルとしてダウンロードすることもできます。詳細については、『AWS Billing and Cost Management ユーザーガイド』の「[キャパシティーの予約 明細項目](#)」を参照してください。

## キャパシティーの予約 を使用する

キャパシティーの予約の使用を開始するには、必要なアベイラビリティゾーンにキャパシティーの予約を作成します。次に、インスタンスをリザーブドキャパシティーに起動し、そのキャパシティーの使用率をリアルタイムで表示して、必要に応じてキャパシティーを増減することができます。

デフォルトでは、キャパシティーの予約 は、新しいインスタンスと、一致する属性 (インスタンスタイプ、プラットフォーム、およびアベイラビリティゾーン) を持つ実行中のインスタンスを自動的に一致させます。つまり、一致する属性を持つインスタンスがキャパシティーの予約で自動的に実行されます。ただし、特定のワークロードに対して キャパシティーの予約 を指定することもできます。これにより、リザーブドキャパシティーで実行できるインスタンスを明示的に制御できます。

予約が終了する方法を指定できます。手動で キャパシティーの予約 をキャンセルするか指定した時刻に自動的に終了させるかを選択できます。終了時間を指定する場合、キャパシティーの予約 は指定した時刻の 1 時間以内にキャンセルされます。たとえば、2019 年 5 月 31 日、13:30:55 を指定すると、キャパシティーの予約 は 2019 年 5 月 31 日の 13:30:55 と 14:30:55 の間に終了することが保証されます。予約 が終了すると、インスタンスを キャパシティーの予約 のターゲットにすることはできなくなります。リザーブドキャパシティーで実行されているインスタンスは、中断されずに引き続き実行されます。キャパシティーの予約 をターゲットにしているインスタンスが停止している場合は、キャパシティーの予約 ターゲット設定を削除するか、別の キャパシティーの予約 をターゲットに設定するまで再開できません。

### 目次

- ・ [キャパシティーの予約 の作成 \(p. 434\)](#)
- ・ [既存の キャパシティーの予約 へのインスタンスの起動 \(p. 436\)](#)
- ・ [キャパシティーの予約 の変更 \(p. 436\)](#)
- ・ [インスタンスの キャパシティーの予約 設定を変更する \(p. 437\)](#)
- ・ [キャパシティーの予約 の表示 \(p. 438\)](#)
- ・ [キャパシティーの予約 のキャンセル \(p. 438\)](#)

## キャパシティーの予約 の作成

キャパシティーの予約 を作成すると、すぐにキャパシティーが利用可能になります。このキャパシティーは、キャパシティーの予約 がアクティブであれば、使用のために予約されており、いつでもインスタンスを起動することができます。キャパシティーの予約 がオープンの場合、新しいインスタンスと一致する属性を持つ既存のインスタンスは キャパシティーの予約 のキャパシティーで自動的に実行されます。キャパシティーの予約 が targeted の場合、インスタンスはそれがリザーブドキャパシティーで実行されるように具体的に設定する必要があります。

次のいずれかが当てはまる場合、キャパシティーの予約 を作成するリクエストは失敗する可能性があります。

- ・ Amazon EC2 には、リクエストに対応する十分なキャパシティ - がありません。時間をおいてからもう一度試すか、別のアベイラビリティゾーンを試すか、キャパシティ - を小さくしてみてください。イ

ンスタンスタイプとサイズに応じてアプリケーションに柔軟性がある場合は、別のインスタンス属性を試してみてください。

- リクエストされた数量は、選択したインスタンスマミリーに対するオンデマンドインスタンスの上限を超えていません。インスタンスマミリーに対するオンデマンドインスタンスの上限を上げて、もう一度試してください。詳細については、「[オンデマンドインスタンスの制限 \(p. 277\)](#)」を参照してください。

コンソールを使用して キャパシティーの予約 を作成するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- [キャパシティーの予約]、[作成キャパシティーの予約] の順に選択します。
- キャパシティーの予約 の作成ページの、[Instance Details (インスタンスの詳細)] セクションで、以下の設定を指定します。起動するインスタンスのインスタンスタイプ、プラットフォーム、アベイラビリティゾーンは、ここで指定するインスタンスタイプ、プラットフォーム、アベイラビリティゾーンと一致する必要があります。一致しない場合、キャパシティーの予約 は適用されません。たとえば、開いているキャパシティーの予約が一致しない場合、このキャパシティーの予約を明示的に対象とするインスタンスの起動は失敗します。
  - [Instance Type (インスタンスのタイプ)] — リザーブドキャパシティーに起動するインスタンスのタイプ。
  - [Launch EBS-optimized instances (EBS 最適化インスタンスを起動する)] — EBS 最適化インスタンスのキャパシティーを予約するかどうかを指定します。このオプションは、一部のインスタンスタイプではデフォルトで選択されています。EBS 最適化インスタンスの詳細については、「[Amazon Elastic Block Store \(p. 929\)](#)」を参照してください。
  - [Attach instance store at launch (起動時にインスタンストアをアタッチ)] — キャパシティーの予約 に起動されたインスタンスが一時的なブロックレベルのストレージを使用するかどうかを指定します。インスタンストアボリューム上のデータは、関連付けられたインスタンスの運用中のみ維持されます。
  - [プラットフォーム] — インスタンスのオペレーティングシステム。
  - [アベイラビリティゾーン] — キャパシティーを予約するアベイラビリティゾーン。
  - [テナント] — 共有ハードウェア (デフォルト) を実行するか専有インスタンスを実行するかを指定します。
  - [数量] — キャパシティーを予約するインスタンスの数。選択したインスタンスタイプの残りの オンデマンドインスタンス 制限を超える数量を指定すると、そのリクエストは拒否されます。
- [Reservation details (予約の詳細)] セクションで次のように設定します。
  - [Reservation Ends (予約終了)] — 次のいずれかのオプションを選択します。
    - [Manually (手動)] — 明示的にキャンセルするまで容量を予約してください。
    - [Specific time (特定の時間)] — 指定された日時にキャパシティーの予約を自動的に解除します。
  - [Instance eligibility (インスタンスの利用資格)] — 次のいずれかのオプションを選択します。
    - [open (開く)] — (デフォルト) キャパシティーの予約 は、一致する属性 (インスタンスタイプ、プラットフォーム、およびアベイラビリティゾーン) を持つインスタンスに一致します。一致する属性を持つインスタンスを起動すると、そのインスタンスはリザーブドキャパシティーに自動的に配置されます。
    - [targeted (指定済み)] — キャパシティーの予約 は、一致する属性 (インスタンスタイプ、プラットフォーム、およびアベイラビリティゾーン) を持つインスタンスのみを受け入れ、明示的に予約を行います。
- [Request reservation (リクエスト予約)] を選択します。

AWS CLI を使用して キャパシティーの予約 を作成するには

[create-capacity-reservation](#) コマンドを使用します。

```
aws ec2 create-capacity-reservation --instance-type instance_type --instance-platform platform_type --availability-zone az --instance-count quantity
```

## 既存の キャパシティーの予約 へのインスタンスの起動

一致する属性(インスタンスタイプ、プラットフォーム、およびアベイラビリティーゾーン)と十分なキャパシティーがある場合に、インスタンスを既存の キャパシティーの予約 に起動することができます。キャパシティーの予約 にインスタンスを起動すると、起動されたインスタンスの数だけ使用可能なキャパシティーが減少します。たとえば、3 つのインスタンスを起動すると、キャパシティーの予約 の使用可能なキャパシティーは 3 つ減少します。

コンソールを使用して既存の キャパシティーの予約 でインスタンスを起動するには

1. [ダッシュボード] または [インスタンス] から [インスタンスの起動] を選択して、起動インスタンス ウィザードを開きます。
2. Amazon Machine Image (AMI) とインスタンスタイプを選択します。
3. [Configure Instance Details (インスタンスの詳細の設定)] ページに入力します。[キャパシティーの予約] で、以下のいずれかのオプションを選択します。
  - [Open (開く)] — 選択したインスタンスの数に対して一致する属性と十分なキャパシティ - のある キャパシティーの予約 にインスタンスを起動します。十分なキャパシティーを持つ、一致する キャパシティーの予約 がない場合は、インスタンスはオンデマンドのキャパシティーを使用します。
  - <キャパシティーの予約> — インスタンスをこのキャパシティーの予約に起動します。この キャパシティーの予約 に選択したインスタンスの数に対して十分なキャパシティ - がない場合、インスタンスの起動に失敗します。
  - [なし] — インスタンスが キャパシティーの予約 に起動しないようにします。
4. インスタンスを起動する残りのステップを完了します。

AWS CLI を使用して既存の キャパシティーの予約 でインスタンスを起動するには

[run-instances](#) コマンドを使用して --capacity-reservation-specification パラメータを指定します。

次の例では、属性と使用可能なキャパシティーが一致する任意の開いている キャパシティーの予約 で t2.micro インスタンスを起動します。

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --availability-zone us-east-1b --capacity-reservation-specification CapacityReservationPreference=open
```

次の例では、t2.micro インスタンスをtargetedのキャパシティーの予約に起動します。

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --availability-zone us-east-1b --capacity-reservation-specification CapacityReservationTarget=[{CapacityReservationId=cr-a1234567}]
```

## キャパシティーの予約 の変更

アクティブな キャパシティーの予約 の属性は、作成後に変更することができます。期限が切れた後、または明示的にキャンセルした後で、キャパシティーの予約 を変更することはできません。

キャパシティーの予約 を変更する際は、数量を増減するだけで、解放される方法を変更することができます。キャパシティーの予約 のインスタンスタイプ、EBS 最適化、インスタンスストア設定、プラット

フォーム、アベイラビリティーゾーン、またはインスタンスの利用資格は変更できません。これらの属性を変更する必要がある場合は、予約をキャンセルし、必要な属性を持つ新しいものを作成することをお勧めします。

選択したインスタンスタイプの残りの オンデマンドインスタンス 制限を超える新しい数量を指定すると、その更新は失敗します。

コンソールを使用して キャパシティーの予約 を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [キャパシティーの予約] を選択し、キャパシティーの予約 を選択して、次に [Edit (編集)] を選択します。
3. 必要に応じて、[Quantity (数量)] または [Reservation ends (予約終了)] オプションを選択し、[Save changes (変更の保存)] を選択します。

AWS CLI を使用して キャパシティーの予約 を変更するには

`modify-capacity-reservations` コマンドを使用します。

```
aws ec2 modify-capacity-reservation --capacity-reservation-id reservation_id --instance-count quantity --end-date-type limited/unlimited --end-date expiration_date
```

## インスタンスの キャパシティーの予約 設定を変更する

停止したインスタンスの次のキャパシティーの予約設定は、いつでも変更できます。

- 一致する属性 (インスタンスタイプ、プラットフォーム、およびアベイラビリティーゾーン) と使用可能なキャパシティーを持つ任意の キャパシティーの予約 で起動します。
- 特定の キャパシティーの予約 でインスタンスを起動します。
- インスタンスが キャパシティーの予約 で起動しないようにします。

コンソールを使用して、インスタンスの キャパシティーの予約 設定を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンス] を選択し、変更するインスタンスを選択します。インスタンスをまだ停止していない場合は、停止します。
3. [アクション]、[キャパシティーの予約 設定を変更する] の順に選択します。
4. [キャパシティーの予約] で、以下のいずれかのオプションを選択します。
  - [Open (開く)] ——一致する属性 (インスタンスタイプ、プラットフォーム、およびアベイラビリティーゾーン) と使用可能なキャパシティーを持つ任意のオープンな キャパシティーの予約 でインスタンスを起動します。使用できるキャパシティーを持つ、一致する キャパシティーの予約 がない場合は、インスタンスはオンデマンドのキャパシティーを使用します。
  - <キャパシティーの予約> — このキャパシティーの予約でインスタンスを実行します。インスタンス属性 (インスタンスタイプ、プラットフォーム、およびアベイラビリティーゾーン) が選択した キャパシティーの予約 のインスタンス属性と一致しない場合、または選択した キャパシティーの予約 に十分なキャパシティーがない場合、インスタンスの起動は失敗します。
  - [なし] — キャパシティーの予約 でインスタンスを実行しないようにします。

AWS CLI を使用して、インスタンスの キャパシティーの予約 設定を変更するには

`modify-instance-capacity-reservation-attributes` コマンドを使用します。

```
aws ec2 modify-instance-capacity-reservation-attributes --instance-id instance_id --  
capacity-reservation-specification 'CapacityReservationPreference=none|open'
```

## キャパシティーの予約 の表示

キャパシティーの予約には次の状態があります。

- active—キャパシティー - を使用できます。
- expired—キャパシティーの予約は、予約リクエストで指定された日時に自動的に有効期限が切れました。リザーブドキャパシティーも使用できなくなります。
- cancelled—キャパシティーの予約は手動でキャンセルされました。リザーブドキャパシティーも使用できなくなります。
- pending—キャパシティーの予約リクエストは成功しましたが、キャパシティーのプロビジョニングはまだ保留中です。
- failed—キャパシティーの予約リクエストは失敗しました。無効なリクエストパラメータ、キャパシティー制約、またはインスタンス制限の制約のため、リクエストが失敗する可能性があります。失敗したリクエストを60分間表示できます。

コンソールを使用してキャパシティーの予約を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [キャパシティーの予約] を選択して、表示するキャパシティーの予約を選択します。
3. [この予約の起動インスタンスを表示する]。

AWS CLI を使用してキャパシティーの予約を表示するには

`describe-capacity-reservations` コマンドを使用します。

```
aws ec2 describe-capacity-reservations
```

## キャパシティーの予約 のキャンセル

リザーブドキャパシティーが不要になったら、いつでもキャパシティーの予約をキャンセルできます。キャパシティーの予約をキャンセルすると、キャパシティーが解放され、使用のために予約されなくなります。

空のキャパシティーの予約と実行中のインスタンスがあるキャパシティーの予約をキャンセルすることができます。実行中のインスタンスがあるキャパシティーの予約をキャンセルすると、インスタンスはキャパシティー予約外において標準のオンデマンドインスタンス料金で、あるいは一致する Savings Plan またはリージョンリザーブドインスタンスがある場合は割引料金で、正常に動作し続けます。

キャパシティーの予約をキャンセルすると、それをターゲットとするインスタンスは起動できなくなります。これらのインスタンスを異なるキャパシティーの予約をターゲットに設定するように変更し、一致する属性と十分なキャパシティーでオープンなキャパシティーの予約に起動するか、キャパシティーの予約への起動を回避します。詳細については、「[インスタンスのキャパシティーの予約 設定を変更する \(p. 437\)](#)」を参照してください。

コンソールを使用してキャパシティーの予約をキャンセルするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [キャパシティーの予約] を選択し、キャンセルするキャパシティーの予約を選択します。

- [Cancel reservation (予約をキャンセル)] 選択し、[Cancel reservation (予約をキャンセル)] を選択します。

AWS CLI を使用して キャパシティーの予約 をキャンセルするには

[cancel-capacity-reservation](#) コマンドを使用します。

```
aws ec2 cancel-capacity-reservation --capacity-reservation-id reservation_id
```

## 共有 キャパシティーの予約 の使用

キャパシティーの予約 共有を使用すると、キャパシティーの予約 の所有者がリザーブドキャパシティーを他の AWS アカウントと共有することも、AWS 組織内で共有することもできます。これにより、キャパシティーの予約 の作成と管理を一元的に行い、リザーブドキャパシティーを複数の AWS アカウント間または AWS 組織内で共有できます。

このモデルでは、キャパシティーの予約 を所有する AWS アカウント (所有者) が他の AWS アカウント (コンシューマー) との共有を行います。コンシューマーは、自身のアカウントで所有しているキャパシティーの予約にインスタンスを起動する場合と同じように、共有を受けているキャパシティーの予約にインスタンスを起動できます。キャパシティーの予約 の所有者は、共有したキャパシティーの予約 と、そこで起動したインスタンスを管理します。所有者は、共有したキャパシティーの予約 でコンシューマーが起動したインスタンスを変更することはできません。コンシューマーは、共有を受けているキャパシティーの予約 で起動したインスタンスを管理します。コンシューマーが、キャパシティーの予約 の所有者 や他のコンシューマーが所有するインスタンスを表示したり変更したりすることはできません。

キャパシティーの予約 の所有者が キャパシティーの予約 を共有できる相手は次のとおりです。

- AWS 組織内または組織外の特定の AWS アカウント
- AWS 組織内の組織単位
- AWS 組織全体

### コンテンツ

- [キャパシティーの予約 を共有するための前提条件 \(p. 439\)](#)
- [関連サービス \(p. 440\)](#)
- [アベイラビリティーゾーン間の共有 \(p. 440\)](#)
- [キャパシティーの予約 の共有 \(p. 440\)](#)
- [共有 キャパシティーの予約 の共有解除 \(p. 441\)](#)
- [共有 キャパシティーの予約 の特定 \(p. 442\)](#)
- [キャパシティーの予約 の使用状況の表示 \(p. 442\)](#)
- [共有 キャパシティーの予約 のアクセス許可 \(p. 442\)](#)
- [請求と使用量測定 \(p. 443\)](#)
- [インスタンス制限 \(p. 443\)](#)

## キャパシティーの予約 を共有するための前提条件

- 共有する キャパシティーの予約 は、AWS アカウント内で所有している必要があります。自身が共有を受けている キャパシティーの予約 を他者に共有することはできません。
- 共有テナントインスタンスの キャパシティーの予約 のみ共有できます。専用テナントインスタンス の キャパシティーの予約 は共有できません。

- 新規の AWS アカウントや、請求制限履歴のある AWS アカウントは、キャパシティーの予約 共有を使用できません。認定マスター(支払者)アカウントにリンクされているか、AWS 組織経由でリンクされている新規アカウントは、この制約を受けません。
- AWS 組織や AWS 組織内の組織単位と キャパシティーの予約 を共有するには、AWS Organizations との共有を有効にする必要があります。詳細については、AWS RAM ユーザーガイド の「[Enable Sharing with AWS Organizations](#)」を参照してください。

## 関連サービス

キャパシティーの予約 は AWS Resource Access Manager (AWS RAM) と連携します。AWS RAM は、任意の AWS アカウントを対象にするか AWS Organizations 経由で AWS リソースを共有するためのサービスです。AWS RAM を使用すると、リソース共有を作成することで、自身が所有するリソースを共有できます。リソース共有では、共有対象のリソースと、共有先となるコンシューマーを指定します。コンシューマーには、個人の AWS アカウントや、AWS Organizations 内の組織単位または組織全体を指定できます。

AWS RAM の詳細については、「[AWS RAM ユーザーガイド](#)」を参照してください。

## アベイラビリティーゾーン間の共有

リソースがリージョンの複数のアベイラビリティーゾーンに分散されるようにするために、アベイラビリティーゾーンは各アカウントの名前に個別にマッピングされます。このため、アカウントが異なると、アベイラビリティーゾーンの命名方法が異なる場合があります。たとえば、AWS アカウントのアベイラビリティーゾーン `us-east-1a` の場所は、別の AWS アカウントのアベイラビリティーゾーン `us-east-1a` の場所と異なる可能性があります。

自身のアカウントを基準にして キャパシティーの予約 の場所を特定するには、アベイラビリティーゾーン ID (AZ ID) を使用する必要があります。AZ ID は、すべての AWS アカウントで同じアベイラビリティーゾーンを一貫して示すための一意の識別子です。たとえば、`use1-az1` は `us-east-1` リージョンの AZ ID であり、すべての AWS アカウントで同じ場所を示します。

アカウントのアベイラビリティーゾーンの AZ ID を表示するには

1. AWS RAM コンソール (<https://console.aws.amazon.com/ram>) を開きます。
2. 現在のリージョンの AZ ID は、画面の右側にある [お客様の AZ ID] パネルに表示されます。

## キャパシティーの予約 の共有

自分が所有する キャパシティーの予約 を他の AWS アカウントに共有する場合、共有を受けたアカウントは、リザーブドキャパシティー内でインスタンスを起動できます。オープンな キャパシティーの予約 を共有する場合は、意図しない形で キャパシティーの予約 が使用されないよう、次の点に注意してください。

- キャパシティーの予約 の属性に一致するインスタンスをコンシューマーが実行している場合に、`CapacityReservationPreference` パラメータが `open` に設定され、リザーブドキャパシティー内の実行がまだであれば、共有 キャパシティーの予約 が自動的に使用されます。
- 属性 (インスタンスタイプ、プラットフォーム、アベイラビリティーゾーン) が一致するインスタンスをコンシューマーが起動する場合、`CapacityReservationPreference` パラメータが `open` に設定されていれば、自動的に共有 キャパシティーの予約 で起動されます。

キャパシティーの予約 を共有するには、リソース共有に追加する必要があります。リソース共有とは、自身のリソースを AWS アカウント間で共有するための AWS RAM リソースです。リソース共有では、共有対象のリソースと、共有先のコンシューマーを指定します。Amazon EC2 コンソールを使用して キャパシティーの予約 を共有すると、既存のリソース共有に追加されます。キャパシティーの予約 を新しいリソース共有に追加するには、[AWS RAM コンソール](#)を使用してリソース共有を作成する必要があります。

AWS Organizations 組織に属していて、組織内での共有が有効になっている場合、組織内のコンシューマーには共有 キャパシティーの予約に対するアクセス許可が自動的に付与されます。これに該当しない場合、コンシューマーはリソースへの参加の招待を受け取り、その招待を受け入れた後で、共有 キャパシティーの予約に対するアクセス許可が付与されます。

自身が所有する キャパシティーの予約 は、Amazon EC2 コンソール、AWS RAM コンソール、または AWS CLI を使用して共有できます。

Amazon EC2 コンソールを使用して、自身が所有する キャパシティーの予約 を共有するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[キャパシティーの予約] を選択します。
3. 共有する キャパシティーの予約 を選択し、[アクション]、[Share reservation] の順に選択します。
4. キャパシティーの予約 の追加先となるリソース共有を選択し、[Share キャパシティーの予約] を選択します。

コンシューマーから共有 キャパシティーの予約 にアクセスできるようになるまでに、数分かかることがあります。

AWS RAM コンソールを使用して、自身が所有する キャパシティーの予約 を共有するには

AWS RAM ユーザーガイド の「リソース共有の作成」を参照してください。

AWS CLI を使用して、自身が所有する キャパシティーの予約 を共有するには

`create-resource-share` コマンドを使用します。

## 共有 キャパシティーの予約 の共有解除

キャパシティーの予約 の所有者は、共有した キャパシティーの予約 をいつでも共有解除できます。共有した キャパシティーの予約 の共有解除を行う場合、以下のルールが適用されます。

- コンシューマーが所有し、共有解除の時点では共有キャパシティー内で実行されていたインスタンスは、リザーブドキャパシティー外で正常に実行を継続します。キャパシティーは、Amazon EC2 キャパシティーの可用性に応じて キャパシティーの予約 に復元されます。
- キャパシティーの予約 の共有先コンシューマーが、このリザーブドキャパシティーで新たにインスタンスを起動することはできません。

自身が所有している共有 キャパシティーの予約 の共有解除を行うには、まずリソース共有から削除する必要があります。この操作は、Amazon EC2 コンソール、AWS RAM コンソール、または AWS CLI を使用して行うことができます。

Amazon EC2 コンソールを使用して、自身が所有する共有 キャパシティーの予約 の共有解除を行うには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[キャパシティーの予約] を選択します。
3. 共有解除対象の キャパシティーの予約 を選択し、[共有] タブを選択します。
4. [共有] タブに、キャパシティーの予約 の追加先のリソース共有が一覧表示されます。キャパシティーの予約 を削除する対象のリソース共有を選択し、[リソース共有から削除] を選択します。

AWS RAM コンソールを使用して、自身が所有する共有 キャパシティーの予約 の共有解除を行うには

AWS RAM ユーザーガイド の「リソース共有の更新」を参照してください。

AWS CLI を使用して、自身が所有する共有 キャパシティーの予約 の共有解除を行うには

[disassociate-resource-share](#) コマンドを使用します。

## 共有 キャパシティーの予約 の特定

所有者とコンシューマーは、Amazon EC2 コンソールまたは AWS CLI を使用して、共有 キャパシティー の予約 を特定できます。

Amazon EC2 コンソールを使用して共有 キャパシティーの予約 を特定するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[キャパシティーの予約] を選択します。この画面には、自身が所有する キャパシティーの予約 と共有を受けている キャパシティーの予約 が一覧表示されます。[所有者] 列には、キャパシティーの予約 所有者の AWS アカウント ID が示されます。AWS アカウント ID の横に (me) と表示されている場合は、自身が所有者であることを示します。

AWS CLI を使用して共有 キャパシティーの予約 を特定するには

[describe-capacity-reservations](#) コマンドを使用します。このコマンドでは、自身が所有する キャパシティーの予約 および共有を受けている キャパシティーの予約 が返されます。OwnerId では、キャパシティーの予約 の所有者の AWS アカウント ID が返されます。

## キャパシティーの予約 の使用状況の表示

共有 キャパシティーの予約 の所有者は、Amazon EC2 コンソールまたは AWS CLI を使用して、いつでも 使用状況を表示できます。

Amazon EC2 コンソールを使用して キャパシティーの予約 の使用状況を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[キャパシティーの予約] を選択します。
3. 使用状況を表示する キャパシティーの予約 を選択し、[使用状況] タブを選択します。

[AWS アカウント ID] 列には、現在 キャパシティーの予約 を使用しているコンシューマーのアカウント ID が表示されます。[起動したインスタンス] 列には、リザーブドキャパシティー内で各コンシューマーが現在実行しているインスタンスの数が表示されます。

AWS CLI を使用して キャパシティーの予約 の使用状況を表示するには

[get-capacity-reservation-usage](#) コマンドを使用します。AccountId では、キャパシティーの予約 を使用しているアカウントのアカウント ID が表示されます。UsedInstanceCount では、リザーブドキャパシティー内でコンシューマーが現在実行しているインスタンスの数が表示されます。

## 共有 キャパシティーの予約 のアクセス許可

### 所有者のアクセス許可

共有 キャパシティーの予約 の管理とキャンセルは、所有者が行います。所有者は、共有 キャパシティー の予約 内で実行されており他のアカウントが所有するインスタンスを変更することはできません。共有 キャパシティーの予約 で起動されされたインスタンスは、所有者が管理します。

### コンシューマーのアクセス許可

コンシューマーは、共有 キャパシティーの予約 で実行している自身のインスタンスを管理します。コンシューマーは、共有 キャパシティーの予約 をどのように方法で変更することもできません。また、他のコ

ンシューマーまたは キャパシティーの予約 の所有者が所有するインスタンスを表示または変更することもできません。

## 請求と使用量測定

キャパシティーの予約 の共有に追加料金はかかりません。

キャパシティーの予約 の所有者には、キャパシティーの予約 内で自身が実行するインスタンスと、使用されていないリザーブドキャパシティーに対する料金が請求されます。コンシューマーには、共有 キャパシティーの予約 内で自身が実行するインスタンスに対する料金が請求されます。

## インスタンス制限

キャパシティーの予約 の使用量はすべて、キャパシティーの予約 の所有者の オンデマンドインスタンス制限の対象としてカウントされます。ここでは次の点について説明します。

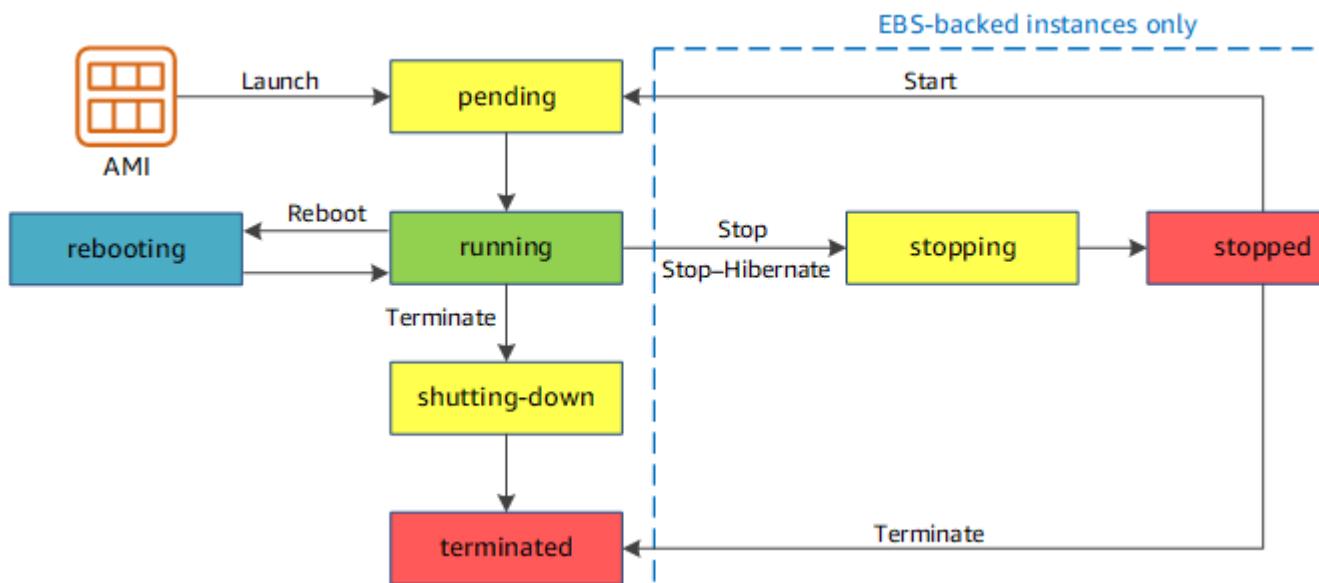
- 使用されていないリザーブドキャパシティ
- キャパシティーの予約 の所有者が所有するインスタンスによる使用量
- コンシューマーが所有するインスタンスによる使用量

共有キャパシティー内でコンシューマーによって起動されたインスタンスは、キャパシティーの予約 の所有者の オンデマンドインスタンス 制限の対象としてカウントされます。コンシューマーのインスタンス制限は、コンシューマー自身が所有する オンデマンドインスタンス の制限と、コンシューマーがアクセスできる共有 キャパシティーの予約 内で使用可能なキャパシティーの合計です。

# インスタンスのライフサイクル

インスタンスを起動した瞬間から終了まで、Amazon EC2 を使用してインスタンスを管理することにより、インスタンスでホストするアプリケーションまたはサイトを利用するお客様に最高の体験を提供することができます。

次の図は、インスタンス状態の遷移を示しています。instance store-backed インスタンスは停止および起動できることに注意してください。instance store-backed インスタンスの詳細については、「ルートデバイスのストレージ (p. 96)」を参照してください。



各インスタンスの状態の概要と、請求有無を次の表に示します。

Note

この表では、インスタンス使用のみの請求を示します。Amazon EBS ボリュームや Elastic IP アドレスなどの一部の AWS リソースでは、インスタンスの状態に関係なく、料金がかかります。詳細については、『AWS Billing and Cost Management ユーザーガイド』の「[予想外の料金の回避](#)」を参照してください。

インスタンスの状態	説明	インスタンス使用の請求
pending	インスタンスは <code>running</code> 状態への移行準備中です。初めて起動する場合、または <code>pending</code> 状態になってから起動する場合、インスタンスは <code>stopped</code> 状態になります。	課金されない
running	インスタンスは実行中で、使用できる状態です。	請求済み付
stopping	インスタンスは停止または停止休止の準備中です。	停止準備中の場合、課金されません 休止準備中の場合、課金されます
stopped	インスタンスはシャットダウンされているため、使用できません。インスタンスはいつでも起動できます。	課金されない
shutting down	インスタンスは削除準備中です。	課金されない
terminated	インスタンスは完全に削除されているため、起動することはできません。	課金されない  Note  終了したインスタンスに適用されるリザーブドインスタンスは、支払いオプションに従って、契約期間末まで請求が発生します。詳細については、「 <a href="#">リザーブドインスタンス (p. 279)</a> 」を参照してください。

Note

インスタンスは `running` 状態のため、インスタンスを再起動しても新しいインスタンスの請求期間が開始されることはありません。

## インスタンスの作成

インスタンスを起動すると、インスタンスは `pending` 状態に移行します。起動時に指定したインスタンスタイプによって、インスタンスのホストコンピュータのハードウェアが決定します。起動時に指定された Amazon Machine Image (AMI) を使って、インスタンスを再作成します。インスタンスの準備ができる

と、running 状態へ移行します。実行中のインスタンスに接続して、自分の前にあるコンピュータと同じように使用することができます。

インスタンスが running 状態に移行するとすぐに、インスタンスの実行時間に応じて（インスタンスがアイドル状態のままで、接続されていなくても）課金（秒単位、最低 1 分間分）が発生します。

詳細については、「[インスタンスの起動 \(p. 448\)](#)」および「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。

## インスタンスの停止と起動 (Amazon EBS-Backed インスタンスのみ)

インスタンスのステータスチェックに失敗するか、インスタンスでアプリケーションが想定通りに動作しておらず、インスタンスのルートボリュームが Amazon EBS である場合、インスタンスの停止と起動を行い、問題が解決するか試してみることができます。

インスタンスを停止した場合、インスタンスは stopping 状態に移行してから、stopped 状態になります。停止後のインスタンスに対しての使用料金やデータ転送料金が課金されることはありませんが、Amazon EBS ボリュームのストレージについては課金されます。インスタンスが stopped 状態の間、インスタンスタイプなど、インスタンスの特定の属性を変更できます。

インスタンスを起動すると、pending 状態に移行し、ほとんどの場合は新しいホストコンピュータに移動されます（ホストコンピュータに問題がない場合、インスタンスは同じホストコンピュータに残る可能性があります）。インスタンスの停止と起動を行うと、前のホストコンピュータ上のインスタンスストアボリューム上に存在していたすべてのデータが失われます。

プライベート IPv4 アドレスは保持されます。つまり、プライベート IPv4 アドレスまたはネットワークインターフェイスに関連付けられていた Elastic IP アドレスは、インスタンスとの関連付けが継続されるということです。インスタンスに IPv6 アドレスがある場合、IPv6 アドレスは保持されます。

インスタンスを stopped から running に移行するたびに、インスタンスの実行中は 1 秒単位で課金されます。インスタンスを起動するたびに、1 分間分の最低料金が請求されます。

詳細については、「[インスタンスの停止と起動 \(p. 529\)](#)」を参照してください。

## インスタンスの休止 (Amazon EBS Backed インスタンスのみ)

インスタンスを休止すると、オペレーティングシステムに休止を実行するように合図します（ディスクの停止）。これにより、内容がインスタンスのメモリ（RAM）から Amazon EBS ルートボリュームに保存されます。インスタンスの Amazon EBS ルートボリュームとアタッチされた Amazon EBS データボリュームは保持されます。インスタンスを起動すると、Amazon EBS ルートボリュームは以前の状態に復元され、RAM の内容が再ロードされます。以前にアタッチされたデータボリュームは再アタッチされ、インスタンスはそのインスタンス ID を保持します。

インスタンスを休止した場合、インスタンスは stopping 状態に移行してから、stopped 状態になります。休止状態にあるインスタンスが stopped 状態にある間はに課金しませんが、休止せずに [インスタンスを停止 \(p. 445\)](#) したときとは異なり、stopping 状態にある間は課金します。データ転送料金に対して使用料を課金しませんが、RAM データのストレージを含め、Amazon EBS ボリュームのストレージに対する課金します。

休止したインスタンスを起動すると、そのインスタンスは pending 状態に移行し、ほとんどの場合、新しいホストコンピュータに移動されます。ホストコンピュータに問題がない場合、インスタンスは同じホストコンピュータに残る可能性があります。

プライベート IPv4 アドレスは保持されます。つまり、プライベート IPv4 アドレスまたはネットワークインターフェイスに関連付けられていた Elastic IP アドレスは、インスタンスとの関連付けが継続されるとということです。インスタンスに IPv6 アドレスがある場合、IPv6 アドレスは保持されます。

詳細については、「[Linux インスタンスの休止 \(p. 532\)](#)」を参照してください。

## インスタンスの再起動

Amazon EC2 コンソール、コマンドラインツール、Amazon EC2 API を使って、インスタンスを再起動できます。インスタンスからオペレーティングシステムの再起動コマンドを実行する代わりに、Amazon EC2 を使ってインスタンスを再起動することをお勧めします。

インスタンスの再起動は、オペレーティングシステムの再起動と同等です。インスタンスは同じホストコンピュータに残り、そのパブリック DNS 名、プライベート IP アドレス、およびその他のデータをインスタンスストアボリュームに維持します。通常、再起動が完了するまでに数分かかりますが、再起動に必要な時間は、インスタンスの設定によって異なります。

インスタンスを再起動しても、新しいインスタンスの課金時間(秒単位、最低 1 分間分の課金はなし)は開始されません。

詳細については、「[インスタンスの再起動 \(p. 542\)](#)」を参照してください。

## インスタンスのリタイア

インスタンスをホストしている基盤のハードウェアで回復不可能な障害が検出されると、AWS によってインスタンスのリタイアが予定されます。予定されたリタイア日になると、インスタンスは AWS によって停止または削除されます。インスタンスのルートデバイスが Amazon EBS ボリュームである場合、インスタンスは停止されますが、その後いつでも再び起動できます。インスタンスのルートデバイスがインスタンスストアボリュームである場合、インスタンスは削除し、再び使用することはできません。

詳細については、「[インスタンスのリタイア \(p. 543\)](#)」を参照してください。

## インスタンスの削除

インスタンスが必要なくなったら、削除することができます。インスタンスのステータスが `shutting-down` または `terminated` に変わったら、そのインスタンスへの課金は停止します。

停止保護が有効な場合、コンソール、CLI、または API を使用してインスタンスを削除することはできません。

インスタンスの削除後、インスタンスはしばらくの間コンソールに表示されたままで、エントリは自動的に削除されます。CLI および API を使って、削除したインスタンスを記述することもできます。(タグなどの) リソースは削除されたインスタンスから徐々に関連付けが解除されるため、しばらくすると、削除されたインスタンスで表示されなくなる可能性があります。削除したインスタンスへの接続や復旧はできません。

Amazon EBS-Backed インスタンスはそれぞれ、`InstanceInitiatedShutdownBehavior` 属性をサポートしています。この属性は、インスタンス自体からシャットダウンを開始した場合 (Linux で `shutdown` コマンドを使用した場合など)、インスタンスを停止または終了するかを制御します。デフォルトの動作は、インスタンスの停止です。インスタンスの実行中または停止中に、この属性の設定を変更できます。

各 Amazon EBS ボリュームは `DeleteOnTermination` 属性をサポートします。この属性は、アタッチされたインスタンスを終了するときには、ボリュームの削除や保持を制御します。デフォルトでは、ルートデバイスボリュームを削除し、それ以外に EBS ボリュームがあれば保持します。

詳細については、「[インスタンスの終了 \(p. 545\)](#)」を参照してください。

## 再起動、停止、休止、終了の違い

次の表に、インスタンスの再起動、停止、休止、終了の主な違いをまとめました。

特徴	再起動	停止/開始 (Amazon EBS-Backed インスタンスのみ)	休止 (Amazon EBS Backed インスタンスのみ)	終了
ホストコンピュータ	インスタンスは、同じホストコンピュータで保持される	ほとんどの場合、インスタンスは新しいホストコンピュータに移動されます。(ホストコンピュータに問題がない場合、インスタンスは同じホストコンピュータに残る可能性があります)。	ほとんどの場合、インスタンスは新しいホストコンピュータに移動されます。(ホストコンピュータに問題がない場合、インスタンスは同じホストコンピュータに残る可能性があります)。	なし
プライベート IPv4 アドレスとパブリック IPv4 アドレス	同一のまま保持される	インスタンスはプライベート IPv4 アドレスを保持します。インスタンスは、Elastic IP アドレス (停止/起動の際に変更されない) を持っていない限り、新しいパブリック IPv4 アドレスを取得します。	インスタンスはプライベート IPv4 アドレスを保持します。インスタンスは、Elastic IP アドレス (停止/起動の際に変更されない) を持っていない限り、新しいパブリック IPv4 アドレスを取得します。	なし
Elastic IP アドレス (IPv4)	Elastic IP アドレスは、インスタンスに関連付けられたまま維持される	Elastic IP アドレスは、インスタンスに関連付けられたまま維持される	Elastic IP アドレスは、インスタンスに関連付けられたまま維持される	Elastic IP アドレスはインスタンスの関連付けが解除される
IPv6 アドレス	アドレスは同一のまま保持される	インスタンスは、IPv6 アドレスを保持する	インスタンスは、IPv6 アドレスを保持する	なし
インスタンスストアボリューム	データは保持される	データは消去される	データは消去される	データは消去される
ルートデバイスボリューム	ボリュームは保持される	ボリュームは保持される	ボリュームは保持される	ボリュームはデフォルトで削除される
RAM (メモリの内容)	RAM は消去される	RAM は消去される	RAM はルートボリュームにあるファイルに保存される	RAM は消去される
請求	インスタンスの課金時間は変更されません。	インスタンスの状態が <code>stopping</code> になるとすぐに、そのインスタンスへの課金が停止されます。インスタンスが <code>stopped</code> 状態から <code>running</code> 状態に	インスタンスが <code>stopping</code> 状態にある間は課金されますが、そのインスタンスが <code>stopped</code> 状態にある場合、課金は停止します。インスタン	インスタンスの状態が <code>shutting-down</code> に変わるとすぐに、そのインスタンスへの課金が停止されます。

特徴	再起動	停止/開始 (Amazon EBS-Backed インスタンスのみ)	休止 (Amazon EBS Backed インスタンスのみ)	終了
		移行するたびに、新しいインスタンスの課金(起動ごとに最低料金1分間分)が開始されます。	スが stopped 状態から running 状態に移行するたびに、新しいインスタンスの課金(起動ごとに最低料金1分間分)が開始されます。	

オペレーティングシステムのシャットダウンコマンドを実行すると、instance store-backed インスタンスは必ず停止されます。オペレーティングシステムのシャットダウンコマンドによって Amazon EBS-backed インスタンスを停止または終了するかどうかを制御できます。詳細については、「[インスタンスによって起動されたシャットダウン動作の変更 \(p. 548\)](#)」を参照してください。

## インスタンスの起動

インスタンスとは AWS クラウドにある仮想サーバーです。Amazon Machine Image (AMI) からインスタンスを起動します。AMI はインスタンスに対して、オペレーティングシステム、アプリケーションサーバー、およびアプリケーションを提供します。

AWS にサインアップすると、[AWS 無料利用枠](#)を使って、Amazon EC2 を無料で開始することができます。無料利用枠を利用すれば、12か月無料でマイクロインスタンスを起動および使用できます。無料利用枠に含まれないインスタンスを起動する場合は、そのインスタンスの通常の Amazon EC2 使用料がかかります。詳細については、「[Amazon EC2 料金表](#)」を参照してください。

次の方法を使用してインスタンスを起動できます。

方法	ドキュメント
[Amazon EC2 コンソール] インスタンス起動ウィザードを使用して、起動パラメータを指定します。	<a href="#">インスタンス起動ウィザードを使用してインスタンスを起動する (p. 449)</a>
[Amazon EC2 コンソール] 起動テンプレートを作成して、起動テンプレートからインスタンスを起動します。	<a href="#">起動テンプレートからのインスタンスの起動 (p. 454)</a>
[Amazon EC2 コンソール] 既存のインスタンスを基本として使用します。	<a href="#">既存のインスタンスのパラメータを使用してインスタンスを起動 (p. 465)</a>
[Amazon EC2 コンソール] 作成した Amazon EBS スナップショットを使用します。	<a href="#">バックアップからの Linux インスタンスの起動 (p. 466)</a>
[Amazon EC2 コンソール] AWS Marketplace から購入した AMI を使用します。	<a href="#">AWS Marketplace インスタンスの起動 (p. 467)</a>
[AWS CLI] 選択した AMI を使用します。	<a href="#">AWS CLI で Amazon EC2 を使用する</a>
[AWS Tools for Windows PowerShell] 選択した AMI を使用します。	<a href="#">AWS Tools for Windows PowerShell からの Amazon EC2</a>
[AWS CLI] EC2 フリートを使用し、さまざまな EC2 インスタンスタイプおよびアベイラビリティーゾーン間で、および オンデマンドインスタ	<a href="#">EC2 フリートの起動 (p. 468)</a>

方法	ドキュメント
ンス、リザーブドインスタンス、スポットインスタンス 購入モデル間で容量をプロビジョニングします。	

インスタンスを起動する場合、次のいずれかのリソースに関連付けられているサブネットでインスタンスを起動できます。

- ・アベイラビリティーゾーン - このオプションはデフォルトです。
- ・ローカルゾーン - ローカルゾーンでインスタンスを起動するには、この機能にオプトインする必要があります。詳細については、「[ローカルゾーンへのオプトイン](#)」を参照してください。
- ・アウトポスト - アウトポストでインスタンスを起動するには、アウトポストを作成する必要があります。アウトポストの作成方法については、AWS Outposts ユーザーガイドの「[AWS Outposts の開始方法](#)」を参照してください。

インスタンスを起動した後、インスタンスに接続して使用できます。最初、インスタンスの状態は `pending` です。インスタンスの状態が `running` の場合、インスタンスは起動を開始します。インスタンスに接続するまで、少し時間がかかることがあります。インスタンスは、パブリック DNS 名を受信します。この DNS 名はインターネットからインスタンスに接続する場合に使用できます。また、インスタンスはプライベート DNS 名も受け取ります。これは、同じ VPC 内の他のインスタンスがインスタンスに接続するために使用できます。インスタンスへの接続の詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。

インスタンスを使い終わったら、必ずインスタンスを終了してください。詳細については、「[インスタンスの終了 \(p. 545\)](#)」を参照してください。

## インスタンス起動ウィザードを使用してインスタンスを起動する

インスタンスを起動する前に、セットアップが終了していることを確認してください。詳細については、「[Amazon EC2 でのセットアップ \(p. 22\)](#)」を参照してください。

### Important

AWS 無料利用枠に含まれないインスタンスを起動すると、アイドル状態であっても、インスタンスの実行中は料金が発生します。

## AMI からのインスタンスの起動

インスタンスを起動するときに、Amazon Machine Image (AMI) と呼ばれる設定を選択する必要があります。AMI には、新しいインスタンスの作成に必要な情報が含まれています。たとえば、ある AMI にはウェブサーバーとして動作するために必要なソフトウェア (Linux、Apache、ウェブサイトなど) が格納されます。

### Tip

インスタンスの起動を高速化するには、大きなリクエストをより小さなバッチに分割します。たとえば、1つの起動リクエストに 500 インスタンスが含まれている場合は、それを 5 つの起動リクエスト (各 100 インスタンス) に分割します。

## インスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 画面の上のナビゲーションバーで、現在のリージョンが表示されます (例: 米国東部 (オハイオ))。ニーズを満たすインスタンスのリージョンを選択します。一部の Amazon EC2 リソースはリージョン間で

共有できるため、この選択は重要です。詳細については、「[リソースの場所 \(p. 1110\)](#)」を参照してください。

3. Amazon EC2 コンソールダッシュボードで、[Launch Instance] を選択します。
4. [Choose an Amazon Machine Image (AMI)] ページで、次のように AMI を選択します。
  - a. 左ペインで、使用する AMI のタイプを選択します。

#### クリックスタート

すぐに作業を開始できるように、一般的な AMI を選択します。無料利用枠の対象となる AMI を選択するには、左ペインで [無料利用枠のみ] を選択しますこれらの AMI は [Free tier eligible] と表示されます。

#### マイ AMI

お客様が所有しているプライベート AMI、またはお客様が共有しているプライベート AMI。お客様と共に共有されている AMI を表示するには、左ペインの [Shared with me] を選択します。

#### AWS Marketplace

AMI も含めて、AWS で実行するソフトウェアを購入できるオンラインストア。AWS Marketplace からのインスタンスの起動の詳細については、[AWS Marketplace インスタンスの起動 \(p. 467\)](#) を参照してください。

#### コミュニティ AMI

AWS コミュニティのメンバーが、メンバー以外でも使用できるようにした AMI。オペレーティングシステムを条件として AMI のリストをフィルタリングするには、[Operating system] の該当するチェックボックスをオンにします。アーキテクチャおよびルートデバイスタイプを条件としてフィルタリングすることもできます。

- b. 各 AMI の [Root device type] を確認します。必要なタイプはどの AMI かに注意してください。タイプは `ebs` (Amazon EBS でバックアップ) または `instance-store` (インスタンスストアでバックアップ) です。詳細については、「[ルートデバイスのストレージ \(p. 96\)](#)」を参照してください。
  - c. 各 AMI の [Virtualization type] を確認します。必要なタイプはどの AMI かに注意してください。タイプは `hvm` または `paravirtual` です。たとえば、一部のインスタンスタイプには HVM が必要です。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。
  - d. ニーズを満たす AMI を選択し、[Select] を選択します。
5. [Choose an Instance Type] ページで、起動するインスタンスのハードウェア設定とサイズを選択します。インスタンスタイプが大きくなると、CPU およびメモリも増えます。詳細については、「[インスタンスタイプ \(p. 183\)](#)」を参照してください。

無料利用枠の対象とするには、[t2.micro] インスタンスタイプを選択します。詳細については、「[バースト可能パフォーマンスインスタンス \(p. 199\)](#)」を参照してください。

デフォルトでは、ウィザードには現行世代のインスタンスタイプが表示され、お客様が選択した AMI に基づいて使用可能な最初のインスタンスタイプが選択されます。旧世代のインスタンスタイプを表示するには、フィルタリストから [All generations] を選択します。

#### Note

テスト目的でインスタンスをすばやくセットアップする必要がある場合は、[Review and Launch] を選択し、デフォルトの設定を受け入れてインスタンスを起動します。それ以外の場合は、インスタンスをさらに設定するために、[Next: Configure Instance Details] を選択します。

6. [Configure Instance Details] ページで、必要に応じて次の設定を変更し (すべての設定を表示するには [Advanced Details] を展開)、[Next: Add Storage] を選択します。
  - [Number of instances]: 起動するインスタンスの数を入力します。

- (オプション) アプリケーションで需要を処理するためにインスタンスの正しい数を確実に維持するには、[Launch into Auto Scaling Group (Auto Scaling グループに作成する)] を選択して起動設定と Auto Scaling グループを作成します。Auto Scaling によって、指定どおりにグループのインスタンス数がスケーリングされます。詳細については、「[Amazon EC2 Auto Scaling ユーザーガイド](#)」を参照してください。
- [購入のオプション]: [スポットインスタンスのリクエスト] を選択してスポットインスタンスを起動します。このページからオプションの追加と削除を行います。上限価格を設定し、必要に応じてリクエストタイプ、中断動作、およびリクエストの有効性を更新します。詳細については、「[スポットインスタンス リクエストを作成する \(p. 339\)](#)」を参照してください。
- [Network]: VPC を選択します。新しい VPC を作成するには、[Create new VPC] を選択して Amazon VPC コンソールに移動します。終了したらウィザードに戻り、[Refresh] を選択して一覧に VPC を読み込みます。
- [サブネット]: インスタンスは、アベイラビリティゾーン、ローカルゾーン、またはアウトポストに関連付けられたサブネットで起動できます。

アベイラビリティゾーンでインスタンスを起動するには、インスタンスを起動するサブネットを選択します。[指定なし] を選択して、AWS で任意のアベイラビリティゾーンのデフォルトサブネットを自動的に選択できます。新しいサブネットを作成するには、[Create new subnet] を選択して Amazon VPC コンソールに移動します。終了したらウィザードに戻り、[Refresh] を選択して一覧にサブネットを読み込みます。

ローカルゾーンでインスタンスを起動するには、ローカルゾーンで作成したサブネットを選択します。

アウトポストでインスタンスを起動するには、アウトポストに関連付けられた VPC 内のサブネットを選択します。

- [Auto-assign Public IP]: インスタンスがパブリック IPv4 アドレスを受け取るかどうかを指定します。デフォルトで、デフォルトのサブネットにあるインスタンスはパブリック IPv4 アドレスを受け取り、デフォルト以外のサブネットにあるインスタンスは受け取れません。[Enable] または [Disable] を選択すると、これがサブネットのデフォルト設定より優先されます。詳細については、「[パブリック IPv4 アドレスと外部 DNS ホスト名 \(p. 686\)](#)」を参照してください。
- [Auto-assign IPv6 IP]: インスタンスがサブネットの範囲から IPv6 アドレスを受け取るかどうかを指定します。[Enable] または [Disable] を選択すると、これによりサブネットのデフォルト設定がオーバーライドされます。このオプションは IPv6 CIDR ブロックを VPC とサブネットに関連付けた場合にのみ使用できます。詳細については、Amazon VPC ユーザーガイドの「[VPC とサブネット](#)」を参照してください。
- キャパシティーの予約: インスタンスを共有キャパシティーに起動するか既存の キャパシティーの予約に起動するかを指定します。詳細については、「[既存の キャパシティーの予約へのインスタンスの起動 \(p. 436\)](#)」を参照してください。
- [IAM ロール]: インスタンスに関連付ける AWS Identity and Access Management (IAM) ロールを選択します。詳細については、「[Amazon EC2 の IAM ロール \(p. 888\)](#)」を参照してください。
- CPU オプション: 起動中に [CPU オプションを指定] を選択して、カスタム数の vCPU を指定します。CPU コアの数とコアごとのスレッド数を設定します。詳細については、「[CPU オプションの最適化 \(p. 571\)](#)」を参照してください。
- [Shutdown behavior]: シャットダウン時にインスタンスを停止するか終了するかを選択します。詳細については、「[インスタンスによって起動されたシャットダウン動作の変更 \(p. 548\)](#)」を参照してください。
- [Stop - Hibernate behavior]: 休止を有効にするには、このチェックボックスをオンにします。このオプションは、インスタンスが休止の前提条件を満たしている場合にのみ使用できます。詳細については、「[Linux インスタンスの休止 \(p. 532\)](#)」を参照してください。
- [Enable termination protection]: 偶発的な終了を防ぐには、このチェックボックスをオンにします。詳細については、「[インスタンスの削除保護の有効化 \(p. 547\)](#)」を参照してください。

- [Monitoring]: Amazon CloudWatch を使用したインスタンスの詳細モニタリングを有効にするには、このチェックボックスをオンにします。追加の変更が適用されます。詳細については、「[CloudWatch を使用したインスタンスのモニタリング \(p. 642\)](#)」を参照してください。
- [EBS 最適化インスタンス]: Amazon EBS 最適化インスタンスは、最適化された設定スタックを使用し、Amazon EBS I/O に対して追加の専用の容量を提供します。インスタンスタイプがこの機能をサポートしている場合は、このチェックボックスをオンにして有効にします。追加の変更が適用されます。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。
- [Tenancy]: VPC でインスタンスを起動する場合、独立した専用のハードウェア ([Dedicated]) または Dedicated Host ([Dedicated host]) を選択できます。追加料金が適用される場合があります。詳細については、「[ハードウェア専有インスタンス \(p. 425\)](#)」および「[Dedicated Hosts \(p. 395\)](#)」を参照してください。
- [T2/T3 無制限]: このチェックボックスをオンにすると、アプリケーションがベースラインを越えて必要なだけバーストできるようになります。追加料金が適用される場合があります。詳細については、「[バースト可能パフォーマンスインスタンス \(p. 199\)](#)」を参照してください。
- [Network interfaces]: 特定のサブネットを選択すると、インスタンスに対して最大 2 つのネットワークインターフェイスを指定できます。
  - [Network Interface] で、[New network interface] を選択して AWS によって新しいインターフェイスを作成するか、既存の使用できるネットワークインターフェイスを選択します。
  - [Primary IP] で、サブネットの範囲からプライベート IPv4 アドレスを入力するか、[Auto-assign] をデフォルトのままにしてプライベート IPv4 アドレスが自動的に選択されるようにします。
  - 選択したネットワークインターフェイスに対して複数のプライベート IPv4 アドレスを割り当てるには、[Secondary IP addresses] で [Add IP] を選択します。
  - (IPv6 のみ) [IPv6 IP] で、[Add IP] を選択し、サブネットの範囲から IPv6 アドレスを入力するか、[Auto-assign] をデフォルトのままにして IPv6 アドレスが自動的に選択されるようにします。
  - [Add Device] を選択して、セカンダリネットワークインターフェイスを追加します。セカンダリネットワークインターフェイスは、インスタンスと同じアベイラビリティーゾーンにある場合は、VPC の別のサブネットに存在できます。

詳細については、「[Elastic Network Interface \(p. 713\)](#)」を参照してください。複数のネットワークインターフェイスを指定した場合、インスタンスはパブリック IPv4 アドレスを受け取ることはできません。さらに、eth0 に既存のネットワークインターフェイスを指定した場合、[Auto-assign Public IP] を使用してサブネットのパブリック IPv4 設定をオーバーライドする操作は禁止されます。詳細については、「[インスタンス起動時のパブリック IPv4 アドレスの割り当て \(p. 691\)](#)」を参照してください。

- [カーネル ID]: (準仮想化 (PV) AMI でのみ有効) 特定のカーネルを使用する場合を除き、[デフォルトを使用] を選択します。
- [RAM ディスク ID]: (準仮想化 (PV) AMI でのみ有効) 特定の RAM ディスクを使用する場合を除き、[デフォルトを使用] を選択します。カーネルを選択した場合は、サポートするドライバーとともに特定の RAM ディスクを選択しなければならない可能性があります。
- [プレイスメントグループ]: プレイスマントグループは、インスタンスの配置戦略を決定します。既存のプレイスメントグループを選択するか、新しいグループを作成します。このオプションは、プレイスメントグループをサポートするインスタンスタイプを選択した場合にのみ使用できます。詳細については、「[プレイスメントグループ \(p. 791\)](#)」を参照してください。
- [ユーザーデータ]: 起動時にインスタンスを設定するユーザーデータ、または設定スクリプトを実行するユーザーデータを指定できます。ファイルを添付するには、[As file] オプションを選択し、添付するファイルを参照します。

7. 選択した AMI には、ルートデバイスピリュームを含む、1 つまたは複数のストレージボリュームが含まれます。[Add Storage] ページで、[Add New Volume] を選択することにより、インスタンスにアタッチする追加ボリュームを指定できます。各ボリュームを次のように設定し、[Next: Add Tags (次へ: タグの追加)] を選択します。

- [Type (タイプ)]: インスタンスと関連付けるインスタンスストアまたは Amazon EBS ボリュームを選択します。一覧で利用できるボリュームの種類は、選択したインスタンスタイプに応じて異なります。詳細については、「[Amazon EC2 インスタンスストア \(p. 1076\)](#)」および「[Amazon EBS ボリューム \(p. 931\)](#)」を参照してください。
  - [Device [デバイス]]: ボリュームで利用できるデバイス名の一覧から選択します。
  - [Snapshot (スナップショット)]: ボリュームを復元するスナップショットの名前または ID を入力します。[Snapshot (スナップショット)] フィールドにテキストを入力して、利用できる共有スナップショットとパブリックスナップショットを検索することもできます。スナップショットの説明では大文字と小文字が区別されます。
  - [Size (サイズ)]: EBS ボリュームの場合、ストレージサイズを指定できます。無料利用枠の対象となる AMI とインスタンスを選択した場合でも、無料利用枠内に収めるには、合計ストレージを 30 GiB 以下に維持する必要があります。詳細については、「[EBS ボリュームのサイズと設定の制限 \(p. 946\)](#)」を参照してください。
  - [Volume Type (ボリュームタイプ)]: EBS ボリュームの場合、ボリュームタイプを選択します。詳細については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。
  - [IOPS]: プロビジョンド IOPS SSD ボリュームタイプを選択した場合は、ボリュームがサポートできる I/O オペレーション/秒 (IOPS) を入力できます。
  - [Delete on Termination (終了時に削除)]: Amazon EBS ボリュームについては、インスタンスが終了したときにボリュームを削除するには、このチェックボックスをオンにします。詳細については、「[インスタンスの削除で Amazon EBS ボリュームを保持する \(p. 549\)](#)」を参照してください。
  - [Encrypted (暗号化)]: インスタンスタイプが EBS 暗号化をサポートしている場合、ボリュームの暗号化状態を指定できます。このリージョンでデフォルトで暗号化を有効にした場合、デフォルトの CMK が選択されます。別のキーを選択するか、暗号化を無効にすることができます。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。
8. [Add Tags] ページで、キーと値の組み合わせを [タグ \(p. 1120\)](#) として指定します。インスタンス、ボリューム、またはその両方にタグ付けできます。スポットインスタンスでは、スポットインスタンスリクエストにのみタグ付けできます。リソースに複数のタグを追加するには、[Add another tag] を選択します。完了したら、[次の手順: セキュリティグループの設定] を選択します。
  9. [Configure Security Group] ページで、セキュリティグループを使用してインスタンスのファイアウォールルールを定義しますこのルールでは、どの着信ネットワークトラフィックをインスタンスに配信するかを指定します。他のトラフィックはすべて無視されます。(セキュリティグループの詳細については、「[Linux インスタンスの Amazon EC2 セキュリティグループ \(p. 911\)](#)」を参照してください)。以下のようにセキュリティグループを選択または作成して、[Review and Launch] を選択します。
    - a. 既存のセキュリティグループを選択するには、[Select an existing security group (既存のセキュリティグループの選択)] を選択してから、セキュリティグループを選択します既存のセキュリティグループのルールを編集することはできません。しかし、[Copy to new (コピーして新規作成)] を選択して、新しいグループにルールをコピーすることはできます。その後、次の手順で説明しているように、ルールを追加できます。
    - b. 新しいセキュリティグループを作成するには、[Create a new security group (新しいセキュリティグループの作成)] を選択します。このウィザードでは、launch-wizard-x セキュリティグループが自動的に定義され、SSH (ポート 22) を介したインスタンスへの接続を許可するインバウンドルールが作成されます。
    - c. ニーズに応じたルールを追加できます。たとえば、インスタンスがウェブサーバーである場合は、ポート 80 (HTTP) とポート 443 (HTTPS) を開いて、インターネットトラフィックを許可します。

ルールを追加するには、[Add Rule] を選択し、プロトコルを選択してネットワークトラフィックを開いてから、ソースを指定します。[Source] リストから [My IP] を選択し、ウィザードでコンピュータのパブリック IP アドレスを追加します。ただし、ISP 経由で、またはファイアウォールの内側から静的な IP アドレスなしで接続している場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。

### Warning

すべての IP アドレス (0.0.0.0/0) に SSH または RDP を介したインスタンスへのアクセスを許可するルールは、この短期間の実習では許容されますが、本番稼働用環境では安全ではありません。特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを限定してください。

10. [Review Instance Launch] ページで、インスタンスの詳細をチェックし、適切な [Edit] リンクを選択して必要な変更を加えます。

準備ができたら、[Launch] を選択します。

11. [Select an existing key pair or create a new key pair] ダイアログボックスで、既存のキーペアを選択するか、新しいキーペアを作成できます。たとえば、[Choose an existing key pair] を選択し、セットアップ中に作成したキーペアを選択します。

インスタンスを起動するには、確認のチェックボックスをオンにし、続いて [Launch Instances] を選択します。

### Important

[Proceed without key pair] オプションを選択した場合、ユーザーが別の方でログインすることを許可するように設定された AMI を選択した場合でなければ、インスタンスに接続できなくなります。

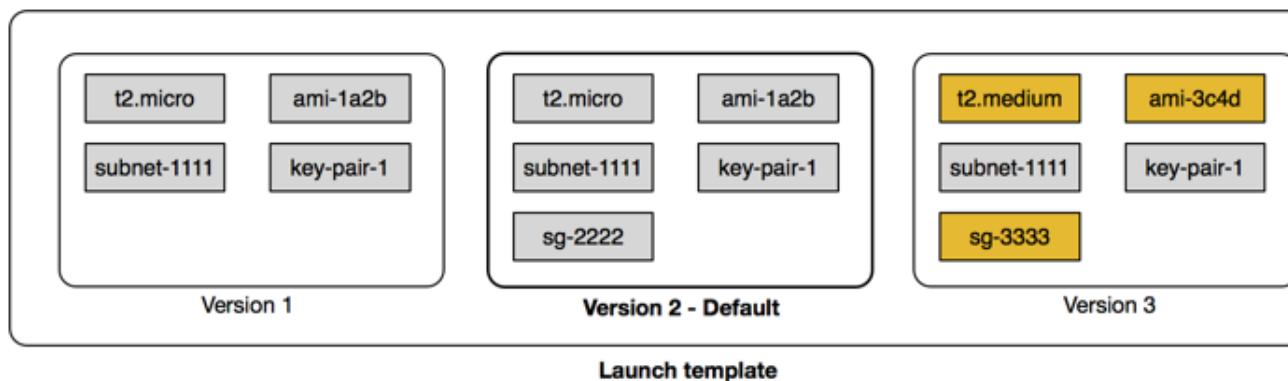
12. (オプション) インスタンスのステータスチェックアラームを作成することもできます(追加料金がかかります)。(不明な場合は、後からいつでも追加できます。) 確認画面で、[Create status check alarms] を選択して、指示にしたがいます。詳細については、「[ステータスチェックアラームの作成と編集 \(p. 631\)](#)」を参照してください。
13. インスタンスが起動しないか、状態が `terminated` ではなくすぐに `running` になる場合は、「[インスタンスの起動に関する問題のトラブルシューティング \(p. 1133\)](#)」を参照してください。

## 起動テンプレートからのインスタンスの起動

インスタンスを起動するための設定情報を含む起動テンプレートを作成できます。起動テンプレートにより起動パラメータを格納でき、インスタンスを起動するたびに指定する必要がなくなります。たとえば、AMI ID やインスタンスタイプ、通常インスタンスの起動に使用しているネットワーク設定を起動テンプレートに含めることができます。Amazon EC2 コンソール、AWS SDK、またはコマンドラインツールを使用してインスタンスを起動する場合、使用的起動テンプレートを指定できます。

各起動テンプレートについて、1つ以上の番号付きの起動テンプレートのバージョンを作成できます。各バージョンに異なる起動パラメータを指定できます。起動テンプレートからインスタンスを起動する際、起動テンプレートのいずれかのバージョンを使用できます。バージョンを指定しない場合は、デフォルトバージョンが使用されます。いずれかの起動テンプレートをデフォルトバージョンとして設定できます—デフォルトでは、起動テンプレートの最初のバージョンです。

以下の図は、3つのバージョンの起動テンプレートを示しています。最初のバージョンでは、インスタンスの起動に使用するインスタンスタイプ、AMI ID、サブネット、およびキーペアが指定されています。2番目のバージョンは最初のバージョンに基づいており、インスタンスのセキュリティグループも指定しています。3番目のバージョンは、パラメータの一部に異なる値を使用しています。バージョン 2 がデフォルトバージョンとして設定されています。この起動テンプレートからインスタンスを起動すると、他のバージョンを指定しない限りバージョン 2 の起動パラメータが使用されます。



## コンテンツ

- [起動テンプレートの制限 \(p. 455\)](#)
- [起動テンプレートを使用して起動パラメータを制御する \(p. 455\)](#)
- [起動テンプレートの使用の管理 \(p. 456\)](#)
- [起動テンプレートの作成 \(p. 456\)](#)
- [起動テンプレートのバージョンの管理 \(p. 461\)](#)
- [起動テンプレートからのインスタンスの起動 \(p. 463\)](#)
- [Amazon EC2 Auto Scaling を使用して起動テンプレートを使用する \(p. 464\)](#)
- [EC2 フリート を使用して起動テンプレートを使用する \(p. 464\)](#)
- [スポットフリート を使用して起動テンプレートを使用する \(p. 464\)](#)
- [起動テンプレートの削除 \(p. 464\)](#)

## 起動テンプレートの制限

起動テンプレートおよび起動テンプレートのバージョンには次のルールが適用されます。

- 1 つのリージョンあたり 5,000 の起動テンプレート、1 つの起動テンプレートあたり 10,000 のバージョンに作成が制限されています。
- 起動テンプレートのパラメータはオプションです。ただし、テンプレートには、インスタンス起動のためのリクエストに必要なすべてのパラメータが含まれている必要があります。たとえば、起動テンプレートに AMI ID が含まれていない場合、インスタンスの起動時に起動テンプレートと AMI ID の両方を指定する必要があります。
- 起動テンプレートパラメータは、起動テンプレート作成の際には検証されません。パラメータに正しい値を指定したか、およびサポートされているパラメータの組み合わせを使用しているかを確認します。たとえば、プレイスメントグループ内でインスタンスを起動するには、サポートされているインスタンスタイプを指定する必要があります。
- 起動テンプレートにはタグ付けできますが、起動テンプレートのバージョンにはタグ付けできません。
- 起動テンプレートのバージョンには、作成された順序で番号が付けられます。起動テンプレートのバージョンを作成する場合、自分でバージョン番号を指定することはできません。

## 起動テンプレートを使用して起動パラメータを制御する

起動テンプレートには、インスタンスを起動するためのパラメータすべてまたは一部を含めることができます。起動テンプレートを使用してインスタンスを起動するときは、起動テンプレートで指定されたパラメータを上書きできます。または、起動テンプレートにない追加のパラメータを指定できます。

#### Note

起動時に起動テンプレートパラメータを削除することはできません（たとえば、パラメータに null 値を指定することはできません）。パラメータを削除するには、起動テンプレートの新しいバージョンをパラメータなしで作成し、そのバージョンを使用してインスタンスを起動します。

インスタンスを起動するには、IAM ユーザーは `ec2:RunInstances` アクションを使用するためのアクセス許可が必要です。また、インスタンスに作成または関連付けられたリソースを作成または使用するアクセス許可が必要です。`ec2:RunInstances` アクションのリソースレベルのアクセス権限を使用して、ユーザーが指定できる起動パラメータを管理できます。または、起動テンプレートを使用してインスタンスを起動するアクセス権限をユーザーに付与することもできます。これにより、IAM ポリシーではなくむしろ起動テンプレートで起動パラメータを管理できるようになります。インスタンス起動の権限を付与するための手段として起動テンプレートを使用できます。たとえば、ユーザーがインスタンスの起動に必ず起動テンプレートを使用し、また特定の起動テンプレートだけを使用するように指定できます。また、ユーザーが起動テンプレートで上書きする起動パラメータを制御することもできます。エンドポイントポリシーの例については、「[起動テンプレート \(p. 870\)](#)」を参照してください。

## 起動テンプレートの使用の管理

デフォルトでは、IAM ユーザーには起動テンプレートを使用するためのアクセス許可がありません。IAM ユーザーポリシーを作成して、起動テンプレートと起動テンプレートのバージョンの作成、変更、記述、削除を行うアクセス許可をユーザーに付与することができます。一部の起動テンプレートアクションにリソースレベルのアクセス許可を適用して、ユーザーがこれらのアクションに対する特定のリソースを使用する機能を制御することもできます。詳細については、「[例: 起動テンプレートの使用 \(p. 877\)](#)」のポリシー例を参照してください。

ユーザーに `ec2>CreateLaunchTemplate` および `ec2>CreateLaunchTemplateVersion` のアクションを使用するアクセス許可を付与するには注意が必要です。リソースレベルのアクセス許可を使用して、ユーザーが起動テンプレートで指定できるリソースを制御することはできません。インスタンス起動のために使用されるリソースを制限するには、起動テンプレートと起動テンプレートのバージョンを作成するアクセス許可を、適切な管理者のみに付与していることを確認してください。

## 起動テンプレートの作成

ユーザー定義のパラメータを使用して新しい起動テンプレートを作成するか、既存の起動テンプレートまたはインスタンスをベースにして新しい起動テンプレートを作成します。

定義されたパラメータを使用して新しい起動テンプレートを作成するには（コンソール）

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[起動テンプレート]、[起動テンプレートの作成] を選択します。
3. [起動テンプレート名] に、起動テンプレートのわかりやすい名前を入力します。作成時に起動テンプレートにタグを付けるには、[Show Tags]、[タグの追加] の順に選択し、タグキーと値のペアを入力します。
4. [テンプレートバージョンの説明] に、起動テンプレートバージョンの短い説明を入力します。
5. [起動テンプレートの内容] で、次の情報を指定します。
  - [AMI ID]: インスタンスを起動する AMI ID。利用可能な AMI すべてを検索するには、[Search for AMI (AMI を検索)] を選択します。一般的に使用される AMI を選択するには、[クイックスタート] を選択します。または、[AWS Marketplace] か [コミュニティ AMI] を選択します。所有している AMI を使用するか [適切な AMI を検索 \(p. 100\)](#) します。
  - [インスタンスタイプ]: インスタンスタイプが、指定した AMI と互換性があることを確認します。詳細については、「[インスタンスタイプ \(p. 183\)](#)」を参照してください。
  - [キーペア名]: インスタンスのキーペア。詳細については、「[Amazon EC2 のキーペア \(p. 899\)](#)」を参照してください。
  - [ネットワークタイプ]: 該当する場合は、インスタンスを VPC または EC2-Classic 内のどちらに起動するか。VPC を選択した場合は、[ネットワークインターフェイス] セクションでサブネットを指

定します。[Classic] を選択した場合、指定のインスタンスタイプが EC2-Classic でサポートされていることを確認し、インスタンスのアベイラビリティーゾーンを指定します。

- [セキュリティグループ]: インスタンスに関連付ける 1 つ以上のセキュリティグループ。詳細については、「[Linux インスタンスの Amazon EC2 セキュリティグループ \(p. 911\)](#)」を参照してください。
- 6. [ネットワークインターフェイス] で、インスタンスに対して最大 2 つのネットワークインターフェイス ([p. 713](#))を指定できます。
  - [デバイス]: ネットワークインターフェイスのデバイス番号。たとえば、プライマリネットワークインターフェイスなら eth0 です。フィールドに何も指定しない場合、AWS がプライマリネットワークインターフェイスを作成します。
  - [ネットワークインターフェイス]: ネットワークインターフェイス ID。または、空白のままにして AWS により新しいネットワークインターフェイスを作成します。
  - [説明]: (オプション) 新しいネットワークインターフェイスの説明。
  - [サブネット]: 新しいネットワークインターフェイスを作成するサブネット。プライマリネットワークインターフェイス (eth0) の場合、これはインスタンスが起動する先のサブネットです。eth0 に既存のネットワークインターフェイスを入力すると、インスタンスはネットワークインターフェイスが存在するサブネット内で起動します。
  - [自動割り当てパブリック IP]: eth0 のデバイスインデックスを持つネットワークインターフェイスに、自動的にパブリック IP アドレスを割り当てるかどうか。この設定は、1 つの新しいネットワークインターフェイスにのみ有効にすることができます。
  - [プライマリ IP]: サブネットの範囲からのプライマリプライベート IPv4 アドレス。AWS で自分用のプライベート IPv4 を選択するには、空白のままにします。
  - [セカンダリ IP]: サブネットの範囲からのセカンダリプライベート IPv4 アドレス。AWS で選択するには、空白のままにします。
  - [IPv6 のみ] [IPv6 IP]: サブネットの範囲の IPv6 アドレス。
  - [セキュリティグループ ID]: ネットワークインターフェイスに関連付ける VPC のセキュリティグループの ID。
  - [終了時に削除]: インスタンス終了時にネットワークインターフェイスを削除するかどうか。
  - Elastic Fabric Adapter: ネットワークインターフェイスが Elastic Fabric Adapter かどうかを示します。詳細については、「[Elastic Fabric Adapter](#)」を参照してください。
- 7. [ストレージ (ボリューム)] で、AMI によって指定されるボリューム以外に、インスタンスにアタッチするボリュームを指定します。
  - [ボリュームタイプ]: インスタンスと関連付けるインスタンスストアまたは Amazon EBS ボリューム。ボリュームの種類は、選択したインスタンスタイプに応じて異なります。詳細については、「[Amazon EC2 インスタンスストア \(p. 1076\)](#)」および「[Amazon EBS ボリューム \(p. 931\)](#)」を参照してください。
  - [デバイス名]: ボリュームのデバイス名。
  - [スナップショット]: ボリュームを作成するスナップショットの ID。
  - [サイズ]: Amazon EBS ボリュームの場合は、ストレージサイズ。
  - [ボリュームタイプ]: Amazon EBS ボリュームについては、ボリュームタイプを指定します。詳細については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。
  - [IOPS]: プロビジョンド IOPS SSD ボリュームタイプの場合、ボリュームがサポートできる 1 秒あたりの入力/出力オペレーションの数 (IOPS)。
  - [終了時に削除]: Amazon EBS ボリュームの場合、インスタンスの終了時にボリュームを削除するかどうか。詳細については、「[インスタンスの削除で Amazon EBS ボリュームを保持する \(p. 549\)](#)」を参照してください。
  - [Encrypted (暗号化)]: インスタンスタイプが EBS 暗号化をサポートしている場合、ボリュームの暗号化を有効にできます。このリージョンでデフォルトで暗号化を有効にした場合、暗号化は有効になります。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

- [Key (キー)]: EBS 暗号化に使用する CMK。AWS Key Management Service を使用して作成したカスタマーマスターキー (CMK) の ARN を指定できます。CMK を指定する場合は、[Encrypted (暗号化)] を使用して暗号化を有効にする必要があります。
8. [タグ] で、キーと値の組み合わせを [タグ \(p. 1120\)](#) として指定します。インスタンス、ボリューム、またはその両方にタグ付けできます。
9. [Advanced Details] で、セクションを開いてフィールドを表示し、インスタンスの追加パラメータを指定します。
- [購入のオプション]: 購入モデル。[スポットインスタンスのリクエスト] を選択して、オンデマンド価格を上限とするスポット料金でスポットインスタンスをリクエストし、[Customize Spot parameters (スポットパラメータのカスタマイズ)] を選択して、デフォルトのスポットインスタンス 設定を変更します。スポットインスタンスをリクエストしない場合、EC2 はデフォルトで オンデマンドインスタンスを起動します。詳細については、「[スポットインスタンス \(p. 320\)](#)」を参照してください。
  - [IAM インスタンスプロファイル]: インスタンスに関連付ける AWS Identity and Access Management (IAM) インスタンスプロファイル。詳細については、「[Amazon EC2 の IAM ロール \(p. 888\)](#)」を参照してください。
  - [Shutdown behavior]: シャットダウン時にインスタンスを停止するか終了するかどうか。詳細については、「[インスタンスによって起動されたシャットダウン動作の変更 \(p. 548\)](#)」を参照してください。
  - [Stop - Hibernate behavior (停止 - 休止動作)]: インスタンスに対して休止を有効にできたかどうか。このフィールドは、休止の前提条件を満たすインスタンスにのみ有効です。詳細については、「[Linux インスタンスの休止 \(p. 532\)](#)」を参照してください。
  - [終了保護]: 偶発的な終了を防ぐかどうか。詳細については、「[インスタンスの削除保護の有効化 \(p. 547\)](#)」を参照してください。
  - [モニタリング]: Amazon CloudWatch を使用したインスタンスの詳細モニタリングを有効にするかどうか。追加の変更が適用されます。詳細については、「[CloudWatch を使用したインスタンスのモニタリング \(p. 642\)](#)」を参照してください。
  - [T2/T3 Unlimited (T2/T3 無制限)]: アプリケーションがベースラインを越えて必要なだけバーストできるかどうか。このフィールドは、T2 および T3 インスタンスでのみ有効です。追加料金が適用される場合があります。詳細については、「[バースト可能パフォーマンスインスタンス \(p. 199\)](#)」を参照してください。
  - [プレイスメントグループ名]: インスタンスを起動する先のプレイスメントグループを指定します。すべてのインスタンスタイプがプレイスメントグループ内で起動できるわけではありません。詳細については、「[プレイスメントグループ \(p. 791\)](#)」を参照してください。
  - [EBS 最適化インスタンス]: Amazon EBS I/O 専用の追加容量を提供します。すべてのインスタンスタイプがこの機能をサポートしているわけではなく、追加料金が適用されます。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。
  - [テナンシー]: インスタンスを共有ハードウェア ([共有])、独立した専有ハードウェア ([専有])、あるいは Dedicated Host ([Dedicated host (専有ホスト)]) で実行するかを選択します。Dedicated Host でインスタンスを起動する場合は、インスタンスをホストリソースグループ内で起動するかどうかを指定できます。または、特定の Dedicated Host をターゲットとして設定できます。追加料金が適用される場合があります。詳細については、「[ハードウェア専有インスタンス \(p. 425\)](#)」および「[Dedicated Hosts \(p. 395\)](#)」を参照してください。
  - [RAM disk ID]: インスタンスの RAM ディスク。カーネルを指定した場合は、サポートするドライバーとともに特定の RAM ディスクを指定する必要がある可能性があります。準仮想化 (PV) AMI にのみ有効。
  - [ユーザーデータ]: 起動時にインスタンスを設定するユーザーデータ、または設定スクリプトを実行するユーザーデータを指定できます。詳細については、「[Linux インスタンスでの起動時のコマンドの実行 \(p. 588\)](#)」を参照してください。
10. [起動テンプレートの作成] を選択します。

### 既存の起動テンプレートから起動テンプレートを作成するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Launch Templates] を選択します。
3. [起動テンプレートの作成] を選択します。起動テンプレートの名前、説明、およびタグを指定します。
4. [ソーステンプレート] で、新しい起動テンプレートのベースとなる起動テンプレートを選択します。
5. [ソーステンプレートのバージョン] で、新しい起動テンプレートのベースとなる起動テンプレートのバージョンを選択します。
6. 必要に応じて起動パラメータを調整し、[起動テンプレートの作成] を選択します。

### インスタンスから起動テンプレートを作成するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選び、[アクション]、[Create Template From Instance (インスタンスからテンプレートを作成)] の順に選択します。
4. 名前、説明、およびタグを入力し、必要に応じて起動パラメータを調整します。

#### Note

インスタンスから起動テンプレートを作成するとき、そのインスタンスのネットワークインターフェイス ID と IP アドレスはテンプレートに含まれません。

5. [Create template from instance (インスタンスからテンプレートを作成)] を選択します。

### 起動テンプレートを作成するには (AWS CLI)

- `create-launch-template` (AWS CLI) コマンドを使用します。次の例では、以下を指定する起動テンプレートを作成します。
  - 起動テンプレートのタグ (`purpose=production`)
  - 起動するインスタンスタイプ (`r4.4xlarge`) と AMI (`ami-8c1be5f6`)
  - 合計 8 vCPU (4 コア × 2 スレッド) のコア数 (4) とコアごとのスレッド数 (2)
  - インスタンスを起動するサブネット (`subnet-7b16de0c`)

テンプレートは、パブリック IP アドレスと IPv6 アドレスをインスタンスに割り当て、インスタンスのタグ (`Name=webserver`) を作成します。

```
aws ec2 create-launch-template --launch-template-name TemplateForWebServer
--version-description WebVersion1 --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=purpose,Value=production}]' --launch-template-data file://template-
data.json
```

`template-data.json` ファイルの例を次に示します。

```
{
  "NetworkInterfaces": [
    {
      "AssociatePublicIpAddress": true,
      "DeviceIndex": 0,
      "Ipv6AddressCount": 1,
      "SubnetId": "subnet-7b16de0c"
    }
  ],
  "ImageId": "ami-8c1be5f6",
  "InstanceType": "r4.4xlarge",
```

```
"TagSpecifications": [{"ResourceType": "instance", "Tags": [{"Key": "Name", "Value": "webserver"}]}, {"CpuOptions": {"CoreCount": 4, "ThreadsPerCore": 2}}]
```

出力例を次に示します。

```
{ "LaunchTemplate": { "LatestVersionNumber": 1, "LaunchTemplateId": "lt-01238c059e3466abc", "LaunchTemplateName": "TemplateForWebServer", "DefaultVersionNumber": 1, "CreatedBy": "arn:aws:iam::123456789012:root", "CreateTime": "2017-11-27T09:13:24.000Z" } }
```

起動テンプレートのインスタンステータを取得するには (AWS CLI)

- [get-launch-template-data](#) (AWS CLI) コマンドを使用して、インスタンス ID を指定します。出力をベースとして使用して、新しい起動テンプレートや起動テンプレートのバージョンを作成できます。デフォルトでは、起動テンプレートデータで指定できない最上位レベルの LaunchTemplateData オブジェクトが output に含まれています。このオブジェクトを除外するには、--query オプションを使用します。

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query "LaunchTemplateData"
```

出力例を次に示します。

```
{ "Monitoring": {}, "ImageId": "ami-8c1be5f6", "BlockDeviceMappings": [ { "DeviceName": "/dev/xvda", "Ebs": { "DeleteOnTermination": true } } ], "EbsOptimized": false, "Placement": { "Tenancy": "default", "GroupName": "", "AvailabilityZone": "us-east-1a" }, "InstanceType": "t2.micro", "NetworkInterfaces": [ { "Description": "" } ] }
```

```
"NetworkInterfaceId": "eni-35306abc",
"PrivateIpAddresses": [
    {
        "Primary": true,
        "PrivateIpAddress": "10.0.0.72"
    }
],
"SubnetId": "subnet-7b16de0c",
"Groups": [
    "sg-7c227019"
],
"Ipv6Addresses": [
    {
        "Ipv6Address": "2001:db8:1234:1a00::123"
    }
],
"PrivateIpAddress": "10.0.0.72"
}
]
```

出力を次のようにファイルに直接書き込むことができます。

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
"LaunchTemplateData" >> instance-data.json
```

## 起動テンプレートのバージョンの管理

特定の起動テンプレートの起動テンプレートのバージョンを作成し、デフォルトバージョンを設定し、不要になったバージョンを削除することができます。

### タスク

- [起動テンプレートのバージョンの作成 \(p. 461\)](#)
- [デフォルトの起動テンプレートのバージョンの設定 \(p. 462\)](#)
- [起動テンプレートのバージョンの削除 \(p. 462\)](#)

### 起動テンプレートのバージョンの作成

起動テンプレートのバージョンを作成する際、新しいバージョンに新しい起動パラメータを指定するか、または既存のバージョンをベースとして使用できます。起動パラメータの詳細については、「[起動テンプレートの作成 \(p. 456\)](#)」を参照してください。

#### 起動テンプレートのバージョンを作成するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Launch Templates] を選択します。
3. [起動テンプレートの作成] を選択します。
4. [What would you like to do (該当するものを選択してください。)] で、[新規テンプレートバージョンの作成] を選択します。
5. [起動テンプレート名] で、リストから既存の起動テンプレートの名前を選択します。
6. [テンプレートバージョンの説明] に、起動テンプレートバージョンの説明を入力します。
7. (オプション) 新しい起動テンプレートバージョンのベースとして使用する起動テンプレートのバージョン、または別の起動テンプレートのバージョンを選択します。新しい起動テンプレートバージョンは、この起動テンプレートバージョンから起動パラメータを継承します。

- 必要に応じて起動パラメータを変更し、[起動テンプレートの作成] を選択します。

#### 起動テンプレートのバージョンを作成するには (AWS CLI)

- `create-launch-template-version` (AWS CLI) コマンドを使用します。新しいバージョンのベースとなるソースバージョンを指定できます。新しいバージョンはこのバージョンの起動パラメータを継承し、`--launch-template-data` を使用してパラメータを上書きできます。次の例では、起動テンプレートのバージョン 1 に基づいて新しいバージョンを作成し、異なる AMI ID を指定します。

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123 --version-description WebVersion2 --source-version 1 --launch-template-data "ImageId=ami-c998b6b2"
```

#### デフォルトの起動テンプレートのバージョンの設定

起動テンプレートにデフォルトバージョンを設定できます。起動テンプレートからインスタンスを起動し、バージョンを指定しない場合、インスタンスはデフォルトバージョンのパラメータを使用して起動されます。

#### デフォルトの起動テンプレートのバージョンを設定するには (コンソール)

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Launch Templates] を選択します。
- 起動テンプレートを選択し、[アクション]、[デフォルトバージョンの設定] を選択します。
- [デフォルトバージョン] でバージョン番号を選択し、[デフォルトバージョンとして設定] を選択します。

#### デフォルトの起動テンプレートのバージョンを設定するには (AWS CLI)

- `modify-launch-template` (AWS CLI) コマンドを使用して、デフォルトとして設定するバージョンを指定します。

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-version 2
```

#### 起動テンプレートのバージョンの削除

起動テンプレートのバージョンが不要になった場合には、それを削除することができます。削除後にバージョン番号を置き換えることはできません。起動テンプレートのデフォルトバージョンは削除できません。まずデフォルトとして別のバージョンを割り当てる必要があります。

#### 起動テンプレートのバージョンを削除するには (コンソール)

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Launch Templates] を選択します。
- 起動テンプレートを選択し、[アクション]、[テンプレートのバージョンの削除] を選択します。
- 削除するバージョンを選択し、[起動テンプレートのバージョンの削除] を選択します。

#### 起動テンプレートのバージョンを削除するには (AWS CLI)

- `delete-launch-template-versions` (AWS CLI) コマンドを使用して、削除するバージョン番号を指定します。

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --  
versions 1
```

## 起動テンプレートからのインスタンスの起動

起動テンプレートに含まれているパラメータを使用してインスタンスを起動できます。インスタンスを起動する前に、オプションで起動パラメータを上書きまたは追加できます。

起動テンプレートを使用して起動されたインスタンスには、aws:ec2:launchtemplate:id と aws:ec2:launchtemplate:version のキーを使用して自動的に 2 つのタグが割り当てられます。これらのタグを削除したり、編集することはできません。

起動テンプレートからインスタンスを起動するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Launch Templates] を選択します。
3. 起動テンプレートを選択し、[アクション]、[テンプレートからインスタンスを起動する] を選択します。
4. 使用する起動テンプレートのバージョンを選択します。
5. (オプション) [インスタンスの詳細] セクションでパラメータを変更または追加すると、起動テンプレートパラメータを上書きまたは追加することができます。
6. [テンプレートからインスタンスを起動する] を選択します。

起動テンプレートからインスタンスを起動するには (AWS CLI)

- `run-instances` AWS CLI コマンドを使用して `--launch-template` パラメータを指定します。オプションで、使用する起動テンプレートのバージョンを指定します。バージョンを指定しない場合は、デフォルトバージョンが使用されます。

```
aws ec2 run-instances --launch-template LaunchTemplateId=lt-0abcd290751193123,Version=1
```

- 起動テンプレートパラメータを上書きするには、`run-instances` コマンドでパラメータを指定します。次の例では、起動テンプレートで指定されたインスタンスタイプを上書きします (ある場合)。

```
aws ec2 run-instances --launch-template LaunchTemplateId=lt-0abcd290751193123 --instance-type t2.small
```

- 複雑な構造の一部である入れ子状のパラメータを指定した場合、インスタンスは、起動テンプレートで指定された複雑な構造および、指定した入れ子状の追加パラメータを使用して起動されます。

次の例で、インスタンスは、タグ `Owner=TeamA` および起動テンプレートで指定された他のタグを使用して起動されます。起動テンプレートに既存の `Owner` のキーのタグがある場合、値は `TeamA` に置き換えられます。

```
aws ec2 run-instances --launch-template LaunchTemplateId=lt-0abcd290751193123 --tag-specifications "ResourceType=instance,Tags=[{Key=Owner,Value=TeamA}]"
```

次の例で、インスタンスは、`/dev/xvdb` という名前のデバイス名を持つボリューム、および起動テンプレートで指定された他のブロックデバイスマッピングを使用して起動されます。起動テンプレートに `/dev/xvdb` 用に定義された既存のボリュームがある場合、値は指定された値で置き換えられます。

```
aws ec2 run-instances --launch-template LaunchTemplateId=lt-0abcd290751193123 --block-device-mappings "DeviceName=/dev/xvdb,Ebs={VolumeSize=20,VolumeType=gp2}"
```

インスタンスが起動しないか、状態が `terminated` ではなくすぐに `running` になる場合は、「[インスタンスの起動に関する問題のトラブルシューティング \(p. 1133\)](#)」を参照してください。

## Amazon EC2 Auto Scaling を使用して起動テンプレートを使用する

Auto Scaling グループを作成して、グループに使用する起動テンプレートを指定できます。Auto Scaling グループ内で Amazon EC2 Auto Scaling がインスタンスを起動する際、関連する起動テンプレートで定義された起動パラメータが使用されます。

詳細については、『Amazon EC2 Auto Scaling ユーザーガイド』の「[起動テンプレートを使用した Auto Scaling グループの作成](#)」を参照してください。

起動テンプレートを使用して Amazon EC2 Auto Scaling グループを作成または更新するには (AWS CLI)

- `create-auto-scaling-group` または `update-auto-scaling-group` AWS CLI コマンドを使用して `--launch-template` パラメータを指定します。

## EC2 フリート を使用して起動テンプレートを使用する

EC2 フリート リクエストを作成して、インスタンス設定で起動テンプレートを指定できます。Amazon EC2 は、EC2 フリート リクエストを満たす際、関連する起動テンプレートで定義された起動パラメータを使用します。起動テンプレートで指定されたパラメータの一部を上書きすることができます。

詳細については、「[EC2 フリートを作成する \(p. 488\)](#)」を参照してください。

起動テンプレートで EC2 フリートを作成するには (AWS CLI)

- `create-fleet` AWS CLI コマンドを使用します。`--launch-template-configs` パラメータを使用して、起動テンプレートと起動テンプレートの上書きを指定します。

## スポットフリート を使用して起動テンプレートを使用する

スポットフリート リクエストを作成して、インスタンス設定で起動テンプレートを指定できます。Amazon EC2 は、スポットフリート リクエストを満たす際、関連する起動テンプレートで定義された起動パラメータを使用します。起動テンプレートで指定されたパラメータの一部を上書きすることができます。

詳細については、「[スポットフリート リクエスト \(p. 345\)](#)」を参照してください。

起動テンプレートで スポットフリート リクエストを作成するには (AWS CLI)

- `request-spot-fleet` AWS CLI コマンドを使用します。`LaunchTemplateConfigs` パラメータを使用して、起動テンプレートと起動テンプレートの上書きを指定します。

## 起動テンプレートの削除

起動テンプレートが不要になった場合には、それを削除することができます。起動テンプレートを削除すると、すべてのバージョンが削除されます。

起動テンプレートを削除するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Launch Templates] を選択します。
3. 起動テンプレートを選択し、[アクション]、[テンプレートの削除] を選択します。

- [起動テンプレートの削除] を選択します。

起動テンプレートを削除するには (AWS CLI)

- `delete-launch-template` (AWS CLI) コマンドを使用して、起動テンプレートを指定します。

```
aws ec2 delete-launch-template --launch-template-id lt-01238c059e3466abc
```

## 既存のインスタンスのパラメータを使用してインスタンスを起動

Amazon EC2 コンソールには、現在のインスタンスを他のインスタンスを起動するためのベースとして使用可能にする、[Launch More Like This] ウィザードオプションが用意されています。このオプションでは、Amazon EC2 起動ウィザードで、選択されたインスタンスから自動的に特定の設定が入力されます。

### Note

[Launch More Like This] ウィザードオプションでは、選択されたインスタンスは複製されません。一部の設定が複製されるのみです。インスタンスのコピーを作成するには、最初にインスタンスから AMI を作成して、AMI からさらに多くのインスタンスを起動します。  
または、インスタンスの起動パラメータを格納するための [起動テンプレート \(p. 454\)](#) を作成します。

次の設定詳細は、選択されたインスタンスから起動ウィザードにコピーされます。

- AMI ID
- インスタンスタイプ
- アベイラビリティゾーン、または選択されたインスタンスがある VPC とサブネット
- パブリック IPv4 アドレス。選択されたインスタンスの IPv4 アドレスが現在パブリック IPv4 アドレスの場合、選択されたインスタンスのパブリック IPv4 アドレスのデフォルト設定に関係なく、新しいインスタンスはパブリック IPv4 アドレスを受け取ります。パブリック IPv4 アドレスの詳細については、「[パブリック IPv4 アドレスと外部 DNS ホスト名 \(p. 686\)](#)」を参照してください。
- プレイスメントグループ (該当する場合)
- 該当する場合は、インスタンスに関連付けられた IAM ロール
- シャットダウン動作の設定 ( 停止または終了 )
- 終了保護設定 ( true または false )
- CloudWatch モニタリング ( 有効または無効 )
- Amazon EBS 最適化設定 (true または false)
- VPC ( 共有または専用 ) に起動する場合は、テナント設定
- 該当する場合は、カーネル ID および RAM ディスク ID
- ユーザーデータ ( 指定された場合 )
- 該当する場合は、インスタンスに関連付けられたタグ
- インスタンスに関連付けられたセキュリティグループ

次の設定の詳細は選択されたインスタンスからコピーされず、代わりにウィザードがデフォルトの設定または動作を適用します。

- ネットワークインターフェイスの数: デフォルトでは、1つのネットワークインターフェイス、つまりブライマリネットワークインターフェイス (eth0) です。
- ストレージ: デフォルトのストレージ設定は AMI およびインスタンスタイプによって決まります。

現在のインスタンスをテンプレートとして使用するには

1. [インスタンス] ページで、使用するインスタンスを選択します。
  2. [Actions] を選択し、[Launch More Like This] を選択します。
  3. [Review Instance Launch] ページで起動ウィザードが開きます。インスタンスの詳細をチェックし、適切な [Edit] リンクをクリックして、必要な変更を行うことができます。
- 準備ができたら、[Launch] を選択してキーペアを選択し、インスタンスを起動します。
4. インスタンスが起動しないか、状態が `terminated` ではなくすぐに `running` になる場合は、「[インスタンスの起動に関する問題のトラブルシューティング \(p. 1133\)](#)」を参照してください。

## バックアップからの Linux インスタンスの起動

Amazon EBS-backed Linux インスタンスを使用すると、スナップショットを作成することで、インスタンスのルートデバイスピリュームをバックアップできます。インスタンスのルートデバイスピリュームのスナップショットがある場合、そのインスタンスを終了して、後でスナップショットから新しいインスタンスを起動できます。インスタンスの起動元のオリジナルの AMI がないけれども、同じイメージを使ってインスタンスを起動する必要がある場合に、これは便利です。

コンソールを使用してインスタンスのルートボリュームから AMI を作成するには、次の手順に従います。必要に応じて、[register-image](#) コマンド (AWS CLI) または Register-EC2Image コマンド (AWS Tools for Windows PowerShell) を代わりに使用することもできます。ブロックデバイスマッピングを使用してスナップショットを指定します。

コンソールを使用してルートボリュームから AMI を作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic Block Store]、[Snapshots] の順に選択します。
3. [スナップショットの作成] を選択します。
4. [Volumes] で、ルートボリュームの名前または ID の入力を開始し、オプションのリストから選択します。
5. 先ほど作成したスナップショットを選択し、[Actions]、[Create Image] の順に選択します。
6. [Create Image from EBS Snapshot] ダイアログボックスで、以下の情報を指定して [Create] を選択します。親インスタンスを再作成する場合は、親インスタンスと同じオプションを選択します。
  - Architecture: 32 ビットの場合は [i386] を、64 ビットの場合は [x86\_64] を選択します。
  - Root device name: ルートボリュームの適切な名前を入力します。詳細については、「[Linux インスタンスでのデバイスの名前付け \(p. 1098\)](#)」を参照してください。
  - Virtualization type: この AMI から起動されるインスタンスで準仮想化 (PV) またはハードウェア仮想マシン (HVM) のいずれの仮想化を使用するかを選択します。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。
  - (PV 仮想化タイプのみ) Kernel ID および RAM disk ID: リストから AKI と ARI を選択します。デフォルトの AKI を選択するか、AKI を選択しない場合、この AMI を使用してインスタンスを起動するたびに AKI を指定するように要求されます。また、デフォルトの AKI にインスタンスとの互換性がない場合、インスタンスのヘルスチェックが失敗する可能性があります。
  - (オプション) Block Device Mappings: ボリュームを追加するか、AMI のルートボリュームのデフォルト容量を増やします。ボリュームの容量を増やした場合のインスタンスのファイルシステムのサイズ変更の詳細については、「[ボリュームサイズ変更後の Linux ファイルシステムの拡張 \(p. 1011\)](#)」を参照してください。
7. ナビゲーションペインで [AMIs] を選択します。
8. 作成した AMI を選択し、[Launch] を選択します。ウィザードに従って、インスタンスを起動します。ウィザードの各ステップの設定方法については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。

## AWS Marketplace インスタンスの起動

AWS Marketplace 製品を受信登録し、Amazon EC2 起動ウィザードを使用して、この製品の AMI からインスタンスを起動できます。有料の AMI の詳細については、[有料 AMI \(p. 112\)](#) を参照してください。起動後に受信登録をキャンセルするには、初めに受信登録から、実行されているすべてのインスタンスを削除する必要があります。詳細については、「[AWS Marketplace サブスクリプションの管理 \(p. 115\)](#)」を参照してください。

起動ウィザードを使用して AWS Marketplace からインスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. Amazon EC2 ダッシュボードから、[Launch Instance] を選択します。
3. [Choose an Amazon Machine Image (AMI)] ページで、左の [AWS Marketplace] カテゴリを選択します。カテゴリを参照するか、検索機能を使用して適切な AMI を見つけます。[Select] を選択して製品を選択します。
4. ダイアログに、選択した製品の概要が表示されます。価格情報と、ベンダーが提供したその他の情報を表示できます。準備が完了したら、[Continue] を選択します。

### Note

AMI でインスタンスを起動するまで、製品の使用料は発生しません。ウィザードの次のページでは、インスタンスタイプの選択が求められるため、サポートされているインスタンスタイプの料金をメモしておいてください。追加の税金が製品に適用される場合があります。

5. [Choose an Instance Type] ページで、起動するインスタンスのハードウェア設定とサイズを選択します。終了したら、[Next: Configure Instance Details] を選択します。
6. ウィザードの次のページでは、インスタンスの設定、ストレージの追加、およびタグの追加を行うことができます。設定できるさまざまなオプションの詳細については、[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#) を参照してください。[Configure Security Group] ページが表示されるまで、[Next] を選択します。

製品に関するベンダーの仕様にしたがって、新しいセキュリティグループが作成されます。セキュリティグループには、Linux の SSH (ポート 22) または Windows の RDP (ポート 3389) すべての IPv4 アドレス (0.0.0.0/0) を許可するルールが含まれる場合があります。これらのルールを調整して、特定のアドレスまたはアドレスの範囲のみが、これらのポート経由でインスタンスにアクセスできるようにすることをお勧めします。

準備ができたら、[Review and Launch] を選択します。

7. [Review Instance Launch] ページで、インスタンスを起動しようとしている AMI の詳細と、ウィザードでセットアップするその他の設定の詳細をチェックします。準備ができたら、[Launch] を選択してキーペアを選択または作成し、インスタンスを起動します。
8. 受信登録した製品によっては、インスタンスの起動には数分以上かかります。インスタンスが起動する前に、まず製品に登録されます。クレジットカードの詳細に問題がある場合は、アカウントの詳細を更新するように求められます。起動確認のページが表示されたら、[View Instances] を選択して [Instances] ページに移動します。

### Note

インスタンスが実行されている限り、アイドル状態であっても、受信登録費用が発生します。インスタンスが停止している場合でも、ストレージに対して課金されることあります。

9. インスタンスの状態が [running] の場合、そのインスタンスに接続することができます。そのためには、一覧でインスタンスを選択し、[Connect] を選択します。ダイアログの指示にしたがいます。インスタンスへの接続の詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。

### Important

インスタンスにログインするには、特定のユーザー名を使用しなければならない場合があるため、ベンダーの使用手順を慎重に確認してください。受信登録の詳細へのアクセスについては、[AWS Marketplace サブスクリプションの管理 \(p. 115\)](#) を参照してください。

10. インスタンスが起動しないか、状態が `terminated` ではなくすぐに `running` になる場合は、「[インスタンスの起動に関する問題のトラブルシューティング \(p. 1133\)](#)」を参照してください。

## API と CLI を使用した AWS Marketplace AMI インスタンスの起動

API またはコマンドラインツールを使用して、AWS Marketplace 製品からインスタンスを起動するには、まず製品に登録していることを確認します。次の方法を使用して、製品の AMI ID でインスタンスを起動できます。

方法	ドキュメント
AWS CLI	<a href="#">run-instances コマンドを使用するか、詳細について「<b>インスタンスの起動</b>」を参照してください。</a>
AWS Tools for Windows PowerShell	<a href="#">New-EC2Instance コマンドを使用するか、詳細について<a href="#">Windows PowerShell を使用した Amazon EC2 インスタンスの起動</a>を参照してください。</a>
クエリ API	<a href="#">RunInstances リクエストを使用します。</a>

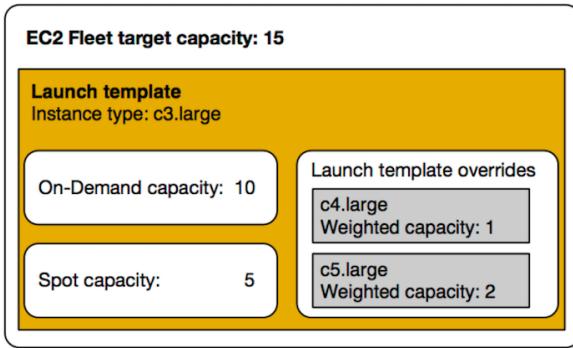
## EC2 フリートの起動

EC2 フリートには、インスタンスのフリート (つまり、グループ) を起動するための設定情報が含まれています。単一の API コールでは、フリートが オンデマンドインスタンス、リザーブドインスタンス、スポットインスタンス の購入オプションと一緒に使用して、複数のアベイラビリティーゾーンにまたがって複数のインスタンスタイプを起動できます。EC2 フリートを使用して、以下のことができます。

- オンデマンドとスポットのターゲット目標を別個に定義し、さらに 1 時間あたりの支払い上限料金を定義する
- アプリケーションに最適なインスタンスタイプを指定する
- 各購入オプション内でフリート容量を Amazon EC2 で分散する方法を指定する

フリートに対する 1 時間あたりの支払い上限容量を設定し、上限料金に達するまで EC2 フリートでインスタンスを起動することもできます。支払い上限料金に達すると、ターゲット容量に満たない場合でも、フリートはインスタンスの起動を停止します。

EC2 フリートは、リクエストで指定したターゲット容量を満たすために必要なインスタンス数の起動を試みます。1 時間あたりの上限の合計料金を指定すると、支払いの上限料金に達するまで、容量が満たされます。また、スポットインスタンス が中断した場合、フリートはスポットのターゲット容量を維持しようとします。詳細については、「[スポットインスタンスの仕組み \(p. 324\)](#)」を参照してください。



EC2 フリートごとにインスタンスタイプを無制限に指定できます。これらのインスタンスタイプは、オンデマンドおよびスポット購入オプションの両方を使用してプロビジョニングできます。複数のアベイラビリティーゾーンを指定し、インスタンスごとに異なる最大スポット料金を指定し、フリートごとに追加のスポットオプションを選択することもできます。Amazon EC2 は、インスタンスが起動したときに、指定したオプションを使用して容量をプロビジョニングします。

フリートの実行中に、価格の値上げまたはインスタンスの失敗のために Amazon EC2 がスポットインスタンスを再利用する場合、EC2 フリートは指定するインスタンスタイプのいずれかで、そのインスタンスを置き換えようとします。これにより、スポット料金の急激な増加中に容量を再取得することが容易になります。フリートごとに、柔軟で順応性に富むリソース戦略を作成できます。たとえば、特定のフリート内で、プライマリ容量に、利用できる場合はより安価なスポット容量をオンデマンドで補足することができます。

リザーブドインスタンスがあり、フリートで オンデマンドインスタンスを指定した場合、EC2 フリートは リザーブドインスタンスを使用します。たとえば、フリートが オンデマンドインスタンスを c4.large として指定し、c4.large 用の リザーブドインスタンスがある場合、リザーブドインスタンス料金表が送信されます。

EC2 フリートは追加料金なしで使用できます。フリートが起動した EC2 インスタンスに対してのみお支払いいただきます。

#### コンテンツ

- [EC2 フリート の制約事項 \(p. 469\)](#)
- [EC2 フリート の制限 \(p. 469\)](#)
- [EC2 フリート の設定戦略 \(p. 470\)](#)
- [EC2 フリート の管理 \(p. 480\)](#)

## EC2 フリート の制約事項

以下の制限が EC2 フリートに適用されます。

- EC2 フリートは、API または AWS CLI を通じてのみ使用できます。
- EC2 フリート リクエストは、複数の AWS リージョンにまたがることはできません。リージョンごとに別個の EC2 フリートを作成する必要があります。
- EC2 フリート リクエストは、同じアベイラビリティーゾーンから複数の異なるサブネットにまたがることはできません。

## EC2 フリート の制限

EC2 フリートによって起動されるインスタンスには、Amazon EC2 の通常の制限 (スポットリクエスト価格制限、インスタンス制限、容量制限など) が適用されます。また、以下の制限も適用されます。

- AWS リージョンあたりのアクティブな EC2 フリート の数: 1,000 \* †
- フリートあたりの起動仕様の数: 50 †
- 起動仕様内のユーザーデータのサイズ: 16 KB †
- EC2 フリート: あたりのターゲット容量: 10,000
- リージョン内のすべての EC2 フリート におけるターゲット容量: 100,000 \*

ターゲット容量のデフォルトの制限を超える容量が必要な場合は、AWS サポートセンターの[ケースの作成](#)フォームから制限の引き上げをリクエストできます。[制限のタイプ] で、[EC2 フリート] を選択して、リージョンを選択し、[フリートごとのターゲットフリート容量(単位)] か、[リージョンごとのターゲットフリート容量(単位)]、または両方を選択します。

\* これらの制限は、EC2 フリートとスポットフリートの両方に適用されます。

† これらはハード制限です。この制限の引き上げをリクエストすることはできません。

## T3スポットインスタンス

すぐに T3 スポットインスタンス を短期間使用する予定で、CPU クレジットを獲得するためのアイドル時間がない場合は、より高い費用を支払うことを避けるために、T3 スポットインスタンスを [standard \(p. 211\)](#) モードを起動することをお勧めします。

T3 スポットインスタンスを [unlimited \(p. 203\)](#) モードで起動し、すぐに CPU を破棄すると、余分なクレジットが破棄されます。インスタンスを短期間使用すると、インスタンスは CPU クレジットが発生して余剰クレジットを払う時間がなくなり、インスタンスを終了するときに余剰クレジットが課金されます。

T3 スポットインスタンスの Unlimited モードは、CPU クレジットをバーストするためにインスタンスが十分長く実行される場合にのみ適しています。それ以外の場合、余剰クレジットを支払うこと、T3 スポットインスタンスは M5 または C5 インスタンスよりも高価になります。詳細については、「[Unlimited モードと固定 CPU を使用する場合 \(p. 205\)](#)」を参照してください。

## T2スポットインスタンス

起動クレジットは、インスタンスを構成するために十分なコンピューティングリソースを提供し、T2 インスタンスの初期起動を効率的に実現することを意図しています。T2 インスタンスの起動を繰り返して新しい起動クレジットを利用することは許可されていません。CPU が持続的に必要な場合、(一定期間のアイドリングにより) クレジットを獲得して [T2 無制限 \(p. 203\)](#) を使用するか、専用 CPU (c4.largeなど) があるインスタンスタイプを使用します。

## EC2 フリート の設定戦略

EC2 フリート は、オンデマンドインスタンスとスポットインスタンスのグループです。

EC2 フリート は、フリートのリクエストで指定したターゲット容量を満たすために必要なインスタンス数の起動を試みます。フリートは、オンデマンドインスタンスのみ、またはスポットインスタンスのみで構成するか、オンデマンドインスタンスとスポットインスタンスを組み合わせて構成できます。スポットインスタンスへのリクエストは、利用可能な容量があり、リクエストで指定した 1 時間あたりの上限料金がスポット料金を超えている場合に達成されます。また、スポットインスタンスが中断した場合、フリートはターゲット容量を維持しようとします。

フリートに対する 1 時間あたりの支払い上限容量を設定し、上限料金に達するまで EC2 フリートでインスタンスを起動することもできます。支払い上限料金に達すると、ターゲット容量に満たない場合でも、フリートはインスタンスの起動を停止します。

スポットインスタンス プールは、同様のインスタンスタイプ、オペレーティングシステム、アベイラビリティーゾーン、ネットワークプラットフォームの一連の使われていない EC2 インスタンスです。EC2 フリートを作成する場合に複数の起動条件を含めることができ、これにはインスタンスタイプ、アベイラビ

リティゾーン、サブネット、上限価格があります。フリートは、リクエストとそのリクエストの設定を含む起動条件に基づいてリクエストを満たすために使用されるスポットインスタンスプールを選択します。スポットインスタンスは選択されたプールから取得されます。

EC2 フリートでは、コアまたはインスタンスの数やメモリの量に基づいてアプリケーションにとって意味がある大量の EC2 容量をプロビジョニングできます。たとえば、EC2 フリートが 200 インスタンス（そのうち 130 が オンデマンドインスタンスで、残りが スpotトインスタンス）のターゲット容量を起動するように指定できます。または、コアあたりの RAM が 2 GB 以上の 1,000 コアをリクエストできます。フリートにより、絶対的に低コストでその容量を起動できる Amazon EC2 オプションの組み合わせが決まります。

ニーズを満たす EC2 フリートを作成するのに適切な設定戦略を使用してください。

## コンテンツ

- [EC2 フリート の計画 \(p. 471\)](#)
- [EC2 フリート のリクエストタイプ \(p. 471\)](#)
- [スポットインスタンス の配分戦略 \(p. 472\)](#)
- [EC2 フリート でのオンデマンドバックアップの設定 \(p. 474\)](#)
- [上限価格の優先 \(p. 474\)](#)
- [使用量の管理 \(p. 475\)](#)
- [EC2 フリート インスタンスの分量指定 \(p. 475\)](#)
- [チュートリアル: EC2 フリート を使ってインスタンスの分量を指定する \(p. 477\)](#)
- [チュートリアル: プライマリ容量がオンデマンドの EC2 フリート を使用する \(p. 479\)](#)

## EC2 フリート の計画

EC2 フリートを計画するときは、次の操作を実行することをお勧めします。

- 目的のターゲット容量の同期または非同期のワンタイムリクエストを送信する EC2 フリートと、ターゲット容量の継続した維持を行うスポットフリートのどちらを作成するかを決定します。詳細については、「[EC2 フリート のリクエストタイプ \(p. 471\)](#)」を参照してください。
- アプリケーションの要件を満たすインスタンスタイプを決定します。
- EC2 フリートにスポットインスタンスを含める予定の場合、フリートを作成する前に「[Spot Best Practices](#)」を確認してください。フリートを計画するときにこれらのベストプラクティスを使用して、できるだけ低価格でインスタンスをプロビジョニングできるようにします。
- EC2 フリートのターゲット容量を決定します。インスタンスまたはカスタムユニットでターゲット容量を設定できます。詳細については、「[EC2 フリート インスタンスの分量指定 \(p. 475\)](#)」を参照してください。
- EC2 フリートのターゲットキャパシティーのどの部分がオンデマンド容量およびスポット容量となるかを決定します。オンデマンド容量とスポット容量のいずれか、または両方に対して 0 を指定できます。
- インスタンス分量指定を使用している場合は、ユニット当たりの料金を決定します。インスタンス時間当たりの料金の計算は、インスタンス時間当たりの料金をそのインスタンスが表すユニット数（または分量）で割って算出します。インスタンス分量指定を使用する場合、ユニット当たりのデフォルトの料金は 1 インスタンス時間当たりの料金となります。
- フリートに支払う 1 時間あたりの上限料金を設定します。詳細については、「[使用量の管理 \(p. 475\)](#)」を参照してください。
- EC2 フリートに対して可能なオプションを確認します。詳細については、「[EC2 フリート JSON 設定ファイルリファレンス \(p. 485\)](#)」を参照してください。EC2 フリートの設定の例については、「[EC2 フリート 設定例 \(p. 495\)](#)」を参照してください。

## EC2 フリート のリクエストタイプ

EC2 フリート リクエストには、次の 3 つの種類があります。

#### instant

リクエストタイプを `instant`、EC2 フリートとして設定した場合、希望する容量に同期ワンタイムリクエストを配置します。API レスポンスで、起動したインスタンスとともに起動できなかったインスタンスのエラーを返します。

#### request

リクエストタイプを `request`、EC2 フリートとして設定した場合、希望する容量に非同期ワンタイムリクエストを配置します。それ以降は、スポットの中断のために容量が減少した場合、フリートはスポットインスタンスを補充しようとせず、容量が利用できない場合に代替スポットインスタンスプールでリクエストを送信しません。

#### maintain

(デフォルト) リクエストタイプを `maintain` として設定した場合、EC2 フリートは希望する容量に非同期リクエストを配置し、中断されたスポットインスタンスを自動的に補充して、容量を維持します。

送信後に `instant` または `request` EC2 フリート リクエストのターゲット容量を変更することはできません。`instant` または `request` フリート リクエストのターゲット容量を変更するには、フリートを削除して新しいフリートを作成します。

3 つのタイプすべてのリクエストが、配分戦略の恩恵を受けます。詳細については、「[スポットインスタンスの配分戦略 \(p. 472\)](#)」を参照してください。

### スポットインスタンスの配分戦略

EC2 フリートの配分戦略は、起動条件によるスポットインスタンス プールからどのようにスポットインスタンスのリクエストを満たすかについて決定します。以下に、フリートで指定できる配分戦略を示します。

#### lowest-price

スポットインスタンスは、最低価格のプールから取得されます。これはデフォルトの戦略です。

#### diversified

スポットインスタンスはすべてのプールに分散されます。

#### capacity-optimized

スポットインスタンスは、起動するインスタンスの数に最適な容量のスポットインスタンス プールから取得されます。

#### InstancePoolsToUseCount

スポットインスタンスは、指定した数のスポットプールに分散されます。このパラメータは `lowest-price` と組み合わせて使用する場合にのみ有効です。

### ターゲット容量を維持する

スポット料金やスポットインスタンス プールの使用可能な容量の変動に伴ってスポットインスタンスが終了すると、`maintain` 型の EC2 フリートによって代替スポットインスタンスが起動されます。配分戦略が `lowest-price` である場合、スポット群は、スポット料金が現在最低値のプールに代替インスタンスを起動します。配分戦略が `lowest-price` と `InstancePoolsToUseCount` の組み合わせである場合、フリートは最低価格のスポットプールを選択し、指定した数のスポットプールでスポットインスタンスを起動します。配分戦略が `capacity-optimized` である場合、フリートは、利用可能なスポットインスタンス 容量が最大のプールで交換インスタンスを起動します。配分戦略が `diversified` である場合には、フリートは残りのプールに代替スポットインスタンスを分散します。

## コスト最適化のための EC2 フリート の設定

スポットインスタンス の使用コストを最適化するには、`lowest-price` 配分戦略を指定し、EC2 フリート が現在のスポット料金に基づいてインスタンスタイプおよびアベイラビリティーボーンの最も安価な組み合わせを自動的にデプロイするようにします。

オンデマンドインスタンス のターゲット容量では、EC2 フリート は スpotトインスタンス の配分戦略 (`lowest-price`、`capacity-optimized`、または `diversified`) を引き続き採用しながら、公開オンデマンド価格に基づいて最も安いインスタンスタイプを常に選択します。

## コスト最適化と分散のための EC2 フリート の設定

安価で同時に分散型の スpotトインスタンス のフリートを作成するには、`lowest-price` 配分戦略を `InstancePoolsToUseCount` と組み合わせて使用します。EC2 フリート は、現在のスポット料金に基づく最も安価なインスタンスタイプとアベイラビリティーボーンの組み合わせを、指定した数のスポットプールに自動的にデプロイします。この組み合わせを使用することで、最も高価な スpotトインスタンス を回避できます。

## 容量最適化のための EC2 フリート の設定

スspotトインスタンス では、価格は需要と供給の長期的な傾向に基づいて時間の経過とともに緩やかに変動しますが、容量はリアルタイムで変動します。`capacity-optimized` 戰略では、リアルタイムの容量データを調べ、可用性の最も高いプールを予測することで、そのプールから スpotトインスタンス を自動的に起動します。この戦略は、作業の再開とチェックポイントの設定に関連する中断のコストが高くなる可能性のあるワークロード (ビッグデータと分析、画像とメディアのレンダリング、機械学習、ハイパフォーマンスコンピューティングなど) に適しています。中断の可能性を低くすることにより、`capacity-optimized` 戰略ではワークロードの全体的なコストを削減できます。

## 適切な配分戦略の選択

ユースケースに基づいてフリートを最適化できます。

フリートが小さい場合、または短時間の実行である場合、すべてのインスタンスが単一の スpotトインスタンス プールにあるとしても、スspotトインスタンス が中断される可能性は低くなります。これにより、`lowest-price` 戰略は、低コストを提供している期間に条件に合いやすくなります。

フリートが大サイズ、または長期間実行される場合には、複数のプールに スpotトインスタンス を分散することでフリートの可用性を改善できます。たとえば、EC2 フリート の条件が 10 プールとして、ターゲット容量が 100 インスタンスとすると、フリートはプールごとに 10 個の スpotトインスタンス を起動します。1 つのプールのスポット料金がこのプールの上限料金を超える場合、フリートの 10% のみに影響がおよびます。この戦略を使用すると、いずれのプールにおいても経時的にフリートが受けるスポット料金の上昇の影響を減少させます。

`diversified` 戰略では、EC2 フリート は、[オンデマンド価格](#)以上のスポット料金のいずれのプールにもスspotトインスタンス を起動しません。

安価で分散型のフリートを作成するには、`lowest-price` 戰略を `InstancePoolsToUseCount` と組み合わせて使用します。スspotトインスタンス には少数または多数のスポットプールを選択して割り当てることができます。たとえば、バッチ処理を実行する場合は、少数のスポットプール (`InstancePoolsToUseCount=2` など) を指定することをお勧めします。これにより、キューのコンピューティング性能を常に確保しながら、削減額を最大化できます。ウェブサービスを実行する場合は、多数のスポットプール (`InstancePoolsToUseCount=10` など) を指定し、スspotトインスタンス が一時に使用不可になった場合の影響を最小限に抑えることをお勧めします。

作業の再開とチェックポイント設定に関連する中断に伴うコストが高くなる可能性があるワークロードをフリートで実行している場合は、`capacity-optimized` 戰略を使用します。この戦略では中断の可能性を低くすることにより、ワークロードの全体的なコストを削減できます。

## EC2 フリート でのオンデマンドバックアップの設定

重大なニュースイベントや試合の開始時にニュースウェブサイトをスケールする必要があるなど、予測できない緊急のスケーリングニーズが生じた場合、希望するオプションに十分な容量がないときは、オンデマンドインスタンスの代替インスタンスタイプを指定することをお勧めします。たとえば、c5.2xlarge オンデマンドインスタンスを希望するが使用可能な容量が十分でない場合、ピーク負荷時に c4.2xlarge インスタンスを使用できます。この場合、EC2 フリート は c5.2xlarge インスタンスを使用してすべてのターゲット容量を満たそうとしますが、容量が十分でない場合、c4.2xlarge インスタンスを自動的に起動してターゲット容量を満たします。

## オンデマンド容量に基づくインスタンスタイプの優先順位付け

EC2 フリート でオンデマンド容量を達成する場合、デフォルトで最低価格のインスタンスタイプが最初に起動されます。AllocationStrategy を prioritized に設定すると、EC2 フリート は優先度に従って、オンデマンド容量を達成するために最初に使用するインスタンスタイプを決定します。優先度は起動テンプレートの上書きに割り当てられ、最も高い優先度が最初に起動されます。

たとえば、3 つの起動テンプレートの上書きにそれぞれ異なるインスタンスタイプとして c3.large、c4.large、c5.large を設定したとします。c5.large のオンデマンド価格は、c4.large の価格より低くなります。c3.large が最低価格です。順番の決定に優先度を使用しない場合、フリートはオンデマンド容量を達成するために最初に c3.large を起動し、次に c5.large を起動します。c4.large のリザーブドインスタンスは未使用のことが多いため、起動テンプレートの上書きの優先度を設定し、c4.large、c3.large、c5.large の順にすることができます。

## オンデマンドインスタンス 用の キャパシティーの予約 の使用

容量予約の使用戦略を use-capacity-reservations-first に設定することで、オンデマンドインスタンス の起動時に最初に オンデマンドキャパシティー予約 を使用するようにフリートを設定できます。この設定は、オンデマンドインスタンス (lowest-price または prioritized) の配分戦略と組み合わせて使用できます。

未使用容量予約がオンデマンド容量を満たすために使用される場合:

- フリートは、未使用容量予約を使用して、目標オンデマンド容量までのオンデマンド容量を満たします。
- 複数のインスタンスプールに未使用容量予約がある場合、オンデマンド配分戦略 (lowest-price または prioritized) が適用されます。
- 未使用容量予約の数が目標オンデマンド容量より少ない場合、残りの目標オンデマンド容量は、オンデマンド配分戦略 (lowest-price または prioritized) に従って満たされます。

タイプ instant のフリートには未使用 オンデマンドキャパシティー予約 のみを使用できます。

オンデマンド容量を満たすために キャパシティーの予約 を使用するようにフリートを設定する方法の例については、「[EC2 フリート 設定例 \(p. 495\)](#)」を参照してください。詳細については、「[オンデマンドキャパシティー予約 \(p. 431\)](#)」および「[オンデマンド容量予約に関するよくある質問](#)」を参照してください。

## 上限価格の優先

各 EC2 フリート には、グローバルな上限料金を含めるか、デフォルト (オンデマンド価格) を使用できます。フリートは、これを起動条件のデフォルト上限料金として使用します。

任意で 1 つまたは複数の起動条件に上限料金を指定することができます。これは、起動条件に指定された料金です。起動条件に特定の料金が含まれる場合、EC2 フリート は起動条件の上限料金としてこの料金を使用し、全体の上限料金に優先することになります。特定の上限料金を含まないそのほかの起動条件は、全体の上限料金を引き続き使用することにご注意ください。

## 使用量の管理

EC2 フリート は、`TotalTargetCapacity` パラメータまたは `MaxTotalPrice` パラメータ (支払い上限料金) のいずれかに達すると、インスタンスの起動を停止します。フリートに支払う 1 時間あたりの料金を管理するには、`MaxTotalPrice` を指定できます。上限の合計料金に達すると、ターゲット容量に満たない場合でも、EC2 フリート はインスタンスの起動を停止します。

以下の例は、2 つの異なるシナリオを示しています。最初の例では、ターゲット容量に達すると、EC2 フリート はインスタンスの起動を停止します。2 番目の例では、支払い上限料金 (`MaxTotalPrice`) に達すると、EC2 フリート はインスタンスの起動を停止します。

例: ターゲット容量に達したときにインスタンスの起動を停止する

`m4.large` オンデマンドインスタンス に対するリクエストの内容が以下のとおりとします。

- オンデマンド料金: 1 時間あたり 0.10 USD
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: 1.50 USD

EC2 フリート は 10 オンデマンドインスタンス を起動します。合計料金 1.00 USD (10 インスタンス × 0.10 USD) は オンデマンドインスタンス の `MaxTotalPrice` (1.50 USD) を超えないためです。

例: 最大の合計料金に達したときにインスタンスの起動を停止する

`m4.large` オンデマンドインスタンス に対するリクエストの内容が以下のとおりとします。

- オンデマンド料金: 1 時間あたり 0.10 USD
- `OnDemandTargetCapacity`: 10
- `MaxTotalPrice`: 0.80 USD

EC2 フリート がオンデマンドターゲット容量 (10 オンデマンドインスタンス) を起動した場合、1 時間あたりの合計コストは 1.00 USD になります。これは オンデマンドインスタンス の `MaxTotalPrice` として指定した料金 (0.80 USD) を超えます。支払い可能な額を超えないように、EC2 フリート は 8 オンデマンドインスタンス (オンデマンドターゲット容量未満) を起動します。これを超えて起動すると、オンデマンドインスタンス の `MaxTotalPrice` を超えてしまいます。

## EC2 フリート インスタンスの分量指定

EC2 フリート を作成する場合、各インスタンスタイプがアプリケーションのパフォーマンスに寄与する容量単位を定義できます。次に、インスタンスの分量指定を使用して、起動仕様ごとの上限料金を調整できます。

デフォルトでは、指定する料金は 1 インスタンス時間あたりの料金となります。インスタンスの分量指定機能を使用すると、指定した料金は ユニット時間ごとの料金となります。ユニット時間あたりの入札価格は、インスタンスタイプの入札価格をそのユニットの数で割って計算できます。EC2 フリート では、ターゲット容量をインスタンス分量で割ることで起動するインスタンスの数を計算します。その結果が整数でなければ、フリートはその数を次の整数に切り上げ、これによりフリートのサイズがターゲット容量以上になります。起動されたインスタンスの容量がリクエストされたターゲット容量を超えた場合でも、フリートは起動仕様で指定したどのプールでも選択できます。

次の表には、10 のターゲット容量の EC2 フリート のユニット当たり入札価格を特定するために計算の例が含まれています。

インスタンスタイプ	インスタンスの分量	ターゲット容量	起動されたインスタンスの数	インスタンス時間あたりのスポット料金	ユニット時間あたりの価格
r3.xlarge	2	10	5 ( $10 \div 2$ )	0.05 USD	0.025 USD ( $.05 \div 2$ )
r3.8xlarge	8	10 ( $10 \div 8$ 、結果切り上げ)	2	0.10 USD	0.0125 USD ( $.10 \div 8$ )

次に示すように、EC2 フリートを使用して、受理時のユニットごとの最低価格のプールに指定するターゲット容量をプロビジョニングします。

1. EC2 フリート のターゲット容量を、インスタンス(デフォルト)あるいは仮想 CPU、メモリ、ストレージまたはスループットからご希望のユニットで設定します。
2. ユニットあたりの料金を設定します。
3. 各起動条件で、インスタンスタイプがターゲット容量に対して必要なユニット数である分量を指定します。

#### インスタンスの分量指定例

次の設定の EC2 フリート を検討します。

- ターゲット容量 24
- r3.2xlarge のインスタンスタイプの起動条件と分量 6
- c3.xlarge のインスタンスタイプの起動条件と分量 5

分量とは、インスタンスタイプがターゲット容量に対して必要なユニット数を表します。最初の起動条件がユニットあたりの料金を最低値で提供する場合(インスタンス時間あたりの r3.2xlarge の料金を 6 で割ったもの)、EC2 フリートはこれらのインスタンスから 4 つを起動します(24 を 6 で割ったもの)。

2 番目の起動条件がユニットあたりの料金を最低値で提供する場合(インスタンス時間あたりの c3.xlarge の料金を 5 で割ったもの)、EC2 フリートはこれらのインスタンスから 5 つを起動します(24 を 5 で割ったもの、結果が切り上げられる)。

#### インスタンスの分量指定と配分戦略

次の設定の EC2 フリート を検討します。

- ターゲット容量 30 スポットインスタンス
- c3.2xlarge のインスタンスタイプの起動条件と分量 8
- m3.xlarge のインスタンスタイプの起動条件と分量 8
- r3.xlarge のインスタンスタイプの起動条件と分量 8

EC2 フリート は、4 つのインスタンスを起動します(30 を 8 出割ったもの、結果を切り上げ)。lowest-price 戰略では、すべての 4 つのインスタンスはユニットあたりの最低価格を提供するプールから取得されます。diversified 戰略では、フリートは 3 プールごとに 1 つのインスタンスを起動し、そしてこの 3 つのプールのいずれかから取得された 4 つ目のインスタンスがユニットあたりの最低価格を提供することになります。

## チュートリアル: EC2 フリート を使ってインスタンスの分量を指定する

このチュートリアルでは、サンプル株式会社という名の架空会社で、インスタンス分量指定を使った EC2 フリート リクエストのプロセスを説明します。

### 目的

製薬会社であるサンプル株式会社は、癌と闘うために使用される可能性のある化合物を選別するために Amazon EC2 の計算処理能力を使用したいと考えています。

### 計画

サンプル株式会社はまず、「[Spot Best Practices](#)」を参照します。次に、サンプル株式会社は EC2 フリートに関する要件を確認します。

#### インスタンスタイプ

サンプル株式会社には、60 GB 以上のメモリと 8 つの仮想 CPU (vCPU) で最適に実行される、計算能力とメモリに負担がかかるアプリケーションがあります。同社は、できるだけ低価格でアプリケーション用のこれらのリソースを最大化したいと考えています。サンプル株式会社は、以下のいずれかの EC2 インスタンスタイプがそのニーズを満たすと判断します。

インスタンスタイプ	メモリ (GiB)	vCPU
r3.2xlarge	61	8
r3.4xlarge	122	16
r3.8xlarge	244	32

#### ユニット単位の目標容量

インスタンスの分量指定を使用すると、ターゲット容量はインスタンスの数 (デフォルト)、またはコア (vCPU)、メモリ (GiB) とストレージ (GB) との要素の組み合わせで表すことができます。アプリケーションのベース (60 GB の RAM と 8 個の vCPU) を 1 ユニットとして考えることで、サンプル株式会社はこの量の 20 倍で十分ニーズに合うと決定します。これにより、会社は EC2 フリート リクエストのターゲット容量を 20 に設定します。

#### インスタンスの分量

ターゲット容量の決定後、サンプル株式会社はインスタンスの分量を計算します。各インスタンスタイプのインスタンスの分量を計算することは、以下のように、ターゲット容量に達するために必要な各インスタンスタイプのユニットの数を決定することです。

- r3.2xlarge (61.0 GB、8 個の vCPU) = 1/20 ユニット
- r3.4xlarge (122.0 GB、16 個の vCPU) = 2/20 ユニット
- r3.8xlarge (244.0 GB、32 個の vCPU) = 4/20 ユニット

これよりサンプル株式会社は、1、2 と 4 のインスタンス分量を EC2 フリート リクエストのそれぞれの起動設定に割り当てます。

#### ユニット時間あたりの価格

サンプル株式会社は、料金の出発点としてインスタンス時間あたりの「[オンデマンド料金](#)」を使用します。最近のスポット料金または 2 つの組み合わせを使用することもできます。ユニット時間あたりの料金を計算するために、インスタンス時間あたりの出発点の料金を分量で割ります。以下に例を示します。

インスタンスタイプ	オンデマンド価格	インスタンスの分量	ユニット時間あたりの価格
r3.2xLarge	\$0.7	1	\$0.7
r3.4xLarge	\$1.4	2	\$0.7
r3.8xLarge	\$2.8	4	\$0.7

サンプル株式会社は、ユニット時間あたりのグローバルな料金として 0.7 USD を使用し、3 つのインスタンスタイプすべてで競争力を高めることもできます。また、`r3.8xlarge` の起動条件のなかで、1 ユニット時間あたりの全体料金を 0.7 USD、そして 1 ユニット時間あたりの指定入力料金を 0.9 USD とすることもできます。

### アクセス許可の確認

EC2 フリートを作成する前に、サンプル株式会社は必要なアクセス許可の IAM ロールがあることを確認します。詳細については、「[EC2 フリートの前提条件 \(p. 481\)](#)」を参照してください。

### EC2 フリートの作成

サンプル株式会社は、その EC2 フリートのために次の設定の config.json ファイルを作成します。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-07b3bc7625cdab851",
                "Version": "1"
            },
            "Overrides": [
                {
                    "InstanceType": "r3.2xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "r3.4xlarge",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 2
                },
                {
                    "InstanceType": "r3.8xlarge",
                    "MaxPrice": "0.90",
                    "SubnetId": "subnet-482e4972",
                    "WeightedCapacity": 4
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 20,
        "DefaultTargetCapacityType": "spot"
    }
}
```

サンプル株式会社は、次の `create-fleet` コマンドを使用して EC2 フリートを作成します。

```
aws ec2 create-fleet --cli-input-json file://config.json
```

詳細については、「[EC2 フリートを作成する \(p. 488\)](#)」を参照してください。

## 受理

配分戦略は、スポットインスタンスが取得されるスポットインスタンスプールを決定します。

`lowest-price` 戰略(デフォルトの戦略)では、受理時にユニットあたりの料金が最低値であるプールからスポットインスタンスが取得されます。20 ユニットの容量を提供するためには、20 の `r3.2xlarge` インスタンス ( $20 \div 1$ )、10 の `r3.4xlarge` インスタンス ( $20 \div 2$ )、あるいは 5 の `r3.8xlarge` インスタンス ( $20 \div 4$ ) が EC2 フリートから起動されることになります。

サンプル株式会社が `diversified` 戰略を採用する場合、スポットインスタンスは 3 つのすべてのプールから取得されます。EC2 フリートは、6 つの `r3.2xlarge` インスタンス(6 ユニットを提供)、3 つの `r3.4xlarge` インスタンス(6 ユニットを提供)、そして 2 つの `r3.8xlarge` インスタンス(8 ユニットを提供)の全部で 20 ユニットを起動します。

## チュートリアル: プライマリ容量がオンデマンドの EC2 フリートを使用する

このチュートリアルでは、ABC Online と呼ばれる架空の会社を使用して、プライマリ容量および使用可能な場合はスポット容量がオンデマンドの EC2 フリートをリクエストするプロセスを説明します。

### 目的

レストラン配達会社である ABC Online は、EC2 インスタンスタイプおよび購入オプション間で Amazon EC2 容量をプロビジョニングし、必要なスケール、パフォーマンス、コストを実現したいと思っています。

### 計画

ABC Online には、ピーク期間も機能する固定容量が必要ですが、低価格の容量増加の恩恵を受けたいと思っています。ABC Online は、EC2 フリートについて以下の要件を設定します。

- ・ オンデマンドインスタンス 容量 – ABC Online には、ピーク期間のトラフィックに対応できるように 15 オンデマンドインスタンスが必要です。
- ・ スpot インスタンス 容量 – ABC Online は、5 スpot インスタンスをプロビジョニングすることで、低価格でパフォーマンスを改善したいと思っています。

### アクセス許可の確認

EC2 フリートを作成する前に、ABC Online は必要なアクセス許可の IAM ロールがあることを確認します。詳細については、「[EC2 フリートの前提条件 \(p. 481\)](#)」を参照してください。

### EC2 フリートの作成

ABC Online は、その EC2 フリートのために次の設定の config.json ファイルを作成します。

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-07b3bc7625cdab851",  
                "Version": "2"  
            }  
        }  
    ],  
    "TargetCapacitySpecification": {  
        "TotalTargetCapacity": 20,  
        "OnDemandTargetCapacity": 15,  
        "DefaultTargetCapacityType": "spot"  
    }  
}
```

ABC Online は、次の `create-fleet` コマンドを使用して EC2 フリートを作成します。

```
aws ec2 create-fleet --cli-input-json file://config.json
```

詳細については、「[EC2 フリートを作成する \(p. 488\)](#)」を参照してください。

## 受理

配分戦略により、オンデマンド容量が常に満たされるが、ターゲット容量が容量と可用性がある場合にスポットで満たされることが決定されます。

## EC2 フリート の管理

EC2 フリート を使用するには、合計ターゲット容量、オンデマンド容量、スポット容量、インスタンスの 1 つ以上の起動仕様、希望上限価格などを指定したリクエストを作成します。フリートリクエストには、フリートがインスタンスの起動に必要とする情報 (AMI、インスタンスタイプ、サブネットまたはアベイラビリティーゾーン、そして 1 つ以上のセキュリティグループ) を定義する起動テンプレートを含める必要があります。お客様は、インスタンスタイプ、サブネット、アベイラビリティーゾーン、支払い上限価格の起動条件オーバーライドを指定でき、各起動条件オーバーライドに加重容量を割り当てることができます。

フリートに スポットインスタンス が含まれている場合、Amazon EC2 はスポット料金の変更に応じてフリートのターゲット容量を維持しようと試みることができます。

EC2 フリート リクエストは、期限切れになるかお客様によって削除されるまで、アクティブのままになります。フリートを削除するときは、削除によってそのフリートのインスタンスを終了するかどうかを指定できます。

## 目次

- [EC2 フリート リクエストの状態 \(p. 480\)](#)
- [EC2 フリート の前提条件 \(p. 481\)](#)
- [EC2 フリート ヘルスチェック \(p. 483\)](#)
- [EC2 フリート JSON 設定ファイルの生成 \(p. 484\)](#)
- [EC2 フリートを作成する \(p. 488\)](#)
- [EC2 フリート にタグを付ける \(p. 491\)](#)
- [EC2 フリート のモニタリング \(p. 480\)](#)
- [EC2 フリート の変更 \(p. 493\)](#)
- [EC2 フリート の削除 \(p. 494\)](#)
- [EC2 フリート 設定例 \(p. 495\)](#)

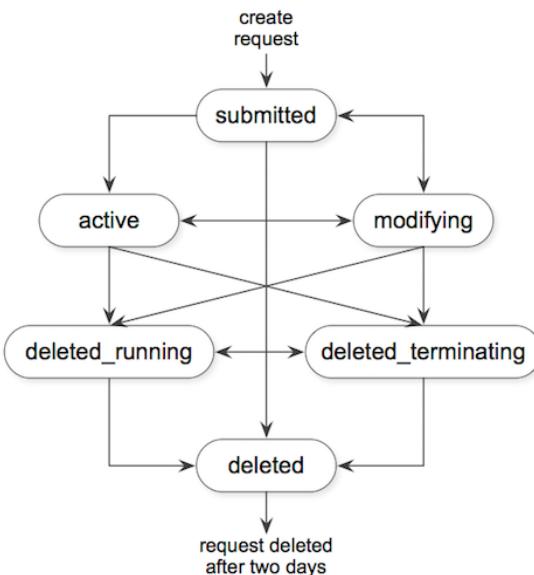
## EC2 フリート リクエストの状態

EC2 フリート リクエストは、次に示す状態のいずれかになります。

- `submitted` – EC2 フリート リクエストは評価中です。Amazon EC2 はターゲット数のインスタンスを起動する準備をしています。これには、オンデマンドインスタンス と スポットインスタンス のどちらか、または両方が含まれる場合があります。
- `active` – EC2 フリート リクエストは検証済みです。Amazon EC2 は実行中のインスタンスをターゲット数分、確保しようとしています。リクエストは、変更または削除されるまで、この状態のままになります。
- `modifying` – EC2 フリート リクエストは変更中です。リクエストは、変更が完全に処理されるか、リクエストが削除されるまで、この状態のままになります。`maintain` リクエストタイプのみ、変更できます。この状態はその他のリクエストタイプには適用されません。

- **deleted\_running** – EC2 フリート リクエストが削除され、追加のインスタンスは起動されません。その既存のインスタンスは、中断または終了されるまで実行され続けます。リクエストは、すべてのインスタンスが中断されるか終了されるまで、この状態のままになります。
- **deleted\_terminating** – EC2 フリート リクエストが削除され、そのインスタンスが終了します。リクエストは、すべてのインスタンスが終了されるまで、この状態のままになります。
- **deleted** – EC2 フリート が削除され、実行中のインスタンスがありません。リクエストは、そのインスタンスが終了されてから 2 日後に削除されます。

次の図は、EC2 フリート リクエストの状態の遷移を示しています。フリートの制限を超えた場合、リクエストはすぐに削除されます。



## EC2 フリート の前提条件

EC2 フリート を作成するには、以下の前提条件を設定する必要があります。

### 起動テンプレート

起動テンプレートには、インスタンスタイプ、アベイラビリティゾーン、支払い上限価格など、起動するインスタンスの情報が含まれています。詳細については、「[起動テンプレートからのインスタンスの起動 \(p. 454\)](#)」を参照してください。

### EC2 フリート 用のサービスにリンクされたロール

`AWSServiceRoleForEC2Fleet` ルールは、自分の代わりにインスタンスをリクエスト、起動、終了、タグ付けするアクセス許可を EC2 フリート に付与します。Amazon EC2 は、このサービスにリンクされたロールを使用して以下のアクションを実行します。

- `ec2:RequestSpotInstances` – スポットインスタンス をリクエストします。
- `ec2:TerminateInstances` – スポットインスタンス を終了します。
- `ec2:DescribeImages` – スポットインスタンス の Amazon Machine Image (AMI) を表示します。
- `ec2:DescribeInstanceStatus` – スポットインスタンス のステータスを表示します。
- `ec2:DescribeSubnets` – スポットインスタンス のサブネットを表示します。
- `ec2:CreateTags` – スポットインスタンス へのシステムタグの追加

AWS CLI または API を使って EC2 フリートを作成する前にこのロールが存在していることを確認します。ロールを作成するには、IAM コンソールを次のように使用します。

EC2 フリートの AWSServiceRoleForEC2Fleet ロールを作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [Roles] を選択し、続いて [Create role] を選択します。
3. [Select type of trusted entity (信頼されたエンティティのタイプの選択)] で、[AWS サービス] を選択します。
4. [このロールを使用するサービスを選択] で、[EC2 - Fleet] を選択後、[Next: Permissions]、[Next: Tags]、[Next: Review] の順に選択します。
5. [確認] ページで、[ロールの作成] を選択します。

EC2 フリートを使用する必要がなくなった場合は、[AWSServiceRoleForEC2Fleet] ロールを削除することをお勧めします。このロールがアカウントから削除された後で、別のフリートを作成した場合はロールを再度作成できます。

詳細については、IAM ユーザーガイドの「[サービスにリンクされたロールの使用](#)」を参照してください。

### 暗号化された AMI および EBS スナップショット用に CMK へのアクセスを付与する

暗号化された AMI (p. 151) または暗号化された Amazon EBS スナップショット (p. 1014) を EC2 フリートで指定し、カスタマーマネジメント CMK (カスタマーマスターキー) を暗号化に使用する場合は、CMK を使用するアクセス許可を AWSServiceRoleForEC2Fleet ロールに付与して Amazon EC2 がユーザーに代わってインスタンスを起動できるようにする必要があります。これを行うには、次の手順で示すように、CMK に付与を追加する必要があります。

アクセス権限を設定するときは、付与がキー・ポリシーの代わりになります。詳細については、「[許可の使用](#)」と「[AWS KMS でのキー・ポリシーの使用](#)」(AWS Key Management Service Developer Guide) を参照してください。

CMK を使用するアクセス許可を AWSServiceRoleForEC2Fleet ロールに付与するには

- `create-grant` コマンドを使用して CMK に付与を追加し、プリンシパル (AWSServiceRoleForEC2Fleet サービスにリンクされたロール) を指定します。このプリンシパルには、付与が許可するオペレーションを実行するためのアクセス許可が提供されます。CMK を指定するには、key-id パラメータと CMK の ARN を使用します。プリンシパルを指定するには、grantee-principal パラメータと AWSServiceRoleForEC2Fleet サービスにリンクされたロールの ARN を使用します。

次の例は、読みやすいようにフォーマットされています。

```
aws kms create-grant
--region us-east-1
--key-id arn:aws:kms:us-east-1:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab
--grantee-principal arn:aws:iam::111122223333:role/AWSServiceRoleForEC2Fleet
--operations "Decrypt" "Encrypt" "GenerateDataKey" "GenerateDataKeyWithoutPlaintext"
  "CreateGrant" "DescribeKey" "ReEncryptFrom" "ReEncryptTo"
```

### EC2 フリート ユーザーと IAM ユーザー

IAM ユーザーが EC2 フリートを作成または管理する場合、必ず次のようにして必要なアクセス許可を付与してください。

IAM ユーザーに EC2 フリートのアクセス許可を付与するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。

2. ナビゲーションペインで、[Policies (ポリシー)] を選択します。
3. [Create policy] を選択します。
4. [Create policy] ページで、[JSON] タブを選択し、テキストを以下に置き換えて [Review policy] を選択します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:*"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam>ListRoles",  
                "iam:PassRole",  
                "iam>ListInstanceProfiles"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

ec2:\* は、IAM ユーザーにすべての Amazon EC2 API アクションを呼び出すアクセス許可を付与します。特定の Amazon EC2 API アクションに制限するには、代わりにこれらのアクションを指定します。

IAM ユーザーは、既存の IAM ロールを列挙する iam>ListRoles アクション、EC2 フリート ロールを指定する iam:PassRole アクション、および既存のインスタンスプロファイルを列挙する iam>ListInstanceProfiles アクションを呼び出すには、アクセス許可が必要です。

(オプション) IAM ユーザーが IAM コンソールを使用してロールまたはインスタンスプロファイルを作成できるようにするには、次のアクションをポリシーに追加する必要があります。

- iam>AddRoleToInstanceProfile
  - iam:AttachRolePolicy
  - iam>CreateInstanceProfile
  - iam>CreateRole
  - iam:GetRole
  - iam>ListPolicies
5. [ポリシーの確認] ページでポリシーネ名と説明を入力し、[ポリシーの作成] を選択します。
  6. ナビゲーションペインで [Users] を選択し、ユーザーを選択します。
  7. [Permissions] タブで、[Add permissions] を選択します。
  8. [Attach existing policies directly] を選択します。以前に作成したポリシーを選択し、[Next: Review] を選択します。
  9. [Add permissions] を選択します。

## EC2 フリート ヘルスチェック

EC2 フリートは、2 分ごとにフリートのインスタンスのヘルステータスをチェックします。インスタンスのヘルステータスは healthy または unhealthy です。フリートは Amazon EC2 によって提供されるステータスチェックを使用して、インスタンスのヘルステータスを判断します。インスタンス

データスチェックとシステムヘルスチェックのいずれかのステータスが、連続した 3 回のヘルスチェックで `impaired` である場合、インスタンスのヘルスステータスは `unhealthy` になります。それ以外の場合、ヘルスステータスは `healthy` になります。詳細については、「[インスタンスのステータスチェック \(p. 628\)](#)」を参照してください。

異常なインスタンスは置き換えるよう EC2 フリートを設定できます。ヘルスチェックの置換を有効にすると、ヘルスステータスが `unhealthy` と報告された後でインスタンスが置き換えられます。異常なインスタンスを置き換えている間、数分間にわたりフリートがターゲット容量を下回る場合があります。

#### 要件

- ヘルスチェックの置換は、1 回限りのフリートではなく、ターゲット容量を維持する EC2 フリートでのみサポートされます。
- 作成時ののみ異常なインスタンスを置き換えるよう EC2 フリートを設定できます。
- IAM ユーザーは、`ec2:DescribeInstanceStatus` アクションを呼び出すアクセス許可を持っている場合のみ、ヘルスチェックの置換を使用できます。

#### EC2 フリート JSON 設定ファイルの生成

EC2 フリートを作成するには、起動テンプレート、合計ターゲット容量、デフォルトの購入オプションがオンデマンドとスポットのどちらであるかのみ指定する必要があります。パラメータを指定しない場合、フリートはデフォルト値を使用します。フリート設定パラメータの詳細なリストを見るには、JSON ファイルを次のように作成できます。

コマンドラインを使用して使用可能なすべての EC2 フリート パラメータを含む JSON ファイルを生成するには

- `create-fleet` (AWS CLI) コマンドと `--generate-cli-skeleton` パラメータを使用して、EC2 フリート JSON ファイルを生成します。

```
aws ec2 create-fleet --generate-cli-skeleton
```

以下の EC2 フリート パラメータが利用可能です。

```
{
    "DryRun": true,
    "ClientToken": "",
    "SpotOptions": {
        "AllocationStrategy": "lowest-price",
        "InstanceInterruptionBehavior": "hibernate",
        "InstancePoolsToUseCount": 0,
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MaxTotalPrice": 0,
        "MinTargetCapacity": 0
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized",
        "SingleInstanceType": true,
        "SingleAvailabilityZone": true,
        "MaxTotalPrice": 0,
        "MinTargetCapacity": 0
    },
    "ExcessCapacityTerminationPolicy": "termination",
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "",
                "LaunchTemplateName": ""
            }
        }
    ]
}
```

```
        "Version": "",  
    },  
    "Overrides": [  
        {  
            "InstanceType": "t2.micro",  
            "MaxPrice": "",  
            "SubnetId": "",  
            "AvailabilityZone": "",  
            "WeightedCapacity": null,  
            "Priority": null,  
            "Placement": {  
                "AvailabilityZone": "",  
                "Affinity": "",  
                "GroupName": "",  
                "PartitionNumber": 0,  
                "HostId": "",  
                "Tenancy": "dedicated",  
                "SpreadDomain": ""  
            }  
        }  
    ]  
},  
"TargetCapacitySpecification": {  
    "TotalTargetCapacity": 0,  
    "OnDemandTargetCapacity": 0,  
    "SpotTargetCapacity": 0,  
    "DefaultTargetCapacityType": "spot"  
},  
"TerminateInstancesWithExpiration": true,  
"Type": "maintain",  
"ValidFrom": "1970-01-01T00:00:00",  
"ValidUntil": "1970-01-01T00:00:00",  
"ReplaceUnhealthyInstances": true,  
"TagSpecifications": [  
    {  
        "ResourceType": "fleet",  
        "Tags": [  
            {  
                "Key": "",  
                "Value": ""  
            }  
        ]  
    }  
]
```

## EC2 フリート JSON 設定ファイルリファレンス

### Note

すべてのパラメータ値に小文字を使用してください。そうしないと、Amazon EC2 が JSON ファイルを使用して EC2 フリートを起動するときにエラーが表示されます。

#### AllocationStrategy (SpotOptions 用)

(オプション) EC2 フリートにより指定された スポットインスタンス プール間で スポットインスタンス ターゲット容量をどのように配分するかを指定します。有効な値は、lowest-price および diversified です。デフォルト: lowest-price。ニーズを満たす配分戦略を指定します。詳細については、「[スポットインスタンス の配分戦略 \(p. 472\)](#)」を参照してください。

#### InstanceInterruptionBehavior

(オプション) スポットインスタンスが中断された場合の動作。有効な値は、hibernate、stop、terminate です。デフォルトでは、スポットサービスは中断されるとス

ポットインスタンスを削除します。フリートタイプが `maintain` である場合、中断時にスポットインスタンスを休止または停止するように指定できます。

`InstancePoolsToUseCount`

ターゲットスポット容量を割り当てる先のスポットプールの数。スポットの `AllocationStrategy` が `lowest-price` に設定されている場合にのみ有効です。EC2 フリートは、最低価格のスポットプールを選択し、指定した数のスポットプールに均等にターゲットスポット容量を割り当てます。

`SingleInstanceType`

フリートが単一のインスタンスタイプを使用してフリートのすべてのスポットインスタンスを起動することを示します。

`SingleAvailabilityZone`

フリートがすべてのスポットインスタンスを単一のアベイラビリティゾーンで起動することを示します。

`MaxTotalPrice`

スポットインスタンスに支払う 1 時間あたりの上限料金。

`MinTargetCapacity`

フリートのスポットインスタンスの最小ターゲット容量。最小ターゲット容量に達しない場合、フリートはインスタンスを起動しません。

`AllocationStrategy` (OnDemandOptions 用)

オンデマンド容量を達成するために使用する起動テンプレートの上書きの順序。`lowest-price` を指定すると、EC2 フリートは料金に従って順序を決定し、最低価格を最初に起動します。優先度を指定すると、EC2 フリートは、各起動テンプレートの上書きに割り当てられた優先度に従って、最も高い優先度を最初に起動します。値を指定しないと、EC2 フリートはデフォルトで `lowest-price` に従います。

`SingleInstanceType`

フリートが単一のインスタンスタイプを使用してフリートのすべてのオンデマンドインスタンスを起動することを示します。

`SingleAvailabilityZone`

フリートがすべてのオンデマンドインスタンスを単一のアベイラビリティゾーンで起動することを示します。

`MaxTotalPrice`

オンデマンドインスタンスに支払う 1 時間あたりの上限料金。

`MinTargetCapacity`

フリートのオンデマンドインスタンスの最小ターゲット容量。最小ターゲット容量に達しない場合、フリートはインスタンスを起動しません。

`ExcessCapacityTerminationPolicy`

(オプション) EC2 フリートの合計ターゲット容量が EC2 フリートの現在のサイズより小さくなったり場合、実行中のインスタンスが終了されるかどうかを示します。有効な値は、`no-termination` および `termination` です。

`LaunchTemplateId`

使用する起動テンプレートの ID。起動テンプレート ID または起動テンプレート名を指定する必要があります。起動テンプレートでは、Amazon Machine Image (AMI) を指定する必要があります。起動

テンプレート作成の詳細については、「[起動テンプレートからのインスタンスの起動 \(p. 454\)](#)」を参照してください。

#### LaunchTemplateName

使用する起動テンプレートの名前。起動テンプレート ID または起動テンプレート名を指定する必要があります。起動テンプレートでは、Amazon Machine Image (AMI) を指定する必要があります。詳細については、「[起動テンプレートからのインスタンスの起動 \(p. 454\)](#)」を参照してください。

#### Version

起動テンプレートのバージョン番号。

#### InstanceType

(オプション) インスタンスタイプ。入力した場合、この値は起動テンプレートより優先されます。インスタンスタイプは、必要最小限のハードウェア仕様 (vCPU、メモリ、ストレージ) が必要です。

#### MaxPrice

(オプション) お客様が スポットインスタンス に対して支払ってもよいと考えるユーニット時間あたりの上限価格。入力した場合、この値は起動テンプレートより優先されます。デフォルトの上限料金 (オンデマンド価格) を使用するか、支払う予定の上限料金を指定することができます。上限価格が、指定したインスタンスタイプのスポット料金より低い場合、スポットインスタンス は起動されません。

#### SubnetId

(オプション) インスタンスを起動するサブネットの ID。入力した場合、この値は起動テンプレートより優先されます。

新しい VPC を作成するには、Amazon VPC コンソールにアクセスします。完了したら、JSON ファイルに戻って新しいサブネット ID を入力します。

#### AvailabilityZone

(オプション) インスタンスを起動するアベイラビリティーゾーン。デフォルトでは AWS によりスポットインスタンスのゾーンが選択されます。希望する場合には、特定のゾーンを設定できます。入力した場合、この値は起動テンプレートより優先されます。

1 つ以上のアベイラビリティーゾーンを指定します。ゾーンに複数のサブネットがある場合、適切なサブネットを指定します。サブネットを追加するには、Amazon VPC コンソールにアクセスします。完了したら、JSON ファイルに戻って新しいサブネット ID を入力します。

#### WeightedCapacity

(オプション) 単位数は、指定のインスタンスタイプによって提供されます。入力した場合、この値は起動テンプレートより優先されます。

#### 優先度

起動テンプレートの上書きの優先度。AllocationStrategy を prioritized に設定すると、EC2 フリー トは優先度に従って、オンデマンド容量を達成するために最初に使用する起動テンプレートの上書きを決定します。最も高い優先度が最初に起動されます。有効な値は 0 から始まる整数です。数値が小さいほど、優先度が高くなります。数値を設定しないと、上書きの優先度は最低になります。

#### TotalTargetCapacity

起動するインスタンスの数。アプリケーションのワークロードに重要なインスタンスまたはパフォーマンスのプロパティ (vCPU、メモリ、ストレージなど) を選択できます。リクエストタイプが [maintain] の場合、ターゲット容量を 0 に指定して後で容量を追加できます。

#### OnDemandTargetCapacity

(オプション) 起動する オンデマンドインスタンス の数。この数は TotalTargetCapacity 未満にする必要があります。

#### SpotTargetCapacity

(オプション) 起動する スポットインスタンス の数。この数は TotalTargetCapacity 未満にする必要があります。

#### DefaultTargetCapacityType

TotalTargetCapacity の値が OnDemandTargetCapacity と SpotTargetCapacity を組み合わせた値より大きい場合、その差が、ここで指定されたインスタンス購入オプションとして起動されます。有効な値は on-demand または spot です。

#### TerminateInstancesWithExpiration

(オプション) デフォルトでは、EC2 フリート リクエストの有効期限が切れると Amazon EC2 ゲインスタンスを終了します。デフォルト値は true です。リクエストの期限が切れた後に実行を継続するには、このパラメータの値を入力しないでください。

#### タイプ

(オプション) EC2 フリート が、希望する容量の同期ワンタイムリクエストを送信 (instant) するか、希望する容量に非同期ワンタイムリクエスト (希望する容量を維持しようとしない) を送信するか、容量が利用できない場合に代替容量プールでリクエストを送信 (request) するか、非同期のワンタイムリクエストを送信し、中断された スポットインスタンス を自動的に補充して、容量の維持を継続 (maintain) するかを示します。有効な値は、instant、request、maintain です。デフォルト値は maintain です。詳細については、「[EC2 フリート のリクエストタイプ \(p. 471\)](#)」を参照してください。

#### ValidFrom

(オプション) 特定の期間中だけ有効なリクエストを作成するには、開始日に入力します。

#### ValidUntil

(オプション) 特定の期間中だけ有効なリクエストを作成するには、終了日に入力します。

#### ReplaceUnhealthyInstances

(オプション) フリートの maintain が設定された EC2 フリート で正常でないインスタンスを置き換えるには、true と入力します。それ以外の場合、このパラメータは空にしてください。

#### TagSpecifications

(オプション) 作成時に EC2 フリート リクエストをタグ付けするためのキーと値のペアです。ResourceType の値を fleet にしないと、フリートリクエストが失敗します。起動時にインスタンスをタグ付けする場合は、[起動テンプレート \(p. 456\)](#)でタグを指定します。起動後のタグ付けについては「[リソースにタグを付ける \(p. 1121\)](#)」をご覧ください。

### EC2 フリートを作成する

EC2 フリート を作成するとき、インスタンスタイプ、アベイラビリティーゾーン、支払い上限価格など、起動するインスタンスの情報を含む起動テンプレートを指定する必要があります。

起動テンプレートをオーバーライドする複数の起動条件を含む EC2 フリート を作成できます。起動条件は、インスタンスタイプ、アベイラビリティーゾーン、サブネット、上限価格によって異なり、異なる加重容量が含まれていることがあります。

EC2 フリート を作成するときは、JSON ファイルを使用して起動するインスタンスについての情報を指定します。詳細については、「[EC2 フリート JSON 設定ファイルリファレンス \(p. 485\)](#)」を参照してください。

EC2 フリート の作成に使用できるのは、AWS CLI のみです。

## EC2 フリート (AWS CLI) を作成するには

- 次の [create-fleet](#) (AWS CLI) コマンドを使用して EC2 フリートを作成します。

```
aws ec2 create-fleet --cli-input-json file://file_name.json
```

設定ファイルの例については、「[EC2 フリート 設定例 \(p. 495\)](#)」を参照してください。

タイプ `request` またはタイプ `maintain` のフリートの出力例を次に示します。

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"  
}
```

ターゲット容量を起動したタイプ `instant` のフリートの出力例を次に示します。

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",  
    "Errors": [],  
    "Instances": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c5.large",  
                    "AvailabilityZone": "us-east-1a"  
                }  
            },  
            "Lifecycle": "on-demand",  
            "InstanceIds": [  
                "i-1234567890abcdef0",  
                "i-9876543210abcdef9"  
            ],  
            "InstanceType": "c5.large",  
            "Platform": null  
        },  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c4.large",  
                    "AvailabilityZone": "us-east-1a"  
                }  
            },  
            "Lifecycle": "on-demand",  
            "InstanceIds": [  
                "i-5678901234abcdef0",  
                "i-5432109876abcdef9"  
            ],  
            "InstanceType": "c4.large",  
            "Platform": null  
        },  
    ]  
}
```

ターゲット容量の一部を起動し、起動されなかったインスタンスをエラーとするタイプ instant のフリートの出力例を次に示します。

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",  
    "Errors": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c4.xlarge",  
                    "AvailabilityZone": "us-east-1a",  
                }  
            },  
            "Lifecycle": "on-demand",  
            "ErrorCode": "InsufficientInstanceCapacity",  
            "ErrorMessage": "",  
            "InstanceType": "c4.xlarge",  
            "Platform": null  
        },  
    ],  
    "Instances": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c5.large",  
                    "AvailabilityZone": "us-east-1a"  
                }  
            },  
            "Lifecycle": "on-demand",  
            "InstanceIds": [  
                "i-1234567890abcdef0",  
                "i-9876543210abcdef9"  
            ],  
            "InstanceType": "c5.large",  
            "Platform": null  
        },  
    ]  
}
```

インスタンスを起動しなかったタイプ instant のフリートの出力例を次に示します。

```
{  
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE",  
    "Errors": [  
        {  
            "LaunchTemplateAndOverrides": {  
                "LaunchTemplateSpecification": {  
                    "LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",  
                    "Version": "1"  
                },  
                "Overrides": {  
                    "InstanceType": "c4.xlarge",  
                    "AvailabilityZone": "us-east-1a",  
                }  
            },  
            "Lifecycle": "on-demand",  
        }  
    ]  
}
```

```
"ErrorCode": "InsufficientCapacity",
"ErrorMessage": "",
"InstanceType": "c4.xlarge",
"Platform": null
},
{
"LaunchTemplateAndOverrides": {
"LaunchTemplateSpecification": {
"LaunchTemplateId": "lt-01234a567b8910abcEXAMPLE",
"Version": "1"
},
"Overrides": {
"InstanceType": "c5.large",
"AvailabilityZone": "us-east-1a",
}
},
"Lifecycle": "on-demand",
"ErrorCode": "InsufficientCapacity",
"ErrorMessage": "",
"InstanceType": "c5.large",
"Platform": null
},
],
"Instances": []
}
```

## EC2 フリートにタグを付ける

EC2 フリート リクエストを分類および管理しやすくするために、カスタムメタデータでタグ付けすることができます。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。

EC2 フリート タグは、作成時または作成後にリクエストに割り当てることができます。フリートリクエストに指定されているタグは、フリートが起動するインスタンスには割り当てられません。

### 新しい EC2 フリートリクエストにタグを付ける

作成時に EC2 フリート リクエストをタグ付けするには、フリートを作成するために使用した [JSON ファイル \(p. 484\)](#)でキーと値のペアを指定します。 ResourceType の値は fleet にする必要があります。別の値で指定すると、フリートリクエストは失敗します。

### EC2 フリートが起動したインスタンスにタグを付ける

フリートが起動したインスタンスをタグ付けるには、EC2 フリート リクエストで参照される [起動テンプレート \(p. 456\)](#)でタグを指定します。

既存の EC2 フリート リクエストとインスタンスにタグを付けるには (AWS CLI)

既存のリソースにタグを追加するには、次の `create-tags` コマンドを使用します。

```
aws ec2 create-tags --resources fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE i-1234567890abcdef0 --tags Key=purpose,Value=test
```

## EC2 フリートのモニタリング

EC2 フリートは、使用可能な容量があるときは オンデマンドインスタンス を起動し、上限価格がスポット料金を超えていて容量が利用可能なときは スポットインスタンス を起動します。オンデマンドインスタンスは、終了されるまで実行され、スポットインスタンスは中断されるか終了されるまで実行されます。

実行中のインスタンスの返されるリストは定期的に更新されますが、古い可能性もあります。

EC2 フリートを監視するには (AWS CLI)

EC2 フリート の詳細を表示するには、次の [describe-fleets](#) コマンドを使用します。

```
aws ec2 describe-fleets
```

出力例を次に示します。

```
{  
    "Fleets": [  
        {  
            "Type": "maintain",  
            "FulfilledCapacity": 2.0,  
            "LaunchTemplateConfigs": [  
                {  
                    "LaunchTemplateSpecification": {  
                        "Version": "2",  
                        "LaunchTemplateId": "lt-07b3bc7625cdab851"  
                    }  
                }  
            ],  
            "TerminateInstancesWithExpiration": false,  
            "TargetCapacitySpecification": {  
                "OnDemandTargetCapacity": 0,  
                "SpotTargetCapacity": 2,  
                "TotalTargetCapacity": 2,  
                "DefaultTargetCapacityType": "spot"  
            },  
            "FulfilledOnDemandCapacity": 0.0,  
            "ActivityStatus": "fulfilled",  
            "FleetId": "fleet-76e13e99-01ef-4bd6-ba9b-9208de883e7f",  
            "ReplaceUnhealthyInstances": false,  
            "SpotOptions": {  
                "InstanceInterruptionBehavior": "terminate",  
                "InstancePoolsToUseCount": 1,  
                "AllocationStrategy": "lowest-price"  
            },  
            "FleetState": "active",  
            "ExcessCapacityTerminationPolicy": "termination",  
            "CreateTime": "2018-04-10T16:46:03.000Z"  
        }  
    ]  
}
```

指定した EC2 フリート のインスタンスの詳細を表示するには、次の [describe-fleet-instances](#) コマンドを使用します。

```
aws ec2 describe-fleet-instances --fleet-id fleet-73fb2ce-aa30-494c-8788-1cee4EXAMPLE
```

```
{  
    "ActiveInstances": [  
        {  
            "InstanceId": "i-09cd595998cb3765e",  
            "InstanceHealth": "healthy",  
            "InstanceType": "m4.large",  
            "SpotInstanceRequestId": "sir-86k84j6p"  
        },  
        {  
            "InstanceId": "i-09cf95167ca219f17",  
            "InstanceHealth": "healthy",  
            "InstanceType": "m4.large",  
            "SpotInstanceRequestId": "sir-dvxi7fsm"  
        }  
    ]  
}
```

```
    ],
    "FleetId": "fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

指定した EC2 フリート の指定期間の履歴を表示するには、次の [describe-fleet-history](#) コマンドを使用します。

```
aws ec2 describe-fleet-history --fleet-request-id fleet-73fbcd2ce-
aa30-494c-8788-1cee4EXAMPLE --start-time 2018-04-10T00:00:00Z
```

```
{
    "HistoryRecords": [],
    "FleetId": "fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE",
    "LastEvaluatedTime": "1970-01-01T00:00:00.000Z",
    "StartTime": "2018-04-09T23:53:20.000Z"
}
```

## EC2 フリート の変更

状態が `submitted` または `active` の EC2 フリート を変更することができます。フリートを変更すると、そのフリートは `modifying` 状態に移行します。

EC2 フリート の以下のパラメータを変更できます。

- `target-capacity-specification` – `TotalTargetCapacity`、`OnDemandTargetCapacity`、および `SpotTargetCapacity` のターゲット容量を増やすか減らします。
- `excess-capacity-termination-policy` – EC2 フリート の合計ターゲット容量がフリートの現在のサイズより小さくなつた場合、実行中のインスタンスが終了されるかどうか。有効な値は、`no-termination` および `termination` です。

### Note

`Type=maintain` の EC2 フリート のみ変更できます。

ターゲット容量を増やすと、EC2 フリート は `DefaultTargetCapacityType` で指定されたインスタンス購入オプション (オンデマンドインスタンス または スポットインスタンス) に従つて追加のインスタンスを起動します。

`DefaultTargetCapacityType` が `spot` の場合、EC2 フリート はその配分戦略に従つて追加のスポットインスタンス を起動します。配分戦略が `lowest-price` の場合、フリートは、リクエストの最低価格のスポットインスタンス プールからインスタンスを起動します。配分戦略が `diversified` の場合、フリートは、リクエストのプールにインスタンスを分散します。

ターゲット容量を減らす場合、EC2 フリート は新しいターゲット容量を超えるすべてのオーブンリクエストをキャンセルします。フリートのサイズが新しいターゲット容量に達するとフリートのスポットインスタンスが終了されるようにリクエストできます。配分戦略が `lowest-price` である場合は、フリートの最低単価のインスタンスが終了されます。配分戦略が `diversified` である場合は、フリートのプール全体でインスタンスが終了されます。あるいは、EC2 フリート の現在のサイズを保持するようにリクエストすることができますが、中断された スpot インスタンス や手動終了されたインスタンスへの置き換えはできません。

ターゲット容量が減つたために EC2 フリート によって スpot インスタンス が終了される場合、インスタンスは スpot インスタンス の中断通知を受け取ります。

### EC2 フリート を変更するには (AWS CLI)

次の [modify-fleet](#) コマンドを使用して、指定された EC2 フリート のターゲット容量を更新します。

```
aws ec2 modify-fleet --fleet-id fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity-specification TotalTargetCapacity=20
```

ターゲット容量を小さくしてもフリートの現在のサイズを保持する場合は、前のコマンドを以下のように変更できます。

```
aws ec2 modify-fleet --fleet-id fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE --target-capacity-specification TotalTargetCapacity=10 --excess-capacity-termination-policy no-termination
```

## EC2 フリート の削除

EC2 フリート が不要になった場合には、それを削除することができます。フリートを削除すると、新しいインスタンスは起動されません。

EC2 フリート のインスタンスを終了するかどうか指定する必要があります。フリートを削除するときにインスタンスを終了する必要があることを指定した場合、`deleted_terminating` 状態へ移行します。それ以外の場合は `deleted_running` 状態になり、インスタンスは中断または手動終了されるまで、引き続き実行されます。

EC2 フリート を削除するには (AWS CLI)

`delete-fleets` コマンドと `--terminate-instances` パラメータを使用し、指定された EC2 フリート を削除してインスタンスを終了します。

```
aws ec2 delete-fleets --fleet-ids fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE --terminate-instances
```

出力例を次に示します。

```
{  
    "UnsuccessfulFleetDeletions": [],  
    "SuccessfulFleetDeletions": [  
        {  
            "CurrentFleetState": "deleted_terminating",  
            "PreviousFleetState": "active",  
            "FleetId": "fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE"  
        }  
    ]  
}
```

`--no-terminate-instances` パラメータを使用して前のコマンドを変更することで、インスタンスを終了せずに、指定された EC2 フリート を削除できます。

```
aws ec2 delete-fleets --fleet-ids fleet-73fbcd2ce-aa30-494c-8788-1cee4EXAMPLE --no-terminate-instances
```

出力例を次に示します。

```
{  
    "UnsuccessfulFleetDeletions": [],  
    "SuccessfulFleetDeletions": [  
        {  
            "CurrentFleetState": "deleted_running",  
            "PreviousFleetState": "active",  
            "FleetId": "fleet-4b8aaae8-dfb5-436d-a4c6-3dafa4c6b7dcEXAMPLE"  
        }  
    ]  
}
```

```
    ]  
}
```

## EC2 フリート 設定例

以下の例で示しているのは、EC2 フリート を作成するための `create-fleet` コマンドで使用できる起動設定です。[create-fleet パラメータの詳細については、「EC2 フリート JSON 設定ファイルリファレンス \(p. 485\)」を参照してください。](#)

### 例

- [例 1: スポットインスタンス をデフォルト購入オプションとして起動する \(p. 495\)](#)
- [例 2: オンデマンドインスタンス をデフォルト購入オプションとして起動する \(p. 495\)](#)
- [例 3: オンデマンドインスタンス をプライマリ容量として起動する \(p. 496\)](#)
- [例 4: 最低価格の配分戦略を使用して スポットインスタンス を起動する \(p. 496\)](#)
- [例 5: 容量予約と優先配分戦略を使用して オンデマンドインスタンス を起動する \(p. 497\)](#)
- [例 6: 合計ターゲット容量が未使用 キャパシティーの予約 の数を超えたときに キャパシティーの予約 と優先配分戦略を使用して オンデマンドインスタンス を起動する \(p. 499\)](#)
- [例 7: 容量予約と最低料金配分戦略を使用して オンデマンドインスタンス を起動する \(p. 501\)](#)
- [例 8: 合計ターゲット容量が未使用 キャパシティーの予約 の数を超えたときに容量予約と最低料金配分戦略を使用して オンデマンドインスタンス を起動する \(p. 503\)](#)

### 例 1: スポットインスタンス をデフォルト購入オプションとして起動する

次の例では、EC2 フリート で必要な最小限のパラメータ (起動テンプレート、ターゲット容量、デフォルト購入オプション) を指定します。起動テンプレートは、起動テンプレート ID とバージョン番号によって識別されます。フリートのターゲット容量は 2 インスタンスであり、デフォルト購入オプションは `spot` です。この結果、フリートは 2 スポットインスタンス を起動します。

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-0e8c754449b27161c",  
                "Version": "1"  
            }  
        },  
        ],  
        "TargetCapacitySpecification": {  
            "TotalTargetCapacity": 2,  
            "DefaultTargetCapacityType": "spot"  
        }  
    }  
}
```

### 例 2: オンデマンドインスタンス をデフォルト購入オプションとして起動する

次の例では、EC2 フリート で必要な最小限のパラメータ (起動テンプレート、ターゲット容量、デフォルト購入オプション) を指定します。起動テンプレートは、起動テンプレート ID とバージョン番号によって識別されます。フリートのターゲット容量は 2 インスタンスであり、デフォルト購入オプションは `on-demand` です。この結果、フリートは 2 オンデマンドインスタンス を起動します。

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-0e8c754449b27161c",  
                "Version": "1"  
            }  
        },  
        ],  
        "TargetCapacitySpecification": {  
            "TotalTargetCapacity": 2,  
            "DefaultTargetCapacityType": "on-demand"  
        }  
    }  
}
```

```
        "Version": "1"
    }

],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "on-demand"
}
}
```

#### 例 3: オンデマンドインスタンスをプライマリ容量として起動する

次の例では、フリートの合計ターゲット容量を 2 インスタンス、ターゲット容量を 1 オンデマンドインスタンスとして指定します。デフォルト購入オプションは spot です。フリートは指定されたとおり 1 オンデマンドインスタンスを起動しますが、合計ターゲット容量を満たすために、さらに 1 つ以上のインスタンスを起動する必要があります。購入オプションの差は TotalTargetCapacity - OnDemandTargetCapacity = DefaultTargetCapacityType と計算されます。この結果、フリートは 1 スポットインスタンスを起動します。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity": 1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

#### 例 4: 最低価格の配分戦略を使用してスポットインスタンスを起動する

スポットインスタンスの配分戦略を指定しない場合、デフォルト配分戦略である `lowest-price` が使用されます。次の例では、`lowest-price` の配分戦略を使用します。起動テンプレートをオーバーライドする 3 つの起動条件は、インスタンスタイプが異なりますが、加重容量とサブネットは同じです。合計ターゲット容量は 2 インスタンスで、デフォルト購入オプションは spot です。EC2 フリートは、最低価格の起動条件のインスタンスタイプを使用して 2 スポットインスタンスを起動します。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        }
    ],
    "Overrides": [
        {
            "InstanceType": "c4.large",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-a4f6c5d3"
        },
        {
            "InstanceType": "c3.large",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-a4f6c5d3"
        }
    ]
}
```

```
        },
        {
            "InstanceType": "c5.large",
            "WeightedCapacity": 1,
            "SubnetId": "subnet-a4f6c5d3"
        }
    ]
}
],
"TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "spot"
}
}
```

#### 例 5: 容量予約と優先配分戦略を使用して オンデマンドインスタンス を起動する

キャパシティの予約 の使用戦略を `use-capacity-reservations-first` に設定することで、オンデマンドインスタンス の起動時に最初に オンデマンドキャパシティー予約 を使用するようにフリートを設定できます。また、複数のインスタンスプールに未使用 キャパシティーの予約 がある場合、選択したオンデマンド配分戦略が適用されます。この例では、オンデマンド配分戦略は `prioritized` です。

この例では、利用可能な未使用 キャパシティーの予約 が 15 個あります。これは、フリートの目標オンデマンド容量である 12 オンデマンドインスタンス を超えています。

アカウントには、3 つの異なるプールに以下の 15 個の未使用 キャパシティーの予約 があります。各プールの キャパシティーの予約 の数は `AvailableInstanceCount` で示されます。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "c4.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "c3.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "c5.large",
    "InstancePlatform": "Linux/UNIX",
    "AvailabilityZone": "us-east-1a",
    "AvailableInstanceCount": 5,
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

以下のフリート設定は、この例に関連する設定のみを示しています。オンデマンド配分戦略は `prioritized` であり、キャパシティーの予約 の使用戦略は `use-capacity-reservations-first` です。合計ターゲット容量は 12 で、デフォルトのターゲット容量タイプは `on-demand` です。

### Note

フリートタイプは instant であることが必要です。キャパシティーの予約は他のフリートタイプではサポートされていません。

```
{  
    "LaunchTemplateConfigs": [  
        {  
            "LaunchTemplateSpecification": {  
                "LaunchTemplateId": "lt-1234567890abcdefg",  
                "Version": "1"  
            },  
            "Overrides": [  
                {  
                    "InstanceType": "c4.large",  
                    "AvailabilityZone": "us-east-1a",  
                    "WeightedCapacity": 1,  
                    "Priority": 1.0  
                },  
                {  
                    "InstanceType": "c3.large",  
                    "AvailabilityZone": "us-east-1a",  
                    "WeightedCapacity": 1,  
                    "Priority": 2.0  
                },  
                {  
                    "InstanceType": "c5.large",  
                    "AvailabilityZone": "us-east-1a",  
                    "WeightedCapacity": 1,  
                    "Priority": 3.0  
                }  
            ]  
        },  
        {"TargetCapacitySpecification": {  
            "TotalTargetCapacity": 12,  
            "DefaultTargetCapacityType": "on-demand"  
        },  
        "OnDemandOptions": {  
            "AllocationStrategy": "prioritized",  
            "CapacityReservationOptions": {  
                "UsageStrategy": "use-capacity-reservations-first"  
            }  
        },  
        "Type": "instant",  
    ]  
}
```

上記の設定を使用して instant フリートを作成すると、目標容量を満たすために以下の 12 個のインスタンスが起動されます。

- 5 c4.large オンデマンドインスタンス (us-east-1a) – c4.large (us-east-1a) が最初に優先され、利用可能な未使用 c4.large キャパシティーの予約が 5 つあります。
- 5 c3.large オンデマンドインスタンス (us-east-1a) – c3.large (us-east-1a) が 2 番目に優先され、利用可能な未使用 c3.large キャパシティーの予約が 5 つあります。
- 2 c5.large オンデマンドインスタンス (us-east-1a) – c5.large (us-east-1a) は 3 番目に優先され、利用可能な未使用 c5.large キャパシティーの予約が 5 つあります。そのうちの 2 つのみが目標容量を満たすために必要です。

フリートの起動後、[describe-capacity-reservations](#) を実行して、未使用 キャパシティーの予約の数を確認できます。この例では、以下のレスポンスが表示されます。これは、c4.large および c3.large のすべての

キャパシティの予約が使用され、c5.large の 3 つのキャパシティの予約が未使用のままであることを示しています。

```
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c4.large",  
    "AvailableInstanceCount": 0  
}  
  
{  
    "CapacityReservationId": "cr-222",  
    "InstanceType": "c3.large",  
    "AvailableInstanceCount": 0  
}  
  
{  
    "CapacityReservationId": "cr-333",  
    "InstanceType": "c5.large",  
    "AvailableInstanceCount": 3  
}
```

#### 例 6: 合計ターゲット容量が未使用 キャパシティの予約の数を超えたときに キャパシティの予約と優先配分戦略を使用して オンデマンドインスタンスを起動する

キャパシティの予約の使用戦略を `use-capacity-reservations-first` に設定することで、オンデマンドインスタンスの起動時に最初に オンデマンドキャパシティ予約を使用するようにフリートを設定できます。また、未使用 キャパシティの予約の数が目標オンデマンド容量よりも少ない場合、残りのオンデマンド目標容量は、選択したオンデマンド配分戦略に従って満たされます。この例では、オンデマンド配分戦略は `prioritized` です。

この例では、利用可能な未使用 キャパシティの予約が 15 個あります。これは、フリートの目標オンデマンド容量である 16 オンデマンドインスタンスを下回っています。

アカウントには、3 つの異なるプールに以下の 15 個の未使用 キャパシティの予約があります。各プールの キャパシティの予約の数は `AvailableInstanceCount` で示されます。

```
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c4.large",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}  
  
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c3.large",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}  
  
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c5.large",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,
```

```
    "InstanceMatchCriteria": "open",
    "State": "active"
}
```

以下のフリート設定は、この例に関連する設定のみを示しています。オンデマンド配分戦略は prioritized であり、キャパシティーの予約 の使用戦略は use-capacity-reservations-first です。合計ターゲット容量は 16 で、デフォルトのターゲット容量タイプは on-demand です。

Note

フリートタイプは instant であることが必要です。キャパシティーの予約 は他のフリートタイプではサポートされていません。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
        },
        "Overrides": [
            {
                "InstanceType": "c4.large",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1,
                "Priority": 1.0
            },
            {
                "InstanceType": "c3.large",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1,
                "Priority": 2.0
            },
            {
                "InstanceType": "c5.large",
                "AvailabilityZone": "us-east-1a",
                "WeightedCapacity": 1,
                "Priority": 3.0
            }
        ]
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 16,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "prioritized"
    },
    "Type": "instant",
}
```

上記の設定を使用して instant フリートを作成すると、目標容量を満たすために以下の 16 個のインスタンスが起動されます。

- 6 c4.large オンデマンドインスタンス (us-east-1a) – c4.large (us-east-1a) が最初に優先され、利用可能な未使用 c4.large キャパシティーの予約 が 5 つあります。キャパシティーの予約 は、5 つの オンデマンドインスタンス を起動するために最初に使用され、さらにオンデマンド配分戦略 (この例では prioritized) に従って、追加の オンデマンドインスタンス が起動されます。

- 5 c3.large オンデマンドインスタンス (us-east-1a) – c3.large (us-east-1a) が 2 番目に優先され、利用可能な未使用 c3.large キャパシティーの予約が 5つあります。
- 5 c5.large オンデマンドインスタンス (us-east-1a) – c5.large (us-east-1a) が 3 番目に優先され、利用可能な未使用 c5.large キャパシティーの予約が 5つあります。

フリートの起動後、[describe-capacity-reservations](#) を実行して、未使用 キャパシティーの予約 の数を確認できます。この例では、以下のレスポンスが表示されます。これは、すべてのプール内のすべての キャパシティーの予約が使用されたことを示しています。

```
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "c4.large",  
    "AvailableInstanceCount": 0  
}  
  
{  
    "CapacityReservationId": "cr-222",  
    "InstanceType": "c3.large",  
    "AvailableInstanceCount": 0  
}  
  
{  
    "CapacityReservationId": "cr-333",  
    "InstanceType": "c5.large",  
    "AvailableInstanceCount": 0  
}
```

#### 例 7: 容量予約と最低料金配分戦略を使用して オンデマンドインスタンスを起動する

キャパシティーの予約 の使用戦略を `use-capacity-reservations-first` に設定することで、オンデマンドインスタンス の起動時に最初に オンデマンドキャパシティー予約 を使用するようにフリートを設定できます。また、複数のインスタンスプールに未使用 キャパシティーの予約 がある場合、選択したオンデマンド配分戦略が適用されます。この例では、オンデマンド配分戦略は `lowest-price` です。

この例では、利用可能な未使用 キャパシティーの予約 が 15 個あります。これは、フリートの目標オンデマンド容量である 12 オンデマンドインスタンス を超えています。

アカウントには、3 つの異なるプールに以下の 15 個の未使用 キャパシティーの予約 があります。各プールの キャパシティーの予約 の数は `AvailableInstanceCount` で示されます。

```
{  
    "CapacityReservationId": "cr-111",  
    "InstanceType": "m5.large",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}  
  
{  
    "CapacityReservationId": "cr-222",  
    "InstanceType": "m4.xlarge",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}  
  
{  
    "CapacityReservationId": "cr-333",  
    "InstanceType": "t2.medium",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "AvailableInstanceCount": 5,  
    "InstanceMatchCriteria": "open",  
    "State": "active"  
}
```

```
"CapacityReservationId": "cr-333",
"InstanceType": "m4.2xlarge",
"InstancePlatform": "Linux/UNIX",
"AvailabilityZone": "us-east-1a",
"AvailableInstanceCount": 5,
"InstanceMatchCriteria": "open",
"State": "active"
}
```

以下のフリート設定は、この例に関連する設定のみを示しています。オンデマンド配分戦略は `lowest-price` であり、キャパシティーの予約の使用戦略は `use-capacity-reservations-first` です。合計ターゲット容量は 12 で、デフォルトのターゲット容量タイプは `on-demand` です。

この例では、オンデマンドインスタンスの料金は以下のようになります。

- m5.large – 0.096 USD/時間
- m4.xlarge – 0.20 USD/時間
- m4.2xlarge – 0.40 USD/時間

#### Note

フリートタイプは `instant` であることが必要です。キャパシティーの予約は他のフリートタイプではサポートされていません。

```
{
    "LaunchTemplateConfigs": [
        {
            "LaunchTemplateSpecification": {
                "LaunchTemplateId": "lt-0e8c754449b27161c",
                "Version": "1"
            }
            "Overrides": [
                {
                    "InstanceType": "m5.large",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "m4.xlarge",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1
                },
                {
                    "InstanceType": "m4.2xlarge",
                    "AvailabilityZone": "us-east-1a",
                    "WeightedCapacity": 1
                }
            ]
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 12,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    }
},
```

```
    "Type": "instant",  
}
```

上記の設定を使用して instant フリートを作成すると、目標容量を満たすために以下の 12 個のインスタンスが起動されます。

- 5 m5.large オンデマンドインスタンス (us-east-1a) – m5.large (us-east-1a) は最低料金であり、利用可能な未使用 m5.large キャパシティーの予約が 5 つあります。
- 5 m4.xlarge オンデマンドインスタンス (us-east-1a) – m4.xlarge (us-east-1a) は次に低い料金であり、利用可能な未使用 m4.xlarge キャパシティーの予約が 5 つあります。
- 2 m4.2xlarge オンデマンドインスタンス (us-east-1a) – m4.2xlarge (us-east-1a) は 3 番目に低い料金であり、利用可能な未使用 m4.2xlarge キャパシティーの予約は 5 つあります。そのうちの 2 つのみが目標容量を満たすために必要です。

フリートの起動後、[describe-capacity-reservations](#) を実行して、未使用 キャパシティーの予約の数を確認できます。この例では、以下のレスポンスが表示されます。これは、m5.large および m4.xlarge のすべてのキャパシティーの予約が使用され、m4.2xlarge の 3 つのキャパシティーの予約が未使用のままであることを示しています。

```
{  
  "CapacityReservationId": "cr-111",  
  "InstanceType": "m5.large",  
  "AvailableInstanceCount": 0  
}  
  
{  
  "CapacityReservationId": "cr-222",  
  "InstanceType": "m4.xlarge",  
  "AvailableInstanceCount": 0  
}  
  
{  
  "CapacityReservationId": "cr-333",  
  "InstanceType": "m4.2xlarge",  
  "AvailableInstanceCount": 3  
}
```

#### 例 8: 合計ターゲット容量が未使用 キャパシティーの予約の数を超えたときに容量予約と最低料金配分戦略を使用して オンデマンドインスタンスを起動する

キャパシティーの予約の使用戦略を `use-capacity-reservations-first` に設定することで、オンデマンドインスタンスの起動時に最初に オンデマンドキャパシティー予約を使用するようにフリートを設定できます。また、未使用 キャパシティーの予約の数が目標オンデマンド容量よりも少ない場合、残りのオンデマンド目標容量は、選択した オンデマンド配分戦略に従って満たされます。この例では、オンデマンド配分戦略は `lowest-price` です。

この例では、利用可能な未使用 キャパシティーの予約が 15 個あります。これは、フリートの目標オンデマンド容量である 16 オンデマンドインスタンスを下回っています。

アカウントには、3 つの異なるプールに以下の 15 個の未使用 キャパシティーの予約があります。各プールの キャパシティーの予約の数は `AvailableInstanceCount` で示されます。

```
{  
  "CapacityReservationId": "cr-111",  
  "InstanceType": "m5.large",  
  "InstancePlatform": "Linux/UNIX",  
  "AvailabilityZone": "us-east-1a",  
  "AvailableInstanceCount": 5,
```

```
"InstanceMatchCriteria": "open",
"State": "active"
}

{
"CapacityReservationId": "cr-222",
"InstanceType": "m4.xlarge",
"InstancePlatform": "Linux/UNIX",
"AvailabilityZone": "us-east-1a",
"AvailableInstanceCount": 5,
"InstanceMatchCriteria": "open",
"State": "active"
}

{
"CapacityReservationId": "cr-333",
"InstanceType": "m4.2xlarge",
"InstancePlatform": "Linux/UNIX",
"AvailabilityZone": "us-east-1a",
"AvailableInstanceCount": 5,
"InstanceMatchCriteria": "open",
"State": "active"
}
```

以下のフリート設定は、この例に関連する設定のみを示しています。オンデマンド配分戦略は `lowest-price` であり、キャパシティーの予約の使用戦略は `use-capacity-reservations-first` です。合計ターゲット容量は 16 で、デフォルトのターゲット容量タイプは `on-demand` です。

この例では、オンデマンドインスタンスの料金は以下のようになります。

- m5.large – 0.096 USD/時間
- m4.xlarge – 0.20 USD/時間
- m4.2xlarge – 0.40 USD/時間

#### Note

フリートタイプは `instant` であることが必要です。キャパシティーの予約は他のフリートタイプではサポートされていません。

```
{
"LaunchTemplateConfigs": [
{
"LaunchTemplateSpecification": {
"LaunchTemplateId": "lt-0e8c754449b27161c",
"Version": "1"
},
"Overrides": [
{
"InstanceType": "m5.large",
"AvailabilityZone": "us-east-1a",
"WeightedCapacity": 1
},
{
"InstanceType": "m4.xlarge",
"AvailabilityZone": "us-east-1a",
"WeightedCapacity": 1
},
{
"InstanceType": "m4.2xlarge",
"AvailabilityZone": "us-east-1a",
"WeightedCapacity": 1
}
]
}
]
```

```
        }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 16,
        "DefaultTargetCapacityType": "on-demand"
    },
    "OnDemandOptions": {
        "AllocationStrategy": "lowest-price"
        "CapacityReservationOptions": {
            "UsageStrategy": "use-capacity-reservations-first"
        }
    },
    "Type": "instant",
}
```

上記の設定を使用して instant フリートを作成すると、目標容量を満たすために以下の 16 個のインスタンスが起動されます。

- 6 m5.large オンデマンドインスタンス (us-east-1a) – m5.large (us-east-1a) は最低料金であり、利用可能な未使用 m5.large キャパシティーの予約が 5 つあります。キャパシティーの予約は、5 つのオンデマンドインスタンスを起動するために最初に使用され、さらにオンデマンド配分戦略(この例では lowest-price)に従って、追加の オンデマンドインスタンスが起動されます。
- 5 m4.xlarge オンデマンドインスタンス (us-east-1a) – m4.xlarge (us-east-1a) は次に低い料金であり、利用可能な未使用 m4.xlarge キャパシティーの予約が 5 つあります。
- 5 m4.2xlarge オンデマンドインスタンス (us-east-1a) – m4.2xlarge (us-east-1a) は 3 番目に低い料金であり、利用可能な未使用 m4.2xlarge キャパシティーの予約が 5 つあります。

フリートの起動後、[describe-capacity-reservations](#) を実行して、未使用 キャパシティーの予約の数を確認できます。この例では、以下のレスポンスが表示されます。これは、すべてのプール内のすべての キャパシティーの予約が使用されたことを示しています。

```
{
    "CapacityReservationId": "cr-111",
    "InstanceType": "m5.large",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-222",
    "InstanceType": "m4.xlarge",
    "AvailableInstanceCount": 0
}

{
    "CapacityReservationId": "cr-333",
    "InstanceType": "m4.2xlarge",
    "AvailableInstanceCount": 0
}
```

## Linux インスタンスへの接続

起動した Linux インスタンスに接続し、ローカルコンピュータとインスタンスの間でファイルを転送します。

Windows インスタンスに接続する必要がある場合は、Windows インスタンスの Amazon EC2 ユーザーガイドの「[Windows インスタンスへの接続](#)」を参照してください。

## 接続オプション

ローカルコンピュータのオペレーティングシステムによって、ローカルコンピュータから Linux インスタンスに接続する必要があるオプションが決定されます。

### Linux および MacOS X 用のオプション

- [SSH クライアント \(p. 508\)](#)
- [EC2 Instance Connect \(p. 511\)](#)
- [AWS Systems Manager セッションマネージャー](#)

### Windows 用のオプション

- [PuTTY \(p. 520\)](#)
- [SSH クライアント \(p. 508\)](#)
- [AWS Systems Manager Session Manager](#)
- [Windows Subsystem for Linux \(p. 525\)](#)

Amazon EC2 コンソールには、Java SSH クライアントを使用してブラウザからインスタンスに直接接続するオプションがあります。ただし、現在多くのブラウザではサポートされていません。詳細については、「[ブラウザを使用して接続できない \(p. 1142\)](#)」を参照してください。

## インスタンスに接続するための一般的な前提条件

Linux インスタンスに接続する前に、以下の前提条件を満たしていることを確認します。

- [インスタンスに関する情報を取得する \(p. 506\)](#)
- [インスタンスへのインバウンドトラフィックを有効にする \(p. 507\)](#)
- [プライベートキーを見つける \(p. 507\)](#)
- [\(オプション\) インスタンスのフィンガープリントを取得する \(p. 507\)](#)

## インスタンスに関する情報を取得する

- インスタンスの ID を取得する。

Amazon EC2 コンソールを使用して、インスタンスの ID を取得できます ([Instance ID] 列を確認します)。必要に応じて、[describe-instances](#) (AWS CLI) または [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) コマンドを使用することもできます。

- インスタンスのパブリック DNS 名を取得します。

インスタンスのパブリック DNS を取得するには、Amazon EC2 コンソールを使用します。[パブリック DNS (IPv4)] 列を確認します。この列が非表示になっている場合は、[Show/Hide] アイコンを選択してから、[Public DNS (IPv4)] を選択します。必要に応じて、[describe-instances](#) (AWS CLI) または [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) コマンドを使用することもできます。

- (IPv6 のみ) インスタンスの IPv6 アドレスを取得します。

インスタンスに IPv6 アドレスを割り当てた場合は、オプションでパブリック IPv4 アドレスまたはパブリック IPv4 DNS ホスト名でなく IPv6 アドレスを使用して、インスタンスに接続できます。ローカルコンピュータに IPv6 アドレスがあり、IPv6 を使用するように設定されている必要があります。インスタンスの IPv6 アドレスを取得するには、Amazon EC2 コンソールを使用します。[IPv6 IP] フィールドを確認します。必要に応じて、[describe-instances](#) (AWS CLI) または [Get-EC2Instance](#) (AWS Tools for Windows PowerShell) コマンドを使用することもできます。IPv6 の詳細については、「[IPv6 アドレス \(p. 687\)](#)」を参照してください。

- インスタンスの起動に使用した AMI のデフォルトのユーザー名を取得します。
  - Amazon Linux 2 または Amazon Linux AMI の場合は、ユーザー名は `ec2-user` です。

- CentOS AMI の場合、ユーザー名は `centos` です。
- Debian AMI の場合は、ユーザー名は `admin` または `root` です。
- Fedora AMI の場合、ユーザー名は `ec2-user` または `fedora` です。
- RHEL AMI の場合は、ユーザー名は `ec2-user` または `root` のどちらかです。
- SUSE AMI の場合は、ユーザー名は `ec2-user` または `root` のどちらかです。
- Ubuntu AMI の場合は、ユーザー名は `ubuntu` です。
- それ以外の場合で、`ec2-user` および `root` が機能しない場合は、AMI プロバイダーに確認してください。

## インスタンスへのインバウンドトラフィックを有効にする

- IP アドレスからインスタンスへのインバウンド SSH トラフィックを有効にします。

インスタンスに関連付けられているセキュリティグループで、IP アドレスからの受信 SSH トラフィックが許可されることを確認します。VPC のデフォルトのセキュリティグループでは、着信 SSH トラフィックはデフォルトでは許可されません。[インスタンスの起動] ウィザードで作成されたセキュリティグループでは、デフォルトで SSH トラフィックが有効になります。詳細については、「[Linux インスタンス用の受信トラフィックの認可 \(p. 897\)](#)」を参照してください。

## プライベートキーを見つける

- プライベートキーの検索とアクセス許可の確認

インスタンスの起動時に指定したキーペアの `.pem` ファイルの、コンピュータ上の場所への完全修飾パスを取得します。キーペアを作成した方法の詳細については、「[Amazon EC2 を使用してキーペアを作成する](#)」を参照してください。

`.pem` ファイルに対するアクセス許可が 0777 ではなく 0400 であることを確認します。詳細については、「[エラー: Unprotected Private Key File \(保護されていないプライベートキーファイル\) \(p. 1141\)](#)」を参照してください。

## プライベートキーのアクセス権限を設定するには

1. コマンドラインシェルで、インスタンスを起動した時に作成したプライベートキーファイルの場所にディレクトリを変更します。
2. 次のコマンドを使用してプライベートキーファイルのアクセス許可を設定し、お客様以外のユーザーが読み取ることができないようにします。

```
chmod 400 /path/my-key-pair.pem
```

これらのアクセス権限を設定しないと、このキーペアを使用してインスタンスに接続できません。詳細については、「[エラー: Unprotected Private Key File \(保護されていないプライベートキーファイル\) \(p. 1141\)](#)」を参照してください。

## (オプション) インスタンスのフィンガープリントを取得する

中間者攻撃を防ぐため、インスタンスを接続するときに RSA キーフィンガープリントを検証することができます。フィンガープリントの検証は、サードパーティのパブリック AMI からインスタンスを起動した場合に役立ちます。

まず、インスタンスのフィンガープリントを取得します。次に、インスタンスを接続する際に、フィンガープリントを検証するようにプロンプトされます。取得したフィンガープリントと表示されたフィ

ンガープリントを比較して検証できます。これらのフィンガープリントが一致しない場合、「中間者(MITM)」攻撃を受けている可能性があります。一致する場合には、安心してインスタンスに接続できます。

#### インスタンスのフィンガープリントを取得するための前提条件

- インスタンスのフィンガープリントを取得するには、AWS CLI を使用する必要があります。AWS CLI のインストールに関する詳細は、『AWS Command Line Interface ユーザーガイド』の「[AWS コマンドラインインターフェイス](#)」を参照してください。
- インスタンスは `pending` 状態ではなく、`running` 状態であることが必要です。

#### インスタンスのフィンガープリントを取得するには

- 使用するローカルシステム（インスタンス上ではなく）で `get-console-output` (AWS CLI) コマンドを使用して、次に示すようにフィンガープリントを取得します。

```
$ aws ec2 get-console-output --instance-id instance_id
```

以下は、探すセクションを示す例です。

```
-----BEGIN SSH HOST KEY FINGERPRINTS-----  
... 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f ...  
-----END SSH HOST KEY FINGERPRINTS-----
```

- 生成された出力で SSH HOST KEY FINGERPRINTS セクションを見つけ、RSA フィンガープリント（たとえば `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f`）を書き留めます。SSH HOST KEY FINGERPRINTS セクションは、インスタンスの最初の起動後にのみ使用できます。

## SSH を使用した Linux インスタンスへの接続

インスタンスを起動したら、これに接続し、普通のコンピュータと同じように使用できます。

次の手順では、SSH クライアントを使って、インスタンスに接続する方法について説明します。インスタンスの接続でエラーが発生した場合は、「[インスタンスへの接続に関するトラブルシューティング \(p. 1135\)](#)」を参照してください。

### 前提条件

Linux インスタンスに接続する前に、以下の前提条件を満たしていることを確認してください。

インスタンスの準備ができていることを確認する

インスタンスを起動してから接続できるようになるまでには、数分かかる場合があります。インスタンスのステータスチェックが成功していることを確認します。この情報は、[Instances (インスタンス)] ページの [Status Checks (ステータスチェック)] 列で確認できます。

インスタンスに接続するための一般的な前提条件を確認する

詳細については、「[インスタンスに接続するための一般的な前提条件 \(p. 506\)](#)」を参照してください。

必要に応じてローカルコンピューターに SSH クライアントをインストールする

ローカルコンピューターには、デフォルトで SSH クライアントがインストールされている場合があります。これは、コマンドラインに「`ssh`」と入力することで確認できます。ご使用のコンピューターでのこのコマンドが認識されない場合、SSH クライアントをインストールできます。

- 最近バージョンの Windows サーバー 2019 と Windows 10 - OpenSSH がインストール可能なコンポーネントとして含まれています。詳細については、「[Windows での OpenSSH](#)」を参照してください。
- 以前のバージョンの Windows - OpenSSH をダウンロードしてインストールします。詳細については、「[Win32-OpenSSH](#)」を参照してください。
- Linux および MacOS X - OpenSSH をダウンロードしてインストールします。詳細については、<http://www.openssh.com> を参照してください。

## SSH クライアントを使用して Linux インスタンスに接続する

SSH クライアントを使用して Linux インスタンスに接続するには、次の手順に従います。インスタンスの接続でエラーが発生した場合は、「[インスタンスへの接続に関するトラブルシューティング \(p. 1135\)](#)」を参照してください。

SSH を使用してインスタンスに接続するには

- ターミナルウインドウで ssh コマンドを使用して、インスタンスに接続します。プライベートキー (.pem) のパスとファイル名、AMI のユーザー名、およびインスタンスのパブリック DNS 名や IPv6 アドレスを指定します。プライベートキー、AMI のユーザー名、およびインスタンスの DNS 名や IPv6 アドレスの検索方法の詳細については、「[プライベートキーを見つける \(p. 507\)](#)」および「[インスタンスに関する情報を取得する \(p. 506\)](#)」を参照してください。インスタンスに接続するには、次のいずれかの操作を実行します。
  - (パブリック DNS) インスタンスのパブリック DNS を使用して接続するには、次のコマンドを入力します。

```
ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

- (IPv6) インスタンスに IPv6 アドレスがある場合、インスタンスの IPv6 アドレスを使用して接続するには、次のコマンドを入力します。

```
ssh -i /path/my-key-pair.pem ec2-user@2001:db8:1234:1a00:9691:9503:25ad:1761
```

以下のようなレスポンスが表示されます。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established.  
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

- (オプション) セキュリティアラートのフィンガープリントが、[\(オプション\) インスタンスのフィンガープリントを取得する \(p. 507\)](#) で事前に取得したフィンガープリントと一致することを確認します。これらのフィンガープリントが一致しない場合、「中間者 (MITM)」攻撃を受けている可能性があります。一致した場合は、次の手順に進んでください。
- yes と入力します。以下のようないいなレスポンスが表示されます。

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)  
to the list of known hosts.
```

## SCP を使用した Linux から Linux インスタンスへのファイルの転送

ローカルコンピュータと Linux インスタンスの間でファイルを転送する方法の 1 つとして、セキュアコピープロトコル (SCP) を使用します。このセクションでは、SCP でファイルを転送する方法について説明します。この手順は、SSH を使用してインスタンスに接続する手順と似ています。

## 前提条件

- インスタンスにファイルを転送するための一般的な前提条件の確認

インスタンスにファイルを転送するための一般的な前提条件は、インスタンスに接続するための一般的な前提条件と同様です。詳細については、「[インスタンスに接続するための一般的な前提条件 \(p. 506\)](#)」を参照してください。

- SCP クライアントのインストール

ほとんどの Linux、Unix、および Apple コンピュータには、デフォルトで SCP クライアントが含まれています。含まれていない場合は、OpenSSH プロジェクトから、SSH ツールの完全なスイートの無料実装が提供されており、これに SCP クライアントが含まれます。詳細については、<http://www.openssh.org> を参照してください。

SCP を使用してファイルを転送するステップを次に示します。既に SSH でインスタンスに接続し、フィンガープリントの確認が完了している場合は、SCP コマンドを実行するステップ (ステップ 4) から開始できます。

### SCP を使用してファイルを転送するには

- インスタンスのパブリック DNS 名を使って、インスタンスにファイルを転送します。たとえば、プライベートキーの名前が `my-key-pair`、転送するファイルが `SampleFile.txt`、ユーザー名が `ec2-user`、インスタンスのパブリック DNS の名前が `ec2-198-51-100-1.compute-1.amazonaws.com` の場合、次のコマンドを使って、ファイルを `ec2-user` ホームディレクトリにコピーします。

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com:~
```

以下のようなレスポンスが表示されます。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' 
can't be established.
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

- (IPv6 のみ) 別の方法として、インスタンスの IPv6 アドレスを使用してファイルを転送することもできます。IPv6 アドレスは、\でエスケープした角かっこ ([]) で囲む必要があります。

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@
\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~
```

- (オプション) セキュリティアラートのフィンガープリントが、[\(オプション\) インスタンスのフィンガープリントを取得する \(p. 507\)](#) で事前に取得したフィンガープリントと一致することを確認します。これらのフィンガープリントが一致しない場合、「中間者 (MITM)」攻撃を受けている可能性があります。一致した場合は、次の手順に進んでください。
- yes** と入力します。

以下のようなレスポンスが表示されます。

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
Sending file modes: C0644 20 SampleFile.txt
Sink: C0644 20 SampleFile.txt
SampleFile.txt                                         100%    20      0.0KB/s   00:00
```

[bash: scp: command not found] エラーを受け取った場合は、まず Linux インスタンスに scp をインストールする必要があります。一部のオペレーティングシステムでは、これは `openssh-clients`/パッケージに含まれます。Amazon Linux-optimized Amazon ECS などの AMI バリアントでは、以下のコマンドを使用して scp をインストールします。

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

- 逆の方向 (Amazon EC2 インスタンスからローカルコンピュータに) にファイルを転送する場合は、ホストパラメータの順番を逆にします。たとえば、`SampleFile.txt` ファイルを EC2 インスタンスからローカルコンピュータのホームディレクトリに `SampleFile2.txt` として転送するには、ローカルコンピュータで次のコマンドを実行します。

```
scp -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/SampleFile.txt ~/SampleFile2.txt
```

- (IPv6 のみ) 別の方法として、インスタンスの IPv6 アドレスを使用して別の方向にファイルを転送することもできます。

```
scp -i /path/my-key-pair.pem ec2-user@[2001:db8:1234:1a00:9691:9503:25ad:1761]:~/SampleFile.txt ~/SampleFile2.txt
```

## EC2 Instance Connect を使用して Linux インスタンスに接続する

Amazon EC2 Instance Connect は、Secure Shell (SSH) を使用してインスタンスに接続するシンプルで安全な方法を提供します。EC2 Instance Connect では、AWS Identity and Access Management (IAM) ポリシーおよびプリンシパルを使用して SSH からインスタンスへのアクセスをコントロールします。SSH キーを共有および管理する必要はありません。EC2 Instance Connect を使用したすべての接続リクエストは、AWS CloudTrail にログとして記録されるため、接続リクエストを監査できます (p. 683)。

Instance Connect を使用して、ブラウザベースのクライアント、Amazon EC2 Instance Connect CLI、または任意の SSH クライアントを使用して Linux インスタンスに接続できます。

EC2 Instance Connect を使用してインスタンスに接続すると、Instance Connect API から 1 回限り使用的 SSH パブリックキーが [インスタンスマタデータ \(p. 593\)](#) にプッシュされ、60 秒間保持されます。IAM ユーザーにアタッチされた IAM ポリシーにより、IAM ユーザーはパブリックキーをインスタンスマタデータにプッシュすることを許可されます。SSH デーモンは、Instance Connect のインストール時に設定される `AuthorizedKeysCommand` および `AuthorizedKeysCommandUser` により、インスタンスマタデータからパブリックキーを見つけて認証を行い、ユーザーをインスタンスに接続します。

### Note

Windows を実行しているローカルコンピュータから Linux インスタンスに接続する場合は、代わりに「[PuTTY を使用した Windows から Linux インスタンスへの接続 \(p. 520\)](#)」および「[Windows Subsystem for Linux を使用した Windows から Linux インスタンスへの接続 \(p. 525\)](#)」を参照してください。

### コンテンツ

- [EC2 Instance Connect のセットアップ \(p. 512\)](#)
- [EC2 Instance Connect を使用して接続する \(p. 517\)](#)
- [EC2 Instance Connect のアンインストール \(p. 519\)](#)

## EC2 Instance Connect のセットアップ<sup>¶</sup>

Amazon Linux 2 2.0.20190618 以降には、EC2 Instance Connect が事前設定されています。サポートされているその他の Linux ディストリビューションでは、Instance Connect の使用をサポートするインスタンスごとに Instance Connect をセットアップする必要があります。これは、インスタンスごとに 1 回のみ必要なセットアップです。

Instance Connect をセットアップするためのタスク

- [ステップ 1: インスタンスへのネットワークアクセスを設定する \(p. 512\)](#)
- [ステップ 2: EC2 Instance Connect をインスタンスにインストールする \(p. 513\)](#)
- [ステップ 3: \(オプション\) EC2 Instance Connect CLI をインストールする \(p. 515\)](#)
- [ステップ 4: EC2 Instance Connect 用に IAM のアクセス許可を設定する \(p. 515\)](#)

### 制約事項

- 以下の Linux ディストリビューションがサポートされています。
  - Amazon Linux 2 (すべてのバージョン)
  - Ubuntu 16.04 以降
- SSH 認証の `AuthorizedKeysCommand` および `AuthorizedKeysCommandUser` 設定を構成した場合、EC2 Instance Connect をインストールしても更新されません。その結果、Instance Connect を使用することはできません。

### 前提条件

- SSH を使用してインスタンスに接続するための一般的な前提条件を確認します。

詳細については、「[インスタンスに接続するための一般的な前提条件 \(p. 506\)](#)」を参照してください。

- 使用するローカルコンピュータに SSH クライアントをインストールします。

ほとんどの場合、ローカルコンピュータにはデフォルトで SSH クライアントがインストールされています。SSH クライアントがあるかどうかを確認するには、コマンドラインで `ssh` と入力します。使用的ローカルコンピュータでこのコマンドが認識されない場合、SSH クライアントをインストールできます。Linux または macOS X に SSH クライアントをインストールする詳細については、「<http://www.openssh.com>」を参照してください。Windows 10 に SSH クライアントをインストールする詳細については、「[Windows の OpenSSH](#)」を参照してください。

- ローカルコンピュータに AWS CLI をインストールします。

IAM アクセス許可を設定するには、AWS CLI を使用する必要があります。AWS CLI のインストールの詳細については、AWS Command Line Interface ユーザーガイドの「[AWS CLI のインストール](#)」を参照してください。

- [Ubuntu] AWS CLI をインスタンスにインストールします。

EC2 Instance Connect を Ubuntu インスタンスにインストールするには、インスタンスで AWS CLI を使用する必要があります。AWS CLI のインストールの詳細については、AWS Command Line Interface ユーザーガイドの「[AWS CLI のインストール](#)」を参照してください。

### ステップ 1: インスタンスへのネットワークアクセスを設定する

EC2 Instance Connect をインストールして、ユーザーがインスタンスに接続できるように、インスタンスへの次のネットワークアクセスを設定する必要があります。

- インスタンスに関連付けられているセキュリティグループで、IP アドレスからのインバウンド SSH ト ラフィック ([p. 898](#)) がポート 22 で許可されることを確認します。VPC のデフォルトのセキュリティ

ループでは、着信 SSH トラフィックはデフォルトでは許可されません。起動ウィザードで作成されたセキュリティグループは、デフォルトで受信 SSH トラフィックを許可します。詳細については、「[Linux インスタンス用の受信トラフィックの認可 \(p. 897\)](#)」を参照してください。

- (ブラウザベースのクライアント) [サービスに発行された推奨される IP ブロックからのインバウンド SSH トラフィックをインスタンスで許可することをお勧めします。](#) EC2 Instance Connect サブネットの IP 範囲を取得するには、service パラメータの EC2\_INSTANCE\_CONNECT フィルタを使用します。詳細については、Amazon ウェブ サービス全般のリファレンスの「[AWS IP アドレスの範囲](#)」を参照してください。

## ステップ 2: EC2 Instance Connect をインスタンスにインストールする

EC2 Instance Connect をインストールすると、インスタンスに SSH デーモンが設定されます。EC2 Instance Connect をインストールする手順は、Amazon Linux 2 および Ubuntu を使用して起動したインスタンスに応じて異なります。

Amazon Linux 2

EC2 Instance Connect を Amazon Linux 2 で起動したインスタンスにインストールするには

1. SSH を使用してインスタンスに接続します。

インスタンスの起動時にインスタンスに割り当てた SSH キーペアと、インスタンスを起動するために使用した AMI のデフォルトのユーザー名を使用します。Amazon Linux 2 の場合、デフォルトのユーザー名は ec2-user です。

たとえば、Amazon Linux 2 を使用してインスタンスを起動した場合、インスタンスのパブリック DNS は ec2-a-b-c-d.us-west-2.compute.amazonaws.com、キーペアは my\_ec2\_private\_key.pem です。次のコマンドを使用して SSH 経由でインスタンスに接続します。

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

インスタンスへの接続の詳細については、[SSH を使用した Linux インスタンスへの接続 \(p. 508\)](#) を参照してください。

2. インスタンスに EC2 Instance Connect パッケージをインストールします。

Amazon Linux 2 では、yum install コマンドを使用します。

```
[ec2-user ~]$ sudo yum install ec2-instance-connect
```

4 つの新しいファイルが /opt/aws/bin/ フォルダに表示されます。

```
eic_curlAuthorizedKeys  
eic_harvestHostkeys  
eic_parseAuthorizedKeys  
eic_runAuthorizedKeys
```

3. (オプション) Instance Connect がインスタンスに正常にインストールされたことを確認します。

sudo less コマンドを使用して /etc/ssh/sshd\_config ファイルが以下のとおりに正しく更新されたことを確認します。

```
[ec2-user ~]$ sudo less /etc/ssh/sshd_config
```

/etc/ssh/sshd\_config ファイルの AuthorizedKeysCommand 行と AuthorizedKeysCommandUser 行に以下の値が含まれていれば、Instance Connect は正常にインストールされています。

```
AuthorizedKeysCommand /opt/aws/bin/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand は、インスタンスマタデータからキーを探すように eic\_run\_authorized\_keys ファイルを設定します。
- AuthorizedKeysCommandUser は、システムユーザーを ec2-instance-connect として設定します。

#### Note

AuthorizedKeysCommand や AuthorizedKeysCommandUser を設定済みである場合は、Instance Connect をインストールしても値は変更されないため、Instance Connect は使用できません。

#### Ubuntu

EC2 Instance Connect を Ubuntu 16.04 で起動したインスタンスにインストールするには

1. SSH を使用してインスタンスに接続します。

インスタンスの起動時にインスタンスに割り当てた SSH キーペアと、インスタンスを起動するために使用した AMI のデフォルトのユーザー名を使用します。Ubuntu AMI の場合、ユーザー名は ubuntu です。

Ubuntu を使用してインスタンスを起動した場合、インスタンスのパブリック DNS は ec2-a-b-c-d.us-west-2.compute.amazonaws.com、キーペアは my\_ec2\_private\_key.pem です。次のコマンドを使用して SSH 経由でインスタンスに接続します。

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

インスタンスへの接続の詳細については、[SSH を使用した Linux インスタンスへの接続 \(p. 508\)](#) を参照してください。

2. (オプション) インスタンスに最新の Ubuntu AMI があることを確認します。

Ubuntu では、apt-get update コマンドを使用して、インスタンスのすべてのパッケージを更新します。

```
ubuntu:~$ sudo apt-get update
```

3. インスタンスに Instance Connect パッケージをインストールします。

Ubuntu では、sudo apt-get コマンドを使用して .deb パッケージをインストールします。

```
ubuntu:~$ sudo apt-get install ec2-instance-connect
```

4 つの新しいファイルが /usr/share/ec2-instance-connect/ フォルダに表示されます。

```
eic_curl_authorized_keys
eic_harvest_hostkeys
eic_parse_authorized_keys
```

eic\_run\_authorized\_keys

4. (オプション) Instance Connect ガインスタンスに正常にインストールされたことを確認します。

sudo less コマンドを使用して /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf ファイルが以下のとおりに正しく更新されたことを確認します。

```
ubuntu:~$ sudo less /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf
```

/lib/systemd/system/ssh.service.d/ec2-instance-connect.conf ファイルの AuthorizedKeysCommand 行と AuthorizedKeysCommandUser 行に以下の値が含まれていれば、Instance Connect は正常にインストールされています。

```
AuthorizedKeysCommand /usr/share/ec2-instance-connect/eic_run_authorized_keys %u %f
AuthorizedKeysCommandUser ec2-instance-connect
```

- AuthorizedKeysCommand は、インスタンスマタデータからキーを探すように eic\_run\_authorized\_keys ファイルを設定します。
- AuthorizedKeysCommandUser は、システムユーザーを ec2-instance-connect として設定します。

#### Note

AuthorizedKeysCommand や AuthorizedKeysCommandUser を設定済みである場合は、Instance Connect をインストールしても値は変更されないため、Instance Connect は使用できません。

EC2 Instance Connect パッケージの詳細については、GitHub ウェブサイトの「[aws/aws-ec2-instance-connect-config](#)」を参照してください。

#### ステップ 3: (オプション) EC2 Instance Connect CLI をインストールする

EC2 Instance Connect CLI は、標準の SSH 呼び出しと似たようなインターフェイスを提供します。このインターフェイスでは、EC2 インスタンス情報へのクエリの実行、一時的なパブリックキーの生成と発行、単一のコマンド mssh *instance\_id* を介しての SSH 接続の確立を行います。

#### Note

ユーザーがブラウザベースのクライアントまたは SSH クライアントのみを使用してインスタンスに接続する場合は、EC2 Instance Connect CLI をインストールする必要はありません。

EC2 Instance Connect CLI パッケージをインストールするには

pip を使用して、ec2instanceconnectcli パッケージをインストールします。詳細について  
は、GitHub ウェブサイトの「[aws/aws-ec2-instance-connect-cli](#)」、および Python Package Index (PyPI)  
ウェブサイトの「<https://pypi.org/project/ec2instanceconnectcli/>」を参照してください。

```
$ pip install ec2instanceconnectcli
```

#### ステップ 4: EC2 Instance Connect 用に IAM のアクセス許可を設定する

IAM ユーザーが EC2 Instance Connect を使用してインスタンスに接続できるように、パブリックキーをインスタンスにプッシュするアクセス許可を付与する必要があります。詳細については、IAM ユーザーガイドの「[Amazon EC2 Instance Connect のアクション、リソース、および条件キー](#)」を参照してください。

次の手順では、AWS CLI を使用してポリシーを作成し、アタッチする方法を示します。AWS マネジメントコンソールを使用する手順については、IAM ユーザーガイドの「[IAM ポリシーを作成する \(コンソール\)](#)」と「[ポリシーをユーザーに直接アタッチすることによるアクセス権限の追加](#)」を参照してください。

#### 制限

Instance Connect のタグベースの認証は現在サポートされていません。

#### IAM ユーザーに EC2 Instance Connect のアクセス許可を付与するには (AWS CLI)

- 以下を含む JSON ポリシードキュメントを作成します。

- ec2-instance-connect:SendSSHPublicKey アクション。これにより、パブリックキーをインスタンスにプッシュするためのアクセス許可を IAM ユーザーに付与します。ec2-instance-connect:SendSSHPublicKey では、特定の EC2 インスタンスへのアクセスを制限することを検討してください。それ以外の場合、このアクセス許可を持つすべての IAM ユーザーは、すべての EC2 インスタンスに接続できます。
- ec2:osuser 条件。これは、パブリックキーをインスタンスにプッシュできる OS ユーザーの名前を指定します。インスタンスの起動に使用した AMI のデフォルトのユーザー名を使用します。たとえば、Amazon Linux 2 のデフォルトのユーザー名は ec2-user、Ubuntu の場合は ubuntu です。
- ec2:DescribeInstances アクション。これは、ラッパーがこのアクションを呼び出すため、EC2 Instance Connect CLI を使用するときに必要です。IAM ユーザーには、別のポリシーからこのアクションを呼び出すアクセス許可がすでに付与されている場合があります。

以下に、ポリシードキュメントの例を示します。ユーザーが SSH クライアントのみを使用してインスタンスに接続する場合は、ec2:DescribeInstances アクションのステートメントを省略できます。Resource で指定されたインスタンスを置き換えて、EC2 Instance Connect を使用してすべての EC2 インスタンスへのアクセスをユーザーに許可できます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2-instance-connect:SendSSHPublicKey",  
            "Resource": [  
                "arn:aws:ec2:region:account-id:instance/i-1234567890abcdef0",  
                "arn:aws:ec2:region:account-id:instance/i-0598c7d356eba48d7"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:osuser": "ami-username"  
                }  
            },  
            {  
                "Effect": "Allow",  
                "Action": "ec2:DescribeInstances",  
                "Resource": "*"  
            }  
        ]  
    ]  
}
```

- create-policy コマンドを使用して新しい管理ポリシーを作成し、新しいポリシーの内容として使用するために作成した JSON ドキュメントを指定します。

```
$ aws iam create-policy --policy-name my-policy --policy-document file://JSON-file-name
```

- attach-user-policy コマンドを使用して、管理ポリシーを指定した IAM ユーザーにアタッチします。--user-name パラメータで、IAM ユーザーのわかりやすい名前 (ARN ではなく) を指定します。

```
$ aws iam attach-user-policy --policy-arn arn:aws:iam::account-id:policy/my-policy --user-name IAM-friendly-name
```

## EC2 Instance Connect を使用して接続する

次の手順では、EC2 Instance Connect を使用して Linux インスタンスに接続する方法について説明します。

Instance Connect を使用して接続するためのオプション

- ・[ブラウザベースのクライアントを使用した接続 \(p. 518\)](#)
- ・[EC2 Instance Connect CLI を使用して接続する \(p. 518\)](#)
- ・[独自のキーと SSH クライアントを使用して接続する \(p. 518\)](#)

### 制約事項

- ・以下の Linux ディストリビューションがサポートされています。
  - ・Amazon Linux 2 (すべてのバージョン)
  - ・Ubuntu 16.04 以降
- ・ブラウザベースのクライアントを使用して接続するには、インスタンスにパブリック IPv4 アドレスが必要です。
- ・EC2 Instance Connect CLI を使用して接続するため、インスタンスにパブリック IPv4 アドレスは必要ありません。これは、プライベート IP アドレスを使用できるためです。インスタンスにパブリック IP アドレスとプライベート IP アドレスの両方がある場合、API はまずパブリック IP アドレスを使用して接続を試みます。
- ・EC2 Instance Connect は IPv6 アドレスを使用した接続をサポートしていません。
- ・現在、Safari ブラウザはサポートされていません。

### 前提条件

- ・インスタンスに Instance Connect をインストールします。

詳細については、「[EC2 Instance Connect のセットアップ \(p. 512\)](#)」を参照してください。

- ・(オプション) ローカルコンピュータに SSH クライアントをインストールします。

ユーザーがコンソールまたは EC2 Instance Connect CLI のみを使用してインスタンスに接続する場合、SSH クライアントをインストールする必要はありません。ほとんどの場合、ローカルコンピュータにはデフォルトで SSH クライアントがインストールされています。SSH クライアントがあるかどうかを確認するには、コマンドラインで ssh と入力します。使用するローカルコンピュータでこのコマンドが認識されない場合、SSH クライアントをインストールできます。Linux または macOS X に SSH クライアントをインストールする詳細については、「<http://www.openssh.com>」を参照してください。Windows 10 に SSH クライアントをインストールする詳細については、「[Windows の OpenSSH](#)」を参照してください。

- ・(オプション) ローカルコンピュータに EC2 Instance Connect CLI をインストールします。

コンソールや SSH クライアントのみを使用してインスタンスに接続する場合、EC2 Instance Connect CLI をインストールする必要はありません。詳細については、「[ステップ 3: \(オプション\) EC2 Instance Connect CLI をインストールする \(p. 515\)](#)」を参照してください。

## ブラウザベースのクライアントを使用した接続

Amazon EC2 コンソールからインスタンスを選択し、EC2 Instance Connect を使用して接続することを選択することにより、ブラウザベースのクライアントを使用してインスタンスに接続できます。Instance Connect からアクセス許可が付与され、正常に接続されます。

Amazon EC2 コンソールからブラウザベースのクライアントを使用してインスタンスに接続するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[接続] を選択します。
4. [EC2 Instance Connect (ブラウザベースの SSH 接続)]、[接続] の順に選択します。

ウィンドウが開き、インスタンスに接続されます。

## EC2 Instance Connect CLI を使用して接続する

EC2 Instance Connect CLI を使用してインスタンスに接続するには、インスタンス ID のみを指定します。Instance Connect CLI によって次の 3 つのアクションが 1 つの呼び出しで実行されます：1 回限り使用の SSH パブリックキーが生成されます。このキーがインスタンスにプッシュされて 60 秒間保持されます。ユーザーがインスタンスに接続されます。Instance Connect CLI では基本的な SSH/SFTP コマンドを使用できます。

Amazon Linux 2

EC2 Instance Connect CLI を使用してインスタンスに接続するには

次のように、mssh コマンドをインスタンス ID と共に使用します。AMI のユーザー名を指定する必要はありません。

```
$ mssh i-001234a4bf70dec41EXAMPLE
```

Ubuntu

EC2 Instance Connect CLI を使用してインスタンスに接続するには

次のように、mssh コマンドを Ubuntu AMI のインスタンス ID とデフォルトのユーザー名と共に使用します。AMI のユーザー名を指定する必要があります。指定しない場合、Authentication failed というエラーが表示されます。

```
$ mssh ubuntu@i-001234a4bf70dec41EXAMPLE
```

## 独自のキーと SSH クライアントを使用して接続する

EC2 Instance Connect API の使用中に、独自の SSH キーを使用して、選択した SSH クライアントからインスタンスに接続できます。これにより、インスタンスにパブリックキーをプッシュする Instance Connect 機能を活用できます。

要件

サポートされる RSA キータイプは、OpenSSH および SSH2 です。サポートされている長さは 2048 および 4096 です。詳細については、「[独自のパブリックキーを Amazon EC2 にインポートする \(p. 902\)](#)」を参照してください。

独自のキーと任意の SSH クライアントを使用してインスタンスに接続するには

1. (オプション) 新しい SSH プライベートキーおよびパブリックキーを生成します。

新しい SSH プライベートキーおよびパブリックキー (`my_rsa_key` および `my_rsa_key.pub`) は、次のコマンドを使用して生成できます。

```
$ ssh-keygen -t rsa -f my_rsa_key
```

2. SSH パブリックキーをインスタンスにプッシュします。

`send-ssh-public-key` コマンドを使用して、SSH パブリックキーをインスタンスにプッシュします。Amazon Linux 2 を使用してインスタンスを起動した場合、AMI のデフォルトのユーザー名は `ec2-user` です。Ubuntu を使用してインスタンスを起動した場合、AMI のデフォルトのユーザー名は `ubuntu` です。

以下に、`ec2-user` を認証するために、指定されたアベイラビリティゾーンで指定されたインスタンスにパブリックキーをプッシュする例を示しています。

```
$ aws ec2-instance-connect send-ssh-public-key --instance-id i-001234a4bf70dec41EXAMPLE  
--availability-zone us-west-2b --instance-os-user ec2-user --ssh-public-key  
file:///my_rsa_key.pub
```

3. プライベートキーを使用してインスタンスに接続します。

パブリックキーがインスタンスマタデータから削除される前に(削除されるまでの時間は 60 秒です)、プライベートキーを使用してインスタンスに接続するには、ssh コマンドを使用します。パブリックキーに対応するプライベートキー、インスタンスを起動するために使用した AMI のデフォルトのユーザー名、およびインスタンスのパブリック DNS を指定します。

```
$ ssh -i my_rsa_key ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

## EC2 Instance Connect のアンインストール

EC2 Instance Connect を無効にするには、インスタンスに接続し、OS にインストールした `ec2-instance-connect` パッケージをアンインストールします。`sshd` 設定が EC2 Instance Connect をインストールしたときのまま変更されていない場合、`ec2-instance-connect` をアンインストールすると、`sshd` 設定も削除されます。`sshd` 設定が EC2 Instance Connect のインストール後に変更されている場合は、それを手動で更新する必要があります。

Amazon Linux 2

EC2 Instance Connect が事前設定されている Amazon Linux 2 2.0.20190618 以降では EC2 Instance Connect をアンインストールできます。

Amazon Linux 2 で起動したインスタンスの EC2 Instance Connect をアンインストールするには

1. SSH を使用してインスタンスに接続します。インスタンスの起動時に使用した SSH キーペアと Amazon Linux 2 AMI のデフォルトのユーザー名 (`ec2-user`) を指定します。

たとえば、次の ssh コマンドは、キーペア `my_ec2_private_key.pem` を使用して、パブリック DNS 名 `ec2-a-b-c-d.us-west-2.compute.amazonaws.com` でインスタンスに接続します。

```
$ ssh -i my_ec2_private_key.pem ec2-user@ec2-a-b-c-d.us-  
west-2.compute.amazonaws.com
```

2. yum コマンドを使用して ec2-instance-connect パッケージをアンインストールします。

```
[ec2-user ~]$ sudo yum remove ec2-instance-connect
```

#### Ubuntu

Ubuntu AMI で起動したインスタンスの EC2 Instance Connect をアンインストールするには

1. SSH を使用してインスタンスに接続します。インスタンスの起動時に使用した SSH キーペアと、Ubuntu AMI のデフォルトのユーザー名 (ubuntu) を指定します。

たとえば、次の ssh コマンドは、キーペア my\_ec2\_private\_key.pem を使用して、パブリック DNS 名 ec2-a-b-c-d.us-west-2.compute.amazonaws.com でインスタンスに接続します。

```
$ ssh -i my_ec2_private_key.pem ubuntu@ec2-a-b-c-d.us-west-2.compute.amazonaws.com
```

2. apt-get コマンドを使用して ec2-instance-connect パッケージをアンインストールします。

```
ubuntu:~$ sudo apt-get remove ec2-instance-connect
```

## PuTTY を使用した Windows から Linux インスタンスへの接続

インスタンスを起動したら、これに接続し、普通のコンピュータと同じように使用できます。

次の手順では、Windows 用の無料の SSH クライアントである PuTTY を使用して、インスタンスに接続する方法について説明します。インスタンスの接続でエラーが発生した場合は、「[Troubleshooting Connecting to Your Instance](#)」を参照してください。

### 前提条件

PuTTY を使用して Linux インスタンスに接続する前に、以下の前提条件を満たしていることを確認してください。

#### インスタンスの準備ができていることを確認する

インスタンスを起動してから接続できるようになるまでには、数分かかる場合があります。インスタンスのステータスチェックが成功していることを確認します。この情報は、[Instances (インスタンス)] ページの [Status Checks (ステータスチェック)] 列で確認できます。

#### インスタンスに接続するための一般的な前提条件を確認する

詳細については、「[インスタンスに接続するための一般的な前提条件 \(p. 506\)](#)」を参照してください。

#### ローカルコンピューターに PuTTY をインストールする

[PuTTY のダウンロードページ](#)から、PuTTY をダウンロードしてインストールします。すでにインストールされている旧バージョンの PuTTY がある場合は、最新バージョンをダウンロードすることをお勧めします。必ずスイート全体をインストールします。

#### PuTTYgen を使用してプライベートキーを変換する

インスタンスの起動時に指定したキーペアのプライベートキー (.pem ファイル) を見つけます。PuTTY で使用するために、.pem ファイルを .ppk ファイルに変換します。詳細については、次のセクションのステップに従います。

## PuTTYgen を使用したプライベートキーの変換

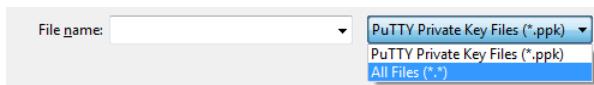
PuTTY は、SSH キーのプライベートキー形式をネイティブにサポートしていません。PuTTY は、キーを PuTTY に必要な形式に変換する PuTTYgen というツールを提供します。PuTTY を使用してインスタンスに接続するには、プライベートキー（.pem ファイル）を次の形式（.ppk ファイル）に変換する必要があります。

### プライベートキーを変換するには

- [スタート] メニューで、[すべてのプログラム]、[PuTTY]、[PuTTYgen] の順に選択します。
- [Type of key to generate (生成するキーのタイプ)] で、[RSA] を選択します。旧バージョンの PuTTYgen を使用している場合は、[SSH-2 RSA] を選択します。



- [Load (ロード)] を選択します。デフォルトでは、PuTTYgen には拡張子 .ppk を持つファイルだけが表示されます。.pem ファイルの場所を特定するには、すべてのタイプのファイルを表示するオプションを選択します。



- インスタンスを起動したときに指定したキーペアの .pem ファイルを選択し、[Open (開く)] を選択します。PuTTYgen によって、.pem ファイルが正常にインポートされたことが通知されます。[OK] を選択します。
- PuTTY が使用できる形式でキーを保存するには、[Save private key (プライベートキーの保存)] を選択します。PuTTYgen に、パスフレーズなしでキーを保存することに関する警告が表示されます。[Yes (はい)] を選択します。

### Note

プライベートキーのパスフレーズは追加の保護レイヤーです。プライベートキーが検出されても、パスフレーズがなければ使用できません。パスフレーズを使用することの欠点は、インスタンスにログオンしたり、ファイルをインスタンスにコピーしたりするのに人間の介入が必要となるため、オートメーションが難しくなることです。

- キーペアに使用した名前と同じ名前（my-key-pair など）をキーに指定します。[Save (保存)] を選択すると、PuTTY によって .ppk ファイル拡張子が自動的に追加されます。

プライベートキーが PuTTY で使用するための正しい形式となりました。これで、PuTTY の SSH クライアントを使用してインスタンスに接続することができます。

## Linux インスタンスへの接続

PuTTY を使用して Linux インスタンスに接続するには、次の手順に従います。秘密キーに作成した .ppk ファイルが必要になります。詳細については、前のセクションの「[PuTTYgen を使用したプライベートキーの変換 \(p. 521\)](#)」を参照してください。インスタンスの接続でエラーが発生した場合は、「[インスタンスへの接続に関するトラブルシューティング](#)」を参照してください。

### PuTTY を使用してインスタンスに接続するには

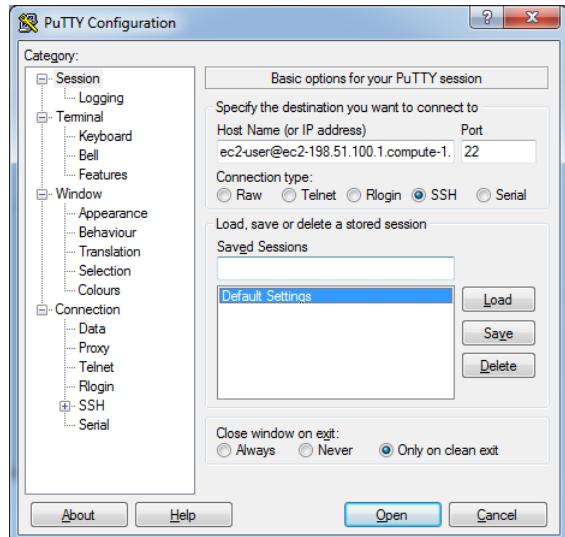
- PuTTY を開始します ([スタート] メニューで [All Programs (すべてのプログラム)、PuTTY、PuTTY] を選択)。
- [Category (カテゴリ)] ペインで [Session (セッション)] を選択し、次のフィールドに入力します。
  - [Host Name (ホスト名)] ボックスで、次のいずれかの操作を行います。

- ・ (パブリック DNS) インスタンスのパブリック DNS を使用して接続するには、「`user_name@public_dns_name`」と入力します。
- ・ (IPv6) インスタンスに IPv6 アドレスがある場合、インスタンスの IPv6 アドレスを使用して接続するには、「`user_name@ipv6_address`」と入力します。

インスタンスのパブリック DNS 名または IPv6 アドレスを取得する方法については、「[インスタンスに関する情報を取得する \(p. 506\)](#)」を参照してください。

`user_name` で、AMI の適切なユーザー名を指定してください。以下に例を示します。

- ・ Amazon Linux 2 または Amazon Linux AMI の場合は、ユーザー名は `ec2-user` です。
  - ・ CentOS AMI の場合、ユーザー名は `centos` です。
  - ・ Debian AMI の場合は、ユーザー名は `admin` または `root` です。
  - ・ Fedora AMI の場合、ユーザー名は `ec2-user` または `fedora` です。
  - ・ RHEL AMI の場合は、ユーザー名は `ec2-user` または `root` のどちらかです。
  - ・ SUSE AMI の場合は、ユーザー名は `ec2-user` または `root` のどちらかです。
  - ・ Ubuntu AMI の場合は、ユーザー名は `ubuntu` です。
  - ・ それ以外の場合で、`ec2-user` および `root` が機能しない場合は、AMI プロバイダーに確認してください。
- a. [Port (ポート)] の値が 22 であることを確認します。
  - b. [Connection type (接続タイプ)] で [SSH] を選択します。



- a. [Browse (参照)] を選択します。
  - b. キーペア用に生成した .ppk ファイルを選択し、[Open (開く)] を選択します。
3. (オプション) セッションをアクティブに保つため、定期的に「キープアライブ」データを自動的に送信するように PuTTY を設定できます。これは、セッションがアイドル状態になった際にインスタンスから切断されないようにするのに便利です。[Category] ペインで [Connection] を選択し、[Seconds between keepalives] フィールドで必要な間隔を入力します。たとえば、10 分間アイドル状態が続いた後にセッションが切断される場合、180 と入力して PuTTY を設定し、キープアライブデータを 3 分ごとに送信するようにします。
  4. [Category] ペインで、[Connection]、[SSH] の順に展開し、[Auth] を選択します。次のように入力します。

- c. (オプション) 後でこのセッションを再度開始する場合、後で使用できるようにセッション情報を保存できます。[Category (カテゴリ)] で [Session (セッション)] を選択し、[Saved Sessions (保存されたセッション)] にセッションの名前を入力して、[Save (保存)] を選択します。
  - d. [Open (開く)] を選択します。
5. このインスタンスに接続するのが初めての場合、PuTTY は接続先のホストを信頼するかどうかを確認するセキュリティアラートダイアログボックスを表示します。
- a. (オプション) セキュリティアラートダイアログボックスのフィンガープリントが、(オプション) インスタンスのフィンガープリントを取得する (p. 507) で前に取得したフィンガープリントと一致することを確認します。これらのフィンガープリントが一致しない場合、「中間者 (MITM)」攻撃を受けている可能性があります。一致した場合は、次の手順に進んでください。
  - b. [Yes (はい)] を選択します。ウィンドウが開き、インスタンスに接続した状態になります。

#### Note

プライベートキーを PuTTY フォーマットに変換するときにパスフレーズを指定した場合は、インスタンスへのログイン時にそのパスフレーズを指定する必要があります。

インスタンスの接続でエラーが発生した場合は、「[インスタンスへの接続に関するトラブルシューティング](#)」を参照してください。

## PuTTY Secure Copy Client を使用した Linux インスタンスへのファイルの転送

PuTTY Secure Copy Client (PSCP) は、Windows コンピュータと Linux インスタンスの間でファイルを転送するために使用できるコマンドラインツールです。グラフィカルユーザーインターフェイス (GUI) を使用する場合は、WinSCP という名前のオープンソース GUI ツールを使用できます。詳細については、「[WinSCP を使用した Linux インスタンスへのファイルの転送 \(p. 523\)](#)」を参照してください。

PSCP を使用するには、「[PuTTYgen を使用したプライベートキーの変換 \(p. 521\)](#)」で生成したプライベートキーが必要です。また、Linux インスタンスのパブリック DNS アドレスも必要です。

次の例では、ファイル Sample\_file.txt を Windows コンピュータの C: ドライブから Amazon Linux インスタンス上の ec2-user ホームディレクトリに転送します。

```
pscp -i C:\\path\\my-key-pair.ppk C:\\path\\Sample_file.txt ec2-user@public_dns:/home/ec2-user/Sample_file.txt
```

(IPv6 のみ) 以下の例では、インスタンスの IPv6 アドレスを使用して Sample\_file.txt ファイルを転送します。IPv6 アドレスは角かっこ ([]) で囲む必要があります。

```
pscp -i C:\\path\\my-key-pair.ppk C:\\path\\Sample_file.txt ec2-user@[ipv6-address]:/home/ec2-user/Sample_file.txt
```

## WinSCP を使用した Linux インスタンスへのファイルの転送

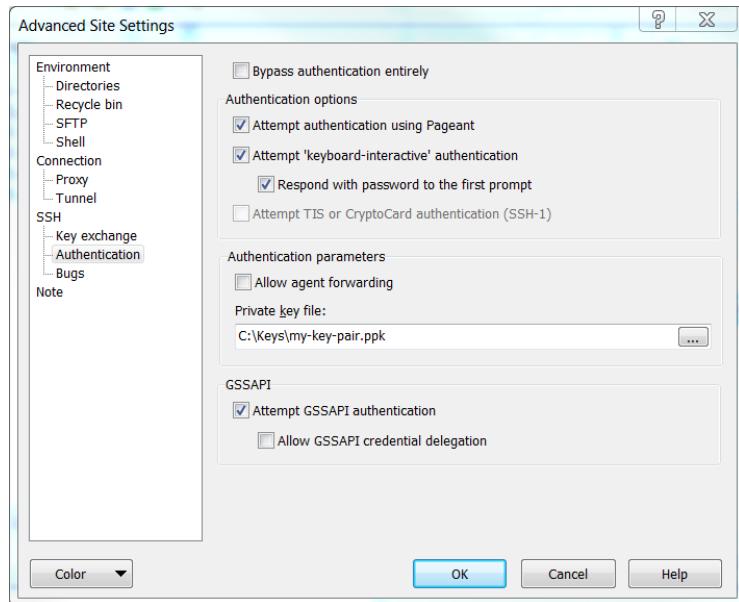
WinSCP は Windows 用の GUI ベースのファイルマネージャで、SFTP、SCP、FTP、および FTPS プロトコルを使って、ファイルをリモートコンピュータにアップロードおよび転送することができます。WinSCP を使用すると、Windows マシンから Linux インスタンスにファイルをドラッグアンドドロップしたり、2 つのシステム間でディレクトリ構造全体を同期させることができます。

WinSCP を使用するには、「[PuTTYgen を使用したプライベートキーの変換 \(p. 521\)](#)」で生成したプライベートキーが必要です。また、Linux インスタンスのパブリック DNS アドレスも必要です。

1. <http://winscp.net/eng/download.php> から WinSCP をダウンロードしてインストールします。ほとんどの場合、デフォルトのインストールオプションでかまいません。

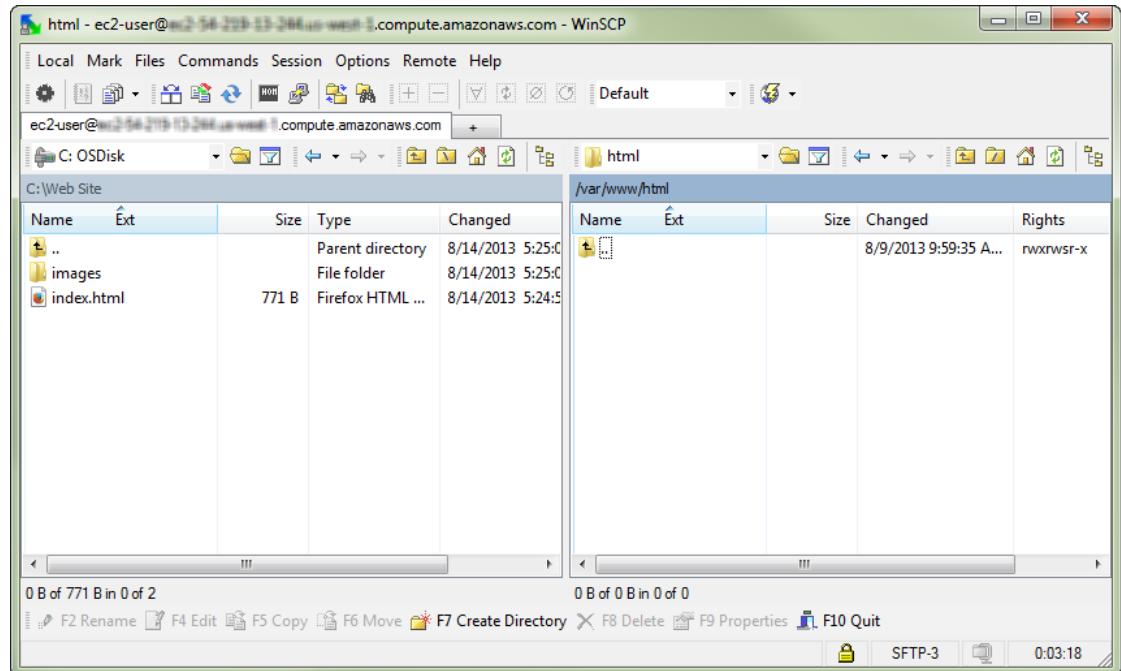
2. WinSCP を起動します。
3. [WinSCP login] 画面で、インスタンスのパブリック DNS ホスト名またはパブリック IPv4 アドレスを [Host name] に入力します。  
(IPv6 のみ) インスタンスの IPv6 アドレスを使用してログインするには、インスタンスの IPv6 アドレスを入力します。
4. [User name] については、AMI のデフォルトユーザー名を入力します。
  - Amazon Linux 2 または Amazon Linux AMI の場合は、ユーザー名は ec2-user です。
  - CentOS AMI の場合、ユーザー名は centos です。
  - Debian AMI の場合は、ユーザー名は admin または root です。
  - Fedora AMI の場合、ユーザー名は ec2-user または fedora です。
  - RHEL AMI の場合は、ユーザー名は ec2-user または root のどちらかです。
  - SUSE AMI の場合は、ユーザー名は ec2-user または root のどちらかです。
  - Ubuntu AMI の場合は、ユーザー名は ubuntu です。
  - それ以外の場合で、ec2-user および root が機能しない場合は、AMI プロバイダーに確認してください。
5. インスタンスのプライベートキーを指定します。[Private key (プライベートキー)] に、プライベートキーへのパスを入力するか、 [...] ボタンを選択して、ファイルを参照します。高度なサイトの設定を開くには、より新しいバージョンの WinSCP で、[設定] を選択します。プライベートキーファイル設定を見つけるには、[SSH] の [認証] を選択します。

次に示すのは、WinSCP バージョン 5.9.4 からのスクリーンショットです。



WinSCP は PuTTY プライベートキーファイル (.ppk) ファイルを必要とします。PuTTYgen を使用して、.pem セキュリティキーファイルを .ppk フォーマットに変換することができます。詳細については、「[PuTTYgen を使用したプライベートキーの変換 \(p. 521\)](#)」を参照してください。

6. (オプション) 左のパネルで、[ディレクトリ] を選択します。[リモートディレクトリ] に、ファイルを追加する先のディレクトリのパスを入力します。より新しいバージョンの WinSCP で高度なサイトの設定を開くには、[設定] を選択します。リモートディレクトリ設定を見つけるには、[環境] の [ディレクトリ] を選択します。
7. [ログイン] を選択します。ホストのフィンガープリントをホストのキャッシュに追加するには、[はい] を選択します。



- 接続確立後、接続ウィンドウには Linux インスタンスが右側、ローカルマシンが左側に表示されます。ローカルマシンからリモートファイルシステムへ、ファイルを直接ドラッグアンドドロップすることができます。WinSCP の詳細については、<http://winscp.net/eng/docs/start> のドキュメントを参照してください。

[Cannot execute SCP to start transfer] エラーを受け取った場合は、まず Linux インスタンスに scp をインストールする必要があります。一部のオペレーティングシステムでは、これは openssh-clients パッケージに含まれます。Amazon Linux-optimized Amazon ECS などの AMI バリエントでは、以下のコマンドを使用して scp をインストールします。

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

## Windows Subsystem for Linux を使用した Windows から Linux インスタンスへの接続

インスタンスを起動したら、これに接続し、普通のコンピュータと同じように使用できます。

次の手順では、Windows Subsystem for Linux (WSL) で Linux ディストリビューションを使用してインスタンスに接続する方法について説明します。WSL は無料でダウンロードでき、Windows でネイティブ Linux コマンドラインツールを直接実行できます。それと同時に、仮想マシンのオーバーヘッドなしに、従来の Windows デスクトップも実行できます。

WSL をインストールすると、PuTTY または PuTTYgen を使用する代わりに、ネイティブ Linux 環境を使用して Linux EC2 インスタンスに接続できます。Linux 環境では、Linux インスタンスにより簡単に接続できます。これは、Linux インスタンスに接続し、.pem キーファイルのアクセス権限を変更するために使用できるネイティブ SSH クライアントが付属しているためです。Amazon EC2 コンソールは、Linux インスタンスに接続するための SSH コマンドを提供します。この SSH コマンドから、トラブルシューティングのために詳細な出力を取得できます。詳細については、[Windows Subsystem for Linux](#) に関する情報を参照してください。

### Note

WSL をインストールした後のすべての必須条件とステップは、「[SSH を使用した Linux インスタンスへの接続 \(p. 508\)](#)」で説明しているものと同じです。また、そのエクスペリエンスはネイティブ Linux の使用と同様です。

インスタンスの接続でエラーが発生した場合は、「[インスタンスへの接続に関するトラブルシューティング](#)」を参照してください。

### 目次

- [前提条件 \(p. 508\)](#)
- [WSL を使用して Linux インスタンスに接続します。 \(p. 526\)](#)
- [SCP を使用した Linux から Linux インスタンスへのファイルの転送 \(p. 527\)](#)
- [WSL のアンインストール \(p. 529\)](#)

## 前提条件

Linux インスタンスに接続する前に、以下の前提条件を満たしていることを確認してください。

インスタンスの準備ができていることを確認する

インスタンスを起動してから接続できるようになるまでには、数分かかる場合があります。インスタンスのステータスチェックが成功していることを確認します。この情報は、[Instances (インスタンス)] ページの [Status Checks (ステータスチェック)] 列で確認できます。

インスタンスに接続するための一般的な前提条件を確認する

詳細については、「[インスタンスに接続するための一般的な前提条件 \(p. 506\)](#)」を参照してください。

ローカルコンピューターに Windows Subsystem for Linux (WSL) と Linux ディストリビューションをインストールする

[Windows 10 インストールガイド](#)の手順を使用して、WSL と Linux ディストリビューションをインストールします。手順の例では、Linux の Ubuntu ディストリビューションをインストールしますが、任意のディストリビューションをインストールできます。コンピュータを再起動して変更を有効にすることが求められます。

プライベートキーを Windows から WSL にコピーする

WSL ターミナルウィンドウで、Windows から WSL に .pem ファイル (インスタンスの起動時に指定したキーペアの場合) をコピーします。インスタンスに接続する際に使用する、WSL の .pem ファイルへの完全修飾パスをメモします。Windows ハードドライブへのパスを指定する方法の詳細については、「[C ドライブにアクセスする方法](#)」を参照してください。

```
cp /mnt/<Windows drive letter>/path/my-key-pair.pem ~/WSL-path/my-key-pair.pem
```

## WSL を使用して Linux インスタンスに接続します。

Windows Subsystem for Linux (WSL) を使用して Linux インスタンスに接続するには、次の手順に従います。インスタンスの接続でエラーが発生した場合は、「[インスタンスへの接続に関するトラブルシューティング](#)」を参照してください。

SSH を使用してインスタンスに接続するには

1. ターミナルウィンドウで ssh コマンドを使用して、インスタンスに接続します。インスタンスのプライベートキー (.pem) ファイル、AMI のユーザー名、およびパブリック DNS 名を指定します。たとえ

ば、Amazon Linux 2 または Amazon Linux AMI を使用した場合、ユーザー名は `ec2-user` です。AMI のユーザー名およびインスタンスの DNS 名を見つける詳しい方法については、「[インスタンスに関する情報を取得する \(p. 506\)](#)」を参照してください。

```
sudo ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

以下のようなレスポンスが表示されます。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established.  
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

2. (IPv6 のみ) 別の方法として、IPv6 アドレスを使用してインスタンスに接続することもできます。ssh コマンドで、プライベートキー (.pem) ファイルへのパス、適切なユーザー名、および IPv6 アドレスを指定します。たとえば、Amazon Linux 2 または Amazon Linux AMI を使用した場合、ユーザー名は `ec2-user` です。

```
sudo ssh -i /path/my-key-pair.pem ec2-user@2001:db8:1234:1a00:9691:9503:25ad:1761
```

3. (オプション) セキュリティアラートのフィンガープリントが、(オプション) インスタンスのフィンガープリントを取得する (p. 507) で事前に取得したフィンガープリントと一致することを確認します。これらのフィンガープリントが一致しない場合、「中間者 (MITM)」攻撃を受けている可能性があります。一致した場合は、次の手順に進んでください。
4. yes と入力します。

以下のようなレスポンスが表示されます。

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)  
to the list of known hosts.
```

## SCP を使用した Linux から Linux インスタンスへのファイルの転送

ローカルコンピュータと Linux インスタンスの間でファイルを転送する方法の 1 つとして、セキュアコピープロトコル (SCP) を使用します。このセクションでは、SCP でファイルを転送する方法について説明します。この手順は、SSH を使用してインスタンスに接続する手順と似ています。

### 前提条件

- インスタンスにファイルを転送するための一般的な前提条件の確認

インスタンスにファイルを転送するための一般的な前提条件は、インスタンスに接続するための一般的な前提条件と同様です。詳細については、「[インスタンスに接続するための一般的な前提条件 \(p. 506\)](#)」を参照してください。

- SCP クライアントのインストール

ほとんどの Linux、Unix、および Apple コンピュータには、デフォルトで SCP クライアントが含まれています。含まれていない場合は、OpenSSH プロジェクトから、SSH ツールの完全なスイートの無料実装が提供されており、これに SCP クライアントが含まれます。詳細については、<http://www.openssh.org> を参照してください。

SCP を使用してファイルを転送するステップを次に示します。既に SSH でインスタンスに接続し、フィンガープリントの確認が完了している場合は、SCP コマンドを実行するステップ (ステップ4) から開始できます。

### SCP を使用してファイルを転送するには

1. インスタンスのパブリック DNS 名を使って、インスタンスにファイルを転送します。たとえば、プライベートキーファイルの名前が `my-key-pair`、転送するファイルが `SampleFile.txt`、ユーザー名が `ec2-user`、インスタンスのパブリック DNS の名前が `ec2-198-51-100-1.compute-1.amazonaws.com` の場合、次のコマンドを使って、ファイルを `ec2-user` ホームディレクトリにコピーします。

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-
user@ec2-198-51-100-1.compute-1.amazonaws.com:~
```

以下のようなレスポンスが表示されます。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' 
can't be established.
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

2. (IPv6 のみ) 別の方法として、インスタンスの IPv6 アドレスを使用してファイルを転送することもできます。IPv6 アドレスは、\でエスケープした角かっこ ([]) で囲む必要があります。

```
scp -i /path/my-key-pair.pem /path/SampleFile.txt ec2-user@
\[2001:db8:1234:1a00:9691:9503:25ad:1761\]:~
```

3. (オプション) セキュリティアラートのフィンガープリントが、(オプション) インスタンスのフィンガープリントを取得する (p. 507) で事前に取得したフィンガープリントと一致することを確認します。これらのフィンガープリントが一致しない場合、「中間者 (MITM)」攻撃を受けている可能性があります。一致した場合は、次の手順に進んでください。
4. **yes** と入力します。

以下のようなレスポンスが表示されます。

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
Sending file modes: C0644 20 SampleFile.txt
Sink: C0644 20 SampleFile.txt
SampleFile.txt                                         100%    20      0.0KB/s   00:00
```

[bash: scp: command not found] エラーを受け取った場合は、まず Linux インスタンスに `scp` をインストールする必要があります。一部のオペレーティングシステムでは、これは `openssh-clients`/パッケージに含まれます。Amazon Linux-optimized Amazon ECS などの AMI バリアントでは、以下のコマンドを使用して `scp` をインストールします。

```
[ec2-user ~]$ sudo yum install -y openssh-clients
```

5. 逆の方向 (Amazon EC2 インスタンスからローカルコンピュータに) にファイルを転送する場合は、ホストパラメータの順番を逆にします。たとえば、`SampleFile.txt` ファイルを EC2 インスタンスからローカルコンピュータのホームディレクトリに `SampleFile2.txt` として転送するには、ローカルコンピュータで次のコマンドを実行します。

```
scp -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/
SampleFile.txt ~/SampleFile2.txt
```

6. (IPv6 のみ) 別の方法として、インスタンスの IPv6 アドレスを使用して別の方向にファイルを転送することもできます。

```
scp -i /path/my-key-pair.pem ec2-user@[2001:db8:1234:1a00:9691:9503:25ad:1761]:~/  
SampleFile.txt ~/SampleFile2.txt
```

## WSL のアンインストール

Windows Subsystem for Linux のアンインストールの詳細については、「[WSL ディストリビューションをアンインストールする方法](#)」について参照してください。

## Session Manager を使用した Linux インスタンスへの接続

Session Manager は AWS Systems Manager の完全マネージド型の機能であり、ブラウザベースのインターフェイスなワンクリックシェルまたは AWS CLI を介して Amazon EC2 インスタンスを管理できます。Session Manager を使用して、アカウント内のインスタンスとのセッションを開始できます。セッションの開始後、他の接続タイプと同様、bash コマンドを実行できます。Session Manager の詳細については、AWS Systems Manager ユーザーガイドの「[AWS Systems Manager Session Manager](#)」を参照してください。

Session Manager を使用してインスタンスに接続する前に、必要な設定手順が完了していることを確認してください。詳細と手順については、「[Session Manager の開始方法](#)」を参照してください。

Amazon EC2 コンソールで Session Manager を使用して Linux インスタンスに接続するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[接続] を選択します。
4. [Connection method (接続方法)] で、[Session Manager] を選択します。
5. [接続] を選択します。

### トラブルシューティング

1 つ以上の Systems Manager アクション (ssm:[command-name](#)) の実行を承認されていないというエラーが表示された場合は、Amazon EC2 コンソールからセッションを開始できるようにポリシーを更新する必要があります。詳細については、AWS Systems Manager ユーザーガイドの「[Session Manager のクイックスタートのデフォルト IAM ポリシー](#)」を参照してください。

## インスタンスの停止と起動

インスタンスにルートデバイスとして Amazon EBS ボリュームがある場合、そのインスタンスを停止して起動できます。インスタンスにはそのインスタンス ID が保持されますが、「[概要 \(p. 530\)](#)」セクションで述べられているように変更される可能性があります。

ユーザーがインスタンスを停止すると、インスタンスはシャットダウンされます。停止しているインスタンスの使用料またはデータ転送料は課金されませんが、Amazon EBS ボリュームのストレージに対しては課金されます。停止したインスタンスを起動するたびに、最低 1 分間分の使用料が課金されます。1 分経過した後は、使用した秒数のみ課金されます。たとえば、インスタンスを 20 秒間実行して停止した場合は、1 分間分課金されます。インスタンスを 3 分 40 秒実行した場合は、ちょうど 3 分 40 秒間分課金されます。

インスタンスが停止している間、他のボリュームと同様にそのルートボリュームを扱い、変更することができます（ファイルシステムの問題を修復したり、ソフトウェアを更新したりするなど）。停止しているインスタンスからボリュームを接続解除し、それを実行中のインスタンスに接続して、変更を行い、実行中のインスタンスから接続解除して、停止しているインスタンスに再接続します。インスタンスのロックデバイスマッピングにルートデバイスとして指定されたストレージデバイス名を使用して、ボリュームを接続解除していることを確認します。

インスタンスが必要なくなったら、終了することができます。インスタンスの状態が `shutting-down` または `terminated` に変わったら、そのインスタンスへの課金は停止します。詳細については、「[インスタンスの終了 \(p. 545\)](#)」を参照してください。インスタンスを休止する場合は、「[Linux インスタンスの休止 \(p. 532\)](#)」を参照してください。詳細については、「[再起動、停止、休止、終了の違い \(p. 447\)](#)」を参照してください。

## コンテンツ

- [概要 \(p. 530\)](#)
- [インスタンスを停止するとどうなるか \(API\) \(p. 531\)](#)
- [インスタンスの停止と起動 \(p. 531\)](#)
- [停止されているインスタンスの変更 \(p. 532\)](#)
- [トラブルシューティング \(p. 532\)](#)

## 概要

停止できるのは Amazon EBS-Backed インスタンスだけです。インスタンスのルートデバイスタイプを確認するには、インスタンスを記述し、そのルートボリュームのデバイスタイプが `ebs` (Amazon EBS-Backed インスタンス) か `instance store` (Instance store-Backed インスタンス) かをチェックします。詳細については、「[AMI のルートデバイスタイプの判別 \(p. 97\)](#)」を参照してください。

実行中のインスタンスを停止すると、次の処理が実行されます。

- インスタンスは正常なシャットダウンを実行し、実行を停止します (ステータスは `stopping`、次に `stopped` に変わります)。
- Amazon EBS ボリュームはインスタンスに接続されたままとなり、そのデータは保持されます。
- ホストコンピュータの RAM またはホストコンピュータのインスタンスストアボリュームに保存されたデータはなくなります。
- 略どの場合、インスタンスは基盤となる新しいホストコンピュータが起動したときに移行されます。
- インスタンスの停止および起動時に、インスタンスにはプライベート IPv4 アドレスと任意の IPv6 アドレスが保持されます。インスタンスの起動時に、パブリック IPv4 アドレスを解放し、新しいアドレスを割り当てます。
- インスタンスには関連付けられた Elastic IP アドレスが保持されます。停止されているインスタンスに関連付けられた Elastic IP アドレスに対して課金されます。EC2-Classic を利用した場合、インスタンスを停止すると、Elastic IP アドレスとインスタンスの関連付けが解除されます。詳細については、「[EC2-Classic \(p. 804\)](#)」を参照してください。
- Windows インスタンスを停止して起動すると、アタッチされた Amazon EBS ボリュームのドライブ文字の変更などのタスクが EC2Config サービスによりインスタンスで実行されます。これらのデフォルトおよび変更方法については、「[EC2Config サービスを使用した Windows インスタンスの設定](#)」(Windows インスタンスの Amazon EC2 ユーザーガイド) を参照してください。
- インスタンスが Auto Scaling グループにある場合、Amazon EC2 Auto Scaling サービスはインスタンスを異常と判断して停止し、場合によってはそれを終了して代わりのインスタンスを起動します。詳細については、「[Amazon EC2 Auto Scaling ユーザーガイド](#)」の「[Auto Scaling インスタンスのヘルスチェック](#)」を参照してください。
- ClassicLink インスタンスを停止すると、今までリンクされていた VPC とのリンクが解除されます。インスタンスを起動した後に再び VPC にリンクする必要があります。ClassicLink の詳細については、「[ClassicLink \(p. 812\)](#)」を参照してください。

詳細については、「[再起動、停止、休止、終了の違い \(p. 447\)](#)」を参照してください。

以下のインスタンスの属性は停止されると、変更できます。

- インスタンスタイプ
- ユーザーデータ

- Kernel
- RAM ディスク

インスタンスの実行中にこれらの属性を変更しようとすると、Amazon EC2 が `IncorrectInstanceState` エラーを返します。

## インスタンスを停止するとどうなるか (API)

`stop-instances` コマンドを使用して EC2 インスタンスを停止すると、OS レベルで以下が登録されます。

- API リクエストは、ボタンのクリックイベントをゲストに送信します。
- ボタンのクリックイベントの結果、さまざまなシステムサービスが停止されます。systemd はシステムの適切なシャットダウンを処理します。適切なシャットダウンは、ハイパーテーバイザーからの ACPI シャットダウンボタンのクリックイベントによってトリガーされます。
- ACPI のシャットダウンが開始されます。
- このインスタンスは、適切なシャットダウンプロセスが終了したときにシャットダウンされます。設定可能な OS シャットダウン時間はありません。

## インスタンスの停止と起動

コンソールまたはコマンドラインを使用して、Amazon EBS-Backed インスタンスを起動および停止できます。

デフォルトでは、Amazon EBS-backed インスタンスからシャットダウンを開始すると (`shutdown` または `poweroff` コマンドを使用)、インスタンスが停止します。この動作を変更して、インスタンスの停止ではなく終了させることができます。詳細については、「[インスタンスによって起動されたシャットダウン動作の変更 \(p. 548\)](#)」を参照してください。

コンソールを使用して Amazon EBS-Backed インスタンスを停止および起動するには

1. ナビゲーションペインで `[Instances]` を選択し、インスタンスを選択します。
2. `[Actions]`、`[Instance State]`、`[Stop]` の順に選択します。`[Stop]` が無効になっている場合は、インスタンスが既に停止しているか、またはルートボリュームがインスタンスストアボリュームです。

### Warning

インスタンスを停止すると、インスタンスストアボリューム上のデータは消去されます。インスタンスストアボリュームのデータを保持するには、このデータを永続的ストレージに必ずバックアップしてください。

3. 確認ダイアログボックスで `[Yes, Stop]` を選択します。インスタンスが停止するまで、数分かかる場合があります。
4. インスタンスが停止されている間、特定のインスタンス属性を変更できます。詳細については、「[停止されているインスタンスの変更 \(p. 532\)](#)」を参照してください。
5. 停止されているインスタンスを再起動するには、インスタンスを選択後、`[Actions (アクション)]`、`[インスタンスの状態]`、`[Start (起動)]` の順に選択します。
6. 確認ダイアログボックスで `[Yes, Start]` を選択します。インスタンスが `running` 状態になるまで、数分かかる場合があります。

コマンドラインを使用して Amazon EBS-Backed インスタンスを停止および起動するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [stop-instances](#) および [start-instances](#) (AWS CLI)
- [Stop-EC2Instance](#) および [Start-EC2Instance](#) (AWS Tools for Windows PowerShell)

## 停止されているインスタンスの変更

AWS マネジメントコンソールまたはコマンドラインインターフェイスを使用して、停止されているインスタンスのインスタンスタイプ、ユーザーデータ、および EBS 最適化属性を変更できます。AWS マネジメントコンソールを使用して、DeleteOnTermination、カーネル、または RAM ディスクの属性を変更することはできません。

インスタンス属性を変更するには

- インスタンスタイプを変更するには、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。
- インスタンスのユーザーデータを変更するには、「[インスタンスユーザーデータの使用 \(p. 607\)](#)」を参照してください。
- インスタンスの EBS 最適化を有効または無効にするには、「[EBS 最適化の変更 \(p. 1043\)](#)」を参照してください。
- インスタンスのルートボリュームの DeleteOnTermination 属性を変更するには、「[実行中のインスタンスのブロックデバイスマッピングの更新 \(p. 1107\)](#)」を参照してください。

コマンドラインを使用してインスタンス属性を変更するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

## トラブルシューティング

Amazon EBS-Backed インスタンスを停止し、`stopping` 状態に "stuck" が表示されている場合、インスタンスを強制終了できます。詳細については、「[インスタンスの停止に関するトラブルシューティング \(p. 1143\)](#)」を参照してください。

## Linux インスタンスの休止

インスタンスを休止すると、オペレーティングシステムに対して休止の実行 (suspend-to-disk) が指示されます。休止に伴って、インスタンスマモリ (RAM) の内容が Amazon EBS ルートボリュームに保存されます。インスタンスの Amazon EBS ルートボリュームとアタッチされた Amazon EBS データボリュームは保持されます。インスタンスを再起動すると、以下のようになります。

- Amazon EBS ルートボリュームが前の状態に復元されます。
- RAM の内容が再ロードされます。
- インスタンスで以前に実行されていたプロセスが再開されます。
- 以前にアタッチされていたデータボリュームが再アタッチされ、インスタンスがそのインスタンス ID を保持します。

インスタンスは、[休止が有効になっており \(p. 538\)](#)、[休止の前提条件 \(p. 534\)](#)を満たしている場合のみ、休止状態にすることができます。

インスタンスまたはアプリケーションが、ブートストラップし、メモリフトプリントを構築して完全に生産性を発揮するのに時間がかかる場合は、休止を使用してインスタンスを事前ウォーミングできます。インスタンスを事前ウォーミングするには、次の操作を行います。

1. 休止を有効にしてインスタンスを起動します。
2. インスタンスを必要な状態に移行させます。
3. インスタンスを休止し、必要に応じて同じ状態に回復されるようにします。

インスタンスが `stopped` 状態にあるときは、休止されているインスタンスの使用料は課金しません。RAM の内容が Amazon EBS ルートボリュームに転送されて、インスタンスが `stopping` 状態にある間は、インスタンスの使用量に対して課金されます。(この点は、休止せずに [インスタンスを停止 \(p. 529\)](#) した場合と異なります。) 使用量はデータ転送料の課金対象外です。ただし、Amazon EBS ボリュームのストレージは、RAM の内容のストレージも含めて、課金対象になります。

インスタンスが必要なくなった場合、`stopped` (休止) 状態にある場合を含め、いつでも終了することができます。詳細については、「[インスタンスの終了 \(p. 545\)](#)」を参照してください。

Note

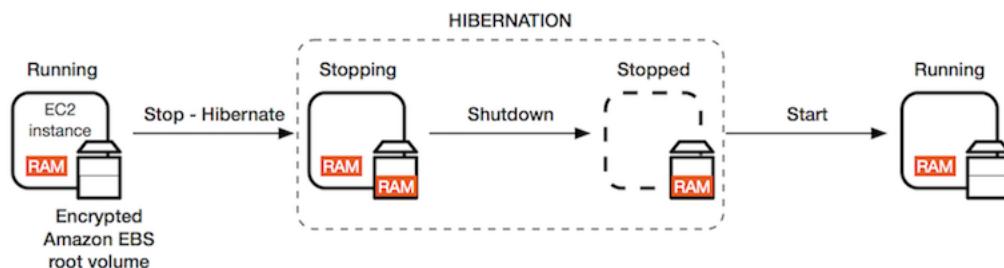
Windows インスタンスでの休止の使用については、「[Windows インスタンスの休止](#)」( Windows インスタンスの Amazon EC2 ユーザーガイド) を参照してください。

コンテンツ

- [休止の概要 \(p. 533\)](#)
- [休止の前提条件 \(p. 534\)](#)
- [制約事項 \(p. 535\)](#)
- [休止をサポートするように既存の AMI を設定する \(p. 535\)](#)
- [インスタンスの休止の有効化 \(p. 538\)](#)
- [インスタンスでの KASLR の無効化 \(Ubuntu のみ\) \(p. 539\)](#)
- [インスタンスを休止する \(p. 540\)](#)
- [休止したインスタンスの起動 \(p. 541\)](#)
- [休止のトラブルシューティング \(p. 542\)](#)

## 休止の概要

次の図は、休止処理の基本的な概要を示しています。



実行中のインスタンスを休止すると、次の処理が実行されます。

- 休止プロセスを開始すると、インスタンスは `stopping` 状態に移行します。オペレーティングシステムに対して休止の実行 (suspend-to-disk) が指示されます。休止に伴ってすべてのプロセスがフリーズされ、RAM の内容が Amazon EBS ルートボリュームに保存されます。その後に、通常のシャットダウンが実行されます。
- シャットダウンプロセスが完了した後、インスタンスは `stopped` 状態に移行します。

- Amazon EBS ボリュームはインスタンスに接続されたままとなり、保存された RAM の内容を含め、データは保持されます。
- 殆どの場合、インスタンスは基盤となる新しいホストコンピュータが起動したときに移行されます。これは、インスタンスを停止して起動した場合と同じです。
- インスタンスを起動すると、インスタンスのブートアッププロセスが実行され、オペレーティングシステムが Amazon EBS ルートボリュームから RAM の内容を読み取ります。次に、プロセスのフリーズが解除されて以前の状態が回復されます。
- インスタンスの休止および起動時に、インスタンスにはプライベート IPv4 アドレスと任意の IPv6 アドレスが保持されます。インスタンスの起動時に、パブリック IPv4 アドレスを解放し、新しいアドレスを割り当てます。
- インスタンスには関連付けられた Elastic IP アドレスが保持されます。休止されているインスタンスに関連付けられた Elastic IP アドレスに対して課金されます。EC2-Classic を利用した場合、インスタンスを休止すると、Elastic IP アドレスとインスタンスの関連付けが解除されます。詳細については、「[EC2-Classic \(p. 804\)](#)」を参照してください。
- ClassicLink インスタンスを休止すると、今までリンクされていた VPC とのリンクが解除されます。インスタンスを起動した後に再び VPC にリンクする必要があります。詳細については、「[ClassicLink \(p. 812\)](#)」を参照してください。

休止と再起動、停止、および終了の違いについては、「[再起動、停止、休止、終了の違い \(p. 447\)](#)」を参照してください。

## 休止の前提条件

インスタンスを休止するには、以下の前提条件を設定する必要があります。

- サポートされているインスタンスファミリー - C3、C4、C5、M3、M4、M5、R3、R4、および R5
- インスタンス RAM サイズ - 150 GB 未満である必要があります。
- インスタンスサイズ - ベアメタルインスタンスにはサポートされていません。
- サポートされる AMI (休止状態をサポートする HVM AMI である必要があります):
  - Amazon Linux 2 AMI (2019 年 8 月 29 日以降にリリース)
  - Amazon Linux AMI 2018.03 (2018 年 11 月 16 日以降にリリース)
  - Ubuntu 18.04 LTS - Bionic AMI (シリアル 20190722.1 以降でリリース)\*
  - Ubuntu 16.04 LTS - Xenial AMI。\*([追加設定](#) は必須です。)

\*Ubuntu 18.04 LTS - Bionic and Ubuntu 16.04 LTS - Xenial を使用するインスタンスでは、KASLR を無効にすることをお勧めします。詳細については、「[インスタンスでの KASLR の無効化 \(Ubuntu のみ\) \(p. 539\)](#)」を参照してください。

独自の AMI が休止をサポートするように設定するには、「[休止をサポートするように既存の AMI を設定する \(p. 535\)](#)」を参照してください。

他のバージョンの Ubuntu および他のオペレーティングシステムはまもなくサポートされる予定です。

Windows のサポートされている AMI の詳細については、「[休止の前提条件](#)」(Windows インスタンスの Amazon EC2 ユーザーガイド) を参照してください。

- ルートボリュームタイプ - インスタンスストアボリュームではなく、Amazon EBS ボリュームにする必要があります。
- Amazon EBS ルートボリュームサイズ - RAM の内容を保存し、OS またはアプリケーションなど、予想される使用量に対応できる大きさである必要があります。休止を有効にすると、RAM を保存するために起動時にルートボリュームでスペースが割り当てられます。
- Amazon EBS ルートボリュームの暗号化 - 休止を使用するには、休止時にメモリにある機密性の高いコンテンツを保護するため、ルートボリュームを暗号化する必要があります。RAM データが Amazon EBS のルートボリュームに移動されるときは、常に暗号化されます。ルートボリュームの暗号化は、イ

ンスタンスの起動時に適用されます。ルートボリュームが暗号化された Amazon EBS ボリュームであることを確認するには、次の 3 つのオプションのいずれかを使用します。

- EBS の「シングルステップ」暗号化: 1 回の run-instances API 呼び出しで、暗号化されていない AMI から暗号化された EBS-Backed EC2 インスタンスを起動し、同時に休止機能を有効にすることができます。詳細については、「[EBS-Backed AMI での暗号化の利用 \(p. 151\)](#)」を参照してください。
- デフォルトでの EBS 暗号化: EBS 暗号化をデフォルトで有効にして、AWS アカウントで作成されたすべての新しい EBS ボリュームを暗号化できます。この方法では、インスタンスの起動時に暗号化のインテントを指定することなく、インスタンスの休止を有効にすることができます。詳細については、「[デフォルトでの暗号化 \(p. 1017\)](#)」を参照してください。
- 暗号化された AMI: 暗号化された AMI を使用してインスタンスを起動することで、EBS 暗号化を有効にすることができます。暗号化されたルートスナップショットが AMI ない場合は、それを新しい AMI にコピーして暗号化をリクエストできます。詳細については、「[コピー時に暗号化されていないイメージを暗号化する \(p. 155\)](#)」および「[AMI のコピー \(p. 159\)](#)」を参照してください。
- 起動時に休止を有効にする - 既存のインスタンス(実行中または停止状態)で休止を有効にすることはできません。詳細については、「[インスタンスの休止の有効化 \(p. 538\)](#)」を参照してください。
- 購入オプション - この機能は オンデマンドインスタンス および リザーブドインスタンス でのみ使用できます。スポットインスタンスでは使用できません。詳細については、「[中断したスポットインスタンスの休止 \(p. 387\)](#)」を参照してください。

## 制約事項

- 以下のアクションは、休止ではサポートされません。
  - 休止したインスタンスのタイプまたはサイズを変更する
  - 休止が有効にされているインスタンスからスナップショットまたは AMI を作成する
  - 休止したインスタンスからスナップショットまたは AMI を作成する
- instance store-backed インスタンスは停止または休止できません。\*
- RAM が 150 GB を超えるインスタンスを休止することはできません。
- Auto Scaling グループであるインスタンスまたは Amazon ECS が使用中のインスタンスを休止することはできません。インスタンスが Auto Scaling グループにあり、そのインスタンスを休止しようとしている場合、Amazon EC2 Auto Scaling サービスは停止したインスタンスを異常と判断し、そのインスタンスを終了して代わりのインスタンスを起動する場合があります。詳細については、Amazon EC2 Auto Scaling ユーザーガイドの「[Auto Scaling インスタンスのヘルスチェック](#)」を参照してください。
- 60 日間以上にわたる休止はサポートしていません。60 日より長くインスタンスを保持するには、休止したインスタンスを起動し、停止して、また起動する必要があります。
- 当社では、継続的にプラットフォームをアップグレードやセキュリティパッチで更新しており、休止されている既存のインスタンスと競合する可能性があります。シャットダウンまたは再起動を実行して必要なアップグレードとセキュリティパッチを適用できるように、休止されているインスタンスの起動が必要になる重要な更新については、通知を受け取ります。

\*休止が有効にされている C3 および R3 インスタンスには、インスタンスストアボリュームを使用しないでください。

## 休止をサポートするように既存の AMI を設定する

独自の AMI を使用して起動したインスタンスを休止するには、最初に休止を有効にするように AMI を設定する必要があります。詳細については、「[インスタンスソフトウェアの更新 \(p. 554\)](#)」を参照してください。

[サポートされる AMI \(p. 534\)](#) (Ubuntu 16.04 LTS を除く) のいずれかを使用する場合、またはサポートされる AMI のいずれかに基づいて AMI を作成する場合は、休止をサポートするように AMI を設定する必要はありません。これらの AMI は、休止をサポートするように事前に設定されています。休止をサポートするために Ubuntu 16.04 LTS を設定するには、linux-aws-hwe カーネルパッケージバージョン 4.15.0-1058-

aws 以降および ec2-hibinit-agent をインストールする必要があります。設定ステップについては、下の [Ubuntu 16.04 - Xenial] タブを選択します。

#### Amazon Linux 2

Amazon Linux 2 AMI で休止がサポートされるように設定するには

1. 次のコマンドを使用して、最新のカーネル 4.14.138-114.102 以降に更新します。

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

2. 次のコマンドを使用してリポジトリから ec2-hibinit-agent パッケージをインストールします。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 次のコマンドを実行して、インスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

4. 次のコマンドを実行して、カーネルバージョンが 4.14.138-114.102 以降に更新されていることを確認します。

```
[ec2-user ~]$ uname -a
```

5. インスタンスを停止し、AMI を作成します。詳細については、「[インスタンスからの Linux AMI の作成 \(p. 117\)](#)」を参照してください。

#### Amazon Linux

Amazon Linux AMI で休止がサポートされるように設定するには

1. 次のコマンドを使用して、最新のカーネルを 4.14.77-70.59 以降に更新します。

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

2. 次のコマンドを使用してリポジトリから ec2-hibinit-agent パッケージをインストールします。

```
[ec2-user ~]$ sudo yum install ec2-hibinit-agent
```

3. 次のコマンドを実行して、インスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

4. 次のコマンドを実行して、カーネルバージョンが 4.14.77-70.59 以降に更新されていることを確認します。

```
[ec2-user ~]$ uname -a
```

5. インスタンスを停止し、AMI を作成します。詳細については、「[インスタンスからの Linux AMI の作成 \(p. 117\)](#)」を参照してください。

Ubuntu 18.04 - Bionic

Ubuntu 18.04 LTS AMI で休止がサポートされるように設定するには

1. 次のコマンドを使用して、最新のカーネルを 4.15.0-1044 以降に更新します。

```
[ec2-user ~]$ sudo apt update  
[ec2-user ~]$ sudo apt dist-upgrade
```

2. 次のコマンドを使用してリポジトリから ec2-hibinit-agent パッケージをインストールします。

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 次のコマンドを実行して、インスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

4. 次のコマンドを実行して、カーネルバージョンが 4.15.0-1044 以降に更新されていることを確認します。

```
[ec2-user ~]$ uname -a
```

Ubuntu 16.04 - Xenial

Ubuntu 16.04 LTS AMI で休止がサポートされるように設定するには

1. 次のコマンドを使用して、最新のカーネルを 4.15.0-1058-aws 以降に更新します。

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

Note

linux-aws-hwe カーネルパッケージは、Canonical によって完全にサポートされています。このパッケージは、Ubuntu 16.04 LTS の標準サポートが 2021 年 4 月に終了するまで、定期的な更新を受け取り、拡張セキュリティメンテナンスサポートが 2024 年に終了するまで、追加のセキュリティ更新プログラムを受け取ります。詳細については、「[Ubuntu 16.04 LTS 用 Amazon EC2 の休止機能が利用可能に](#)」を参照してください。

2. 次のコマンドを使用してリポジトリから ec2-hibinit-agent パッケージをインストールします。

```
[ec2-user ~]$ sudo apt install ec2-hibinit-agent
```

3. 次のコマンドを実行して、インスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

4. 次のコマンドを実行して、カーネルバージョンが 4.15.0-1058-aws 以降に更新されていることを確認します。

```
[ec2-user ~]$ uname -a
```

## インスタンスの休止の有効化

インスタンスを休止するには、最初に休止を有効にする必要があります。休止を有効にするには、インスタンスの起動時に有効にする必要があります。

### Important

インスタンスの起動後に、そのインスタンスの休止を有効または無効にすることはできません。

### Console

コンソールを使用して休止を有効にするには

1. 「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」の手順に従います。
2. [Amazon マシンイメージ (AMI)] ページで、休止をサポートする AMI を選択します。サポート対象の AMI の詳細については、「[休止の前提条件 \(p. 534\)](#)」を参照してください。
3. [インスタンスタイプの選択] ページで、サポート対象のインスタンスタイプを選択し、[次の手順: インスタンスの詳細の設定] を選択します。サポート対象のインスタンスタイプの詳細については、「[休止の前提条件 \(p. 534\)](#)」を参照してください。
4. [インスタンスの詳細設定] ページの [Stop - Hibernate Behavior (停止 - 休止動作)] で、[Enable hibernation as an additional stop behavior (追加の停止動作として休止を有効にする)] チェックボックスをオンにします。
5. ウィザードに従って続行します。[Review Instance Launch (インスタンス作成の確認)] ページでオプションの確認が終了したら、[Launch (起動)] を選択します。詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。

### AWS CLI

AWS CLI を使用して休止を有効にするには

`run-instances` コマンドを使用して、インスタンスを起動します。休止を有効にするには、`-- hibernation-options Configured=true` パラメータを使用します。

```
aws ec2 run-instances --image-id ami-0abcdef1234567890 --instance-type m5.large -- hibernation-options Configured=true --count 1 --key-name MyKeyPair
```

### AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell を使用して休止を有効にするには

`New-EC2Instance` コマンドを使用してインスタンスを起動します。`- HibernationOptions_Configured $true` パラメータを使用して休止を有効にします。

```
New-EC2Instance -ImageId ami-0abcdef1234567890 -InstanceType m5.large - HibernationOptions_Configured $true -MinCount 1 -MaxCount 1 -KeyName MyKeyPair
```

### Console

コンソールを使用して、インスタンスで休止が有効かどうかを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。

- インスタンスを選択し、詳細ペインの [Stop - Hibernation behavior (停止 - 休止動作)] を確認します。[有効] は、インスタンスが休止に対して有効であることを示します。

#### AWS CLI

AWS CLI を使用して、インスタンスで休止が有効かどうかを表示するには

`describe-instances` コマンドを使用し、`--filters "Name=hibernation-options.configured,Values=true"` パラメータを指定して、休止が有効になっているインスタンスをフィルタリングします。

```
aws ec2 describe-instances --filters "Name=hibernation-options.configured,Values=true"
```

次の出力フィールドは、インスタンスで休止が有効になっていることを示しています。

```
"HibernationOptions": {  
    "Configured": true  
}
```

#### AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell を使用して、インスタンスで休止が有効になっていることを確認するには

`Get-EC2Instance` コマンドを使用し、`-Filter @{ Name="hibernation-options.configured"; Value="true" }` パラメータを指定して、休止が有効になっているインスタンスをフィルタリングします。

```
Get-EC2Instance -Filter @{ Name="hibernation-options.configured"; Value="true" }
```

休止が有効になっている EC2 インスタンスが出力に一覧表示されます。

## インスタンスでの KASLR の無効化 (Ubuntu のみ)

Ubuntu 16.04 LTS - Xenial または Ubuntu 18.04 LTS - Bionic (シリアル 20190722.1 以降でリリース) で新しく起動されたインスタンスで休止を使用するには、KASLR (Kernel Address Space Layout Randomization) を無効にするようお勧めします。Ubuntu 16.04 LTS または Ubuntu 18.04 LTS では、デフォルトで KASLR が有効になっています。KASLR は、Linux カーネルに対する標準的なセキュリティ機能であり、カーネルのベースアドレス値をランダム化することにより、未知のメモリアクセス脆弱性による露出と影響を軽減するために役立ちます。KASLR が有効になっている場合は、インスタンスを休止後に再開できないこともあります。

KASLR の詳細については、[Ubuntu の機能に関する記述](#)を参照してください。

Ubuntu で起動したインスタンスで KASLR を無効にするには

- SSH を使用してインスタンスに接続します。詳細については、「[SSH を使用した Linux インスタンスへの接続 \(p. 508\)](#)」を参照してください。
- 適切なエディタで、`/etc/default/grub.d/50-cloudimg-settings.cfg` ファイルを開きます。次の例のように、`GRUB_CMDLINE_LINUX_DEFAULT` 行を編集して、行末に `nokaslr` オプションを追加します。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0 nvme_core.io_timeout=4294967295  
nokaslr"
```

- ファイルを保存し、エディタを終了します。

4. grub 設定を再構築するには、次のコマンドを実行します。

```
[ec2-user ~]$ sudo update-grub
```

5. インスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

6. 次のコマンドの実行時に、nokaslr が追加されていることを確認します。

```
[ec2-user ~]$ cat /proc/cmdline
```

コマンドの出力には、nokaslr オプションが含まれている必要があります。

## インスタンスを休止する

インスタンスは、[休止が有効になっており \(p. 538\)](#)、[休止の前提条件 \(p. 534\)](#)を満たしている場合に、休止することができます。インスタンスを休止できない場合、通常のシャットダウンが実行されます。

### Console

コンソールを使用して Amazon EBS-Backed インスタンスを休止するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[操作]、[インスタンスの状態]、[停止] の順に選択します。[Stop - Hibernate (停止 - 休止)] が無効になっている場合は、インスタンスが既に休止または停止しているか、休止できません。詳細については、「[休止の前提条件 \(p. 534\)](#)」を参照してください。
4. 確認ダイアログボックスで [Yes, Stop - Hibernate (停止する - 休止)] を選択します。インスタンスが休止するまで、数分かかる場合があります。インスタンスが休止に入ると、インスタンスの状態が [停止中] に変わり、インスタンスが休止すると [停止] になります。

### AWS CLI

AWS CLI を使用して Amazon EBS-Backed インスタンスを休止するには

`stop-instances` コマンドを使用して `--hibernate` パラメータを指定します。

```
aws ec2 stop-instances --instance-ids i-1234567890abcdef0 --hibernate
```

### AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell を使用して Amazon EBS-Backed インスタンスを休止するには

`Stop-EC2Instance` コマンドを使用して、`-Hibernate $true` パラメータを指定します。

```
Stop-EC2Instance -InstanceId i-1234567890abcdef0 -Hibernate $true
```

### Console

コンソールを使用して、インスタンスで休止が開始されたかどうかを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、詳細ペインの [状態遷移の理由メッセージ] を確認します。Client.UserInitiatedHibernate: User initiated hibernate というメッセージは、インスタンスで休止が開始されたことを示します。

#### AWS CLI

AWS CLI を使用して、インスタンスで休止が開始されたかどうかを表示するには

`describe-instances` コマンドを使用して、`state-reason-code` フィルターを指定し、休止が開始されたインスタンスを確認します。

```
aws ec2 describe-instances --filters "Name=state-reason-code,Values=Client.UserInitiatedHibernate"
```

出力の次のフィールドは、そのインスタンスで休止が開始されたことを示しています。

```
"StateReason": {  
    "Code": "Client.UserInitiatedHibernate"  
}
```

#### AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell を使用して、インスタンスで休止が開始されたかどうかを確認するには

`Get-EC2Instance` コマンドを使用し、`state-reason-code` フィルタを指定して休止が開始されたインスタンスを確認します。

```
Get-EC2Instance -Filter @{Name="state-reason-code";Value="Client.UserInitiatedHibernate"}
```

休止が開始された EC2 インスタンスが出力に一覧表示されます。

## 休止したインスタンスの起動

休止したインスタンスは、停止したインスタンスを起動するのと同じ方法で起動します。

#### Console

コンソールを使用して、休止したインスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. 休止したインスタンスを選択し、[操作]、[インスタンスの状態]、[開始] の順に選択します。インスタンスが `running` 状態になるまで、数分かかる場合があります。この間、インスタンスの [ステータスチェック \(p. 629\)](#) では、インスタンスが起動するまで、インスタンスは失敗状態にあるように表示されます。

#### AWS CLI

AWS CLI を使用して、休止したインスタンスを起動するには

`start-instances` コマンドを使用します。

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

#### AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell を使用して、休止したインスタンスを起動するには

[Start-EC2Instance](#) コマンドを使用します。

```
Start-EC2Instance -InstanceId i-1234567890abcdef0
```

## 休止のトラブルシューティング

次の情報を使用して、インスタンスを休止するときに発生する可能性がある問題の診断や修復を行います。

### 起動直後に休止できません

インスタンスの起動後にすぐ休止しようとすると、エラーが発生します。

起動後、休止するまで約 2 分待つ必要があります。

### stopping から stopped への移行に時間がかかりすぎ、起動後にメモリ状態が復元されません

休止しているインスタンスが stopping 状態から stopped に移行するのに時間がかかり過ぎ、メモリの状態が起動後に復元されない場合は、休止が正しく設定されていない可能性があります。

インスタンスのシステムログをチェックして、休止に関連するメッセージを探します。システムログにアクセスするには、インスタンスに接続 (p. 505) するか、[get-console-output](#) コマンドを使用します。hibinit-agent からログ行を見つけます。ログ行が失敗を示している場合、またはログ行がない場合、起動時に休止の設定に失敗している可能性が高いと思われます。

たとえば、メッセージ「hibinit-agent: Insufficient disk space. Cannot create setup for hibernation. Please allocate a larger root device.」は、インスタンスのルートボリュームの大きさが十分ではないことを示しています。

hibinit-agent の最後のログ行が hibinit-agent: Running: swapoff /swap である場合、休止は正常に設定されています。

これらのプロセスで何もログが表示されない場合、AMI が休止をサポートしていない可能性があります。サポート対象の AMI の詳細については、「[休止の前提条件 \(p. 534\)](#)」を参照してください。独自の AMI を使用した場合は、[休止をサポートするように既存の AMI を設定する \(p. 535\)](#) に関する指示に従っていることを確認します。

### インスタンスが stopping 状態で止まりました

インスタンスを休止し、stopping 状態で止まったように見える場合は、インスタンスを強制終了できます。詳細については、「[インスタンスの停止に関するトラブルシューティング \(p. 1143\)](#)」を参照してください。

## インスタンスの再起動

インスタンスの再起動は、オペレーティングシステムの再起動と同等です。ほとんどの場合、インスタンスの再起動には数分しかかかりません。インスタンスを再起動すると、インスタンスのパブリック DNS

名 (IPv4)、プライベート IPv4 アドレス、IPv6 アドレス (該当する場合)、およびインスタンストアボリューム上のすべてのデータが保持されます。

インスタンスを再起動しても、インスタンスの停止と起動とは異なり、新しいインスタンスの課金 (最低料金 1 分間分) は開始されません。

再起動を必要とする更新の適用など、必要なメンテナンスのために、インスタンスの再起動を予定する場合があります。ユーザーが操作する必要はありません。予定されている時間帯に自動的に行われる再起動まで待つことをお勧めします。詳細については、「[インスタンスの予定されたイベント \(p. 633\)](#)」を参照してください。

インスタンスからオペレーティングシステムの再起動コマンドを実行する代わりに、Amazon EC2 コンソール、コマンドラインツール、または Amazon EC2 API を使用してインスタンスを再起動することをお勧めします。Amazon EC2 コンソール、コマンド行ツール、または Amazon EC2 API を使用してインスタンスを再起動する場合、インスタンスが 4 分以内に完全にシャットダウンしないと、ハードリブートが実行されます。AWS CloudTrail を使用した場合は、Amazon EC2 を使用してインスタンスを再起動すると、インスタンスがいつ再起動されたかについて API レコードも作成されます。

コンソールを使用してインスタンスを再起動するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選び、[Actions]、[Instance State]、[Reboot] の順に選択します。
4. 確認を求めるメッセージが表示されたら、[Yes, Reboot] を選択します。インスタンスは [実行中] 状態のままになります。

コマンドラインを使用してインスタンスを再起動するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `reboot-instances` (AWS CLI)
- `Restart-EC2Instance` (AWS Tools for Windows PowerShell)

## インスタンスのリタイア

インスタンスをホストしている基盤のハードウェアで回復不可能な障害が検出されると、AWS によってインスタンスのリタイアが予定されます。予定されたリタイア日になると、インスタンスは AWS によって停止または削除されます。インスタンスのルートデバイスが Amazon EBS ボリュームである場合、インスタンスは停止されますが、その後いつでも再び起動できます。停止したインスタンスを開始すると、新しいハードウェアに移行されます。インスタンスのルートデバイスがインスタンストアボリュームである場合、インスタンスは削除し、再び使用することはできません。

コンテンツ

- [リタイアが予定されているインスタンスの特定 \(p. 543\)](#)
- [リタイアが予定されているインスタンスの操作 \(p. 544\)](#)

インスタンスイベントのタイプの詳細については、「[インスタンスの予定されたイベント \(p. 633\)](#)」を参照してください。

## リタイアが予定されているインスタンスの特定

インスタンスのリタイアが予定された場合、イベントの前に、当該のインスタンス ID とリタイア日を記載したメールが送信されます。このメールは、アカウントに関連付けられているアドレスに送信されます。

これは、AWS マネジメントコンソールへのログインに使用するメールアドレスと同じです。定期的に確認しないメールアカウントを使用している場合は、Amazon EC2 コンソールまたはコマンドラインを使用して、いずれかのインスタンスにリタイアが予定されているかどうかを判断できます。アカウントの連絡先情報を更新するには、「[アカウント設定](#)」ページに移動します。

コンソールを使用してリタイアが予定されているインスタンスを特定するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[EC2 ダッシュボード] を選択します。[Scheduled Events] に、Amazon EC2 インスタンスおよびボリュームに関連付けられたイベントが表示されます。

## Scheduled Events



US East (N. Virginia):

[1 instances have scheduled events](#)

3. インスタンスに予定されたイベントが表示されている場合は、リージョン名の下のリンクを選択して [Events] ページにアクセスします。
4. [Events] ページには、すべてのリソースとそれに関連付けられたイベントが一覧表示されます。リタイアが予定されているインスタンスを表示するには、1 つ目のフィルタリストから [Instance resources] を選択し、2 つ目のフィルタリストから [Instance stop or retirement] を選択します。
5. フィルタの結果にインスタンスのリタイアが予定されていることが表示されたら、当該のインスタンスを選択し、詳細ペインの [Start time] フィールドの日時を書き留めます。これがインスタンスのリタイア日です。

コマンドラインを使用してリタイアが予定されているインスタンスを特定するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-instance-status](#) (AWS CLI)
- [Get-EC2InstanceState](#) (AWS Tools for Windows PowerShell)

## リタイアが予定されているインスタンスの操作

インスタンスのリタイアが予定されているときに実行できるアクションはいくつかあります。実行するアクションは、インスタンスのルートデバイスが Amazon EBS ボリュームであるかインスタンストアボリュームであるかによって異なります。インスタンスのルートデバイスタイプが不明な場合は、Amazon EC2 コンソールまたはコマンドラインを使用して調べることができます。

### インスタンスのルートデバイスタイプの判別

コンソールを使用してインスタンスのルートデバイスタイプを判別するには

1. ナビゲーションペインの [Events] を選択します。前述の手順「[リタイアが予定されているインスタンスの特定 \(p. 544\)](#)」で説明したように、フィルタリストを使用してリタイアが予定されているインスタンスを特定します。
2. [Resource Id] 列でインスタンス ID を選択すると、[Instances] ページに移動します。
3. インスタンスを選択し、[Description] タブで [Root device type] フィールドを探します。この値が ebs の場合、インスタンスは EBS-Backed です。この値が instance-store の場合、インスタンスは、Instance Store-Backed です。

コマンドラインを使用してインスタンスのルートデバイスタイプを判別するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-instances \(AWS CLI\)](#)
- [Get-EC2Instance \(AWS Tools for Windows PowerShell\)](#)

## リタイアが予定されているインスタンスの管理

リタイアが予定されているインスタンスのデータを維持するには、以下に挙げるいずれかのアクションを実行します。予期しないダウントIMEやデータ消失を防ぐために、インスタンスのリタイア日より前にこのアクションを実行することが重要です。

### Warning

Instance Store-Backed インスタンスの場合、リタイア日を過ぎるとインスタンスが終了し、インスタンスやインスタンスに格納されていたデータを復元できなくなります。インスタンストアボリュームのデータは、インスタンスのルートデバイスにかかわらず、EBS-Backed インスタンスにアタッチされている場合でも、インスタンスがリタイアされると失われます。

インスタンスのルートデバイスタイプ	アクション
EBS	バックアップがあるように、インスタンスから EBS-backed AMI を作成します。予定されたリタイア日を待ちます - その日になるとインスタンスが停止します - または、リタイア日の前に自分でインスタンスを停止します。インスタンスはいつでも再び開始することができます。インスタンスの停止と開始、インスタンスを停止したときに予想される影響 (インスタンスに関連付けられたパブリック IP アドレス、プライベート IP アドレス、および Elastic IP アドレスへの影響など) の詳細については、「 <a href="#">インスタンスの停止と起動 (p. 529)</a> 」を参照してください。
EBS	インスタンスから EBS-Backed AMI を作成し、代替インスタンスを起動します。詳細については、「 <a href="#">Amazon EBS-Backed Linux AMI の作成 (p. 116)</a> 」を参照してください。
インスタンストア	AMI ツールを使用してインスタンスから Instance-Store Backed AMI を作成し、代替インスタンスを起動します。詳細については、「 <a href="#">Instance Store-Backed Linux AMI の作成 (p. 119)</a> 」を参照してください。
インスタンストア	データを EBS ボリュームに転送し、ボリュームのスナップショットを作成し、スナップショットから AMI を作成することによって、EBS-Backed インスタンスをインスタンスに変換します。新しい AMI から代替インスタンスを起動できます。詳細については、「 <a href="#">Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換する (p. 131)</a> 」を参照してください。

## インスタンスの終了

不要になったインスタンスは削除できます。これは、インスタンスの削除と呼ばれます。インスタンスの状態が `shutting-down` または `terminated` に変わったら、そのインスタンスへの課金は停止します。

インスタンスを削除した後に、接続または起動することはできません。ただし、同じ AMI から別のインスタンスを起動することができます。インスタンスを停止および起動するか、または休止する場合は、「[インスタンスの停止と起動 \(p. 529\)](#)」または「[Linux インスタンスの休止 \(p. 532\)](#)」を参照してください。詳細については、「[再起動、停止、休止、終了の違い \(p. 447\)](#)」を参照してください。

## コンテンツ

- [インスタンスの削除 \(p. 546\)](#)
- [インスタンスを削除するとどうなるか \(API\) \(p. 546\)](#)
- [インスタンスを削除する \(p. 547\)](#)
- [インスタンスの削除保護の有効化 \(p. 547\)](#)
- [インスタンスによって起動されたシャットダウン動作の変更 \(p. 548\)](#)
- [インスタンスの削除で Amazon EBS ボリュームを保持する \(p. 549\)](#)
- [トラブルシューティング \(p. 551\)](#)

## インスタンスの削除

インスタンスの削除後、インスタンスはしばらくの間コンソールに表示されたままであるが、エントリは自動的に削除されます。終了したインスタンスのエントリを自分で削除することはできません。インスタンスを削除すると、タグやボリュームなどのリソースはインスタンスから徐々に関連付けが解除され、しばらくすると削除されたインスタンスでこれらのリソースが表示されなくなる可能性があります。

インスタンスが終了すると、そのインスタンスに関連付けられたすべてのインスタンストアボリュームのデータが削除されます。

デフォルトでは、インスタンスの削除時に Amazon EBS のルートデバイスボリュームが自動的に削除されます。ただし、起動時にアタッチした追加の EBS ボリューム、または既存のインスタンスにアタッチした EBS ボリュームがある場合、デフォルトでは、インスタンスの削除後もそれらのボリュームは保持されます。この動作はボリュームの `DeleteOnTermination` 属性によって制御されますが、変更できます。詳細については、「[インスタンスの削除で Amazon EBS ボリュームを保持する \(p. 549\)](#)」を参照してください。

AWS マネジメントコンソール、CLI、および API を使用している他のユーザーによって、誤ってインスタンスを終了されないようにできます。この機能は、Amazon EC2 instance store-backed インスタンスと Amazon EBS-backed インスタンスの両方で使用できます。各インスタンスには、デフォルト値の `false` である `DisableApiTermination` 属性があります（インスタンスは Amazon EC2 によって終了される場合があります）。インスタンスの実行中または停止中に、このインスタンス属性を変更できます（Amazon EBS-backed インスタンスの場合）。詳細については、「[インスタンスの削除保護の有効化 \(p. 547\)](#)」を参照してください。

システムをシャットダウンするオペレーティングシステムコマンドを使用して、インスタンスからシャットダウンが開始されたときに、インスタンスを停止または終了するかどうかを制御できます。詳細については、「[インスタンスによって起動されたシャットダウン動作の変更 \(p. 548\)](#)」を参照してください。

インスタンスの終了時にスクリプトを実行した場合、シャットダウンスクリプトが実行されることを証する方法がないため、異常な終了が発生する場合があります。Amazon EC2 はインスタンスを正常にシャットダウンして、システムシャットダウンスクリプトが実行されるように試みますが、特定のイベント（ハードウェア障害など）ではシステムシャットダウンスクリプトが実行されないことがあります。

## インスタンスを削除するとどうなるか (API)

`terminate-instances` コマンドを使用して EC2 インスタンスが削除された場合、OS レベルで以下が登録されています。

- API リクエストは、ボタンのクリックイベントをゲストに送信します。
- ボタンのクリックイベントの結果、さまざまなシステムサービスが停止されます。systemd はシステムの適切なシャットダウンを処理します。適切なシャットダウンは、ハイパーテーバイザーからの ACPI シャットダウンボタンのクリックイベントによってトリガーされます。
- ACPI のシャットダウンが開始されます。
- このインスタンスは、適切なシャットダウンプロセスが終了したときにシャットダウンされます。設定可能な OS シャットダウン時間はありません。

## インスタンスを削除する

インスタンスは AWS マネジメントコンソール またはコマンドラインを使用して終了できます。

コンソールを使用してインスタンスを終了するには

1. インスタンスを終了する前に、終了時に Amazon EBS ボリュームが削除され、必要な任意のデータをインスタンスストアボリュームから Amazon EBS または Amazon S3 にコピーしていることを確認して、データが失われないことを確認します。
2. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
3. ナビゲーションペインで、[インスタンス] を選択します。
4. 該当インスタンスを選択し、[Actions]、[Instance State]、[Terminate] の順に選択します。
5. 確認を求めるメッセージが表示されたら、[Yes, Terminate] を選択します。

コマンドラインを使用してインスタンスを削除するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [terminate-instances \(AWS CLI\)](#)
- [Stop-EC2Instance \(AWS Tools for Windows PowerShell\)](#)

## インスタンスの削除保護の有効化

デフォルトでは、Amazon EC2 コンソール、コマンドラインインターフェイス、または API を使用して、インスタンスを終了できます。Amazon EC2 を使用してインスタンスを誤って終了できないようにするには、インスタンスの削除保護を有効にできます。DisableApiTermination 属性は、インスタンスがコンソール、CLI、または API を使用して終了できるかどうかを制御します。デフォルトでは、インスタンスの削除保護は無効になっています。インスタンスが実行中またはインスタンスが停止中に、インスタンスを起動する際に、この属性の値を設定できます (Amazon EBS-backed インスタンスの場合)。

DisableApiTermination 属性が設定された場合、InstanceInitiatedShutdownBehavior 属性はインスタンスからシャットダウンを開始して (システムシャットダウン用のオペレーティングシステムコマンドを使用)、インスタンスを終了できます。詳細については、「[インスタンスによって起動されたシャットダウン動作の変更 \(p. 548\)](#)」を参照してください。

### 制約事項

スポットインスタンス の削除保護を有効にすることはできません。スポットインスタンス は、スポット料金が スポットインスタンス への支払金額を超えると終了します。しかし、スポットインスタンス の中断を処理するようにアプリケーションを準備できます。詳細については、「[スポットインスタンス の中断 \(p. 385\)](#)」を参照してください。

DisableApiTermination 属性では、Amazon EC2 Auto Scaling によるインスタンスの終了は防止されません。Auto Scaling グループ内のインスタンスについては、Amazon EC2 の終了の防止ではなく Amazon EC2 Auto Scaling の次の機能を使用します。

- Auto Scaling グループ内のインスタンスがスケールイン時に終了されないようにするには、インスタンスの保護を使用します。詳細については、「[インスタンスの保護](#)」(Amazon EC2 Auto Scaling ユーザーガイド) を参照してください。
- Amazon EC2 Auto Scaling による異常なインスタンスの終了を防止するには、ReplaceUnhealthy ポートセスを停止します。詳細については、「[Amazon EC2 Auto Scaling ユーザーガイド](#)」の「[スケーリングプロセスの中止と再開](#)」を参照してください。

- Amazon EC2 Auto Scaling によってどのインスタンスを最初に終了するかを指定するには、終了ポリシーを選択します。詳細については、「[終了ポリシーのカスタマイズ](#)」(Amazon EC2 Auto Scaling ユーザーガイド) を参照してください。

起動時にインスタンスに対する終了保護を有効にするには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ダッシュボードで、[Launch Instance] を選択し、ウィザードの指示に従います。
- [Configure Instance Details] ページで、[Enable termination protection] チェックボックスをオンにします。

実行中または停止中のインスタンスの削除保護を有効にするには

- インスタンスを選択してから、[Actions (アクション)]、[インスタンスの設定]、[削除保護の変更] の順に選択します。
- [はい、有効化する] を選択します。

実行中または停止中のインスタンスの削除保護を無効にするには

- インスタンスを選択してから、[Actions (アクション)]、[インスタンスの設定]、[削除保護の変更] の順に選択します。
- [Yes, Disable] を選択します。

コマンドラインを使用して終了保護を有効または無効にするには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [modify-instance-attribute \(AWS CLI\)](#)
- [Edit-EC2InstanceAttribute \(AWS Tools for Windows PowerShell\)](#)

## インスタンスによって起動されたシャットダウン動作の変更

デフォルトでは、Amazon EBS-backed インスタンスからシャットダウンを開始すると (shutdown や poweroff などのコマンドを使用)、インスタンスが停止します (halt を使用しても、poweroff コマンドは実行されません。使用した場合も、インスタンスは終了しません。代わって、CPU が HLT に配置され、インスタンスは実行されたままになります)。代わりに終了できるように、インスタンスの InstanceInitiatedShutdownBehavior 属性を使用して、この動作を変更できます。インスタンスの実行中または停止中に、この属性を更新できます。

InstanceInitiatedShutdownBehavior 属性は Amazon EC2 コンソールまたはコマンドラインを使用して更新できます。InstanceInitiatedShutdownBehavior 属性は、インスタンス自体のオペレーティングシステムからシャットダウンを実行する場合にのみ適用されます。これは StopInstances API または Amazon EC2 コンソールを使用してインスタンスを停止する場合には適用されません。

コンソールを使用してインスタンスのシャットダウン動作を変更するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[インスタンス] を選択します。
- インスタンスを選択してから、[Actions (アクション)]、[インスタンスの設定]、[シャットダウン動作の変更] の順に選択します。現在の動作は既に選択されています。
- 動作を変更するには、[シャットダウン動作] リストからオプションを選択してから、[Apply (適用)] を選択します。



コマンドラインを使用してインスタンスのシャットダウン動作を変更するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [modify-instance-attribute](#) (AWS CLI)
- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

## インスタンスの削除で Amazon EBS ボリュームを保持する

インスタンスが終了すると、Amazon EC2 はアタッチされた各 Amazon EBS ボリュームの `DeleteOnTermination` 属性の値を使用して、ボリュームを保持するか削除するかを決定します。`DeleteOnTermination` 属性のデフォルト値は、ボリュームがインスタンスのルートボリュームであるかどうかによって異なります。

デフォルトでは、インスタンスのルートボリュームの `DeletionOnTermination` 属性は `true` に設定されます。したがって、デフォルトではインスタンスの削除時に、インスタンスのルートボリュームが削除されます。この `DeletionOnTermination` 属性は、AMI の作成者とインスタンスを起動するユーザーが設定できます。AMI の作成者またはインスタンスを起動したユーザーによって属性が変更された場合、元の AMI のデフォルト設定は新しい設定に上書きされます。AMI でインスタンスを起動したら、`DeletionOnTermination` 属性のデフォルト設定を確認することをお勧めします。

デフォルトでは、インスタンスに EBS ボリュームをアタッチするときは、`DeleteOnTermination` 属性が `false` に設定されます。したがって、デフォルトではこれらのボリュームが保持されます。インスタンスが終了したら、保持されたボリュームのスナップショットを作成するか、別のインスタンスにアタッチできます。不要な料金の発生を防ぐために、ボリュームを削除する必要があります。詳細については、「[Amazon EBS ボリュームの削除 \(p. 969\)](#)」を参照してください。

使用中の EBS ボリュームの `DeleteOnTermination` 属性の値を確認するには、インスタンスのブロックデバイスマッピングを参照します。詳細については、「[インスタンスブロックデバイスマッピングの EBS ボリュームの表示 \(p. 1107\)](#)」を参照してください。

インスタンスの起動時またはインスタンスの実行中に、ボリュームの `DeleteOnTermination` 属性の値を変更できます。

例

- コンソールを使用した起動時のルートボリュームの永続的な変更 (p. 550)
- コマンドラインを使用した起動時のルートボリュームの永続的な変更 (p. 550)
- コマンドラインを使用して実行中のインスタンスのルートボリュームが存続するように変更する (p. 551)

## コンソールを使用した起動時のルートボリュームの永続的な変更

コンソールを使用して、インスタンスの起動時に `DeleteOnTermination` 属性を変更できます。実行中のインスタンスのこの属性を変更するには、コマンドラインを使用する必要があります。

コンソールを使用して、起動時にインスタンスのルートボリュームが存続するように変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. コンソールダッシュボードで、[Launch Instance] を選択します。
3. [Choose an Amazon Machine Image (AMI)] ページで、AMI を選択し、[Select] を選択します。
4. ウィザードにしたがって [Choose an Instance Type] ページと [Configure Instance Details] ページを設定します。
5. [Add Storage] ページで、ルートボリュームの [Delete On Termination] チェックボックスの選択を解除します。
6. ウィザードの残りのページを完了した後、[Launch] を選択します。

インスタンスの詳細ペインでルートデバイスピリュームの詳細を表示することにより、設定を確認できます。[Block devices] の隣にあるルートデバイスピリュームのエントリを選択します。デフォルトでは、[Delete on termination] は [True] です。デフォルトの動作を変更した場合は、[Delete on termination] が [False] になっています。

## コマンドラインを使用した起動時のルートボリュームの永続的な変更

EBS-backed インスタンスの起動時に、次のコマンドのいずれかを使用して、ルートデバイスピリュームが存続するように変更することができます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `run-instances` (AWS CLI)
- `New-EC2Instance` (AWS Tools for Windows PowerShell)

たとえば、`run-instances` コマンドに次のオプションを追加します。

```
--block-device-mappings file://mapping.json
```

`mapping.json` で、以下を指定します。

```
[  
  {  
    "DeviceName": "/dev/sda1",  
    "Ebs": {  
      "DeleteOnTermination": false,  
      "SnapshotId": "snap-1234567890abcdef0",  
      "VolumeType": "gp2"  
    }  
  }  
]
```

## コマンドラインを使用して実行中のインスタンスのルートボリュームが存続するように変更する

次のいずれかのコマンドを使用して、実行中の EBS-backed インスタンスのルートデバイスボリュームを永続化するように変更できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [modify-instance-attribute \(AWS CLI\)](#)
- [Edit-EC2InstanceAttribute \(AWS Tools for Windows PowerShell\)](#)

たとえば、以下のコマンドを使用します。

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

mapping.json で、以下を指定します。

```
[  
 {  
     "DeviceName": "/dev/sda1",  
     "Ebs": {  
         "DeleteOnTermination": false  
     }  
 }  
]
```

## トラブルシューティング

インスタンスが通常より長く `shutting-down` 状態になっている場合、最終的に Amazon EC2 サービス内の自動プロセスによってクリーンアップ(終了)されます。詳細については、「[インスタンスの削除\(シャットダウン\)のトラブルシューティング \(p. 1145\)](#)」を参照してください。

## インスタンスの復旧

Amazon CloudWatch インスタンスをモニタリングし、基になるハードウェア障害または AWS による修復を必要とする問題によりインスタンスが正常に機能しなくなった場合に、自動的にインスタンスを復旧する Amazon EC2 アラームを作成できます。終了したインスタンスは復旧できません。復旧されたインスタンスは、インスタンス ID、プライベート IP アドレス、Elastic IP アドレス、すべてのインスタンスマタデータを含め、元のインスタンスと同じです。障害のあるインスタンスがプレイスメントグループ内にある場合、回復されたインスタンスはそのプレイスメントグループ内で実行されます。インスタンスを復旧する Amazon CloudWatch アラームの使用の詳細については、「[Amazon CloudWatch アラームへの復旧アクションの追加 \(p. 667\)](#)」を参照してください。インスタンスの復旧の失敗に関する問題のトラブルシューティングを行うには、「[インスタンスの復旧の失敗のトラブルシューティング \(p. 552\)](#)」を参照してください。

`StatusCheckFailed_System` アラームがトリガーされ、復旧アクションが開始されると、アラームを作成したときに選択し、復旧アクションに関連付けた Amazon SNS トピックによって通知されます。インスタンスを復旧する際、インスタンスを再起動するときにインスタンスは移行され、メモリ内にあるデータは失われます。プロセスが完了すると、情報はアラームに設定された SNS トピックに発行されます。この SNS トピックにサブスクライブされるすべてのユーザーは、復旧処理のステータスと、それ以降の手順を含むメールの通知を受け取ります。復旧されたインスタンスでインスタンスが再起動されたことがわかります。

システムステータスチェックの失敗の原因となる問題には、次のようなものがあります。

- ネットワーク接続の喪失

- ・システム電源の喪失
- ・物理ホストのソフトウェアの問題
- ・ネットワーク到達可能性に影響する、物理ホスト上のハードウェアの問題

インスタンスにパブリック IPv4 アドレスが割り当てられている場合、そのパブリック IPv4 アドレスは復旧後も保持されます。

## 要件

復旧アクションは、次のような特性を持つインスタンスでのみサポートされています。

- ・インスタンスタイプである A1、C3、C4、C5、C5n、Inf1、M3、M4、M5、M5a、M5n、P3、R3、R4、R5、R5a、R5n、T2、T3、T3a、X1、または X1e のいずれかを使用する
- ・virtual private cloud (VPC) 内で実行される
- ・default または dedicated インスタンステナントを使用する
- ・EBS ボリュームのみ持っている (インスタンストアボリュームは設定しない)

## インスタンスの復旧の失敗のトラブルシューティング

以下の問題が、インスタンスの自動復旧の失敗の原因になる場合があります。

- ・代替ハードウェアの一時的な容量不足。
- ・インスタンスにインスタンストアストレージがアタッチされていますが、自動インスタンス復旧の設定がサポートされていません。
- ・進行中の Service Health Dashboard イベントがあり、復旧プロセスの正常な実行が妨げられています。サービスの可用性に関する最新情報については、「<http://status.aws.amazon.com/>」を参照してください。
- ・インスタンスが、1 日に許可されている 3 回の復旧試行回数に達しました。

自動復旧プロセスは、1 日で 3 回まで別個のエラーについてインスタンスの復旧を試みます。インスタンスのシステムステータスチェックの失敗が続く場合は、インスタンスを手動で停止および開始することをお勧めします。詳細については、「[インスタンスの停止と起動 \(p. 529\)](#)」を参照してください。

インスタンスは、その後、自動復旧が失敗し、元のシステムステータスチェックの失敗の根本原因がハードウェアの機能低下であると判断された場合、リタイアすることができます。

## Amazon Linux インスタンスを設定する

Amazon Linux インスタンスを正常に起動し、ログインしたら、変更できます。特定のアプリケーションのニーズを満たすためにインスタンスを設定するには、多くの方法があります。ここでは、初めて作業する場合の一般的なタスクについて説明します。

### コンテンツ

- ・一般的な設定シナリオ (p. 553)
- ・Linux インスタンスでのソフトウェアの管理 (p. 553)
- ・Linux インスタンスでのユーザー アカウントの管理 (p. 559)
- ・EC2 インスタンスタイプのプロセッサのステート制御 (p. 561)
- ・Linux インスタンスの時刻の設定 (p. 567)

- CPU オプションの最適化 (p. 571)
- Linux インスタンスのホスト名の変更 (p. 583)
- Linux インスタンスの動的な DNS のセットアップ (p. 586)
- Linux インスタンスでの起動時のコマンドの実行 (p. 588)
- インスタンスマターデータとユーザーデータ (p. 593)

## 一般的な設定シナリオ

Amazon Linux のベースのディストリビューションには、基本的なサーバー操作に必要なソフトウェア パッケージとユーティリティが数多く含まれています。ただし、さまざまなソフトウェアリポジトリでさらに多くのソフトウェアパッケージを利用できます。また、ソースコードから、さらに多くのパッケージ ソースコードを作成できます。これらの場所からソフトウェアをインストールし、作成する方法についての詳細は、[Linux インスタンスでのソフトウェアの管理 \(p. 553\)](#) を参照してください。

Amazon Linux インスタンスには、`ec2-user` アカウントが事前設定されていますが、スーパーユーザー 権限を持たない他のユーザーアカウントを追加することができます。ユーザーアカウントの追加と削除についての詳細は、[Linux インスタンスでのユーザーアカウントの管理 \(p. 559\)](#) を参照してください。

Amazon Linux インスタンスのデフォルトの時間設定では、Amazon Time Sync Service を利用し、システム時間をインスタンスに設定します。デフォルトの時間帯は UTC です。インスタンスの時間帯の設定または独自のタイムサーバーの利用についての詳細は、[Linux インスタンスの時刻の設定 \(p. 567\)](#) を参照してください。

お客様がネットワークを所有し、それにドメイン名を登録している場合、インスタンスのホスト名を変更して、そのドメインに含まれる一部としてインスタンスを識別できます。また、システムプロンプトを変更して、より意味のある名前を表示することもできます。ホスト名設定を変更する必要はありません。詳細については、「[Linux インスタンスのホスト名の変更 \(p. 583\)](#)」を参照してください。動的 DNS サービスプロバイダを使用するようにインスタンスを設定できます。詳細については、「[Linux インスタンスの動的な DNS のセットアップ \(p. 586\)](#)」を参照してください。

Amazon EC2 でインスタンスを起動するとき、起動後にそのインスタンスにユーザーデータを渡し、一般的な設定タスクを実行したり、スクリプトを実行したりできます。2 つのタイプのユーザーデータを Amazon EC2 に渡すことができます。cloud-init ディレクティブとシェルスクリプトです。詳細については、「[Linux インスタンスでの起動時のコマンドの実行 \(p. 588\)](#)」を参照してください。

## Linux インスタンスでのソフトウェアの管理

Amazon Linux のベースのディストリビューションには、基本的なサーバー操作に必要なソフトウェア パッケージとユーティリティが数多く含まれています。ただし、さまざまなソフトウェアリポジトリでさらに多くのソフトウェアパッケージを利用できます。また、ソースコードから、さらに多くのパッケージ ソースコードを作成できます。

### コンテンツ

- [インスタンスソフトウェアの更新 \(p. 554\)](#)
- [リポジトリの追加 \(p. 555\)](#)
- [Amazon Linux でソフトウェアパッケージを見つける \(p. 556\)](#)
- [ソフトウェアパッケージのインストール \(p. 557\)](#)
- [ソフトウェアのコンパイル準備 \(p. 558\)](#)

ソフトウェアを最新の状態に維持することが重要です。Linux ディストリビューションの多くのパッケージは頻繁に更新されます。これにより、バグが修正され、機能が追加されて、セキュリティ上の弱点に対する防御措置が行われます。詳細については、「[インスタンスソフトウェアの更新 \(p. 554\)](#)」を参照してください。

デフォルトで、Amazon Linux インスタンスは、次のリポジトリを有効にして起動します。

- Amazon Linux 2: `amzn2-core`、および `amzn2extra-docker`
- Amazon Linux AMI: `amzn-main`、および `amzn-updates`

これらのリポジトリには、Amazon Web Services が更新するさまざまなパッケージがありますが、別のリポジトリに含まれるパッケージをインストールしたい場合があるかもしれません。詳細については、「[リポジトリの追加 \(p. 555\)](#)」を参照してください。有効なリポジトリでパッケージを探す方法については、[Amazon Linux でソフトウェアパッケージを見つける \(p. 556\)](#) を参照してください。Amazon Linux インスタンスにソフトウェアをインストールする方法については、[ソフトウェアパッケージのインストール \(p. 557\)](#) を参照してください。

リポジトリに保管されているソフトウェアパッケージで、すべてのソフトウェアが利用できるわけではありません。一部のソフトウェアは、そのソースコードからインスタンスでコンパイルする必要があります。詳細については、「[ソフトウェアのコンパイル準備 \(p. 558\)](#)」を参照してください。

Amazon Linux インスタンスは、yum パッケージマネージャを利用してソフトウェアを管理します。yum パッケージマネージャはソフトウェアをインストール、削除、更新し、各パッケージのすべての依存関係を管理できます。Ubuntu のような Debian ベースの Linux ディストリビューションは、apt-get コマンドと dpkg パッケージマネージャを使用するため、次のセクションの yum の例は、このようなディストリビューションには該当しません。

## インスタンスソフトウェアの更新

ソフトウェアを最新の状態に維持することが重要です。Linux ディストリビューションの多くのパッケージは頻繁に更新されます。これにより、バグが修正され、機能が追加されて、セキュリティ上の弱点に対する防御措置が行われます。最初に Amazon Linux インスタンスを起動して接続するときに、メッセージが表示され、セキュリティ上の目的からソフトウェアパッケージを更新するように求められる場合があります。このセクションでは、システム全体またはパッケージを 1 つだけ更新する方法を紹介します。

### Important

この情報は Amazon Linux に適用されます。その他のディストリビューションの情報については、各ドキュメントを参照してください。

Amazon Linux インスタンスのすべてのパッケージを更新するには

1. (オプション) シェルウインドウで screen セッションを開始します。ネットワークが遮断され、インスタンスへの SSH 接続が切離されることがあります。この状態が長時間のソフトウェア更新中に発生した場合、インスタンスは混乱した状態になりますが、復元できます。screen セッションを開始しておけば、接続が遮断された場合でも更新を続行でき、後で問題なくセッションに再接続できます。

- a. セッションを開始するには screen コマンドを実行します。

```
[ec2-user ~]$ screen
```

- b. セッションが中止された場合、インスタンスにログインし直し、利用できる画面を表示します。

```
[ec2-user ~]$ screen -ls
There is a screen on:
  17793.pts-0.ip-12-34-56-78 (Detached)
  1 Socket in /var/run/screen/S-ec2-user.
```

- c. screen -r コマンドと前のコマンドのプロセス ID を使用して、画面に再接続します。

```
[ec2-user ~]$ screen -r 17793
```

- d. screen の使用が終わったら、exit コマンドを使用してセッションを閉じます。

```
[ec2-user ~]$ exit  
[screen is terminating]
```

2. yum update コマンドを実行します。オプションで、**--security** フラグを追加すれば、セキュリティ更新のみを適用できます。

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

3. 表示されたパッケージを確認したら、**y** と入力し、Enter を押して更新を承認します。システムのパッケージをすべて更新するには数分かかります。実行中、yum 出力には更新のステータスが表示されます。
4. (オプション) 更新によって最新のパッケージおよびライブラリを使用していることを確実にするためにインスタンスを再起動してください。カーネル更新は再起動が発生するまでロードされません。glibc ライブラリを更新した後も再起動が必要です。サービスを制御する更新の場合は、更新を取得するにはサービスの再起動で十分かもしれません、システムを再起動することで、それ以前のすべてのパッケージとライブラリの更新を確実に完了できます。

Amazon Linux インスタンスの 1 つのパッケージを更新するには

システム全体ではなく、1 つのパッケージ (とその依存関係) を更新するには、この手順を使用します。

1. 更新するパッケージの名前を指定して、yum update コマンドを実行します。

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

2. リストにあるパッケージ情報を確認したら、**y** と入力し、Enter を押して更新を承認します。時折、解決する必要があるパッケージ依存関係ある場合、リストには複数のパッケージがあります。実行中、yum 出力には更新のステータスが表示されます。
3. (オプション) 更新によって最新のパッケージおよびライブラリを使用していることを確実にするためにインスタンスを再起動してください。カーネル更新は再起動が発生するまでロードされません。glibc ライブラリを更新した後も再起動が必要です。サービスを制御する更新の場合は、更新を取得するにはサービスの再起動で十分かもしれません、システムを再起動することで、それ以前のすべてのパッケージとライブラリの更新を確実に完了できます。

## リポジトリの追加

デフォルトで、Amazon Linux インスタンスは、amzn-main と amzn-updates の 2 つのリポジトリを有効にして起動します。これらのリポジトリには、Amazon Web Services が更新するさまざまなパッケージがありますが、別のリポジトリに含まれるパッケージをインストールしたい場合があるかもしれません。

### Important

この情報は Amazon Linux に適用されます。他のディストリビューションの情報については、各ドキュメントを参照してください。

yum で異なるリポジトリからパッケージをインストールには、/etc/yum.conf ファイル、または **repository.repo** ディレクトリにあるお客様の /etc/yum.repos.d ファイルに、リポジトリ情報を追加する必要があります。これは手動で行えますが、ほとんどの yum リポジトリのリポジトリ URL で、独自の **repository.repo** ファイルが提供されています。

既にインストールされている yum レポジトリを調べるには

- 次のコマンドで、インストール済みの yum リポジトリを表示します。

```
[ec2-user ~]$ yum repolist all
```

生成される出力には、インストール済みのリポジトリが一覧表示され、それぞれのステータスが報告されます。有効なリポジトリには、そこに含まれているパッケージの数が表示されます。

yum リポジトリを `/etc/yum.repos.d` に追加するには

1. `.repo` ファイルの場所を見つけます。場所は、追加しているリポジトリによって異なります。この例では、`.repo` ファイルは、`https://www.example.com/repository.repo` にあります。
2. `yum-config-manager` コマンドを使用してリポジトリを追加します。

```
[ec2-user ~]$ sudo yum-config-manager --add-repo https://www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/yum.repos.d/repository.repo
repository.repo                                         | 4.0 kB     00:00
repo saved to /etc/yum.repos.d/repository.repo
```

リポジトリをインストールしたら、次の手順で説明するように有効にする必要があります。

yum リポジトリを `/etc/yum.repos.d` で有効にするには

- `yum-config-manager` フラグを付けて `--enable repository` コマンドを使用します。次のコマンドを使用すると、Fedora プロジェクトの Extra Packages for Enterprise Linux (EPEL) リポジトリが有効になります。デフォルトでは、このリポジトリは Amazon Linux AMI インスタンスの `/etc/yum.repos.d` にありますが、有効になっていません。

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

#### Note

Amazon Linux 2 で EPEL レポジトリを有効にするには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

他のディストリビューションの EPEL リポジトリ (Red Hat や CentOS など) を有効にする方法については、EPEL ドキュメント (<https://fedoraproject.org/wiki/EPEL>) を参照してください。

## Amazon Linux でソフトウェアパッケージを見つける

`yum search` コマンドを使用すると、設定したリポジトリで利用できるパッケージの説明を検索できます。これは特に、インストールするパッケージの正確な名前がわからない場合に便利です。キーワード検索をコマンドに追加します。複数の単語を検索するには、引用符で検索クエリを囲みます。

#### Important

この情報は Amazon Linux に適用されます。その他のディストリビューションの情報については、各ドキュメントを参照してください。

```
[ec2-user ~]$ sudo yum search "find"
```

Amazon Linux 2 の出力例を次に示します。

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
=====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
gedit-plugin-findinfiles.x86_64 : gedit findinfiles plugin
ocaml-findlib-devel.x86_64 : Development files for ocaml-findlib
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
    File::Find
robotfindskitten.x86_64 : A game/zen simulation. You are robot. Your job is to find kitten.
mlocate.x86_64 : An utility for finding files by name
ocaml-findlib.x86_64 : Objective CAML package manager and build helper
perl-Devel-Cycle.noarch : Find memory cycles in objects
perl-Devel-EnforceEncapsulation.noarch : Find access violations to blessed objects
perl-File-Find-Rule-Perl.noarch : Common rules for searching for Perl things
perl-File-HomeDir.noarch : Find your home and other directories on any platform
perl-IPC-Cmd.noarch : Finding and running system commands made easy
perl-Perl-MinimumVersion.noarch : Find a minimum required version of perl for Perl code
texlive-xesearch.noarch : A string finder for XeTeX
valgrind.x86_64 : Tool for finding memory management bugs in programs
valgrind.i686 : Tool for finding memory management bugs in programs
```

Amazon Linux の出力例を次に示します。

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
=====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
    File::Find
perl-Module-Find.noarch : Find and use installed modules in a (sub)category
libpuzzle.i686 : Library to quickly find visually similar images (gif, png, jpg)
libpuzzle.x86_64 : Library to quickly find visually similar images (gif, png, jpg)
mlocate.x86_64 : An utility for finding files by name
```

引用符で囲まれた複数の単語検索クエリは、正確なクエリに一致する結果のみを返します。予想されたパッケージが表示されない場合、キーワードを1つに絞って検索し、結果をスキャンします。キーワードの同義語を試して、検索の幅を広げることもできます。

Amazon Linux 2 および Amazon Linux のパッケージの詳細については、以下を参照してください。

- [パッケージリポジトリ \(p. 168\)](#)
- [Extras Library \(Amazon Linux 2\) \(p. 170\)](#)

## ソフトウェアパッケージのインストール

yum パッケージマネージャは、ソフトウェアをインストールするための優れたツールです。有効になっているすべてのリポジトリからさまざまなソフトウェアパッケージを検索し、ソフトウェアインストールプロセスに伴う依存関係を処理できます。

### Important

この情報は Amazon Linux に適用されます。その他のディストリビューションの情報については、各ドキュメントを参照してください。

リポジトリからパッケージをインストールするには、`yum install package` コマンドを使用します。#### #にはインストールするソフトウェアの名前を置き換えます。たとえば、links テキストベースウェブブラウザをインストールするには、次のコマンドを入力します。

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

また、yum install を使用して、インターネットからダウンロードした RPM パッケージファイルをインストールすることもできます。その場合、リポジトリパッケージ名の代わりに、RPM ファイルのパス名をインストールコマンドに追加します。

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

## ソフトウェアのコンパイル準備

インターネットには事前コンパイルされていないオープンソースのソフトウェアが豊富に存在します。パッケージリポジトリからダウンロードできます。いずれは、そのソースコードから、自分でコンパイルする必要があるソフトウェアパッケージが判明することがあります。システムでソフトウェアのコンパイルを可能にするには、make、gcc、autoconf など、いくつかの開発ツールをインストールする必要があります。

### Important

この情報は Amazon Linux に適用されます。その他のディストリビューションの情報については、各ドキュメントを参照してください。

ソフトウェアのコンパイルはすべての Amazon EC2 インスタンスで必要なタスクではないため、そのようなツールはデフォルトでインストールされていません。ただし、「Development Tools」という名前のパッケージグループで利用でき、yum groupinstall コマンドでインスタンスに簡単に追加されます。

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

ソフトウェアのソースコードパッケージは、多くの場合、tarball という圧縮アーカイブファイルとしてダウンロードできます (<https://github.com/> や <http://sourceforge.net/> などのウェブサイトから)。通常、これらの tarball には .tar.gz というファイル拡張子が付いています。これらのアーカイブは tar コマンドで解凍できます。

```
[ec2-user ~]$ tar -xzf software.tar.gz
```

ソースコードパッケージを解凍したら、ソースコードディレクトリで README ファイルまたは INSTALL ファイルを探します。これらのファイルに、ソースコードのコンパイルとインストールに関する詳細な指示があります。

Amazon Linux パッケージのソースコードを取得するには

Amazon Web Services は、保守管理されているパッケージのソースコードを提供します。yumdownloader --source コマンドを使用して、インストールされているパッケージのソースコードをダウンロードできます。

- yumdownloader --source **package** コマンドを実行して、**package** のソースコードをダウンロードします。たとえば、htop パッケージのソースコードをダウンロードするには、次のコマンドを入力します。

```
[ec2-user ~]$ yumdownloader --source htop
Loaded plugins: priorities, update-motd, upgrade-helper
Enabling amzn-updates-source repository
Enabling amzn-main-source repository
amzn-main-source
| 1.9 kB  00:00:00
amzn-updates-source
| 1.9 kB  00:00:00
(1/2): amzn-updates-source/latest/primary_db
| 52 kB   00:00:00
(2/2): amzn-main-source/latest/primary_db
| 734 kB  00:00:00
```

htop-1.0.1-2.3.amzn1.src.rpm

ソース RPM の場所は、コマンドを実行したディレクトリにあります。

## Linux インスタンスでのユーザーアカウントの管理

各 Linux インスタンスは、デフォルトの Linux システムユーザーアカウントで起動されます。デフォルトのユーザー名は、インスタンスを起動したときに指定された AMI によって決まります。Amazon Linux 2 または Amazon Linux AMI の場合は、ユーザー名は `ec2-user` です。CentOS の場合、ユーザー名は `centos` です。Debian の場合は、ユーザー名は `admin` または `root` です。Fedora の場合は、ユーザー名は `ec2-user` または `fedora` です。RHEL の場合は、ユーザー名は `ec2-user` または `root` のどちらかです。SUSE の場合は、ユーザー名は `ec2-user` または `root` のどちらかです。Ubuntu の場合は、ユーザー名は `ubuntu` です。それ以外の場合で、`ec2-user` および `root` が機能しない場合は、ご利用の AMI プロバイダーに確認してください。

### Note

Linux システムユーザーと AWS Identity and Access Management ( IAM ) ユーザーを混同しないでください。詳細については、『IAM ユーザーガイド』の「[IAM ユーザーとグループ](#)」を参照してください。

### コンテンツ

- [考慮事項 \(p. 559\)](#)
- [ユーザーアカウントを作成する \(p. 559\)](#)
- [ユーザーアカウントの削除 \(p. 561\)](#)

## 考慮事項

デフォルトのユーザーアカウントを使用するのが多くのアプリケーションに適しています。ただし、個人が自分のファイルとワークスペースを持つことができるよう、ユーザーアカウントを追加することを選択できます。さらに、新しいユーザー用にユーザーアカウントを作成することは、デフォルトユーザーのアカウントへのアクセス権を複数のユーザーに（経験のないユーザーも含めて）与えるよりも、はるかに安全です。これはデフォルトのユーザーアカウントが不適切に使用された場合、システムにさまざまな損害を与える可能性があるためです。詳細については、「[EC2 インスタンスの保護のヒント](#)」を参照してください。

Linux システムのユーザーアカウントを使用してユーザーが EC2 インスタンスに SSH アクセスできるようになるには、SSH キーをユーザーと共有する必要があります。または、EC2 Instance Connect を使用して、SSH キーを共有および管理せずにユーザーにアクセスを提供できます。詳細については、「[EC2 Instance Connect を使用して Linux インスタンスに接続する \(p. 511\)](#)」を参照してください。

## ユーザーアカウントを作成する

最初にユーザーアカウントを作成してから、ユーザーがインスタンスに接続してログインできるようにする SSH パブリックキーを追加します。

### ユーザーアカウントを作成するには

1. [新しいキーペアを作成します \(p. 901\)](#)。この `.pem` ファイルは、ユーザーアカウントを作成するユーザーに提供する必要があります。ユーザーがインスタンスに接続するには、このファイルを使用する必要があります。
2. 前のステップで作成したキーペアからパブリックキーを取得します。

```
$ ssh-keygen -y -f /path_to_key_pair/key-pair-name.pem
```

コマンドは、次の例に示すように、パブリックキーを返します。

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCLKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706Vhz2ITxCIh
+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/
d6RJhJOI0iBXrlsLnBITntckij7FbtxJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/
cQk+OFzzQaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi
+z7wB3RbBQoQzd8v7yeb7OzlPnW0yN0qFU0XA246RA8QFYiCNYwi3f05p6KLxEXAMPLE
```

3. インスタンスに接続します。
4. adduser コマンドを使用して、ユーザー アカウントを作成し、システムに追加します (/etc/passwd ファイルにエントリが追加されます)。このコマンドでも、グループが作成され、アカウントのホームディレクトリが作成されます。この例では、ユーザー アカウントは **newuser** という名前になります。
  - Amazon Linux および Amazon Linux 2

```
[ec2-user ~]$ sudo adduser newuser
```

- Ubuntu

--disabled-password パラメータを含めて、パスワードなしでユーザー アカウントを作成します。

```
[ubuntu ~]$ sudo adduser newuser --disabled-password
```

5. 新しいアカウントに切り替えて、作成するディレクトリとファイルが適切な所有権を持つようにします。

```
[ec2-user ~]$ sudo su - newuser
```

シェルセッションが新しいアカウントに切り替わったことを示すために ec2-user から **newuser** に変更するように求められます。

6. ユーザー アカウントに SSH パブリックキーを追加します。以下のサブステップで説明しているように、最初に SSH キーファイル用のディレクトリをユーザーのホームディレクトリに作成し、次にキーファイルを作成して、最後に公開キーをキーファイルに貼り付けます。
  - a. .ssh ホームディレクトリに **newuser** ディレクトリを作成し、そのファイルのアクセス許可を 700 (所有者のみ、読み取り、書き込み、削除が可能) に変更します。

```
[newuser ~]$ mkdir .ssh
```

```
[newuser ~]$ chmod 700 .ssh
```

### Important

厳密なファイル権限がなければ、ユーザーはログインできません。

- b. **authorized\_keys** という名前のファイルを .ssh ディレクトリに作成し、そのファイルのアクセス許可を 600 (所有者のみ、読み取りおよび書き込みが可能) に変更します。

```
[newuser ~]$ touch .ssh/authorized_keys
```

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

### Important

厳密なファイル権限がなければ、ユーザーはログインできません。

- c. お好みのテキストエディタ(例: vim や nano)で、`authorized_keys` ファイルを開きます。

```
[newuser ~]$ nano .ssh/authorized_keys
```

ステップ 2 で取得したパブリックキーをファイルに貼り付け、変更を保存します。

### Important

パブリックキーは、必ず 1 つの連続した行に貼り付けてください。パブリックキーを複数行に分割することはできません。

これで、`newuser` ファイルに追加したパブリックキーの対であるプライベートキーを使用して、インスタンスの `authorized_keys` アカウントにログインできるようになりました。Linux インスタンスに接続するさまざまな方法の詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。

## ユーザー アカウントの削除

ユーザー アカウントが不要になった場合、今後使用されないようにそのアカウントを削除できます。

システムからユーザー アカウントを削除するには、`userdel` コマンドを使用します。`-r` パラメータを指定すると、ユーザーのホームディレクトリとメールスプールが削除されます。ユーザーのホームディレクトリとメールスプールを維持するには、`-r` パラメータを省略します。

```
[ec2-user ~]$ sudo userdel -r olduser
```

## EC2 インスタンスタイプのプロセッサのステート制御

C ステートはアイドル時のコアのスリープレベルを制御します。C ステートは、C0 (コアがアクティブで、命令を実行している最も浅い状態) から始まる番号が付けられ、C6 (コアの電源がオフになっている最も深いアイドル状態) まで移行します。P ステートはコアに希望するパフォーマンス (CPU 周波数) を制御します。P ステートは、P0 (コアが Intel Turbo Boost Technology を使用して可能であれば周波数を上げることができる最高パフォーマンスの設定) から始まる番号が付けられ、P1 (最大限のベースライン周波数をリクエストする P ステート) から P15 (最小限の周波数) まで移行します。

次のインスタンスタイプにより、オペレーティングシステムがプロセッサの C ステートと P ステートを制御できるようになります。

- 汎用: m4.10xlarge | m4.16xlarge | m5.metal | m5d.metal
- コンピューティングの最適化: c4.8xlarge | c5.metal | c5n.metal
- メモリの最適化: r4.8xlarge | r4.16xlarge | r5.metal | r5d.metal | u-6tb1.metal | u-9tb1.metal | u-12tb1.metal | x1.16xlarge | x1.32xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge | z1d.metal
- ストレージの最適化: d2.8xlarge | i3.8xlarge | i3.16xlarge | i3.metal | i3en.metal | h1.8xlarge | h1.16xlarge
- 高速コンピューティング: f1.16xlarge | g3.16xlarge | p2.16xlarge | p3.16xlarge

次のインスタンスタイプにより、オペレーティングシステムがプロセッサの C ステートを制御できるようになります。

- 汎用: m5.12xlarge | m5.24xlarge | m5d.12xlarge | m5d.24xlarge | m5n.12xlarge | m5n.24xlarge | m5dn.12xlarge | m5dn.24xlarge
- コンピューティングの最適化: c5.9xlarge | c5.12xlarge | c5.18xlarge | c5.24xlarge | c5d.9xlarge | c5d.12xlarge | c5d.18xlarge | c5d.24xlarge | c5n.9xlarge | c5n.18xlarge
- メモリの最適化: r5.12xlarge | r5.24xlarge | r5d.12xlarge | r5d.24xlarge | r5n.12xlarge | r5n.24xlarge | r5dn.12xlarge | r5dn.24xlarge | z1d.6xlarge | z1d.12xlarge
- ストレージの最適化: i3en.12xlarge | i3en.24xlarge
- 高速コンピューティング: p3dn.24xlarge

プロセッサのパフォーマンスの安定性を向上させたり、レイテンシーを減らしたり、インスタンスを特定のワークロード用に調整するために、C ステートまたは P ステートの設定を変更したいと思う場合があるかもしれません。デフォルトの C ステートおよび P ステートの設定は、ほとんどの作業負荷に対して最適なパフォーマンスを提供します。ただし、アプリケーションにおいて、より高いシングルコアまたはデュアルコアの周波数でレイテンシーを軽減したい場合、またはバースト的な Turbo Boost 周波数よりも低い周波数でより安定したパフォーマンスを維持することを優先する場合、これらのインスタンスで利用可能な C ステートまたは P ステートを試みることを考慮してください。

以下のセクションでは、プロセッサのさまざまなステート設定と、設定の効果をモニタリングする方法について説明します。これらの手順は、Amazon Linux を対象としており、これに適用されますが、バージョン 3.9 以降の Linux カーネルを持つ他の Linux ディストリビューションでも使用できる可能性があります。他の Linux ディストリビューションやプロセッサのステート制御については、システム固有ドキュメントを参照してください。

#### Note

このページの例では、turbostat ユーティリティ (Amazon Linux でデフォルトで使用できる) を使用してプロセッサ周波数と C ステート情報を表示し、stress コマンド (sudo yum install -y stress を実行することによってインストールできる) を使用してワークロードをシミュレートします。出力に C ステート情報が表示されない場合は、コマンド (sudo turbostat --debug stress <options>) に --debug オプションを含めます。

#### コンテンツ

- [最大 Turbo Boost 周波数による最高パフォーマンス \(p. 562\)](#)
- [深い C ステートの制限による高パフォーマンスと低レイテンシー \(p. 563\)](#)
- [変動性が最も低いベースラインパフォーマンス \(p. 565\)](#)

## 最大 Turbo Boost 周波数による最高パフォーマンス

これは、Amazon Linux AMI のデフォルトのプロセッサのステート制御設定であり、ほとんどのワークロードにお勧めします。この設定では、変動性を抑え、最高パフォーマンスを実現します。非アクティブなコアは深いスリープ状態になることができるため、シングルまたはデュアルコアプロセッサが Turbo Boost の潜在能力を最大限に引き出すために必要な発熱量の余裕を実現できます。

次の例は、2 個のコアでアクティブに処理を実行する c4.8xlarge インスタンスが Turbo Boost の最大周波数に到達した状況を示しています。

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [30680] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30680] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
      5.54 3.44 2.90   0   9.18   0.00  85.28   0.00   0.00   0.00   0.00   0.00
 94.04 32.70 54.18  0.00
  0    0    0.12 3.26 2.90   0   3.61   0.00  96.27   0.00   0.00   0.00   0.00
 48.12 18.88 26.02  0.00
```

```

0 0 18 0.12 3.26 2.90 0 3.61
0 1 1 0.12 3.26 2.90 0 4.11 0.00 95.77 0.00
0 1 19 0.13 3.27 2.90 0 4.11
0 2 2 0.13 3.28 2.90 0 4.45 0.00 95.42 0.00
0 2 20 0.11 3.27 2.90 0 4.47
0 3 3 0.05 3.42 2.90 0 99.91 0.00 0.05 0.00
0 3 21 97.84 3.45 2.90 0 2.11
...
1 1 10 0.06 3.33 2.90 0 99.88 0.01 0.06 0.00
1 1 28 97.61 3.44 2.90 0 2.32
...
10.002556 sec

```

この例では、vCPU 21 と 28 が最大 Turbo Boost 周波数で実行され、他のコアは c6 スリープ状態になることで電力を節約し、実行中のコアに電力と発熱の余裕を持たせています。vCPU 3 と 10 (それぞれ vCPU 21 および 28 とプロセッサコアを共有する) は c1 ステートであり、命令を待っています。

以下の例では、18 個のコアはすべてアクティブに処理を実行しているため、Turbo Boost の最大周波数のための余裕はありませんが、すべてのコアが「all core Turbo Boost」の速度である 3.2 GHz で実行されています。

```

[ec2-user ~]$ sudo turbostat stress -c 36 -t 10
stress: info: [30685] dispatching hogs: 36 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30685] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
99.27 3.20 2.90 0 0.26 0.00 0.47 0.00 0.00 0.00 0.00 0.00 0.00
228.59 31.33 199.26 0.00
0 0 0 99.08 3.20 2.90 0 0.27 0.01 0.64 0.00 0.00 0.00 0.00
114.69 18.55 99.32 0.00
0 0 18 98.74 3.20 2.90 0 0.62
0 1 1 99.14 3.20 2.90 0 0.09 0.00 0.76 0.00
0 1 19 98.75 3.20 2.90 0 0.49
0 2 2 99.07 3.20 2.90 0 0.10 0.02 0.81 0.00
0 2 20 98.73 3.20 2.90 0 0.44
0 3 3 99.02 3.20 2.90 0 0.24 0.00 0.74 0.00
0 3 21 99.13 3.20 2.90 0 0.13
0 4 4 99.26 3.20 2.90 0 0.09 0.00 0.65 0.00
0 4 22 98.68 3.20 2.90 0 0.67
0 5 5 99.19 3.20 2.90 0 0.08 0.00 0.73 0.00
0 5 23 98.58 3.20 2.90 0 0.69
0 6 6 99.01 3.20 2.90 0 0.11 0.00 0.89 0.00
0 6 24 98.72 3.20 2.90 0 0.39
...

```

## 深い C ステートの制限による高パフォーマンスと低レイテンシー

C ステートは非アクティブ時のコアのスリープレベルを制御します。C ステートを制御して、システムのレイテンシーとパフォーマンスを調整することができます。コアをスリープ状態にするには時間がかかります。また、スリープ状態のコアによって、別のコアが高い周波数で動作するための余裕が生まれますが、そのスリープ状態にあるコアが再び稼働し処理を実行するのにも時間がかかります。たとえば、ネットワークパケットの中断を処理するように割り当てられたコアがスリープ状態である場合、その中断の処理に遅延が生じる可能性があります。より深い C ステートを使用しないようにシステムを設定できます。これにより、プロセッサの応答のレイテンシーは減少しますが、他のコアの Turbo Boost 用の余裕も減少します。

深いスリープ状態を無効にする一般的なシナリオとして、Redis データベースアプリケーションがあります。このアプリケーションは、最速のクエリ応答時間を探求するために、データベースをシステムメモリ内に格納します。

### Amazon Linux 2 で深いスリープ状態を制限するには

- 適切なエディタで `/etc/default/grub` ファイルを開きます。

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

- GRUB\_CMDLINE\_LINUX\_DEFAULT 行を編集し、`intel_idle.max_cstate=1` オプションを追加して、アイドル状態のコアの最も深い C ステートとして `C1` を設定します。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0  
biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1"  
GRUB_TIMEOUT=0
```

- ファイルを保存し、エディタを終了します。
- 起動設定を再構築するには、次のコマンドを実行します。

```
[ec2-user ~]$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

- 新しい kernel オプションを有効にするためにインスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

### Amazon Linux AMI で深いスリープ状態を制限するには

- 適切なエディタで `/boot/grub/grub.conf` ファイルを開きます。

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

- 最初のエントリの `kernel` 行を編集し、`intel_idle.max_cstate=1` = オプションを追加して、アイドル状態のコアの最も深いステートとして `C1C` を設定します。

```
# created by imagebuilder  
default=0  
timeout=1  
hiddenmenu  
  
title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)  
root (hd0,0)  
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0  
intel_idle.max_cstate=1  
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

- ファイルを保存し、エディタを終了します。
- 新しい kernel オプションを有効にするためにインスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

次の例では、2つのコアが「all core Turbo Boost」コア周波数でアクティブに処理を実行している `c4.8xlarge` インスタンスを示します。

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10  
stress: info: [5322] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd  
stress: info: [5322] successful run completed in 10s  
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7  
Pkg_W RAM_W PKG_% RAM_%
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
プロセッサのステート制御

5.56	3.20	2.90	0	94.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
131.90	31.11	199.47	0.00									
0	0	0	0.03	2.08	2.90	0	99.97	0.00	0.00	0.00	0.00	0.00
67.23	17.11	99.76	0.00									
0	0	18	0.01	1.93	2.90	0	99.99					
0	1	1	0.02	1.96	2.90	0	99.98	0.00	0.00	0.00		
0	1	19	99.70	3.20	2.90	0	0.30					
...												
1	1	10	0.02	1.97	2.90	0	99.98	0.00	0.00	0.00		
1	1	28	99.67	3.20	2.90	0	0.33					
1	2	11	0.04	2.63	2.90	0	99.96	0.00	0.00	0.00		
1	2	29	0.02	2.11	2.90	0	99.98					
...												

この例では、vCPU 19 および 28 のコアは 3.2 GHz で実行中であり、その他のコアは C1 C ステートで、命令を待機しています。稼働中のコアは Turbo Boost の最大周波数には到達していませんが、非アクティブなコアはより深い C6 C ステートにある場合と比べて、新しいリクエストに迅速に応答します。

## 変動性が最も低いベースラインパフォーマンス

P ステートによってプロセッサの周波数の変動性を抑制することができます。P ステートはコアに希望するパフォーマンス (CPU 周波数) を制御します。ほとんどのワークロードでは、Turbo Boost をリクエストする、P0 でパフォーマンスが向上します。ただし、Turbo Boost 周波数が有効であるときに発生する可能性があるバースト的なパフォーマンスではなく、安定したパフォーマンスになるようにシステムを調整することもできます。

Intel Advanced Vector Extensions (AVX または AVX2) のワークロードは低い周波数でもパフォーマンスに優れ、AVX 命令はより多くの処理能力を使用できます。Turbo Boost を無効にして、プロセッサをより低い周波数で実行すると、使用される処理能力が抑えられ、より安定した速度が維持されます。インスタンスの設定の最適化や AVX のワークロードの詳細については、<http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/performance-xeon-e5-v3-advanced-vector-extensions-paper.pdf> を参照してください。

このセクションでは、深いスリープ状態を制限し、(P1 P ステートをリクエストすることにより) Turbo Boost を無効にすることで、これらのタイプのワークロードに対して、低レイテンシーを提供し、プロセッサ速度の変動性を最低限に抑える方法を説明します。

Amazon Linux 2 で深いスリープ状態を制限し、Turbo Boost を無効にするには

1. 適切なエディタで /etc/default/grub ファイルを開きます。

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

2. GRUB\_CMDLINE\_LINUX\_DEFAULT 行を編集し、intel\_idle.max\_cstate=1 オプションを追加して、アイドル状態のコアの最も深い C ステートとして C1 を設定します。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0  
biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1"  
GRUB_TIMEOUT=0
```

3. ファイルを保存し、エディタを終了します。
4. 起動設定を再構築するには、次のコマンドを実行します。

```
[ec2-user ~]$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. 新しい kernel オプションを有効にするためにインスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

6. P1 P ステートによってプロセッサ速度の変動性を抑える必要がある場合は、次のコマンドを実行して Turbo Boost を無効にします。

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

7. ワークロードが終了したら、次のコマンドで Turbo Boost を再度有効にすることができます。

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

Amazon Linux AMI で深いスリープ状態を制限し、Turbo Boost を無効にするには

1. 適切なエディタで /boot/grub/grub.conf ファイルを開きます。

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. 最初のエントリの kernel 行を編集し、intel\_idle.max\_cstate=1 = オプションを追加して、アイドル状態のコアの最も深いステートとして C1C を設定します。

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
    intel_idle.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

3. ファイルを保存し、エディタを終了します。
4. 新しい kernel オプションを有効にするためにインスタンスを再起動します。

```
[ec2-user ~]$ sudo reboot
```

5. P1 P ステートによってプロセッサ速度の変動性を抑える必要がある場合は、次のコマンドを実行して Turbo Boost を無効にします。

```
[ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

6. ワークロードが終了したら、次のコマンドで Turbo Boost を再度有効にすることができます。

```
[ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

次の例では、2 つの vCPU が、Turbo Boost を使用せずに、ベースラインコア周波数でアクティブに処理を行っている c4.8xlarge インスタンスを示します。

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5389] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5389] successful run completed in 10s
pk cor CPU %c0 GHz TSC SMI %c1 %c3 %c6 %c7 %pc2 %pc3 %pc6 %pc7
Pkg_W RAM_W PKG_% RAM_%
      5.59 2.90 2.90   0 94.41  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
128.48 33.54 200.00 0.00
0   0   0   0.04 2.90 2.90   0 99.96  0.00  0.00  0.00  0.00  0.00  0.00
65.33 19.02 100.00 0.00
0   0   18  0.04 2.90 2.90   0 99.96
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
時刻の設定

```
0 1 1 0.05 2.90 2.90 0 99.95 0.00 0.00 0.00
0 1 19 0.04 2.90 2.90 0 99.96
0 2 2 0.04 2.90 2.90 0 99.96 0.00 0.00 0.00
0 2 20 0.04 2.90 2.90 0 99.96
0 3 3 0.05 2.90 2.90 0 99.95 0.00 0.00 0.00
0 3 21 99.95 2.90 2.90 0 0.05
...
1 1 28 99.92 2.90 2.90 0 0.08
1 2 11 0.06 2.90 2.90 0 99.94 0.00 0.00 0.00
1 2 29 0.05 2.90 2.90 0 99.95
```

vCPU 21 および 28 のコアは、ベースラインプロセッサ速度の 2.9 GHz でアクティブに処理を実行し、すべての非アクティブなコアも、c1 C ステートのベースライン速度で実行され、すぐに命令を受け付けることができます。

## Linux インスタンスの時刻の設定

サーバータスクとプロセスの多くには、一貫して正確な時間参照が不可欠です。システムログのほとんどにタイムスタンプが含まれています。これを利用すれば、問題が発生した時刻とイベントの発生順序を判断できます。AWS CLI、または AWS SDK を使用してインスタンスからリクエストを行う場合、これらのツールによって自動的にリクエストに署名されます。インスタンスの日時が正しく設定されていない場合、署名の日付がリクエストの日付と一致しないことがあります。その場合は AWS によってリクエストが却下されます。

Amazon では、Amazon Time Sync Service を提供します。このサービスはすべての EC2 インスタンスからアクセスでき、その他の AWS サービスにも利用されます。このサービスは、各リージョンで衛星接続された原子基準クロックを使用し、ネットワークタイムプロトコル (NTP) を通じて世界標準時 (UTC) の現在の正確な現在時刻を表示します。Amazon Time Sync Service は、UTC に追加されたうるう秒を自動的に均一化します。

Amazon Time Sync Service は、VPC で実行されているすべてのインスタンスの 169.254.169.123 IP アドレスで NTP を介して利用できます。インスタンスはインターネットにアクセスする必要はなく、アクセスを許可するためにセキュリティグループまたはネットワーク ACL ルールを設定する必要はありません。最新バージョンの Amazon Linux 2 と Amazon Linux AMI はデフォルトで Amazon Time Sync Service と同期します。

chrony クライアントを使用して、インスタンスに Amazon Time Sync Service を設定するには、次の手順を実行します。または、外部 NTP ソースを使用できます。NTP やパブリックな時刻ソースの詳細については、<http://www.ntp.org/> を参照してください。インスタンスは、インターネットにアクセスして外部 NTP 時刻ソースを動作させる必要があります。

## Amazon Linux AMI で Amazon Time Sync Service を設定する

### Note

Amazon Linux 2 では、デフォルトの chrony 設定で Amazon Time Sync Service の IP アドレスを使用するように設定されています。

Amazon Linux AMI を使用して、Amazon Time Sync Service のサーバーエントリを追加するには、chrony 設定ファイルを編集する必要があります。

Amazon Time Sync Service を使用してインスタンスを設定するには

1. インスタンスに接続し、NTP サービスをアンインストールします。

```
[ec2-user ~]$ sudo yum erase 'ntp*'
```

2. chrony パッケージをインストールします。

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

- 任意のテキストエディタ(例: vim または nano など)を使って /etc/chrony.conf ファイルを開きます。ファイルに次の行が含まれていることを確認します:

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

この行が存在する場合は、Amazon Time Sync Service が既に設定されており、次の手順に進むことができます。そうでない場合は、すでにファイルに存在する他の server または pool ステートメントの後に行を追加し、変更を保存します。

- chrony デーモン(chronyd) を再起動します。

```
[ec2-user ~]$ sudo service chronyd restart
```

```
Starting chronyd:
```

```
[ OK ]
```

#### Note

RHEL と CentOS (バージョン 6 まで) では、サービス名は chrony ではなく chronyd です。

- chkconfig コマンドを使用して、システムがブートするたびに chronyd が起動するように設定します。

```
[ec2-user ~]$ sudo chkconfig chronyd on
```

- chrony が 169.254.169.123 IP アドレスを使用して時刻を同期させていることを確認します。

```
[ec2-user ~]$ chronyc sources -v
```

```
210 Number of sources = 7

-- Source mode '^' = server, '=' = peer, '#' = local clock.
/- Source state '*' = current synced, '+' = combined , '-' = not combined,
| / '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                               .- xxxx [ yyyy ] +/- zzzz
||       Reachability register (octal) -.          | xxxx = adjusted offset,
||       Log2(Polling interval) --.           |      yyyy = measured offset,
||                                         \ |      zzzz = estimated error.
||                                         | |
MS Name/IP address      Stratum Poll Reach LastRx Last sample
=====
^* 169.254.169.123        3   6    17   43    -30us[ -226us] +/-  287us
^- ec2-12-34-231-12.eu-west>  2   6    17   43    -388us[ -388us] +/-   11ms
^- tshirt.heanet.ie       1   6    17   44    +178us[ +25us] +/- 1959us
^? tbag.heanet.ie        0   6     0    -     +0ns[ +0ns] +/-    0ns
^? bray.walcz.net         0   6     0    -     +0ns[ +0ns] +/-    0ns
^? 2a05:d018:c43:e312:ce77:> 0   6     0    -     +0ns[ +0ns] +/-    0ns
^? 2a05:d018:dab:2701:b70:b> 0   6     0    -     +0ns[ +0ns] +/-    0ns
```

返される出力では、^\* が優先時刻ソースを示します。

- chrony で報告された時刻同期メトリクスを確認します。

```
[ec2-user ~]$ chronyc tracking
```

```
Reference ID      : A9FEA97B (169.254.169.123)
Stratum          : 4
Ref time (UTC)   : Wed Nov 22 13:18:34 2017
System time      : 0.000000626 seconds slow of NTP time
Last offset      : +0.002852759 seconds
RMS offset       : 0.002852759 seconds
Frequency        : 1.187 ppm fast
Residual freq    : +0.020 ppm
Skew              : 24.388 ppm
Root delay       : 0.000504752 seconds
Root dispersion  : 0.001112565 seconds
Update interval  : 64.4 seconds
Leap status       : Normal
```

## Ubuntu で Amazon Time Sync Service を設定する

Amazon Time Sync Service のサーバーエントリを追加するには、`chrony` 設定ファイルを編集する必要があります。

Amazon Time Sync Service を使用してインスタンスを設定するには

1. インスタンスに接続し、`apt` を使用して `chrony` パッケージをインストールします。

```
ubuntu:~$ sudo apt install chrony
```

### Note

必要に応じて、`sudo apt update` を実行してインスタンスを最初に更新します。

2. 任意のテキストエディタ（例：`vim` または `nano` など）を使って `/etc/chrony/chrony.conf` ファイルを開きます。ファイルに既に存在する他の `server` ステートメントや `pool` ステートメントの前に次の行を追加し、変更を保存します。

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

3. `chrony` サービスを再起動します。

```
ubuntu:~$ sudo /etc/init.d/chrony restart
```

```
[ ok ] Restarting chrony (via systemctl): chrony.service.
```

4. `chrony` が 169.254.169.123 IP アドレスを使用して時刻を同期させていることを確認します。

```
ubuntu:~$ chronyc sources -v
```

```
210 Number of sources = 7

--- Source mode '^' = server, '=' = peer, '#' = local clock.
/ .- Source state '*' = current synced, '+' = combined , '-' = not combined,
| / '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||          Reachability register (octal) -.           |     xxxx [ yyyy ] +/- zzzz
||          Log2(Polling interval) --.                 |     xxxx = adjusted offset,
||                                         \             |     yyyy = measured offset,
||                                         |             |     zzzz = estimated error.
||                                         |             \
MS Name/IP address          Stratum Poll Reach LastRx Last sample
```

```
=====
^* 169.254.169.123          3   6    17    12    +15us[  +57us] +/- 320us
^- tbag.heanet.ie           1   6    17    13   -3488us[-3446us] +/- 1779us
^- ec2-12-34-231-12.eu-west- 2   6    17    13    +893us[ +935us] +/- 7710us
^? 2a05:d018:c43:e312:ce77:6 0   6     0   10y    +0ns[ +0ns] +/- 0ns
^? 2a05:d018:d34:9000:d8c6:5 0   6     0   10y    +0ns[ +0ns] +/- 0ns
^? tshirt.heanet.ie         0   6     0   10y    +0ns[ +0ns] +/- 0ns
^? bray.walcz.net           0   6     0   10y    +0ns[ +0ns] +/- 0ns
```

返される出力では、^\* が優先時刻ソースを示します。

5. chrony で報告された時刻同期メトリクスを確認します。

```
ubuntu:~$ chronyc tracking
```

```
Reference ID      : 169.254.169.123 (169.254.169.123)
Stratum          : 4
Ref time (UTC)   : Wed Nov 29 07:41:57 2017
System time       : 0.0000000011 seconds slow of NTP time
Last offset      : +0.000041659 seconds
RMS offset       : 0.000041659 seconds
Frequency        : 10.141 ppm slow
Residual freq    : +7.557 ppm
Skew              : 2.329 ppm
Root delay       : 0.0000544 seconds
Root dispersion  : 0.000631 seconds
Update interval  : 2.0 seconds
Leap status       : Normal
```

## SUSE Linux で Amazon Time Sync Service を設定する

「<https://software.opensuse.org/package/chrony>」から Chrony をインストールします。

任意のテキストエディタ(例: vim または nano など)を使って /etc/chrony.conf ファイルを開きます。ファイルに次の行が含まれていることを確認します。

```
server 169.254.169.123 prefer iburst minpoll 4 maxpoll 4
```

この行が存在しない場合は追加します。他のサーバーまたはプールの行はすべてコメントアウトします。yast を開き、chrony サービスを有効にします。

## Amazon Linux のタイムゾーンの変更

Amazon Linux インスタンスは、デフォルトでは UTC (協定世界時間) 時間帯に設定されていますが、インスタンスの時間を現地時間またはネットワークの別の時間帯に変更したい場合があるでしょう。

### Important

この情報は Amazon Linux に適用されます。その他のディストリビューションの情報については、各ドキュメントを参照してください。

インスタンスの時間帯を変更するには

1. インスタンスで使用する時間帯を特定します。/usr/share/zoneinfo ディレクトリには、タイムゾーンデータファイルの階層が含まれています。その場所でディレクトリ構造を閲覧し、お客様の時間帯のファイルを見つけます。

```
[ec2-user ~]$ ls /usr/share/zoneinfo
```

Africa	Chile	GB	Indian	Mideast	posixrules	US
America	CST6CDT	GB-Eire	Iran	MST	PRC	UTC
Antarctica	Cuba	GMT	iso3166.tab	MST7MDT	PST8PDT	WET
Arctic	EET	GMT0	Israel	Navajo	right	W-SU
...						

この場所にある一部のエントリはディレクトリです (America など)。そのディレクトリには、特定の都市の時間帯ファイルが含まれています。インスタンスに使用する都市 (またはお客様の時間帯と同じ都市) を見つけます。この例では、口サンゼルスのタイムゾーンファイル (/usr/share/zoneinfo/America/Los\_Angeles) を使用できます。

2. 新しいタイムゾーンを適用した /etc/sysconfig/clock ファイルを更新します。
  - a. お好みのテキストエディタ (vim や nano など) で、/etc/sysconfig/clock ファイルを開きます。エディタのコマンドで sudo を使用する必要があります。/etc/sysconfig/clock は root が所有するためです。
  - b. ZONE エントリを特定し、タイムゾーンファイルに変更します (パスの /usr/share/zoneinfo セクションは省略します)。たとえば、口サンゼルスの時間帯に変更するには、ZONE エントリを次のように変更します。

```
ZONE="America/Los_Angeles"
```

#### Note

UTC=true エントリを別の値に変更しないでください。このエントリは、ハードウェアクロックに使用されるため、インスタンスで別のタイムゾーンを設定する場合は調整する必要はありません。

- c. ファイルを保存し、テキストエディタを終了します。
3. インスタンスが現地時間情報を参照するとき、タイムゾーンファイルを見つけられるように、/etc/localtime とタイムゾーンファイルの間にシンボリックリンクを作成します。

```
[ec2-user ~]$ sudo ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

4. システムを再起動し、すべてのサーバーとアプリケーションで新しい時間帯情報を取得します。

```
[ec2-user ~]$ sudo reboot
```

## CPU オプションの最適化

Amazon EC2 インスタンスは、単一の Intel Xeon CPU コアで同時に複数のスレッドを実行できるマルチスレッドをサポートしています。各スレッドは、インスタンスの仮想 CPU (vCPU) として表されます。インスタンスには、インスタンスタイプによって異なるデフォルト数の CPU コアがあります。たとえば、m5.xlarge インスタンスタイプには 2 つの CPU コアがあり、デフォルトでは各コアごとに 2 つのスレッドの合計で 4 つの vCPU があります。—

#### Note

各 vCPU は、T2 インスタンスの場合を除き、CPU コアのスレッドです。

ほとんどの場合、ワークロードに適したメモリと vCPU 数を組み合わせた Amazon EC2 インスタンスタイプがあります。ただし、特定のワークロードまたはビジネスのニーズに合わせて、インスタンスを最適化するために以下の CPU オプションを指定できます。

- CPU コア数: インスタンスの CPU コア数をカスタマイズできます。これによって、大量のメモリを使用するワークロード用に十分な RAM 量がありながら、少ない CPU コアのインスタンスのソフトウェアのライセンスコストを最適化することにつながります。

- コア別のスレッド: マルチスレッドを無効化するには、CPU コアごとに 1 つのスレッドを指定できます。高性能コンピューティング (HPC) のワークロードのような特定のワークロードでこれを使用できます。

この CPU オプションはインスタンスの起動時に指定できます。CPU オプションの指定には、追加あるいは割引課金はありません。デフォルト CPU オプションで起動したインスタンスと同じように課金されます。

#### コンテンツ

- [CPU オプションを指定するためのルール \(p. 572\)](#)
- [インスタンスタイプあたりの CPU コアごとの CPU コアとスレッド \(p. 572\)](#)
- [インスタンスの CPU オプションを変更する \(p. 581\)](#)
- [インスタンスの CPU オプションを表示する \(p. 582\)](#)

## CPU オプションを指定するためのルール

インスタンスで CPU オプションを指定するには、次のルールに注意してください。

- CPU オプションはインスタンスの起動時のみ指定でき、起動後には変更できません。
- インスタンスを起動するときに、CPU コア数およびコアごとのスレッドの両方をリクエストで指定する必要があります。リクエスト例については、「[インスタンスの CPU オプションを変更する \(p. 581\)](#)」を参照してください。
- インスタンスの vCPU の数は、コア別のスレッドで乗算した CPU コアの数です。vCPU のカスタム数を指定するには、インスタンスタイプで CPU およびコア別のスレッドの有効な数を指定する必要があります。インスタンスのデフォルト vCPU の数を超えることはできません。詳細については、「[インスタンスタイプあたりの CPU コアごとの CPU コアとスレッド \(p. 572\)](#)」を参照してください。
- マルチスレッドを無効にするには、コアごとに 1 つのスレッドを指定します。
- 既存のインスタンスの [インスタンスタイプを変更する \(p. 267\)](#) 場合、CPU オプションは自動的に新しいインスタンスタイプのデフォルト CPU オプションに変更されます。
- 指定された CPU オプションは、インスタンスの停止、開始あるいは再起動後にも保持されます。

## インスタンスタイプあたりの CPU コアごとの CPU コアとスレッド

次の表では、CPU オプションの指定をサポートしているインスタンスタイプを一覧表示しています。この表では、各タイプごとにデフォルトおよびサポートされる CPU コアの数とコアごとのスレッドを示しています。

### 高速コンピューティングインスタンス

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
f1.2xlarge	8	4	2	1、2、3、4	1、2
f1.4xlarge	16	8	2	1、2、3、4、5、6、7、8	
f1.16xlarge	64	32	2	2、4、6、8、10、11、12、14、16、18、20、22、24	
g3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	
g3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13	

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
CPU オプションの最適化

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
g3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
g3s.xlarge	4	2	2	1、2	1、2
g4dn.xlarge	4	2	2	1、2	1、2
g4dn.2xlarge	8	4	2	1、2、3、4	1、2
g4dn.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
g4dn.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13	1、2
g4dn.12xlarge	48	24	2	4、6、8、10、12、14、16、18、20、22、24	1、2
g4dn.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
p2.xlarge	4	2	2	1、2	1、2
p2.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13	1、2
p2.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
p3.2xlarge	8	4	2	1、2、3、4	1、2
p3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13	1、2
p3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
p3dn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

### コンピュート最適化インスタンス

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
c4.large	2	1	2	1	1、2
c4.xlarge	4	2	2	1、2	1、2
c4.2xlarge	8	4	2	1、2、3、4	1、2
c4.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
c4.8xlarge	36	18	2	2、4、6、8、10、12、14、16、18	1、2
c5.large	2	1	2	1	1、2
c5.xlarge	4	2	2	2	1、2

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
CPU オプションの最適化

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
c5.2xlarge	8	4	2	2、4	1、2
c5.4xlarge	16	8	2	2、4、6、8	1、2
c5.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18、20、22、24	14、16、18
c5.12xlarge	48	24	2	4、6、8、10、12、14、16、18、20、22、24	1、2
c5.18xlarge	72	36	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36	1、2
c5.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c5d.large	2	1	2	1	1、2
c5d.xlarge	4	2	2	2	1、2
c5d.2xlarge	8	4	2	2、4	1、2
c5d.4xlarge	16	8	2	2、4、6、8	1、2
c5d.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18、20、22、24	14、16、18
c5d.12xlarge	48	24	2	4、6、8、10、12、14、16、18、20、22、24	1、2
c5d.18xlarge	72	36	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36	1、2
c5d.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2
c5n.large	2	1	2	1	1、2
c5n.xlarge	4	2	2	2	1、2
c5n.2xlarge	8	4	2	2、4	1、2
c5n.4xlarge	16	8	2	2、4、6、8	1、2

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
c5n.9xlarge	36	18	2	2、4、6、8、10、12、14、16、18	
c5n.18xlarge	72	36	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36	1、2

### 汎用インスタンス

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
m5.large	2	1	2	1	1、2
m5.xlarge	4	2	2	2	1、2
m5.2xlarge	8	4	2	2、4	1、2
m5.4xlarge	16	8	2	2、4、6、8	1、2
m5.8xlarge	32	16	2	2、4、6、8、10、12、14、16	
m5.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	
m5.16xlarge	64	32	2	4、6、8、10、12、14、16、18、20、22、24	
m5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1、2
m5a.large	2	1	2	1	1、2
m5a.xlarge	4	2	2	2	1、2
m5a.2xlarge	8	4	2	2、4	1、2
m5a.4xlarge	16	8	2	2、4、6、8	1、2
m5a.8xlarge	32	16	2	2、4、6、8、10、12、14、16	
m5a.12xlarge	48	24	2	6, 12, 18, 24	1、2
m5a.16xlarge	64	32	2	4、6、8、10、12、14、16、18、20、22、24	
m5a.24xlarge	96	48	2	12, 18, 24, 36, 48	1、2
m5ad.large	2	1	2	1	1、2
m5ad.xlarge	4	2	2	2	1、2
m5ad.2xlarge	8	4	2	2、4	1、2

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
CPU オプションの最適化

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
m5ad.4xlarge	16	8	2	2、4、6、8	1、2
m5ad.8xlarge	32	16	2	2、4、6、8、10、12、14、16	
m5ad.12xlarge	48	24	2	6、12、18、24	1、2
m5ad.16xlarge	64	32	2	4、6、8、10、12、14、16、18、20、22、24	
m5ad.24xlarge	96	48	2	12、18、24、36、48	1、2
m5d.large	2	1	2	1	1、2
m5d.xlarge	4	2	2	2	1、2
m5d.2xlarge	8	4	2	2、4	1、2
m5d.4xlarge	16	8	2	2、4、6、8	1、2
m5d.8xlarge	32	16	2	2、4、6、8、10、12、14、16	
m5d.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	
m5d.16xlarge	64	32	2	4、6、8、10、12、14、16、18、20、22、24	
m5d.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、30、32、34、36、38、40、42、44、46、48	1、2
m5dn.large	2	1	2	1	1、2
m5dn.xlarge	4	2	2	2	1、2
m5dn.2xlarge	8	4	2	2、4	1、2
m5dn.4xlarge	16	8	2	2、4、6、8	1、2
m5dn.8xlarge	32	16	2	2、4、6、8、10、12、14、16	
m5dn.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	
m5dn.16xlarge	64	32	2	4、6、8、10、12、14、16、18、20、22、24	
m5dn.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、30、32、34、36、38、40、42、44、46、48	1、2
m5n.large	2	1	2	1	1、2
m5n.xlarge	4	2	2	2	1、2
m5n.2xlarge	8	4	2	2、4	1、2

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
CPU オプションの最適化

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
m5n.4xlarge	16	8	2	2、4、6、8	1、2
m5n.8xlarge	32	16	2	2、4、6、8、10、1122	14、16
m5n.12xlarge	48	24	2	2、4、6、8、10、1122	14、16、18、20、22、24
m5n.16xlarge	64	32	2	4、6、8、10、121、14、16、18、20、22、24	
m5n.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1、2
t3.nano	2	1	2	1	1、2
t3.micro	2	1	2	1	1、2
t3.small	2	1	2	1	1、2
t3.medium	2	1	2	1	1、2
t3.large	2	1	2	1	1、2
t3.xlarge	4	2	2	2	1、2
t3.2xlarge	8	4	2	2、4	1、2
t3a.nano	2	1	2	1	1、2
t3a.micro	2	1	2	1	1、2
t3a.small	2	1	2	1	1、2
t3a.medium	2	1	2	1	1、2
t3a.large	2	1	2	1	1、2
t3a.xlarge	4	2	2	2	1、2
t3a.2xlarge	8	4	2	2、4	1、2

### メモリ最適化インスタンス

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
r4.large	2	1	2	1	1、2
r4.xlarge	4	2	2	1、2	1、2
r4.2xlarge	8	4	2	1、2、3、4	1、2
r4.4xlarge	16	8	2	1、2、3、4、5、6、7、8	

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
CPU オプションの最適化

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
r4.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13	
r4.16xlarge	64	32	2	2、4、6、8、10、11、12、13、14、16、18、20、22、24	
r5.large	2	1	2	1	1、2
r5.xlarge	4	2	2	2	1、2
r5.2xlarge	8	4	2	2、4	1、2
r5.4xlarge	16	8	2	2、4、6、8	1、2
r5.8xlarge	32	16	2	2、4、6、8、10、11、12、14、16	
r5.12xlarge	48	24	2	2、4、6、8、10、11、12、14、16、18、20、22、24	
r5.16xlarge	64	32	2	4、6、8、10、12、14、16、18、20、22、24	
r5.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、30、32、34、36、38、40、42、44、46、48	1、2
r5a.large	2	1	2	1	1、2
r5a.xlarge	4	2	2	2	1、2
r5a.2xlarge	8	4	2	2、4	1、2
r5a.4xlarge	16	8	2	2、4、6、8	1、2
r5a.8xlarge	32	16	2	2、4、6、8、10、11、12、14、16	
r5a.12xlarge	48	24	2	6、12、18、24	1、2
r5a.16xlarge	64	32	2	4、6、8、10、12、14、16、18、20、22、24	
r5a.24xlarge	96	48	2	12、18、24、36、48	1、2
r5ad.large	2	1	2	1	1、2
r5ad.xlarge	4	2	2	2	1、2
r5ad.2xlarge	8	4	2	2、4	1、2
r5ad.4xlarge	16	8	2	2、4、6、8	1、2
r5ad.8xlarge	32	16	2	2、4、6、8、10、11、12、14、16	
r5ad.12xlarge	48	24	2	6、12、18、24	1、2
r5ad.16xlarge	64	32	2	4、6、8、10、12、14、16、18、20、22、24	
r5ad.24xlarge	96	48	2	12、18、24、36、48	1、2

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
CPU オプションの最適化

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
r5d.large	2	1	2	1	1、2
r5d.xlarge	4	2	2	2	1、2
r5d.2xlarge	8	4	2	2、4	1、2
r5d.4xlarge	16	8	2	2、4、6、8	1、2
r5d.8xlarge	32	16	2	2、4、6、8、10、1122 14、16	
r5d.12xlarge	48	24	2	2、4、6、8、10、1122 14、16、18、20、22、24	
r5d.16xlarge	64	32	2	4、6、8、10、121、14、16、18、20、22、24、26	
r5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1、2
r5dn.large	2	1	2	1	1、2
r5dn.xlarge	4	2	2	2	1、2
r5dn.2xlarge	8	4	2	2、4	1、2
r5dn.4xlarge	16	8	2	2、4、6、8	1、2
r5dn.8xlarge	32	16	2	2、4、6、8、10、1122 14、16	
r5dn.12xlarge	48	24	2	2、4、6、8、10、1122 14、16、18、20、22、24	
r5dn.16xlarge	64	32	2	4、6、8、10、121、14、16、18、20、22、24、26	
r5dn.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1、2
r5n.large	2	1	2	1	1、2
r5n.xlarge	4	2	2	2	1、2
r5n.2xlarge	8	4	2	2、4	1、2
r5n.4xlarge	16	8	2	2、4、6、8	1、2
r5n.8xlarge	32	16	2	2、4、6、8、10、1122 14、16	
r5n.12xlarge	48	24	2	2、4、6、8、10、1122 14、16、18、20、22、24	
r5n.16xlarge	64	32	2	4、6、8、10、121、14、16、18、20、22、24、26	

# Amazon Elastic Compute Cloud Linux インスタンス用ユーザーガイド CPU オプションの最適化

インスタンスタイプ	デフォルトvCPU	デフォルトのCPUコア	コアごとのデフォルトのスレッド	CPUコアの有効数	コアごとのスレッドの有効数
r5n.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	36, 40, 44, 48
x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	
x1e.xlarge	4	2	2	1, 2	1, 2
x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	
x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	
x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	
x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48	
z1d.large	2	1	2	1	1, 2
z1d.xlarge	4	2	2	2	1, 2
z1d.2xlarge	8	4	2	2, 4	1, 2
z1d.3xlarge	12	6	2	2, 4, 6	1, 2
z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 11, 12, 14, 16, 18, 20, 22, 24	
z1d.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

## ストレージ最適化インスタンス

インスタンスタイプ	デフォルtvCPU	デフォルトのCPUコア	コアごとのデフォルトのスレッド	CPUコアの有効数	コアごとのスレッドの有効数
d2.xlarge	4	2	2	1、2	1、2
d2.2xlarge	8	4	2	1、2、3、4	1、2
d2.4xlarge	16	8	2	1、2、3、4、5、6、7、8	
d2.8xlarge	36	18	2	2、4、6、8、10、1122	14、16、18
h1.2xlarge	8	4	2	1、2、3、4	1、2
h1.4xlarge	16	8	2	1、2、3、4、5、6、7、8	
h1.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13	

インスタンスタイプ	デフォルト vCPU	デフォルトの CPU コア	コアごとのデフォルトのスレッド	CPU コアの有効数	コアごとのスレッドの有効数
h1.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
i3.large	2	1	2	1	1、2
i3.xlarge	4	2	2	1、2	1、2
i3.2xlarge	8	4	2	1、2、3、4	1、2
i3.4xlarge	16	8	2	1、2、3、4、5、6、7、8	1、2
i3.8xlarge	32	16	2	1、2、3、4、5、6、7、8、9、10、11、12、13	1、2
i3.16xlarge	64	32	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
i3en.large	2	1	2	1	1、2
i3en.xlarge	4	2	2	2	1、2
i3en.2xlarge	8	4	2	2、4	1、2
i3en.3xlarge	12	6	2	2、4、6	1、2
i3en.6xlarge	24	12	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
i3en.12xlarge	48	24	2	2、4、6、8、10、12、14、16、18、20、22、24	1、2
i3en.24xlarge	96	48	2	4、6、8、10、12、14、16、18、20、22、24、26、28、30、32、34、36、38、40、42、44、46、48	1、2

## インスタンスの CPU オプションを変更する

インスタンスの起動時に CPU オプションを指定できます。以下の例は、次の [デフォルト値 \(p. 577\)](#) がある `r4.4xlarge` インスタンスタイプの場合です。

- デフォルトの CPU コア: 8
- コアごとのデフォルトのスレッド: 2
- デフォルト vCPU: 16 (8 × 2)
- CPU コアの有効数: 1、2、3、4、5、6、7、8
- コアごとのスレッドの有効数: 1、2

## マルチスレッドの無効化

マルチスレッドを無効にするには、コアごとに 1 つのスレッドを指定します。

インスタンスの起動時にマルチスレッドを無効にするには (コンソール)

- 「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」の手順に従います。
- [CPU オプション] の [インスタンスの詳細設定] ページで、[CPU オプションを指定] を指定します。

3. [Core count (コア数)] では、必要な CPU コア数を選択します。この例では、r4.4xlarge インスタンスにデフォルトの CPU コア数を指定するには、8 を選択します。
4. マルチスレッドを無効にするには、[Threads per core (コアごとのスレッド)] で、[1] を選択します。
5. ウィザードに従って続行します。[Review Instance Launch (インスタンス作成の確認)] ページでオプションの確認が終了したら、[Launch (起動)] を選択します。詳細については、「[インスタンス起動 ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。

インスタンスの起動時にマルチスレッドを無効にするには (AWS CLI)

`run-instances` AWS CLI コマンドを使用して、--cpu-options パラメータの ThreadsPerCore の 1 の値を指定します。[CoreCount] では、CPU コア数を指定します。この例では、r4.4xlarge インスタンスにデフォルトの CPU コア数を指定するには、8 の値を選択します。

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=8,ThreadsPerCore=1" --key-name MyKeyPair
```

## vCPU のカスタム数を指定するには

インスタンスの CPU コア数とコアあたりのスレッドの数をカスタマイズできます。

インスタンス起動中に vCPU のカスタム数を指定するには (コンソール)

次の例では、6 つの vCPU で r4.4xlarge インスタンスを起動します。

1. 「[インスタンス起動 ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」の手順に従います。
2. [CPU オプション] の [インスタンスの詳細設定] ページで、[CPU オプションを指定] を指定します。
3. 6 つの vCPU を取得するには、次のように、3 つの CPU コアとコアごとに 2 つのスレッドを指定します。
  - [Core count (コア数)] には、[3] を選択します。
  - [Threads per core (コアごとのスレッド)] には、[2] を選択します。
4. ウィザードに従って続行します。[Review Instance Launch (インスタンス作成の確認)] ページでオプションの確認が終了したら、[Launch (起動)] を選択します。詳細については、「[インスタンス起動 ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。

インスタンス起動中に vCPU のカスタム数を指定するには (AWS CLI)

次の例では、6 つの vCPU で r4.4xlarge インスタンスを起動します。

`run-instances` AWS CLI コマンドを使用して、--cpu-options パラメータの CPU コア数およびスレッドの数を指定します。6 つの vCPU には、3 つの CPU コアとコアごとに 2 つのスレッドを指定できます。

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=3,ThreadsPerCore=2" --key-name MyKeyPair
```

また、6 つの CPU コアとコアごとに 1 つのスレッドを指定 (マルチスレッドを無効化) して、6 つの vCPU を取得することもできます。

```
aws ec2 run-instances --image-id ami-1a2b3c4d --instance-type r4.4xlarge --cpu-options "CoreCount=6,ThreadsPerCore=1" --key-name MyKeyPair
```

## インスタンスの CPU オプションを表示する

AWS CLI を使用してインスタンスを記述し、Amazon EC2 コンソールで既存のインスタンスの CPU オプションを表示できます。

### インスタンスの CPU オプションを表示するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 左ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. [説明] を選択し、[vCPU の数] フィールドを表示します。
4. にコア数とコアごとのスレッド数を表示するには、[vCPU の数] フィールド値を選択します。

### インスタンスの CPU オプションを表示するには (AWS CLI)

`describe-instances` コマンドを使用します。

```
aws ec2 describe-instances --instance-ids i-123456789abcde123
```

```
...
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": "ec2-198-51-100-5.eu-central-1.compute.amazonaws.com",
            "State": {
                "Code": 16,
                "Name": "running"
            },
            "EbsOptimized": false,
            "LaunchTime": "2018-05-08T13:40:33.000Z",
            "PublicIpAddress": "198.51.100.5",
            "PrivateIpAddress": "172.31.2.206",
            "ProductCodes": [],
            "VpcId": "vpc-1a2b3c4d",
            "CpuOptions": {
                "CoreCount": 34,
                "ThreadsPerCore": 1
            },
            "StateTransitionReason": "",
            ...
        }
    ]
...
}
```

返される出力の `CoreCount` フィールドは、そのインスタンスのコア数を示しています。`ThreadsPerCore` フィールドは、コア別のスレッド数を示します。

また、インスタンスを接続し、のツール (`lscpu` など) を使用して、インスタンスの CPU 情報を表示することもできます。

インスタンスの終了を含む、インスタンスにおける設定変更の記録、判断、監査、評価のために、AWS Config を使用できます。詳細については、『AWS Config Developer Guide』の「[AWS Config の使用開始](#)」を参照してください。

## Linux インスタンスのホスト名の変更

インスタンスを起動すると、プライベートの内部 IPv4 アドレスの形式のホスト名が割り当てられます。典型的な Amazon EC2 プライベート DNS 名は、`ip-12-34-56-78.us-west-2.compute.internal` のような形式になります。この名前は内部ドメイン、サービス (この例では、`compute`)、リージョン、そしてプライベート IPv4 アドレスで構成されます。インスタンスにログインしたとき、このホスト名の一部がシェルプロンプトで表示されます (`ip-12-34-56-78` など)。Amazon EC2 インスタンスを停止し、再起動するたびに (Elastic IP アドレスを使用していない限り)、パブリック IPv4 アドレスが変わり、パブリック DNS 名、システムホスト名、シェルプロンプトも変わります。

### Important

この情報は Amazon Linux に適用されます。その他のディストリビューションの情報については、各ドキュメントを参照してください。

## システムホスト名の変更

インスタンスの IP アドレスにパブリック DNS 名を登録している場合 (`webserver.mydomain.com` など)、インスタンスがそのドメインに含まれているものとして識別されるように、システムホスト名を設定できます。また、システムホスト名を変更すると、シェルプロンプトも変更され、AWS が提供するホスト名の代わりに、新しいシステムホスト名の最初の部分 (`ip-12-34-56-78` など) が表示されます。パブリック DNS 名を登録していない場合でもホスト名は変更できますが、ポートセグが少し違います。

システムホスト名をパブリック DNS 名に変更するには

パブリック DNS 名を登録している場合、この手順を行います。

1. • Amazon Linux 2 の場合: `hostnamectl` コマンドを使用してホスト名を設定し、完全修飾ドメイン名 (`webserver.mydomain.com`) を反映させます。

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.mydomain.com
```

- Amazon Linux AMI の場合: インスタンスで、お好みのテキストエディタを使用して `/etc/sysconfig/network` 設定ファイルを開き、`HOSTNAME` エントリを変更して、完全修飾ドメイン名 (`webserver.mydomain.com` など) を反映させます。

```
HOSTNAME=webserver.mydomain.com
```

2. インスタンスを再起動し、新しいホスト名を取得します。

```
[ec2-user ~]$ sudo reboot
```

または、Amazon EC2 コンソールを使用して再起動できます ([インスタンス] ページで、[アクション]、[Instance State (インスタンスの状態)]、[再起動] を選択します)。

3. インスタンスにログインして、ホスト名が更新されていることを確認します。メッセージには、新しいホスト名が表示されるはずです (最初の「.」まで)。`hostname` コマンドで完全修飾ドメイン名が表示されます。

```
[ec2-user@webserver ~]$ hostname  
webserver.mydomain.com
```

パブリック DNS 名なしでシステムホスト名を変更するには

1. • Amazon Linux 2 の場合: `hostnamectl` コマンドを使用してホスト名を設定し、必要なシステムホスト名 (`webserver` など) を反映させます。

```
[ec2-user ~]$ sudo hostnamectl set-hostname webserver.localdomain
```

- Amazon Linux AMI の場合: インスタンスで、お好みのテキストエディタで `/etc/sysconfig/network` 設定ファイルを開き、`HOSTNAME` エントリを変更して、希望するシステムホスト名を反映させます (例: `webserver`)。

```
HOSTNAME=webserver.localdomain
```

2. お好みのテキストエディタで `/etc/hosts` ファイルを開き、下の例と一致する `127.0.0.1` で始まるエントリを変更します。ホスト名は自分のホスト名に置換します。

```
127.0.0.1 webserver.localdomain webserver localhost4 localhost4.localdomain4
```

- インスタンスを再起動し、新しいホスト名を取得します。

```
[ec2-user ~]$ sudo reboot
```

または、Amazon EC2 コンソールを使用して再起動できます ([インスタンス] ページで、[アクション]、[Instance State (インスタンスの状態)]、[再起動] を選択します)。

- インスタンスにログインして、ホスト名が更新されていることを確認します。メッセージには、新しいホスト名が表示されるはずです (最初の「.」まで)。hostname コマンドで完全修飾ドメイン名が表示されます。

```
[ec2-user@webserver ~]$ hostname  
webserver.localdomain
```

## ホスト名に影響を与えずにシェルプロンプトを変更する

インスタンスのホスト名を変更せずに、AWS が提供するプライベート名 (`webserver` など) よりも便利なシステム名 (`ip-12-34-56-78` など) を表示させる場合、ホスト名の代わりにシステムニックネームを表示するようにシェルプロンプト設定ファイルを編集できます。

シェルプロンプトをホストニックネームに変更するには

- /etc/profile.d で、`NICKNAME` と呼ばれる環境変数を設定するファイルを作成して、シェルプロンプトに表示する値を設定します。たとえば、システムニックネームを `webserver` に設定するには、次のコマンドを実行します。

```
[ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/profile.d/prompt.sh'
```

- お好みのテキストエディタ (`vim` や `nano` など) で、/etc/bashrc (Red Hat) または /etc/bash.bashrc (Debian/Ubuntu) ファイルを開きます。エディタのコマンドで `sudo` を使用する必要があります。/etc/bashrc および /etc/bash.bashrc は `root` が所有するためです。
- ファイルを編集し、ホスト名の代わりにニックネームを表示するようにシェルプロンプト変数 (`PS1`) を変更します。/etc/bashrc または /etc/bash.bashrc でシェルプロンプトを設定する次の行を見つけます (以下には、コンテキストを示すため前後の行も表示されています。[ "\$PS1" で始まる行を探してください)。

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@\h \w\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

その行の `\h` (`hostname` を表す記号) を `NICKNAME` 変数の値に変更します。

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@\$NICKNAME \w\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

- (オプション) シェルウィンドウのタイトルを新しいニックネームに設定するには、次の手順を完了します。

- a. /etc/sysconfig/bash-prompt-xterm という名前のファイルを作成します。

```
[ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
```

- b. 次のコマンドを使用して、ファイルを実行可能にします。

```
[ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
```

- c. お好みのテキストエディタ (vim や nano など) で、/etc/sysconfig/bash-prompt-xterm ファイルを開きます。エディタのコマンドで sudo を使用する必要があります。/etc/sysconfig/bash-prompt-xterm は root が所有するためです。
- d. 次の行をファイルに追加します。

```
echo -ne "\033]0;${USER}@${NICKNAME}: ${PWD/#$HOME/~}\007"
```

5. ログアウトしてから再度ログインし、新しいニックネーム値を取得します。

## 他の Linux ディストリビューションのホスト名の変更

このページの手順は、Amazon Linux のみで使用するためのものです。他の Linux ディストリビューションの詳細については、各ドキュメントおよび次の記事を参照してください。

- RHEL 7 または CentOS 7 を実行するプライベート Amazon EC2 インスタンスに静的ホスト名を割り当てる方法を教えてください。

## Linux インスタンスの動的な DNS のセットアップ

EC2 インスタンスを起動すると、パブリック IP アドレスとパブリック DNS (ドメイン名システム) 名が割り当てられます。それらを使用してインターネットからインスタンスにアクセスできます。Amazon Web Services ドメインには数多くのホストが存在するため、これらのパブリック名はそれぞれの名前を一意にするために、かなり長くする必要があります。典型的な Amazon EC2 パブリック DNS 名は、ec2-12-34-56-78.us-west-2.compute.amazonaws.com のような形式になります。この名前は Amazon Web Services ドメイン、サービス (この例では、compute)、リージョン、パブリック IP アドレスで構成されます。

動的 DNS サービスはそのドメイン領域内でカスタムの DNS ホスト名を提供します。この名前は覚えやすく、ホストのユースケースとの関連性が高くなっています。また、一部の動的 DNS サービスは無料です。Amazon EC2 では動的 DNS プロバイダを利用できます。また、インスタンスを起動するたびに、パブリック DNS 名に関連付けられている IP アドレスを更新するようにインスタンスを設定できます。プロバイダは数多く存在します。また、プロバイダを選択し、それぞれのプロバイダで名前を登録する方法については本ガイドの範囲外です。

### Important

この情報は Amazon Linux に適用されます。その他のディストリビューションの情報については、各ドキュメントを参照してください。

### Amazon EC2 で動的 DNS を使用するには

1. 動的 DNS サービスプロバイダにサインアップし、そのサービスでパブリック DNS 名を登録します。この手順では、[noip.com/free](#) の無料サービスを例として使用します。
2. 動的 DNS 更新クライアントを設定します。動的 DNS サービスプロバイダを選び、そのサービスでパブリック DNS 名を登録したら、その DNS 名をインスタンスの IP アドレスにポイントします。多くのプロバイダ ([noip.com](#) を含む) では、この操作をウェブサイトのアカウントページから手動で実行で

きるようになります。ただし、ソフトウェア更新クライアントもサポートしています。更新クライアントが動作している EC2 インスタンスでは、シャットダウン後に再起動したときなど、IP アドレスが変わるたびに動的 DNS レコードが更新されます。この例では、noip2 クライアントをインストールします。このクライアントは、「[noip.com](http://noip.com)」が提供するサービスで動作します。

- a. noip2 クライアントにアクセスできるように、Extra Packages for Enterprise Linux (EPEL) リポジトリを有効にします。

Note

Amazon Linux インスタンスは、デフォルトでインストールされる EPEL リポジトリに関する GPG キーとリポジトリ情報を保持しています。ただし、Red Hat インスタンスと CentOS インスタンスについては、epel-release パッケージを最初にインストールしてから、EPEL リポジトリを有効にする必要があります。このパッケージの最新バージョンの詳細およびダウンロードについては、「<https://fedoraproject.org/wiki/EPEL>」を参照してください。

- 複数 Amazon Linux 2:

```
[ec2-user ~]$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

- 複数 Amazon Linux AMI:

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

- b. noip パッケージをインストールします。

```
[ec2-user ~]$ sudo yum install -y noip
```

- c. 設定ファイルを作成します。メッセージが表示されたら、ログインおよびパスワード情報を入力して、その後に続く質問に答え、クライアントを設定します。

```
[ec2-user ~]$ sudo noip2 -C
```

3. noip サービスを有効にします。

- 複数 Amazon Linux 2:

```
[ec2-user ~]$ sudo systemctl enable noip.service
```

- 複数 Amazon Linux AMI:

```
[ec2-user ~]$ sudo chkconfig noip on
```

4. noip サービスを開始します。

- 複数 Amazon Linux 2:

```
[ec2-user ~]$ sudo systemctl start noip.service
```

- 複数 Amazon Linux AMI:

```
[ec2-user ~]$ sudo service noip start
```

このコマンドを使用すると、クライアントが起動します。クライアントは前に作成した設定ファイル (/etc/no-ip2.conf) を読み、選択したパブリック DNS 名の IP アドレスを更新します。

- 更新クライアントが動的 DNS 名に正しい IP アドレスを設定したことを確認します。DNS レコードの更新には数分かかります。その後、この手順で設定したパブリック DNS 名により、SSH を使用してインスタンスに接続します。

## Linux インスタンスでの起動時のコマンドの実行

Amazon EC2 でインスタンスを起動するとき、起動後にそのインスタンスにユーザーデータを渡し、一般的な自動設定タスクを実行したり、スクリプトを実行したりできます。2 つのタイプのユーザーデータを Amazon EC2 に渡すことができます。シェルスクリプトと cloud-init ディレクティブです。また、このデータはプレーンテキスト、ファイル（コマンドラインツールを使用してインスタンスを起動する場合に便利です）、または base64 でエンコードされたテキスト（API コールの場合）として、起動ウィザードに渡すこともできます。

より複雑なオートメーションのシナリオに興味がある場合、AWS CloudFormation や AWS OpsWorks のご利用を検討してください。詳細については、「[AWS CloudFormation ユーザーガイド](#)」および「[AWS OpsWorks ユーザーガイド](#)」を参照してください。

Windows インスタンスの起動時にコマンドを実行する方法については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows インスタンスでの起動時のコマンドの実行](#)」および「[Windows インスタンス設定の管理](#)」を参照してください。

次の例では、「[Amazon Linux 2 に LAMP ウェブサーバーをインストールする \(p. 32\)](#)」のコマンドが、シェルスクリプトと、インスタンスの起動時に実行される一連の cloud-init ディレクティブに変換されています。各例では、次のタスクがユーザーデータにより実行されます。

- ディストリビューションソフトウェアパッケージが更新されます。
- 必要なウェブサーバー、php、mariadb パッケージがインストールされます。
- systemctl を介して httpd サービスが開始され、オンになります。
- ec2-user が apache グループに追加されます。
- ウェブディレクトリとその中に含まれるファイルに対して、適切な所有権とファイル権限が設定されます。
- ウェブサーバーと PHP エンジンをテストするために、シンプルなウェブページが作成されます。

### コンテンツ

- [前提条件 \(p. 588\)](#)
- [ユーザーデータとシェルスクリプト \(p. 589\)](#)
- [ユーザーデータおよびコンソール \(p. 589\)](#)
- [ユーザーデータと cloud-init ディレクティブ \(p. 590\)](#)
- [ユーザーデータと AWS CLI \(p. 592\)](#)

## 前提条件

次の例では、インターネットからアクセス可能なパブリック DNS 名がお客様のインスタンスに設定されているものと仮定しています。詳細については、「[ステップ 1: インスタンスを起動する \(p. 27\)](#)」を参照してください。また、SSH (ポート 22)、HTTP (ポート 80)、HTTPS (ポート 443) 接続を許可するように、セキュリティグループを設定する必要があります。前提条件の詳細については、[Amazon EC2 でのセットアップ \(p. 22\)](#) を参照してください。

また、これらの指示は Amazon Linux 2 での使用を意図しています。他の Linux ディストリビューションの場合、コマンドとディレクティブが動作しないことがあります。cloud-init のサポートなど、その他のディストリビューションについての詳細は、該当するディストリビューションの文書を参照してください。

## ユーザーデータとシェルスクリプト

シェルスクリプトに慣れている場合は、この方法が最も簡単で完全に起動時に指示を送信する方法です。起動時にこれらのタスクを追加すると、インスタンスの起動にかかる時間が増えます。タスクが完了するまでさらに数分待ち、それからユーザースクリプトが正常に完了したことをテストしてください。

### Important

デフォルトでは、ユーザーデータのスクリプトおよび cloud-init ディレクティブは、インスタンスの初回起動時の起動サイクル中にのみ実行されます。インスタンスを再起動するたびにユーザーデータスクリプトと cloud-init ディレクティブが実行されるように設定を更新することができます。詳細については、AWS ナリッジセンターの「[EC2 インスタンスを再起動するたびにユーザーデータを実行する方法](#)」を参照してください。

ユーザーデータのシェルスクリプトは、#! の記号と、スクリプトを読み取るインタープリタのパス（通常は /bin/bash）から始める必要があります。シェルスクリプトの概要については、Linux Documentation Project ([tldp.org](http://tldp.org)) の [BASH Programming HOW-TO](#) を参照してください。

ユーザーデータとして入力されたスクリプトは、root ユーザーとして実行されます。そのため、スクリプトでは sudo コマンドを使用しないでください。作成したファイルはすべて root ルートの所有になることを忘れないでください。ルート以外のユーザーにファイルアクセスを与える場合、スクリプトで権限を適宜変更する必要があります。また、スクリプトはインタラクティブに実行されないため、ユーザーフィードバックを必要とするコマンド (-y フラグのない yum update など) を含めることはできません。

cloud-init 出力ログファイル (/var/log/cloud-init-output.log) でコンソール出力がキャプチャされるため、インスタンスが正常に動作しない場合でも、起動後、簡単にスクリプトをデバッグすることができます。

ユーザーデータスクリプトを処理すると、/var/lib/cloud/instances/*instance-id*/ にコピーされ、実行されます。実行後にスクリプトを削除することはできません。必ず /var/lib/cloud/instances/*instance-id*/ のユーザーデータスクリプトを削除してから、インスタンスに AMI を作成してください。それ以外の場合、スクリプトはこの AMI から起動されたインスタンスのこのディレクトリに存在します。

## ユーザーデータおよびコンソール

インスタンスの起動時のインスタンスユーザーデータを指定できます。インスタンスのルートボリュームが EBS ボリュームの場合は、インスタンスを停止してユーザーデータを更新することもできます。

### 起動時にインスタンスユーザーデータを指定する

AMI からのインスタンスの起動 (p. 449) でインスタンスを起動するための手順を行いますが、その手順で Step 6 (p. 450) に到達したら、シェルスクリプトを [User data] フィールドにコピーして、起動手順を完了します。

以下のスクリプト例では、スクリプトがウェブサーバーを作成し、設定します。

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
```

```
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

インスタンスが起動し、スクリプトのコマンドを実行するまで十分待ち、それからスクリプトが意図したタスクを完了したことを見ることができます。

例では、ウェブブラウザにスクリプトが作成した PHP テストファイルの URL を入力します。この URL は、インスタンスのパブリック DNS アドレスにスラッシュとファイル名を追加したものです。

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

PHP 情報ページが表示されるはずです。PHP 情報ページが表示されない場合、使用しているセキュリティグループに HTTP (ポート 80) トラフィックを許可するルールが含まれていることを確認します。詳細については、「[セキュリティグループへのルールの追加 \(p. 916\)](#)」を参照してください。

(オプション) スクリプトが予定のタスクを完了しなかった場合、あるいはスクリプトがエラーせずにタスクを完了したかを確認するには、/var/log/cloud-init-output.log にある cloud-init 出力ログファイルを調べ、エラーメッセージが出力されていないか探します。

デバッグの詳細情報を取得するには、次のディレクティブを指定して cloud-init データセクションを含む Mime マルチパートアーカイブを作成します。

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

このディレクティブにより、スクリプトから /var/log/cloud-init-output.log にコマンド出力が送信されます。cloud-init データ形式と MIME マルチパートアーカイブの作成方法の詳細については、「[cloud-init Formats](#)」を参照してください。

## インスタンスユーザーデータの表示と更新

インスタンスユーザーデータを変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[Actions]、[Instance State]、[Stop] の順に選択します。

### Warning

インスタンスを停止すると、インスタンスストアボリューム上のデータは消去されます。インスタンスストアボリュームのデータを保持するには、このデータを永続的ストレージに必ずバックアップしてください。

4. 確認を求めるメッセージが表示されたら、[Yes, Stop] を選択します。インスタンスが停止するまで、数分かかる場合があります。
5. インスタンスが選択された状態のまま、[Actions(アクション)] を選択し、[Instance Settings(インスタンス設定)] を選択して、[View/Change User Data(ユーザーデータの表示/変更)] を選択します。インスタンスの実行中はユーザーデータを変更できませんが、表示することはできます。
6. [View/Change User Data] ダイアログボックスで、ユーザーデータを更新し、[Save] を選択します。
7. インスタンスを再起動します。新しいユーザーデータは、再起動後にインスタンス上に表示されますが、ユーザーデータスクリプトは実行されません。

## ユーザーデータと cloud-init ディレクティブ

cloud-init パッケージは、新しい Amazon Linux インスタンスが起動したときに、特定の側面を設定します。具体的には、お客様のプライベートキーでログインできるように、ec2-user の .ssh/

authorized\_keys ファイルを設定します。詳細については、「[cloud-init \(p. 171\)](#)」を参照してください。

構文は異なりますが、渡されたスクリプトと同じ方法で cloud-init ユーザーディレクティブを起動時のインスタンスに渡すことができます。cloud-init の詳細については、<http://cloudinit.readthedocs.org/en/latest/index.html> にアクセスしてください。

**Important**

デフォルトでは、ユーザーデータのスクリプトおよび cloud-init ディレクティブは、インスタンスの初回起動時の起動サイクル中にのみ実行されます。インスタンスを再起動するたびにユーザーデータスクリプトと cloud-init ディレクティブが実行されるように設定を更新することができます。詳細については、AWS ナリッジセンターの「[EC2 インスタンスを再起動するたびにユーザーデータを実行する方法](#)」を参照してください。

起動時にこれらのタスクを追加すると、インスタンスの起動にかかる時間が増えます。タスクが完了するまでさらに数分待ち、それからユーザーデータディレクティブが完了したことをテストしてください。

ユーザーデータで cloud-init ディレクティブをインスタンスに渡すには

1. [AMI からのインスタンスの起動 \(p. 449\)](#) でインスタンスを起動するための手順を行いますが、その手順で [Step 6 \(p. 450\)](#) に到達したら、cloud-init ディレクティブテキストを [User data] フィールドに貼り付け、起動手順を完了します。

以下の例では、ディレクティブが Amazon Linux 2 でウェブサーバーを作成し、設定します。一番上の #cloud-config 行は、cloud-init ディレクティブとしてコマンドを識別するために必要です。

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
- httpd
- mariadb-server

runcmd:
- [ sh, -c, "amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2" ]
- systemctl start httpd
- sudo systemctl enable httpd
- [ sh, -c, "usermod -a -G apache ec2-user" ]
- [ sh, -c, "chown -R ec2-user:apache /var/www" ]
- chmod 2775 /var/www
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, \; ]
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, \; ]
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. インスタンスが起動し、ユーザーデータのディレクティブを実行するまで十分待ち、それから意図したタスクをディレクティブが完了したことを確認します。

例では、ウェブブラウザにディレクティブが作成した PHP テストファイルの URL を入力します。この URL は、インスタンスのパブリック DNS アドレスにスラッシュとファイル名を追加したものです。

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

PHP 情報ページが表示されるはずです。PHP 情報ページが表示されない場合、使用しているセキュリティグループに HTTP (ポート 80) トラフィックを許可するルールが含まれていることを確認します。詳細については、「[セキュリティグループへのルールの追加 \(p. 916\)](#)」を参照してください。

3. (オプション) ディレクティブが予定のタスクを完了しなかった場合、あるいはディレクティブがエラーなしでタスクを完了したかを確認するには、/var/log/cloud-init-output.log にある出力

ログファイルを調べ、エラーメッセージが output されていないか探します。デバッグの詳細情報を取得するには、ディレクティブに次の行を追加します:

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

このディレクティブにより、runcmd 出力が `/var/log/cloud-init-output.log` に送信されます。

## ユーザーデータと AWS CLI

AWS CLI を使用して、インスタンスのユーザーデータを指定、変更、表示することができます。インスタンスのメタデータを使用して、インスタンスからユーザーデータを表示する方法については、「[インスタンスユーザーデータを取得する \(p. 607\)](#)」を参照してください。

Windows では、AWS CLI を使用する代わりに AWS Tools for Windows PowerShell を使用できます。詳細については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[ユーザーデータと Tools for Windows PowerShell](#)」を参照してください。

例: ユーザーデータは、起動時に指定します。

インスタンスの起動時にユーザーデータを指定するには、`run-instances` コマンドと `--user-data` パラメータを使用します。`run-instances` で、AWS CLI はユーザーデータの base64 エンコードを実行します。

次の例は、コマンドラインで文字列としてスクリプトを指定する方法を示しています。

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
--key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234 \
--user-data echo user data
```

次の例は、テキストファイルを使用してスクリプトを指定する方法を示しています。ファイルを指定するには、必ず `file://` プレフィックスを使用してください。

```
aws ec2 run-instances --image-id ami-abcd1234 --count 1 --instance-type m3.medium \
--key-name my-key-pair --subnet-id subnet-abcd1234 --security-group-ids sg-abcd1234 \
--user-data file:///my_script.txt
```

シェルスクリプトを使用したテキストファイルの例を次に示します。

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

例: 停止しているインスタンスのユーザーデータを変更します。

停止したインスタンスのユーザーデータは、`modify-instance-attribute` コマンドを使用して変更できます。`modify-instance-attribute` では、AWS CLI はユーザーデータの base64 エンコードを実行しません。

- Linux コンピュータでは、base64 コマンドを使用してユーザーデータをエンコードします。

```
base64 my_script.txt >my_script_base64.txt
```

- Windows コンピュータでは、certutil コマンドを使用してユーザーデータをエンコードします。このファイルを AWS CLI で使用する前に、最初の (証明書の開始) 行と最後の (証明書の終了) 行を削除する必要があります。

```
certutil -encode my_script.txt my_script_base64.txt
```

```
notepad my_script_base64.txt
```

--attribute および --value パラメータを使用して、エンコードされたテキストファイルを使用してユーザーデータを指定します。ファイルを指定するには、必ず file:// プレフィックスを使用してください。

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData --value file:///my_script_base64.txt
```

例: ユーザーデータの表示

インスタンスのユーザーデータを取得するには、[describe-instance-attribute](#) コマンドを使用します。describe-instance-attribute では、AWS CLI はユーザーデータの base64 デコードを実行しません。

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData
```

ユーザーデータが base64 でエンコードされた出力例を次に示します。

```
{  
    "UserData": {  
        "Value"::  
        "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAtOpzZXJ2aWNlIGH0dHBkIHN0YXJ0CmNoa2NvbmZpZyBodHRwZCBvb...  
    },  
    "InstanceId": "i-1234567890abcdef0"  
}
```

- Linux コンピュータでは、--query オプションを使用してエンコードされたユーザーデータを取得し、base64 コマンドを使用してデコードします。

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData --output text --query "UserData.Value" | base64 --decode
```

- Windows コンピュータでは、--query オプションを使用してコード化されたユーザーデータを取得し、certutil コマンドを使用してコードをデコードします。エンコードされた出力はファイルに保存され、デコードされた出力は別のファイルに保存されることに注意してください。

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData --output text --query "UserData.Value" >my_output.txt  
certutil -decode my_output.txt my_output_decoded.txt  
type my_output_decoded.txt
```

出力例を次に示します。

```
#!/bin/bash  
yum update -y  
service httpd start  
chkconfig httpd on
```

## インスタンスマタデータとユーザーデータ

インスタンスマタデータは、インスタンスに関するデータで、実行中のインスタンスを設定または管理するため使用します。インスタンスマタデータは、ホスト名、イベント、およびセキュリティグループなどのカテゴリに分けられます。

インスタンスマタデータを使用して、インスタンスの起動時に指定したユーザーデータにアクセスすることもできます。たとえば、インスタンスを設定するためにパラメータを指定したり、単純なスクリプトを含めたりすることができます。汎用 AMI をビルドし、ユーザーデータを使って起動時に提供された構成ファイルを変更することができます。たとえば、さまざまな小規模ビジネスを対象としたウェブサーバーを実行する場合に、すべてのサーバーで同じ汎用 AMI を使用し、起動時にユーザーデータで指定した Amazon S3 バケットからコンテンツを取得できます。随時新規顧客を追加するには、顧客のバケットを作成し、そのコンテンツを追加し、ユーザーデータのコードに提供された固有のバケット名を使って AMI を起動します。複数のインスタンスを同時に起動する場合、ユーザーデータはその予約においてすべてのインスタンスで使用可能です。同じリザベーションの一部である各インスタンスには固有の ami-launch-index 番号があるため、実行する操作を制御するコードを書くことができます。たとえば、最初のホストがクラスタの最初のマスターノードとして自身を選択する場合があります。詳しい AMI 起動例については、[例: AMI 作成インデックス値 \(p. 609\)](#) を参照してください。

EC2 インスタンスには、インスタンスの起動時に生成されるインスタンスアイデンティティドキュメントなどの動的データも含まれます。詳細については、「[動的データのカテゴリ \(p. 618\)](#)」を参照してください。

#### Important

インスタンスマタデータおよびユーザーデータにはそのインスタンス自体内からのみアクセスできるものの、データは認証または暗号化手法によって保護されていません。インスタンス、そしてインスタンス上で実行される任意のソフトウェアに対して直接アクセス権がある可能性がある人は、メタデータを表示できます。そのため、パスワードまたは存続期間の長い暗号化キーなどの機密データは、ユーザーデータとして保管しないようにしてください。

#### コンテンツ

- [インスタンスマタデータサービスの構成 \(p. 594\)](#)
- [インスタンスマタデータの取得 \(p. 600\)](#)
- [インスタンスユーザーデータの使用 \(p. 607\)](#)
- [動的データの取得 \(p. 608\)](#)
- [例: AMI 作成インデックス値 \(p. 609\)](#)
- [インスタンスマタデータのカテゴリ \(p. 612\)](#)
- [インスタンスアイデンティティドキュメント \(p. 618\)](#)

## インスタンスマタデータサービスの構成

次のいずれかのメソッドを使って、実行中のインスタンスからインスタンスマタデータにアクセスできます。

- [インスタンスマタデータサービスバージョン 1 \(IMDSv1\) – リクエスト/レスポンスマソッド](#)
- [インスタンスマタデータサービスバージョン 2 \(IMDSv2\) – セッション志向メソッド](#)

デフォルトでは、IMDSv1 または IMDSv2 のいずれか、あるいは両方を使用できます。インスタンスマタデータサービスは、所定のリクエストについて、IMDSv2 に固有の PUT または GET ヘッダーがそのリクエストに存在するかどうかによって、IMDSv1 と IMDSv2 リクエストを区別します。

各インスタンスのインスタンスマタデータサービスを、ローカルコードまたはユーザーが IMDSv2 を使用しなければいけないように構成できます。IMDSv2 を使用しなければならないように指定すると、IMDSv1 はもう機能しなくなります。詳細については、「[インスタンスマタデータオプションの構成 \(p. 597\)](#)」を参照してください。

## インスタンスマタデータサービスバージョン 2 の仕組み

IMDSv2 は、セッション志向リクエストを使用します。セッション志向リクエストを使用して、セッション期間 (1 秒 ~ 6 時間) を定義するセッショントークンを作成します。指定した期間中、それ以降のリクエス

トに同じセッショントークンを使用できます。指定した期間が期限切れになった後、将来のリクエストに使用する新しいセッショントークンを作成する必要があります。

次の例では、LinuxシェルスクリプトとIMDSv2を使って、最上位インスタンスマタデータアイテムを取得しています。コマンド例:

- PUTリクエストを使って、6 時間 (21,600 秒) のセッショントークンを作成する
- セッショントークンヘッダーをTOKENという名前の変数に保管する
- トークンを使って最上位メタデータアイテムをリクエストする

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`\&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

トークンを作成した後、期限切れになるまで再使用することができます。次のコマンド例では、インスタンスの起動にAMIのIDが使用されていますが、前の例で\$TOKENに保管されたトークンが再使用されています。

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/ami-id
```

IMDSv2を使ってインスタンスマタデータをリクエストする際は、リクエストに次の項目が含まれている必要があります。

1. PUTリクエストを使って、インスタンスマタデータサービスに対してセッションを開始します。PUTリクエストは、インスタンスマタデータサービスに対するそれ以降のGETリクエストに含まれるべきトークンを返します。このトークンは、IMDSv2を使ってメタデータにアクセスするのに必要です。
2. トークンを、インスタンスマタデータサービスに対するすべてのGETリクエストに含めます。トークン使用が`required`に設定されている場合、有効なトークンがないリクエスト、または有効期限切れのトークンを持つリクエストは401 – Unauthorized HTTP エラーコードを受け取ります。トークン使用要件の変更に関する情報については、AWS CLI Command Referenceの[modify-instance-metadata-options](#)を参照してください。
  - トークンはインスタンス固有のキーです。トークンは他のEC2インスタンスで有効ではなく、生成されたインスタンスの外で使用しようとすると拒否されます。
  - PUTリクエストには、トークンの有効期限 (TTL) を最大6時間 (21,600秒) まで秒単位で指定するヘッダーが含まれている必要があります。トークンは論理的セッションを表します。TTLは、トークンが有効な時間の長さ、つまりセッションの期間を指定します。
  - トークンの期限が切れた後、インスタンスマタデータにアクセスし続けるためには、別のPUTを使って新しいセッションを作成する必要があります。
  - 各リクエストについてトークンを再使用するか、あるいは新しいトークンを作成することを選択できます。少数のリクエストでは、インスタンスマタデータサービスにアクセスする必要があるたびに、トークンを生成してすぐに使用するほうが簡単かもしれません。ただし、効率を重視するなら、インスタンスマタデータをリクエストする必要があるたびにPUTリクエストを書くより、トークン期間を長く指定して再使用することができます。それぞれが独自のセッションを表す同時トークンの数については実際的に制限はありません。ただし、IMDSv2は通常のインスタンスマタデータサービス接続とスロットリングの制限によって制約を受けます。詳細については、「[Throttling \(p. 606\)](#)」を参照してください。

HTTP GETおよびHEADメソッドはIMDSv2インスタンスマタデータリクエストで許可されています。PUTリクエストは、X-Forwarded-Forヘッダーが含まれている場合、拒否されます。

デフォルトで、PUTリクエストに対するレスポンスにはIPプロトコルレベルで1のレスポンスホップリミット(有効期限)があります。ホップリミットを拡大するには、[modify-instance-metadata-options](#)を参照してください。

optionsコマンドを使ってホップリミットを調整できます。たとえば、インスタンスで実行されているコンテナサービスとの下位互換性のためにホップリミットを拡大する必要があるかもしれません。詳細については、AWS CLI Command Referenceの[modify-instance-metadata-options](#)を参照してください。

## インスタンスマタデータサービスバージョン 2 使用への移行

インスタンスマタデータサービスバージョン 2 (IMDSv2) の使用はオプションです。インスタンスマタデータサービスバージョン 1 (IMDSv1) は引き続き無限にサポートされます。IMDSv2の使用に移行することを選択した場合、次のツールと移行パスを使用することができます。

### IMDSv2への移行に役立つツール

お使いのソフトウェアでIMDSv1が使用されている場合、次のツールを使ってIMDSv2を使用するようソフトウェアを再構成することができます。

- AWS ソフトウェア:最新バージョンの AWS SDK および CLI サポートIMDSv2。IMDSv2を使用するには、EC2 インスタンスの AWS SDK および CLI のバージョンが最新であることを確認する必要があります。CLI の更新に関する情報については、AWS Command Line Interface ユーザーガイドの[AWS CLI の最新バージョンのアップグレード](#)を参照してください。
- CloudWatch: IMDSv2 はトークンベースのセッションを使用しますが、IMDSv1 は使用しません。MetadataNoToken CloudWatch メトリクスは、IMDSv1 を使用しているインスタンスマタデータサービスへの呼び出しの数を追跡します。このメトリクスをゼロまでトラッキングすることにより、すべてのソフトウェアがIMDSv2を使用するようアップグレードされたかどうか、そしてそのタイミングを判断できます。詳細については、「[インスタンスマトリクス \(p. 644\)](#)」を参照してください。
- EC2 API および CLI へ更新する: 既存のインスタンスについては、[modify-instance-metadata-options](#) CLI コマンド (または[ModifyInstanceMetadataOptionsAPI](#)) を使用して、IMDSv2の使用を義務付けることができます。新しいインスタンスについては、[run-instances](#) CLI コマンド (または[RunInstancesAPI](#)) およびmetadata-optionsパラメータを使用して、IMDSv2の使用を義務付ける新しいインスタンスを起動できます。

Auto Scaling グループによって起動されるすべての新しいインスタンスで IMDSv2 の使用を必須にするには、Auto Scaling グループで起動テンプレートを使用する必要があります。起動テンプレートを作成するときに、MetadataOptions パラメータを設定して、IMDSv2 の使用を要求します。起動設定を使用する Auto Scaling グループの場合、起動設定を起動テンプレートに置き換えます。起動設定を起動テンプレートに置き換えると、Auto Scaling グループは新しい起動テンプレートを使用して新しいインスタンスを起動しますが、既存のインスタンスには影響しません。既存のインスタンスで IMDSv2 の使用を要求するには、[modify-instance-metadata-options](#) CLI コマンド (または[ModifyInstanceMetadataOptions API](#)) を使用します。または、インスタンスを終了すると、Auto Scaling グループは、起動テンプレートで定義されたインスタンスマタデータオプション設定を使用して新しい代替インスタンスを起動します。

- IAM ポリシーおよび SCP: IAM 条件を使用することで、IAM ユーザーが IMDSv2 を使用しない限り、インスタンスを起動できないよう強制できます。また IAM 条件を使って、IAM ユーザーが実行中のインスタンスを変更してIMDSv1を最有效地化できないように強制し、さらにインスタンスマタデータサービスがインスタンス上で使用できるよう強制することもできます。

IAM 条件キーの `ec2:MetadataHttpTokensec2:MetadataHttpPutResponseHopLimit` と `ec2:MetadataHttpEndpoint` を使用して、[RunInstances API](#) と [ModifyInstanceMetadataOptions API](#) および対応する CLI の使用を制御できます。ポリシーを作成し、条件キーを使用してポリシーに指定した状態と API コールのパラメータが一致しない場合、API コールまたは CLI コールは失敗して `UnauthorizedOperation` レスポンスが返されます。これらの条件キーは、IAM ポリシーまたは AWS Organizations サービスコントロールポリシー (SCP) のいずれかで使用できます。

さらに、追加の保護レイヤーを選択して、IMDSv1からIMDSv2の変更を強制することもできます。EC2 ロール資格情報経由で呼び出された API に関するアクセス管理レイヤーでは、IAMポリシーまたは AWS Organization サービスコントロールポリシー (SCP) で新しい条件キーを使用することができます。具体的には、IAM ポリシーで値 `2.0` を設定してポリシー条件キー `ec2:RoleDelivery` を使用すると、IMDSv1 から取得した EC2 ロールの認証情報を使用した API コールに対して

は、UnauthorizedOperation レスポンスが返されます。同じことは、SCP によって義務付けられる条件を使ってより広く達成できます。これにより、指定した条件と一致しない API コールに対しては UnauthorizedOperation エラーが返されるため、実際に IMDSv1 から取得した認証情報を使用して API を呼び出すことはできなくなります。IAM ポリシーの例は、「[インスタンスマタデータの使用 \(p. 878\)](#)」を参照してください。詳細については、AWS Organizations ユーザーガイドのサービスコントロールポリシーを参照してください。

#### IMDSv2アクセスを必要とする推奨パス

上記のツールを使用する際、IMDSv2への移行にこのパスに従うことを推奨します。

#### ステップ 1: 開始時

SDK、CLI、および EC2 インスタンスでロール資格情報を使用するソフトウェアを、IMDSv2対応バージョンに更新します。CLI の更新に関する情報については、AWS Command Line Interface ユーザーガイドの[AWS CLI の最新バージョンのアップグレード](#)を参照してください。

次に、IMDSv2 リクエストを使ってインスタンスマタデータに直接アクセスする(つまり、SDK を使用しない)ソフトウェアを変更します。

#### ステップ 2: 移行中

CloudWatch の MetadataNoToken メトリクスを使用して、移行の進行状況を追跡します。このメトリクスは、インスタンスで IMDSv1 を使用しているインスタンスマタデータサービスへの呼び出しの数を示します。詳細については、「[インスタンスマトリクス \(p. 644\)](#)」を参照してください。

#### ステップ 3: すべてのインスタンスですべての準備が完了した時点

CloudWatch メトリクスMetadataNoToken が IMDSv1 の使用ゼロを記録した時点で、すべてのインスタンスにおいてすべての準備が完了します。この段階で、次の操作を実行できます。

- 既存のインスタンスの場合: [modify-instance-metadata-options](#) コマンドを通じて IMDSv2 を使用することを要求できます。実行中のインスタンスでこれらの変更を行うことができます。インスタンスを再起動する必要はありません。
- 新しいインスタンスの場合: 新しいインスタンスを起動するときに、[run-instances](#) コマンドを使用して IMDSv2 のみを使用するよう指定できます。

インスタンスマタデータオプションの指定は、API または AWS CLI でのみ使用できます。現在、AWS マネジメントコンソールでは使用できません。詳細については、「[インスタンスマタデータオプションの構成 \(p. 597\)](#)」を参照してください。

#### ステップ 4: すべてのインスタンスが IMDSv2 に移行された時点

IAM 条件キーの `ec2:MetadataHttpTokenSec2:MetadataHttpPutResponseHopLimit` と `ec2:MetadataHttpEndpoint` を使用して、[RunInstances](#) API と [ModifyInstanceMetadataOptions](#) API および対応する CLI の使用を制御できます。ポリシーを作成し、条件キーを使用してポリシーに指定した状態と API コールのパラメータが一致しない場合、API コールまたは CLI コールは失敗して UnauthorizedOperation レスポンスが返されます。IAM ポリシーの例は、「[インスタンスマタデータの使用 \(p. 878\)](#)」を参照してください。

#### インスタンスマタデータオプションの構成

インスタンスマタデータオプションを使用すると、新規または既存のインスタンスで次の操作を実行するように設定できます。

- インスタンスマタデータをリクエストするときに IMDSv2 の使用を要求する

- PUT レスポンスのホップ制限を指定する
- インスタンスマタデータへのアクセスを無効にする

IAM ポリシーまたは SCP で IAM 条件キーを使用して、次の操作を行うこともできます。

- IMDSv2 の使用を要求するようにインスタンスが設定されている場合にのみ、インスタンスの起動を許可する
- ホップの許可数を制限する
- インスタンスマタデータへのアクセスを無効にする

新規または既存インスタンスでインスタンスマタデータオプションを構成するには、AWS SDK または CLI を使用します。詳細については、AWS CLI Command Reference の [run-instances](#) および [modify-instance-metadata-options](#) を参照してください。

Note

注意深く実行し、変更を行う前に慎重なテストを実施する必要があります。以下の情報を記録します。

- IMDSv2 の使用を強制する場合、インスタンスマタデータアクセスのために IMDSv1 を使用するアプリケーションまたはエージェントは休憩します。
- インスタンスマタデータへのアクセスをすべてオフにする場合、インスタンスマタデータアクセスに依存して機能するアプリケーションまたはエージェントは休憩します。

トピック

- 新規インスタンスのインスタンスマタデータオプションの設定 (p. 598)
- 既存インスタンスのインスタンスマタデータオプションの設定 (p. 599)

## 新規インスタンスのインスタンスマタデータオプションの設定

インスタンスの起動時に、インスタンスで IMDSv2 を使用することを要求できます。また、新しいインスタンスで IMDSv2 の使用を要求しない限り、新しいインスタンスの起動をユーザーに禁止する IAM ポリシーを作成することもできます。

新しいインスタンスで IMDSv2 の使用を要求するには

次の [run-instances](#) の例では、`metadata-options` を `HttpTokens=required` に設定して `c3.large` インスタンスを起動します。メタデータの取得リクエストではセキュリティで保護されたトークンヘッダーが `required` に設定されるため、インスタンスマタデータをリクエストするときに IMDSv2 の使用を要求するようにインスタンスが設定されます。

Note

- `HttpTokens` の値を指定する場合は、`HttpEndpoint` を `enabled` に設定することも必要です。
- この例では、`--security-group` パラメータ `--count` とパラメータは含まれていません。`--count` の場合、デフォルトは `1` です。デフォルトの VPC とデフォルトのセキュリティグループがある場合は、これらが使用されます。

```
aws ec2 run-instances \
--image-id ami-1a2b3c4d \
--instance-type c3.large \
```

```
--key-name MyKeyPair \  
--metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

すべての新しいインスタンスでIMDSv2の使用を強制するには

IAM ユーザーがインスタンスマタデータをリクエストする際にIMDSv2の使用を義務付けるインスタンスを起動できるようにするには、IMDSv2を必要とする条件が満たされないとインスタンスを起動できないように指定することができます。IAM ポリシーの例については、「[インスタンスマタデータの使用 \(p. 878\)](#)」を参照してください。

#### 既存インスタンスのインスタンスマタデータオプションの設定

既存インスタンスで IMDSv2 を使用することを要求できます。また、既存インスタンスで PUT レスポンスのホップ制限を変更したり、インスタンスマタデータへのアクセスを無効にしたりすることもできます。また、既存インスタンスでインスタンスマタデータオプションを変更することをユーザーに禁止する IAM ポリシーを作成することもできます。

既存インスタンスでIMDSv2の使用を義務付けるには

既存インスタンスに対して、インスタンスマタデータをリクエストする際に IMDSv2 の使用を義務付けるようオプトインすることができます。[modify-instance-metadata-options](#) CLI コマンドを使って、`http-tokens` パラメータを `required` に設定できます。

##### Note

`http-tokens` の値を指定する場合は、`http-endpoint` を `enabled` に設定することも必要です。

```
aws ec2 modify-instance-metadata-options \  
--instance-id i-1234567898abcdef0 \  
--http-tokens required \  
--http-endpoint enabled
```

既存インスタンスで PUT レスポンスホップリミットを変更するには

既存インスタンスについて、PUT レスポンスホップリミットの設定を変更することができます。[modify-instance-metadata-options](#) CLI コマンドを使って、`http-put-response-hop-limit` パラメータを必要なホップ数に設定できます。以下の例では、ホップリミットが 3 に設定されています。`http-put-response-hop-limit` の値を指定する場合は、`http-endpoint` を `enabled` に設定することも必要です。

```
aws ec2 modify-instance-metadata-options \  
--instance-id i-1234567898abcdef0 \  
--http-put-response-hop-limit 3 \  
--http-endpoint enabled
```

既存インスタンスのインスタンスマタデータへのアクセスをオフにするには

既存インスタンスについて、使用中のインスタンスマタデータサービスのバージョンに関係なく、インスタンスマタデータサービスの HTTP エンドポイントを無効化することによりインスタンスマタデータへのアクセスをオフにすることができます。HTTP エンドポイントを有効化することにより、この変更はいつでも元に戻すことができます。[modify-instance-metadata-options](#) CLI コマンドを使って、`http-endpoint` パラメータを `disabled` に設定できます。

```
aws ec2 modify-instance-metadata-options \  
--instance-id i-1234567898abcdef0 \  
--http-endpoint disabled
```

```
--http-endpoint disabled
```

modify-instance-metadata-options の使用を制御するには

既存インスタンスでインスタンスマタデータオプションを変更できる IAM ユーザーを制御するには、指定したロールを持つユーザー以外のすべてのユーザーに [ModifyInstanceMetadataOptions API](#) の使用を禁止するポリシーを指定できます。IAM ポリシーの例については、「[インスタンスマタデータの使用 \(p. 878\)](#)」を参照してください。

## インスタンスマタデータの取得

インスタンスマタデータは実行中のインスタンスから取得できるため、Amazon EC2 コンソールまたは AWS CLI を使用する必要はありません。これは、インスタンスから実行するスクリプトを記述しているときに便利です。たとえば、インスタンスマタデータからインスタンスのローカル IP アドレスにアクセスして、外部アプリケーションへの接続を管理できます。

インスタンスマタデータはいくつかのカテゴリに分けられます。各インスタンスマタデータカテゴリの説明については、[インスタンスマタデータのカテゴリ \(p. 612\)](#)を参照してください。

実行中のインスタンス内からインスタンスマタデータのすべてのカテゴリを表示するには、次の URI を使用します。

```
http://169.254.169.254/latest/meta-data/
```

IP アドレス 169.254.169.254 は、リンクローカルアドレスで、インスタンスからのみ有効です。詳細については、Wikipedia の [リンクローカルアドレス](#) を参照してください。

インスタンスマタデータおよびユーザーデータの取得に使用する HTTP リクエストに対しては課金されません。

コマンドフォーマットは、IMDSv1とIMDSv2のどちらを使うかによって異なります。デフォルトでは、両方のインスタンスマタデータサービスを使用できます。IMDSv2の使用を義務付けるには、[インスタンスマタデータサービスの構成 \(p. 594\)](#)を参照してください。

次の例のように、cURL などのツールを使用できます。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
```

また、[Instance Metadata Query](#) ツールをダウンロードすることもできます。これにより、インスタンスマタデータサービスバージョン 1 を使用して URI 全体またはカテゴリ名を入力する必要なく、インスタンスマタデータに対してクエリを実行できます。

## レスポンスおよびエラーメッセージ

すべてのインスタンスマタデータがテキスト (HTTP コンテンツタイプ `text/plain`) として返されます。

特定のメタデータリソースに対するリクエストは、適切な値または 404 - Not Found HTTP エラーコード (リソースを使用できない場合) を返します。

一般的なメタデータリソースに対するリクエスト (/ で終わる URI) は、使用可能なリソースのリストまたは 404 - Not Found HTTP エラーコード (使用可能なリソースがない場合) を返します。リスト項目は個別の行に表示され、各行の末尾には改行記号 (ASCII 10) が付いています。

インスタンスマタデータサービスバージョン 2を使って行われたリクエストについては、次の HTTP エラーコードが返されます。

- 400 - Missing or Invalid Parameters-PUTリクエストが無効である。
- 401 - Unauthorized-GETリクエストが無効なトークンを使用している。推奨されるアクションは新しいトークンを生成することです。
- 403 - Forbidden-リクエストが許可されていないか、あるいはインスタンスマタデータサービスがオフです。

## インスタンスマタデータの取得の例

例

- [使用できるインスタンスマタデータのバージョンを取得する \(p. 601\)](#)
- [上位レベルのメタデータ項目を取得する \(p. 602\)](#)
- [使用可能なパブリックキーのリストを取得する \(p. 604\)](#)
- [パブリックキーが使用できるフォーマットを示す \(p. 604\)](#)
- [パブリックキーを取得する \(OpenSSH キーフォーマット\) \(p. 605\)](#)
- [インスタンスのサブネット ID を取得する \(p. 605\)](#)

### 使用できるインスタンスマタデータのバージョンを取得する

次の例では、使用できるインスタンスマタデータのバージョンを取得しています。これらのバージョンは、Amazon EC2 API バージョンと必ずしも関連しているとは限りません。以前のバージョンに存在する構造および情報に依存するスクリプトがある場合は、以前のバージョンを使用することができます。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
1.0
2007-01-19
2007-03-01
2007-08-29
2007-10-10
2007-12-15
2008-02-01
2008-09-01
2009-04-04
2011-01-01
2011-05-01
2012-01-12
2014-02-25
2014-11-05
2015-10-20
2016-04-19
2016-06-30
2016-09-02
latest
```

## IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/  
1.0  
2007-01-19  
2007-03-01  
2007-08-29  
2007-10-10  
2007-12-15  
2008-02-01  
2008-09-01  
2009-04-04  
2011-01-01  
2011-05-01  
2012-01-12  
2014-02-25  
2014-11-05  
2015-10-20  
2016-04-19  
2016-06-30  
2016-09-02  
latest
```

## 上位レベルのメタデータ項目を取得する

次の例では、上位レベルのメタデータ項目を取得しています。詳細については、「[インスタンスマタデータのカテゴリ \(p. 612\)](#)」を参照してください。

## IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
events/  
hostname  
iam/  
instance-action  
instance-id  
instance-type  
local-hostname  
local-ipv4  
mac  
metrics/  
network/  
placement/  
profile  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups  
services/
```

## IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/
```

```
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

次の例では、前の例で取得された最上位メタデータアイテムの値のいくつかを取得しています。IMDSv2 リクエストは、前の例のコマンドで作成された保管済みトークン（期限内であると仮定）を使用します。

#### IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-id
ami-0abcdef1234567890
```

#### IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/reservation-id
r-0efghijk987654321
```

#### IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-hostname
```

```
ip-10-251-50-12.ec2.internal
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-hostname  
ip-10-251-50-12.ec2.internal
```

#### IMDSv2

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/  
latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-hostname  
ec2-203-0-113-25.compute-1.amazonaws.com
```

### 使用可能なパブリックキーのリストを取得する

次の例では、使用できるパブリックキーの一覧を取得しています。

#### IMDSv2

```
[ec2-user ~]$ `curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-  
metadata-token-ttl-seconds: 21600"` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-  
data/public-keys/  
0=my-public-key
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/  
0=my-public-key
```

### パブリックキー0が使用できるフォーマットを示す

次の例は、パブリックキー0のフォーマットを示しています。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-  
ec2-metadata-token-ttl-seconds: 21600"` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-  
data/public-keys/0/  
openssh-key
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/  
openssh-key
```

## パブリックキーを取得する (OpenSSH キーフォーマット)

次の例では、パブリックキーを取得しています (OpenSSH キーフォーマット)。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAFICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxszAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTSBdb25zb2x1MRIwEAYDVQODewlUZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEg5vb251QGFtYXpbvi5jb20wHhcNMTEwNDI1MjAONTIxWhcN
MTIwNDI0MjAONTIxWjCBiDELMAkGA1UEBhMCVVMxszAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSBdb25z
b2x1MRIwEAYDVQODewlUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEg5vb251QGFt
YXpbvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzzswY6786m86gpE
Ibb3OhjZnzcvQAaRHd1QWIMm2nrAgMBAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJ1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa MIICiTCCAFICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxszAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC01BTSBdb25zb2x1MRIwEAYDVQODewlUZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEg5vb251QGFtYXpbvi5jb20wHhcNMTEwNDI1MjAONTIxWhcN
MTIwNDI0MjAONTIxWjCBiDELMAkGA1UEBhMCVVMxszAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSBdb25z
b2x1MRIwEAYDVQODewlUZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEg5vb251QGFt
YXpbvi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzzswY6786m86gpE
Ibb3OhjZnzcvQAaRHd1QWIMm2nrAgMBAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJ1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

## インスタンスのサブネット ID を取得する

次の例では、インスタンスのサブネット ID を取得しています。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:29:96:8f:6a:2d/subnet-id
subnet-be9b61d7
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/
macs/02:29:96:8f:6a:2d/subnet-id
```

subnet-be9b61d7

## Throttling

クエリはインスタンスマタデータサービスでインスタンスごとにスロットリングし、インスタンスからインスタンスマタデータサービスへの同時接続数を制限します。

AWS セキュリティ認証情報を取得するためにインスタンスマタデータサービスを使用している場合、毎回のトランザクションで、または高頻度のスレッドやプロセスから同時に認証情報をクエリしないようにします。スロットリングの原因となる可能性があります。代わりに、認証情報をキャッシュに格納して有効期限が近づくまで待つことをお勧めします。

インスタンスマタデータサービスにアクセスする際にスロットリングした場合、エクスボンシャルパックオフ戦略でクエリを再試行します。

## インスタンスマタデータサービスの制限

ローカルファイアウォールルールを使って、プロセスの一部またはすべてからインスタンスマタデータサービスへのアクセスを無効化することを検討できます。

`iptables` を使ったアクセス制限

次の例では、Linux `iptables` およびその`owner`モジュールを使って、Apache ウェブサーバーが（デフォルトインストールユーザー ID `apache`に基づいて）169.254.169.254 にアクセスするのを防ぐことができます。拒否ルールを使って、そのユーザーとして実行中のプロセスからのインスタンスマタデータリクエスト（IMDSv1またはIMDSv2）をすべて拒否します。

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner --uid-owner apache --jump REJECT
```

また、ルールの許可を使うことで、特定のユーザーまたはグループへのアクセスを許可することを検討できます。ルールの許可は、どのソフトウェアがインスタンスマタデータへのアクセスが必要かについてユーザーが決定しなければならないため、セキュリティ観点からみたときに管理しやすいかもしれません。ルールの許可を使用すると、後にインスタンスのソフトウェアまたは構成を変更した場合に、誤ってソフトウェアがメタデータサービス（アクセスする意図がなかった）にアクセスするのを許可する可能性が低くなります。また、ファイアウォールのルールを変更しなくとも許可されたグループにユーザーを追加/削除できるよう、グループ使用をルールの許可と組み合わせることもできます。

次の例では、ユーザーアカウント`trustworthy-user`で実行中のプロセス以外のすべてのプロセスによるインスタンスマタデータサービスへのアクセスを禁止しています。

```
$ sudo iptables --append OUTPUT --proto tcp --destination 169.254.169.254 --match owner ! --uid-owner trustworthy-user --jump REJECT
```

### Note

- ローカルファイアウォールルールを使用するには、前の例のコマンドをニーズに合わせて変更する必要があります。
- デフォルトでは、`iptables` ルールはシステム再起動全体で永続しません。ここには説明されていない OS 機能を使って永続的にすることができます。
- `iptables owner`モジュールは、グループが所定のローカルユーザーのプライマリグループである場合にのみツールメンバーシップと一致します。他のグループは一致しません。

PF または IPFW を使ってアクセスを制限する

FreeBSD または OpenBSD を使用している場合、PF または IPFW の使用も検討できます。次の例では、インスタンスマタデータサービスへのアクセスをルートユーザーにのみ制限しています。

PF

```
$ block out inet proto tcp from any to 169.254.169.254
```

```
$ pass out inet proto tcp from any to 169.254.169.254 user root
```

IPFW

```
$ allow tcp from any to 169.254.169.254 uid root
```

```
$ deny tcp from any to 169.254.169.254
```

#### Note

PF および IPFW コマンドの順序は重要となります。PF のデフォルトは最後に一致したルールであり、IPFW のデフォルトは最初に一致したルールです。

## インスタンスユーザーデータの使用

インスタンスユーザーデータを使用する場合は、次の点に注意してください。

- ユーザーデータは、base64 でエンコードされている必要があります。Amazon EC2コンソールは、base64 エンコードを実行したり、base64 エンコード入力を受け入れたりできます。
- ユーザーデータは raw 形式の 16 KB に制限されます（以前は base64 エンコード）。base64 エンコード後の文字列の長さサイズ n は、 $\text{ceil}(n/3)*4$  です。
- ユーザーデータを取得するときにユーザーデータを base64 デコードする必要があります。インスタンスのメタデータあるいはコンソールを使用してデータを取得する場合、自動的にデコードされます。
- ユーザーデータは非透過的なデータとして取り扱われ、指定したデータがそのまま返されます。このデータを解釈できるかどうかは、インスタンスによって異なります。
- インスタンスを停止してユーザーデータを変更した場合、インスタンスを起動しても、更新されたユーザーデータは実行されません。

## 起動時にインスタンスユーザーデータを指定する

インスタンスの起動時のユーザーデータを指定できます。詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」および「[Linux インスタンスでの起動時のコマンドの実行 \(p. 588\)](#)」を参照してください。

## インスタンスユーザーデータを変更する

ルートボリュームが EBS ボリュームの場合は、停止状態のインスタンスのユーザーデータを変更することができます。詳細については、「[インスタンスユーザーデータの表示と更新 \(p. 590\)](#)」を参照してください。

## インスタンスユーザーデータを取得する

実行中のインスタンス内からユーザーデータを取得するには、次の URI を使用します。

```
http://169.254.169.254/latest/user-data
```

ユーザーデータのリクエストは、データをそのままの状態で返します(コンテンツタイプ application/octet-stream)。

この例は、カンマで区切られたテキストとして指定されたユーザーデータを返します。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

この例では、スクリプトとして指定されたユーザーデータを返します。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/user-data
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

お使いのコンピュータからインスタンスのユーザーデータを取得するには、「[ユーザーデータと AWS CLI \(p. 592\)](#)」を参照してください。

## 動的データの取得

実行中のインスタンス内から動的データを取得するには、次の URI を使用します。

```
http://169.254.169.254/latest/dynamic/
```

この例では、高レベルのインスタンスアイデンティティカテゴリを取得する方法を表示しています。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/
```

```
rsa2048
pkcs7
document
signature
dsa2048
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/
rsa2048
pkcs7
document
signature
dsa2048
```

動的データの詳細およびその取得方法の例については、「[インスタンスアイデンティティドキュメント \(p. 618\)](#)」を参照してください。

## 例: AMI 作成インデックス値

この例は、ユーザーデータおよびインスタンスマタデータの両方を使用してインスタンスを設定する方法を示しています。

この例では、Alice がお気に入りのデータベース AMI の 4 つのインスタンスを起動したいと考えています。そのうち最初の 1 つはマスターとして、残りの 3 つはレプリカとして動作します。これらのインスタンスを起動するときに、各レプリカントのレプリケーション戦略に関するユーザーデータを追加したいと考えています。このデータはすべての 4 つのインスタンスで使用可能となるので、どの部分が各インスタンスに該当するかをそれぞれが認識できるように、ユーザーデータを構築する必要があります。この構築は、各インスタンスに対して一意となる `ami-launch-index` インスタンスマタデータ値を使用して行うことができます。

Alice が構築したユーザーデータを次に示します。

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

`replicate-every=1min` データは最初のレプリカントの設定を定義し、`replicate-every=5min` は 2 番目のレプリカントの設定を定義するというように、それぞれが定義を行います。Alice は、個別のインスタンスのデータをパイプシンボル (|) で区切って、このデータを ASCII 文字列として指定することにしました。

Alice は、`run-instances` コマンドを使用して 4 つのインスタンスを起動します。このとき、次のユーザーデータを指定します。

```
aws ec2 run-instances --image-id ami-0abcdef1234567890 --count 4 --instance-type t2.micro
--user-data "replicate-every=1min | replicate-every=5min | replicate-every=10min"
```

起動したすべてのインスタンスに、ユーザーデータのコピーと次に示す一般的なメタデータが含まれています。

- AMI ID: ami-0abcdef1234567890
- 予約 ID: r-1234567890abcabc0
- パブリックキー: none
- セキュリティグループ名: default
- インスタンスタイプ: t2.micro

ただし、各インスタンスには所定の一意のメタデータが含まれます。

#### インスタンス 1

メタデータ	値
instance-id	i-1234567890abcdef0
ami-launch-index	0
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-12.ec2.internal
local-ipv4	10.251.50.35

#### インスタンス 2

メタデータ	値
instance-id	i-0598c7d356eba48d7
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

#### インスタンス 3

メタデータ	値
instance-id	i-0ee992212549ce0e7
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

#### インスタンス 4

メタデータ	値
instance-id	i-1234567890abcdef0
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com

メタデータ	値
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice は `ami-launch-index` 値を使用して、ユーザーデータのどの部分が特定のインスタンスに該当するかを判断できます。

1. そのインスタンスの 1 つに接続し、`ami-launch-index` を取得して、それがレプリカントの 1 つであることを確認します。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/meta-data/api/token"  
-H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`\  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-  
data/ami-launch-index  
2
```

次のステップでは、IMDSv2が前のIMDSv2コマンドからの保管済みトークン(期限内であると仮定)を使用するようリクエストします。

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/ami-launch-index  
2
```

2. 変数として `ami-launch-index` を保存します。

#### IMDSv2

```
[ec2-user ~]$ ami_launch_index=`curl -H "X-aws-ec2-metadata-token: $TOKEN" -v  
http://169.254.169.254/latest/meta-data/ami-launch-index`
```

#### IMDSv1

```
[ec2-user ~]$ ami_launch_index=`curl http://169.254.169.254/latest/meta-data/ami-  
launch-index`
```

3. ユーザーデータを変数として保存します。

#### IMDSv2

```
[ec2-user ~]$ user_data=`curl -H "X-aws-ec2-metadata-token: $TOKEN" -v  
http://169.254.169.254/latest/user-data`
```

#### IMDSv1

```
[ec2-user ~]$ user_data=`curl http://169.254.169.254/latest/user-data`
```

4. 最後に、Alice は `cut` コマンドを使用して、そのインスタンスに該当するユーザーデータの部分を抽出します。

#### IMDSv2

```
[ec2-user ~]$ echo $user_data | cut -d'"' -f"$ami_launch_index"replicate-every=5min
```

## IMDSv1

```
[ec2-user ~]$ echo $user_data | cut -d"|" -f"$ami_launch_index"  
replicate-every=5min
```

## インスタンスマタデータのカテゴリ

次の表は、インスタンスマタデータのカテゴリをまとめたものです。

### Important

以下の表のカテゴリ名のいくつかは、インスタンスに固有のデータのプレースホルダーです。たとえば、`mac` はネットワークインターフェイスの MAC アドレスを表します。プレースホルダーを実際の値に置き換える必要があります。

データ	説明	導入されたバージョン
ami-id	インスタンスの起動に使用される AMI ID。	1.0
ami-launch-index	同時に複数のインスタンスを起動した場合、この値はインスタンスが起動された順序を示します。最初に起動されたインスタンスの値は 0 です。	1.0
ami-manifest-path	Amazon S3 での AMI のマニフェストファイルのパス。Amazon EBS-Backed AMI を使用してインスタンスを起動した場合、返される結果は <code>unknown</code> です。	1.0
ancestor-ami-ids	この AMI を作成するために再バンドルされたあらゆるインスタンスの AMI ID。この値は、AMI マニフェストファイルが <code>ancestor-amis</code> キーを含む場合にのみ存在します。	2007-10-10
block-device-mapping/ami	<code>root/boot</code> ファイルシステムを含む仮想デバイス。	2007-12-15
block-device-mapping/ebs N	任意の Amazon EBS ボリュームに関連付けられた仮想デバイス。Amazon EBS ボリュームは、起動の時点またはインスタンスが最後に開始された時点で存在している場合にのみ、メタデータで使用できます。N は、Amazon EBS ボリュームのインデックス ( <code>ebs1</code> や <code>ebs2</code> など) を示します。	2007-12-15
block-device-mapping/eph emeral N	非 NVMe インスタンスストアボリュームの仮想デバイス。N は、各ボリュームのインデックスを示します。ブロックデバイスマッピングのインスタンスストアボリュームの数は、インスタンスのインスタンスストアボ	2007-12-15

データ	説明	導入されたバージョン
	リュームの実際の数に一致しない場合があります。インスタンスに使用可能なインスタンスマッピングのインスタンスマッピングの数が、インスタンスに利用可能な数を超える場合、追加のインスタンスマッピングは無視されます。	
block-device-mapping/root	ルートデバイスに関連付けられた仮想デバイスまたはパーティション、あるいは仮想デバイス上のパーティション。ルート (/ または C:) ファイルシステムは、所定のインスタンスに関連付けられています。	2007-12-15
block-device-mapping/swap	swap に関連付けられた仮想デバイス。存在しない場合もあります。	2007-12-15
elastic-gpus/ associations/ <i>elastic-gpu-id</i>	インスタンスにアタッチされている Elastic GPU がある場合、その ID と接続情報を含めた Elastic GPU に関する情報の JSON 文字列が含まれます。	2016-11-30
elastic-inference/ associations/ <i>eia-id</i>	インスタンスにアタッチされた Elastic Inference アクセラレーターがある場合、その ID とタイプを含めた Elastic Inference アクセラレーターに関する情報の JSON 文字列が含まれます。	2018-11-29
events/maintenance/history	インスタンスの完了またはキャンセルされたメンテナンスイベントがある場合は、イベントに関する情報を含む JSON 文字列を含みます。詳細については、「完了またはキャンセルされたイベントのイベント履歴を表示するには (p. 637)」を参照してください。	2018-08-17
events/maintenance/scheduled	インスタンスがアクティブなメンテナンスイベントがある場合は、イベントに関する情報を含む JSON 文字列を含みます。詳細については、「予定されたイベントの表示 (p. 633)」を参照してください。	2018-08-17
hostname	インスタンスのプライベート IPv4 DNS ホスト名。複数のネットワークインターフェイスが存在する場合、これは eth0 デバイス (デバイス番号が 0 のデバイス) を示します。	1.0

データ	説明	導入されたバージョン
iam/info	インスタンスに関連付けられた IAM ロールがある場合、インスタンスの LastUpdated の日付、InstanceProfileArn、InstanceProfileId など、インスタンスプロファイルが更新された最終時刻に関する情報が格納されます。そうでない場合は、なしになります。	2012-01-12
iam/security-credentials/ role-name	インスタンスに関連付けられた IAM ロールがある場合、 <i>role-name</i> はロールの名前になり、 <i>role-name</i> に、そのロールに関連付けられた一時的なセキュリティ認証情報が格納されます (詳細については、「 <a href="#">インスタンスマタデータからセキュリティ認証情報を取得する (p. 889)</a> 」を参照してください)。そうでない場合は、なしになります。	2012-01-12
identity-credentials/ec2/ info	[内部使用のために留保] Amazon EC2 インフラストラクチャの残りのインスタンスを識別するために AWS が使用する認証情報に関する情報。	2018-05-23
identity-credentials/ec2/ security-credentials/ec2- instance	[内部使用のために留保] Amazon EC2 インフラストラクチャの残りのインスタンスを識別するために AWS が使用する認証情報に関する情報。	2018-05-23
instance-action	バンドルの準備のために再起動する必要があることをインスタンスに伝えます。有効な値: none   shutdown   bundle-pending.	2008-09-01
instance-id	このインスタンスの ID。	1.0
instance-type	インスタンスの種類。詳細については、「 <a href="#">インスタンスタイプ (p. 183)</a> 」を参照してください。	2007-08-29
kernel-id	このインスタンスで起動したカーネルの ID (ある場合)。	2008-02-01
local-hostname	インスタンスのプライベート IPv4 DNS ホスト名。複数のネットワークインターフェイスが存在する場合、これは eth0 デバイス (デバイス番号が 0 のデバイス) を示します。	2007-01-19
local-ipv4	インスタンスのプライベート IPv4 アドレス。複数のネットワークインターフェイスが存在する場合、これは eth0 デバイス (デバイス番号が 0 のデバイス) を示します。	1.0

データ	説明	導入されたバージョン
mac	インスタンスのメディアアクセスコントロール (MAC) アドレス。複数のネットワークインターフェイスが存在する場合、これは eth0 デバイス (デバイス番号が 0 のデバイス) を示します。	2011-01-01
metrics/vhostmd	使用不可	2011-05-01
network/interfaces/macs/mac/device-number	そのインターフェイスに関連付けられた固有のデバイス番号。デバイス番号はデバイス名に対応します。たとえば、2 という device-number は eth2 デバイスを指します。このカテゴリは、Amazon EC2 API で使用される DeviceIndex フィールドと device-index フィールド、および AWS CLI の EC2 コマンドに対応します。	2011-01-01
network/interfaces/macs/mac/interface-id	ネットワークインターフェイスの ID。	2011-01-01
network/interfaces/macs/mac/ipv4-associations/public-ip	各パブリック IP アドレスに関連付けられ、そのインターフェイスに割り当てられたプライベート IPv4 アドレス。	2011-01-01
network/interfaces/macs/mac/ipv6s	インターフェイスに関連付けられた IPv6 アドレス。VPC 内に起動されたインスタンスに対してのみ返されます。	2016-06-30
network/interfaces/macs/mac/local-hostname	インターフェイスのローカルホスト名。	2011-01-01
network/interfaces/macs/mac/local-ipv4s	インターフェイスに関連付けられたプライベート IPv4 アドレス。	2011-01-01
network/interfaces/macs/mac/mac	インスタンスの MAC アドレス。	2011-01-01
network/interfaces/macs/mac/owner-id	ネットワークインターフェイスの所有者の ID。複数インターフェイスの環境では、インターフェイスは Elastic Load Balancing などのサードパーティによってアタッチできます。インターフェイス上のトラフィックは、常にインターフェイス所有者に対して課金されます。	2011-01-01

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
インスタンスマタデータとユーザーデータ

データ	説明	導入されたバージョン
network/interfaces/macs/mac/public-hostname	インターフェイスのパブリック DNS (IPv4)。このカテゴリは、enableDnsHostnames 属性が true に設定されている場合にのみ返されます。詳細については、「 <a href="#">Using DNS with Your VPC</a> 」を参照してください。	2011-01-01
network/interfaces/macs/mac/public-ipv4s	インターフェイスに関連付けられたパブリック IP アドレスまたは Elastic IP アドレス。インスタンスには複数の IPv4 アドレスが存在する場合があります。	2011-01-01
network/interfaces/macs/mac/security-groups	ネットワークインターフェイスが属するセキュリティグループ。	2011-01-01
network/interfaces/macs/mac/security-group-ids	ネットワークインターフェイスが属するセキュリティグループの ID。	2011-01-01
network/interfaces/macs/mac/subnet-id	インターフェイスが存在するサブネットの ID。	2011-01-01
network/interfaces/macs/mac/subnet-ipv4-cidr-block	インターフェイスが存在するサブネットの IPv4 CIDR ブロック。	2011-01-01
network/interfaces/macs/mac/subnet-ipv6-cidr-blocks	インターフェイスが存在するサブネットの IPv6 CIDR ブロック。	2016-06-30
network/interfaces/macs/mac/vpc-id	インターフェイスが存在する VPC の ID。	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-block	VPC のプライマリ IPv4 CIDR ブロック。	2011-01-01
network/interfaces/macs/mac/vpc-ipv4-cidr-blocks	VPC の IPv4 CIDR ブロック。	2016-06-30
network/interfaces/macs/mac/vpc-ipv6-cidr-blocks	インターフェイスが存在する VPC の IPv6 CIDR ブロック。	2016-06-30
placement/availability-zone	インスタンスが起動した利用可能ゾーン。	2008-02-01
product-codes	インスタンスに関連付けられた AWS Marketplace 製品コード (ある場合)。	2007-03-01
public-hostname	インスタンスのパブリック DNS。このカテゴリは、enableDnsHostnames 属性が true に設定されている場合にのみ返されます。詳細については、「Amazon VPC ユーザーガイド」の「 <a href="#">VPC での DNS の使用</a> 」を参照してください。	2007-01-19

データ	説明	導入されたバージョン
public-ipv4	パブリック IPv4 アドレス。インスタンスに Elastic IP アドレスが関連付けられている場合、返される値は Elastic IP アドレスです。	2007-01-19
public-keys/0/openssh-key	パブリックキー。インスタンスの起動時に指定された場合のみ返されます。	1.0
ramdisk-id	起動時に指定された RAM ディスクの ID (該当する場合)。	2007-10-10
reservation-id	予約の ID。	1.0
security-groups	インスタンスに適用されるセキュリティグループの名前。  起動後、インスタンスのセキュリティグループを変更できます。これらの変更は、この場所と network/interfaces/mac/ <i>mac</i> /security-groups に反映されます。	1.0
services/domain	リージョンの AWS リソースのドメイン。	2014-02-25
services/partition	リソースが置かれているパーティションです。標準の AWS リージョンの場合、パーティションは aws です。他のパーティションにリソースがある場合、パーティションは aws- <i>partitionname</i> です。たとえば、中国 (北京) リージョンにあるリソースのパーティションは、aws-cn です。	2015-10-20
spot/instance-action	アクション (休止、停止、または終了) およびアクションのおよその発生時刻 (UTC)。この項目が存在するのは、スポットインスタンスが休止、停止、または終了とマークされた場合のみです。詳細については、「 <a href="#">instance-action (p. 390)</a> 」を参照してください。	2016-11-15

データ	説明	導入されたバージョン
spot/termination-time	スポットインスタンス のオペレーティングシステムがシャットダウン信号を受信するおよその時刻 (UTC)。この項目は、スポットインスタンスに Amazon EC2 による終了のマークが付けられている場合にのみ存在し、時刻値 (たとえば 2015-01-05T18:02:00Z) が含まれます。ユーザー自身がスポットインスタンス を終了した場合、termination-time 項目に時刻は設定されません。 詳細については、「 <a href="#">termination-time (p. 390)</a> 」を参照してください。	2014-11-05

## 動的データのカテゴリ

次の表は、動的データのカテゴリをまとめたものです。

データ	説明	導入されたバージョン
fws/instance-monitoring	顧客が CloudWatch で詳細な 1 分間隔のモニタリングを有効にしているかどうかを示す値。有効な値: enabled   disabled	2009-04-04
instance-identity/document	インスタンス ID、プライベート IP アドレスなど、インスタンスの属性を含む JSON。「 <a href="#">インスタンスアイデンティドキュメント (p. 618)</a> 」を参照してください。	2009-04-04
instance-identity/pkcs7	署名に対してドキュメントの真正性およびコンテンツを確認するために使用されます。「 <a href="#">インスタンスアイデンティドキュメント (p. 618)</a> 」を参照してください。	2009-04-04
instance-identity/signature	オリジンおよび権限を確認するために使用できるデータ。「 <a href="#">インスタンスアイデンティドキュメント (p. 618)</a> 」を参照してください。	2009-04-04

## インスタンスアイデンティドキュメント

インスタンスアイデンティドキュメントは、インスタンスについて説明する JSON ファイルです。インスタンスアイデンティドキュメントには、署名と PKCS7 署名が添付されています。これを使用して、ドキュメント内の情報の正確性、オリジン、および正当性を検証することができます。

インスタンスが起動するときにインスタンスアイデンティドキュメントが生成され、[インスタンスマタデータ \(p. 593\)](#)を通じてインスタンスに公開されます。このドキュメントは、インスタンスサイズ、インスタンスタイプ、オペレーティングシステム、AMI など、インスタンスの属性を検証します。

### Important

インスタンスアイデンティドキュメントと署名には動的な特質があるため、インスタンスマタデータ (p. 593) と署名は、定期的に取得することをお勧めします。

## インスタンスアイデンティティドキュメントと署名の取得

インスタンスアイデンティティドキュメントを取得するには、実行中のインスタンスから以下のコマンドを使用します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/document
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/document
```

出力例を次に示します。

```
{
  "devpayProductCodes" : null,
  "marketplaceProductCodes" : [ "1abc2defghijklm3nopqrs4tu" ],
  "availabilityZone" : "us-west-2b",
  "privateIp" : "10.158.112.84",
  "version" : "2017-09-30",
  "instanceId" : "i-1234567890abcdef0",
  "billingProducts" : null,
  "instanceType" : "t2.micro",
  "accountId" : "123456789012",
  "imageId" : "ami-5fb8c835",
  "pendingTime" : "2016-11-19T16:32:11Z",
  "architecture" : "x86_64",
  "kernelId" : null,
  "ramdiskId" : null,
  "region" : "us-west-2"
}
```

インスタンスの ID 署名を取得するには、実行中のインスタンスから以下のコマンドを使用します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/signature
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/signature
```

出力例を次に示します。

```
dExamplesjNQhhJan7p0RLpLSr7lJEF4V2DhKGlyoYVBoUYrY9njyBCmhEayaGrhtS/AWY+LPx
1VSQURF5n0gwPNCuO6ICT0fNrm5IH7w9ydyaxamplejJw8XvWPxbuRkcN0TAA1p4RtCAqm4ms
x2oALjWSCBExample=
```

PKCS7 署名を取得するには、実行中のインスタンスから以下のコマンドを使用します。

## IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
```

## IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
```

出力例を次に示します。

```
MIICiTCCAFICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAlDbMRawDgYDVQQHEwdTZWF0dGx1M98wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEGB5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCzDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAlDbMRawDgYDVQQHEwdTZWF0dGx1M98wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BTSDb25zb2x1MRIwEAYDVQQDEw1UZXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEGB5vb25lQGFtYXpvbi5jb20wgZ8wDQYJKoZIhvvcNAQEBBQADgY0AMIGJAOGBAMAk0dn+a4GmWIWJ21uUSfwfEvYSwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/Mb0QITxOUSQv7c7ugFFDzQGBzZswY6786m86gPEIbb3OhjZnzcvQAArHd1QWIMm2nrAgMBAAEwDQYJKoZIhvvcNAQEFBQADgYEAtCu4nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgL0FkbFFBjvSfpJ1lJ00zbhNY5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjkx79LjSTbNYiytVbZPQU5Yaxu2jXnimvw3rrszlaEXAMPLE
```

## PKCS7 署名の確認

PKCS7 署名を使用して、適切な AWS パブリック証明書に対して検証することにより、インスタンスを確認できます。

AWS アカウントによって提供されるリージョンの AWS パブリック証明書は、次のとおりです。

```
-----BEGIN CERTIFICATE-----  
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgkqhkiJOOAQDMFwxzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXYXN0aW5ndG9uIFN0YXR1MRawDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnPzY2VzIEzMqzaeFw0xMjAxMDUxMjU2MTJaFw0zODAxMDUxMjU2MTJaMFwxzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXYXN0aW5ndG9uIFN0YXR1MRawDgYDVQQHEwdTZWF0dGx1MSAwHgYDVQQKExdBbWF6b24gV2ViIFNlcnPzY2VzIEzMqzCCAbcwggEsBgcqhkjOOAQBMIIIBhWBgQcjkvcS2bb1VQ4yt/5eih5006kN/n1Lz1lr7D8ZwtQP8f0Epp5E2ng+D6UD1Z1gYipr58Kj3nssSNpI6bX3VylQzK7wLc1nd/YozQNnmgiyZecN7EglK9ITHJLP+x8FtUp3QbyYXJdmVmegN6PhviYt5JH/nY14hh3Pa1HJDskgQIVALVJ3ER11+Ko4tP6nwvHwh6+ERYRAoGBAI1jk+tkqMVHuAFcvAGKocTgsjJem6+5qomzJuKDmbJNu9Qxw3rAotXau8Qe+MBCJ1/Uhh1KHVpCG19fueQ2s6IL0Ca0/buycu1CiYQk40KNHCchFnzbdlx1E9rpUp7bnF1Ra2v1ntMX3caRVDbtPEWmdxSCyfDk4mZrOLBA4GEAAKBgEbmeve5f8LIE/GfMNmP9CM5eoVQOGx5ho8Wqd+Atebs+k2tn92BPqeZqpWRa5P/+jrdKml1qx4llHWMXrs3IgIb6+hUIB+S8dz8/mm00bp76RoZVCYab2CZedFut7qc3WUH9+EUAH5mwvSeDCOUAMYQR7R9LINYwouHIZiqQYMAkGByqGSM44BAMDLwAwLAIUWXBlk40xTwSw7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6Rok0k9K-----END CERTIFICATE-----
```

香港リージョンの AWS パブリック証明書は次のとおりです。

```
-----BEGIN CERTIFICATE-----  
MIIC7zCCAq4CCQC07MJe5Y3VLjAJBgkqhkiJOOAQDMFwxzAJBgNVBAYTA1VTMRkwFwYDVQQIEExBXYXN0aW5ndG9uIFN0YXR1MRawDgYDVQQHEwdTZWF0dGx1MSAwHgYDV
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
インスタンスマタデータとユーザーデータ

```
VQQKExdBbWF6b24gV2ViIFNlcnPzY2VzIExMqZaEfw0xOTAYMDMwMjIxMjFaFw00  
NTAyMDMwMjIxMjFaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQOIExBXYXNoaW5ndG9u  
IFN0YXR1MRAwDgYDVQOHEwdTZWF0dGx1MSAwHgYDVQKExdBbWF6b24gV2ViIFN1  
cnZpY2VzIExMqzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDvQ9RzVvf4MAwGbqfx  
b1CvCoVb99570kLGN/04CowHXJ+vTBR7eyIa6AoXltsQXBOMrJswToFKKxT4gbuw  
jk7s9Q0X4CmTRWcEgO2RxtZSVjOhsUQMh+yf7Ht40VL97LwnNfGsX2cwjCRWHYgi  
71vnubNBzLQHdSEwMNq0Bk76PwIVAMan6XIEEPnwr4e6u/RNnWBGKd9FAoGBAOOG  
eSNmwpW4QFu4p1IAyk6EnTZKKHT87gdXkaKfc5fAfOxxhnE2HezzHp9Ap2tMV5  
8bWNv0PhvoKCQqwfm+OUBLAxAC/3vqoVkl2mG1KgUH9+hrtptMTkwO3RREnKe7150  
x9gDimJpOihrl4I0dYvy9xUooz+dZFAW8+y1WVypA4GFAKbgQDbnBAKSxWr9QHY  
6Dt+EFdGz61AZLedeBKAoP53Z1D034J0C55YbJTwBTFGqPtOLxnUVd1Gid6Gbmc  
80f3jvogPR1mSmGsydbNbZnbUeVWrRhe+y5z3g9qs/DwmDW0deEFvhkWVnLJkFJ  
9pdou/ibRPH11E2nz6pK7GbQoTlyHTAJBgcqhkjOOAQDAzAAMC0CFQCoJlwGtJQC  
cLoM4p/jtVFOj26xbgIUUS4pDKyHaG/eayglTtFpFJqzWhc=  
-----END CERTIFICATE-----
```

バーレーンリージョンの AWS パブリック証明書は次のとおりです。

```
-----BEGIN CERTIFICATE-----  
MIIC7jCCAq4CCQCWVWIgSmP8RhTAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTA1VTMRkw  
FwYDVQOIExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQOHEwdTZWF0dGx1MSAwHgYD  
VQQKExdBbWF6b24gV2ViIFNlcnPzY2VzIExMqZaEfw0xOTAYMDUXMZA2MjFaFw00  
NTAyMDUXMZA2MjFaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQOIExBXYXNoaW5ndG9u  
IFN0YXR1MRAwDgYDVQOHEwdTZWF0dGx1MSAwHgYDVQKExdBbWF6b24gV2ViIFN1  
cnZpY2VzIExMqzCCAbgwggEsBgcqhkjOOAQBMIIBHwKBgQDcwojQfgWdV1Qli0OB  
8n6cLZ38VE7ZmrjZ90QV//Gst6S1h7euhC23YppKXi1zovefSDwFU54zi3/oJ++q  
PH1P1WGL8IZ34BUGRTtG4TVolvp0smjkMvyRu5h1dKtzjv93Ccx15gVgyk+o1IEG  
fZ2Kbw/Dd8JfoPS7KaScmJKxXQIVAIzbIaDFRGa2qcMkW2HWASyND17bAoGBAnTz  
IdhFmq+12I5iofy2oj3HI21Kj3LtZrWEg3W+4rvhL31TmONne1rl9yGujrjQwy5  
Zp9V4A/w9w2010Lx4K6hj3Eefy/aQnZwNdNhv/FQP7Azofju+Y16L13OOHqrL0z  
Q+9cF7zEosekEnBQx3v6psNknKgD3Shgx+GO/LpCA4GFAKbgQCVS7m77nuNALz8  
wvUqcooxXMPkxF154NxAsAul9KP9KN4svm003Zrb7t2F0tXRM8zU3TqMpryq1o5  
mpMPsZDg6RXo9BF7Hn0DoZ6PJTamkFA6md+NyTJWJKvXC7iJ8fGDBJqTciUHuCKr  
12AztQ8bFWsrTgTzPE3p6U5ckcgV1TAJBgcqhkjOOAQDAy8AMCwCFB2NZGwm5ED1  
86ayv3c1PEDukg0IAhOow38rQkN/VwHVeSW9DqEshXHjuQ==  
-----END CERTIFICATE-----
```

AWS GovCloud (US-West) リージョンの AWS パブリック証明書は次のとおりです。

```
-----BEGIN CERTIFICATE-----  
MIIC7TCCAq0CCQCWukjZ5V4aZzAJBgcqhkjOOAQDMFwxCzAJBgNVBAYTA1VTMRkw  
FwYDVQOIExBXYXNoaW5ndG9uIFN0YXR1MRAwDgYDVQOHEwdTZWF0dGx1MSAwHgYD  
VQQKExdBbWF6b24gV2ViIFNlcnPzY2VzIExMqZaEfw0xMjAxMDUXMjU2MTJaFw0z  
ODAxMDUXMjU2MTJaMFwxCzAJBgNVBAYTA1VTMRkwFwYDVQOIExBXYXNoaW5ndG9u  
IFN0YXR1MRAwDgYDVQOHEwdTZWF0dGx1MSAwHgYDVQKExdBbWF6b24gV2ViIFN1  
cnZpY2VzIExMqzCCAbcwggEsBgcqhkjOOAQBMIIBHwKBgQjkvcS2bb1VQ4yt/5e  
ih5006kN/n1Lz1lr7D8ZwtQP8fOEpp5E2ng+D6UD1Z1gYipr58Kj3nssSNpI6bX3  
VyiQzK7wLc1nd/YozqNnmgiyZecN7EglK9ITHJLP+x8FtUpt3QbyYXJdmVMegN6P  
hviYt5JH/nY14hh3Pa1HJdskgQIVALVJ3ER11+Ko4tP6nvvHwh6+ERYRAoGBAI1j  
k+tqMVHuAFcvAGKocTgsjJem6/5qomzJuKDmBjNu9Qxw3rAotXau8Qe+MBcJ1/U  
hhy1KHVpCG19fueQ2s6IL0Ca0/buyC1CiYok40KNHCchFnizbdlx1E9rpUp7bnF  
lRa2v1ntMX3caRVDbdtPEWmdxSCysYFDk4mZrOLBA4GEAKbgEbmeve5f8LIE/Gf  
MNmP9CM5eoV0Gx5ho8WQd+Atebs+k2tn92BPqeZqpWra5P/+jrdKml1qx411HW  
MXrs3IgIb6+hUIB+S8dz8/mm0bpbr76RoZVCXYab2CZedFut7qc3WUH9+EUAH5mw  
vSeDCOUAMYQR7R9LINYwouHiziqQYMAkGByqGSM44BAMDLwAwLAIUWXB1k40xTwSw  
7HX32MxXYruse9ACFBNGmdX2ZBrVNGrN9N2f6Rok0k9K  
-----END CERTIFICATE-----
```

他のリージョンの AWS パブリック証明書を取得するには、[AWS サポート](#)にお問い合わせください。

PKCS7 署名を確認するには

1. インスタンスから、PKCS7 署名用の一時ファイルを作成します。

```
[ec2-user ~]$ PKCS7=$(mktemp)
```

2. -----BEGIN PKCS7----- ヘッダーを一時的な PKCS7 ファイルに追加します。

```
[ec2-user ~]$ echo "-----BEGIN PKCS7-----" > $PKCS7
```

3. インスタンスのメタデータから PKCS7 署名の内容を新しい行に加えます。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7 >> $PKCS7
[ec2-user ~]$ echo "" >> $PKCS7
```

IMDSv1

```
[ec2-user ~]$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/pkcs7
>> $PKCS7
[ec2-user ~]$ echo "" >> $PKCS7
```

4. -----END PKCS7----- フッターを追加します。

```
[ec2-user ~]$ echo "-----END PKCS7-----" >> $PKCS7
```

5. インスタンスアイデンティティドキュメン用の一時ファイルを作成します。

```
[ec2-user ~]$ DOCUMENT=$(mktemp)
```

6. インスタンスのメタデータから一時的なドキュメントファイルにドキュメントのコンテンツを追加します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/dynamic/instance-identity/document > $DOCUMENT
```

IMDSv1

```
[ec2-user ~]$ curl -s http://169.254.169.254/latest/dynamic/instance-identity/document > $DOCUMENT
```

7. テキストエディタを開き、AWSpubkey という名前のファイルを作成します。上記の AWS パブリック証明書の内容をコピーしてファイルに貼り付け、保存します。
8. 次のように、OpenSSL ツールを使用して署名を確認します。

```
[ec2-user ~]$ openssl smime -verify -in $PKCS7 -inform PEM -content $DOCUMENT -certfile AWSpubkey -noverify > /dev/null
Verification successful
```

## Amazon Elastic Inference

Amazon Elastic Inference (EI) は、深層学習 (DL) の推論ワークロードを加速するために Amazon EC2 CPU インスタンスにアタッチできるリソースです。Amazon EI アクセラレーターは複数のサイズで提供され、Amazon EC2 インスタンス上で実行されるアプリケーションにインテリジェントな機能を組み込むための費用対効果の高い方法です。

Amazon EI は、MXNet を介して TensorFlow、Apache MXNet、および Open Neural Network Exchange (ONNX) 形式で定義されたモデルオペレーションを、低コストの DL 推論アクセラレーターとインスタンスの CPU との間で分配します。

Amazon Elastic Inference の詳細については、「[Amazon EI 開発者ガイド](#)」を参照してください。

## EC2 Linux インスタンスを特定する

アプリケーションにより EC2 インスタンスで実行されているかどうかを判断する必要がある場合があります。

Windows インスタンス特定の詳細については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[EC2 Windows インスタンスの特定](#)」を参照してください。

## インスタンスアイデンティティドキュメントの調査

EC2 インスタンスを識別する、暗号により確認された確実な方法については、その署名を含めて、インスタンスアイデンティティドキュメントを参照してください。これらのドキュメントは、ローカルのルーティングできないアドレス <http://169.254.169.254/latest/dynamic/instance-identity/> の各 EC2 インスタンスで入手できます。詳細については、「[インスタンスアイデンティティドキュメント \(p. 618\)](#)」を参照してください。

## システム UUID の確認

システムの UUID を取得して、UUID の最初のオクテットに「ec2」または「EC2」という文字が存在するかどうかを検索することができます。システムが EC2 インスタンスであるかどうかを判断するこの方法は、EC2 インスタンスではないシステムがこれらの文字で始まる UUID を持つ可能性が低いため、迅速でありながら不正確である可能性があります。さらに、EC2 インスタンス向けの Amazon Linux を使用していない場合、SMBIOS のデイストリビューションの実装についてはリトルエンディアン形式で UUID を表すことがあるため、"EC2" の文字は UUID の先頭には使用されません。

Example : ハイパーバイザーから UUID を取得

/sys/hypervisor/uuid が存在する場合は、次のコマンドを使用できます。

```
[ec2-user ~]$ cat /sys/hypervisor/uuid
```

次の出力例では、UUID は「ec2」で始まりますが、これは多くの場合システムが EC2 インスタンスであることを示しています。

```
ec2e1916-9099-7caf-fd21-012345abcdef
```

Example : DMI から UUID を取得 (HVM のインスタンスのみ)

HVM インスタンスの場合のみ、デスクトップ管理インターフェイス (DMI) を使用できます。

`dmidecode` ツールを使用して UUID を返すことができます。Amazon Linux では、次のコマンドを使用して、`dmidecode` ツールがインスタンスにまだインストールされていない場合はインストールします。

```
[ec2-user ~]$ sudo yum install dmidecode -y
```

次に、以下のコマンドを実行します。

```
[ec2-user ~]$ sudo dmidecode --string system-uuid
```

または、以下のコマンドを使用します。

```
[ec2-user ~]$ sudo cat /sys/devices/virtual/dmi/id/product_uuid
```

次の出力例では、UUID は「EC2」で始まりますが、これは多くの場合システムが EC2 インスタンスであることを示しています。

```
EC2E1916-9099-7CAF-FD21-01234ABCDEF
```

次の出力例では、UUID がリトルエンディアン形式で表されています。

```
45E12AEC-DCD1-B213-94ED-01234ABCDEF
```

Nitro インスタンスでは、次のコマンドを使用できます。

```
[ec2-user ~]$ cat /sys/devices/virtual/dmi/id/board_asset_tag
```

これはインスタンス ID を返します。これは EC2 インスタンスに固有のものです。

```
i-0af01c0123456789a
```

# Amazon EC2 のモニタリング

モニタリングは、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスおよび AWS ソリューションの信頼性、可用性、およびパフォーマンスを維持する上で重要な部分です。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。ただし、Amazon EC2 のモニタリングを開始する前に、次の内容を盛り込んだモニタリング計画を作成する必要があります。

- ・モニタリングの目的とは？
- ・モニタリングの対象となるリソースとは？
- ・どのくらいの頻度でこれらのリソースをモニタリングしますか？
- ・使用するモニタリングツールは？
- ・誰がモニタリングタスクを実行しますか？
- ・誰が問題が発生したときに通知を受け取りますか？

モニタリングの目的を定義し、モニタリングの計画を作成したら、次のステップとして、お客様の環境内で通常の Amazon EC2 パフォーマンスのベースラインを確立します。さまざまな時間帯に、さまざまな負荷条件で Amazon EC2 パフォーマンスを測定します。Amazon EC2 をモニタリングしながら、収集したモニタリングデータの履歴を格納する必要があります。現在の Amazon EC2 パフォーマンスをこの履歴データと比較して、通常のパフォーマンスパターンとパフォーマンス異常を識別することで、異常への対処方法を考案することが容易になります。たとえば、EC2 インスタンスの CPU 使用率、ディスク I/O、およびネットワーク使用率をモニタリングすることができます。確立したベースラインからパフォーマンスが外れた場合は、インスタンスの再設定または最適化を行って CPU 使用率の抑制、ディスク I/O の改善、またはネットワークトラフィックの低減を行うことが必要な場合があります。

ベースラインを確立するには、少なくとも、次の項目をモニタリングする必要があります。

モニタリング対象の項目	Amazon EC2 メトリクス	エージェント / CloudWatch Logs のモニタリング
CPU 使用率	<a href="#">CPUUtilization (p. 644)</a>	
ネットワーク使用率	<a href="#">NetworkIn (p. 644)</a> <a href="#">NetworkOut (p. 644)</a>	
ディスクパフォーマンス	<a href="#">DiskReadOps (p. 644)</a> <a href="#">DiskWriteOps (p. 644)</a>	
ディスクの読み書き	<a href="#">DiskReadBytes (p. 644)</a> <a href="#">DiskWriteBytes (p. 644)</a>	
メモリの使用率、ディスクスワップの使用率、ディスクスペースの使用状況、ページファイルの使用状況、ログ収集		[Linux および Windows Server インスタンス] <a href="#">CloudWatch エージェントを使用して Amazon EC2 インスタンスとオンプレミスサーバーからメトリクスを収集する</a> [Windows Server インスタンスでの以前の CloudWatch Logs エージェントからの移行] <a href="#">Windows Server インスタンスのログ収集</a>

モニタリング対象の項目	Amazon EC2 メトリクス	エージェント / CloudWatch Logs のモニタリング
		<a href="#">を CloudWatch エージェントに移行する</a>

## 自動モニタリングと手動モニタリング

AWS では、Amazon EC2 のモニタリングに使用できるさまざまなツールを提供しています。これらのツールの中には、自動モニタリングを設定できるものもあれば、手操作を必要とするものもあります。

### トピック

- [自動モニタリングツール \(p. 626\)](#)
- [手動モニタリングツール \(p. 627\)](#)

## 自動モニタリングツール

次に示す自動化されたモニタリングツールを使用すると、Amazon EC2 の監視が行われ、問題が検出されたときにレポートが返されます。

- [System Status Checks] – インスタンスを使用する際に必要な AWS システムをモニタリングして、AWS システムが正常に実行されていることを確認します。これらのチェックでは、修復には AWS の関与が必要なインスタンスの根本的な問題が検出されます。システムステータスチェックが失敗した場合、AWS によって問題が修正されるのを待つか、自分自身で(たとえば、インスタンスを停止、再起動、終了、置換するなどによって)問題を解決できます。システムステータスチェックの失敗の原因となる問題には、次のようなものがあります。
  - ネットワーク接続の喪失
  - システム電源の喪失
  - 物理ホストのソフトウェアの問題
  - ネットワーク到達可能性に影響する、物理ホスト上のハードウェアの問題

詳細については、「[インスタンスのステータスチェック \(p. 628\)](#)」を参照してください。

- [インスタンスステータスのチェック] – 個々のインスタンスのソフトウェアとネットワークの設定をモニタリングします。これらのチェックでは、ユーザーが関与して修復する必要のある問題が検出されます。インスタンスステータスチェックが失敗した場合、通常はお客様ご自身で(インスタンスの再起動、オペレーティングシステムの修正など)問題を修復する必要があります。インスタンスステータスチェックの失敗の原因となる問題には、次のようなものがあります。
  - 失敗したシステムステータスチェック
  - 誤って設定されたネットワークまたは起動設定
  - メモリの枯渇
  - 破損したファイルシステム
  - 互換性のないカーネル

詳細については、「[インスタンスのステータスチェック \(p. 628\)](#)」を参照してください。

- [Amazon CloudWatch Alarms] – 指定された期間にわたって単一のメトリクスを監視し、複数の期間にわたり既定のしきい値に関連するメトリクス値に基づいて 1 つ以上のアクションを実行します。アクションは、Amazon Simple Notification Service(Amazon SNS) トピックまたは Amazon EC2 Auto Scaling ポリシーに送信される通知です。アラームは、持続している状態変化に対してのみアクションを呼び出します。CloudWatch アラームは、単に特定の状態にあるというだけでアクションを呼び出すわけではありません。状態が変わり、それが指定した数の期間にわたって持続している必要があります。詳細については、「[CloudWatch を使用したインスタンスのモニタリング \(p. 642\)](#)」を参照してください。

- Amazon CloudWatch Events - AWS サービスを自動化し、システムイベントに自動的に応答します。AWS サービスからのイベントはほぼリアルタイムに CloudWatch イベントに提供されます。ユーザーが記述したルールとイベントが一致したときに実行する自動化されたアクションを指定できます。詳細については、「[Amazon CloudWatch Eventsとは何ですか？](#)」を参照してください。
- [Amazon CloudWatch Logs] - Amazon EC2 インスタンス、AWS CloudTrail、またはその他のソースのログファイルの監視、保存、アクセスができます。詳細については、[Amazon CloudWatch Logs User Guide](#) を参照してください。
- [Amazon EC2 Monitoring Scripts] - インスタンスのメモリ、ディスク、スワップファイルの使用状況をモニタリングできる Perl スクリプトです。詳細については、「[Amazon EC2 Linux インスタンスのメモリとディスクのメトリクスのモニタリング](#)」を参照してください。
- [AWS Management Pack for System Center Operations Manager] – Amazon EC2 インスタンスと、これらのインスタンス内で稼働する Microsoft Windows または Linux オペレーティングシステムをリンクします。AWS マネジメントパックは Microsoft System Center Operations Manager 向けの拡張パックです。データセンターの指定されたコンピュータ（監視ノードと呼びます）と Amazon ウェブサービス API を使用して、AWS リソースに関する情報をリモートで検出して収集します。詳細については、「[Microsoft System Center 用 AWS Management Pack](#)」を参照してください。

## 手動モニタリングツール

Amazon EC2 のモニタリングにおけるもう 1 つの重要な部分は、モニタリングスクリプト、ステータスチェック、および CloudWatch アラームで網羅されていない項目を手動でモニタリングすることです。Amazon EC2 および CloudWatch のコンソールダッシュボードには、Amazon EC2 環境の状態が一目でわかるビューが表示されます。

- Amazon EC2 ダッシュボードには次の内容が表示されます。
  - リージョンごとのサービス状態とスケジュールされたイベント
  - インスタンスの状態
  - ステータスチェック
  - アラームステータス
  - インスタンスマトリクスの詳細（ナビゲーションペインで、[Instances] を選択し、インスタンスを選択して、[Monitoring] タブを選択します）
  - ボリュームメトリクスの詳細（ナビゲーションペインの [Volumes] を選択し、ボリュームを選択して、[Monitoring] タブを選択します）
- Amazon CloudWatch ダッシュボードには、次の内容が表示されます。
  - 現在のアラームとステータス
  - アラームとリソースのグラフ
  - サービス状態ステータス

さらに、CloudWatch を使用して次のことが行えます。

- Amazon EC2 モニタリングデータをグラフ化して、問題のトラブルシューティングを行い、傾向を確認する
- AWS リソースのすべてのメトリクスを検索して、参照する
- 問題があることを通知するアラームを作成/編集する
- アラームおよび AWS リソースが一目でわかる概要を表示する

## モニタリングのベストプラクティス

次に示すモニタリングのベストプラクティスを使用すると、Amazon EC2 のモニタリングタスクが容易になります。

- モニタリングの優先順位を設定し、小さな問題が大きな問題に発展する前に阻止します。
- AWS ソリューションのすべての部分からモニタリングデータを収集するモニタリング計画を作成し、実施すると、マルチポイント障害が発生した場合に、その障害をより簡単にデバッグできます。モニタリング計画には、少なくとも、次の質問に対する回答を盛り込む必要があります。
  - モニタリングの目的とは？
  - モニタリングの対象となるリソースとは？
  - これらのリソースをモニタリングする頻度は？
  - 使用するモニタリングツールは？
  - 誰がモニタリングタスクを実行しますか？
  - 誰が問題が発生したときに通知を受け取りますか？
- モニタリングタスクは可能な限り自動化します。
- EC2 インスタンスでログファイルを確認します。

## インスタンスのステータスのモニタリング

インスタンスのステータスをモニタリングして、インスタンスのステータスチェックや、インスタンスにスケジュールされたイベントを表示できます。

ステータスチェックでは、Amazon EC2 によって実行される自動化されたチェックからの情報が提供されます。これらの自動化されたチェックは、特定の問題がインスタンスに影響を与えていたかどうかを検出します。ステータスチェックの情報と、Amazon CloudWatch で提供されるデータによって、各インスタンスの詳細な動作状況を把握できます。

インスタンスに予定されている特定イベントのステータスも表示できます。イベントのステータスは、再起動やリタイヤなど、インスタンスに対して予定されている今後のアクティビティに関する情報を提供します。また、各イベントの予定開始予定時刻および終了時刻も提供されています。

### コンテンツ

- [インスタンスのステータスチェック \(p. 628\)](#)
- [インスタンスの予定されたイベント \(p. 633\)](#)

## インスタンスのステータスチェック

インスタンスのステータスのモニタリングでは、Amazon EC2 によってインスタンスによるアプリケーションの実行が妨げられるような問題が検出されたかどうかをすばやく判断できます。Amazon EC2 は、実行されている各 EC2 インスタンスについて自動化されたチェックを実行して、ハードウェアおよびソフトウェアの問題を識別します。これらのステータスチェックの結果を表示して、具体的で検出可能な問題を識別できます。このイベントステータスデータは、各インスタンス (pending、running、stopping) の状態について Amazon EC2 が既に提供している情報と、Amazon CloudWatch が監視している使用状況メトリクス (CPU 使用率、ネットワークトラフィック、ディスクアクティビティ) を補足するものです。

ステータスチェックは 1 分ごとに実行され、それぞれ成功または失敗のステータスが返ります。すべてのチェックが成功すると、インスタンス全体のステータスが OK になります。1つ以上のチェックが失敗すると、全体のステータスが impaired になります。ステータスチェックは Amazon EC2 に組み込まれています。そのため、無効にしたり、削除したりすることはできません。

ステータスチェックに失敗すると、ステータスチェックの対応する CloudWatch メトリクスは増加します。詳細については、「[ステータスチェックメトリクス \(p. 649\)](#)」を参照してください。このようなメトリクスを使用して、ステータスチェックの結果に基づいてトリガーされる CloudWatch アラームを作成することができます。たとえば、特定のインスタンスでステータスチェックが失敗したときに警告するアラームを作成できます。詳細については、「[ステータスチェックアラームの作成と編集 \(p. 631\)](#)」を参照してください。

また、Amazon EC2 インスタンスをモニタリングし、基になる問題によりインスタンスが正常に機能しなくなった場合に、自動的にインスタンスを復旧する Amazon CloudWatch アラームを作成できます。詳細については、「[インスタンスの復旧 \(p. 551\)](#)」を参照してください。

## コンテンツ

- [ステータスチェックの種類 \(p. 629\)](#)
- [ステータスチェックの表示 \(p. 629\)](#)
- [インスタンスステータスの報告 \(p. 631\)](#)
- [ステータスチェックアラームの作成と編集 \(p. 631\)](#)

## ステータスチェックの種類

ステータスチェックには、システムステータスチェックとインスタンスステータスチェックの 2 種類があります。

### システムステータスのチェック

インスタンスが実行されている AWS システムを監視します。これらのチェックでは、修復には AWS の関与が必要なインスタンスの根本的な問題が検出されます。システムステータスチェックが失敗した場合、AWS が問題を解決するのを待つか、自分で解決できるかを選択できます。Amazon EBS でバックアップされたインスタンスの場合は、インスタンスを自分で停止および起動することができます。通常、インスタンスは新しいホストに移行されます。インスタンスストアによってサポートされているインスタンスの場合、インスタンスを終了して置き換えることができます。

システムステータスチェックの失敗の原因となる問題の例を次に示します。

- ネットワーク接続の喪失
- システム電源の喪失
- 物理ホストのソフトウェアの問題
- ネットワーク到達可能性に影響する、物理ホスト上のハードウェアの問題

### インスタンスステータスのチェック

個々のインスタンスのソフトウェアとネットワークの設定をモニタリングします。Amazon EC2 は、ネットワークインターフェイス (NIC) にアドレス解決プロトコル (ARP) リクエストを送信することによってインスタンスの健全性をチェックします。これらのチェックでは、ユーザーが関与して修復する必要のある問題が検出されます。インスタンスステータスチェックが失敗した場合は通常、自分自身で（たとえば、インスタンスを再起動する、インスタンス設定を変更するなどによって）問題に対処する必要があります。

インスタンスステータスチェックの失敗の原因となる問題の例を次に示します。

- 失敗したシステムステータスチェック
- 正しくないネットワークまたは起動設定
- メモリの枯渇
- 破損したファイルシステム
- 互換性のないカーネル

## ステータスチェックの表示

Amazon EC2 では、いくつかの方法でステータスチェックを表示および操作できます。

## コンソールを使ったステータスの表示

AWS マネジメントコンソールを使用してステータスチェックを表示できます。

ステータスチェックを表示するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. Instances ページで、Status Checks 列には、各インスタンスの動作状況が表示されます。
4. 特定のインスタンスのステータスを表示するには、インスタンスを選択して、[Status Checks] タブを選択します。

The screenshot shows the AWS Management Console interface for an EC2 instance. At the top, there are tabs: Description, Status Checks (which is selected), Monitoring, and Tags. Below the tabs, a message says "Status checks detect problems that may impair this instance from running your applications. [Learn more](#) about status checks." A "Create Status Check Alarm" button is present. The main area is divided into two sections: "System Status Checks" and "Instance Status Checks". The "System Status Checks" section indicates a "System reachability check passed". The "Instance Status Checks" section shows a failure: "Instance reachability check failed at October 7, 2013 11:52:11 AM UTC+2 (16 minutes ago)". There is also a link to "Learn more about this issue". At the bottom, there's a note: "Submit feedback if our checks do not reflect your experience with this instance or if they do not detect the issues you are having. Please note that we will not respond to customer support issues reported via this form. Please post your issue on the [Developer Forums](#) or contact [AWS Support](#) if you need technical assistance with this instance."

ステータスチェックが失敗したインスタンスがあり、そのインスタンスに 20 分以上アクセスできない場合は、[AWS Support] を選択して、サポートのリクエストを送信してください。ご自分でシステムまたはインスタンスのステータスチェック失敗のトラブルシューティングを行う場合は、「[ステータスチェックに失敗したインスタンスのトラブルシューティング \(p. 1146\)](#)」を参照してください。

5. ステータスチェックの CloudWatch メトリクスを確認するには、インスタンスを選択後、[モニタリング] タブを選択します。次のメトリクスのグラフが表示されるまでスクロールします。
  - ステータスチェックに失敗(すべて)
  - ステータスチェックに失敗(インスタンス)
  - ステータスチェックに失敗(システム)

## コマンドラインを使用したステータスの表示

[describe-instance-status](#) (AWS CLI) コマンドを使用すると、実行中のインスタンスのステータスチェックを表示できます。

すべてのインスタンスのステータスを表示するには、次のコマンドを使用します。

```
aws ec2 describe-instance-status
```

インスタンスステータスが impaired であるすべてのインスタンスのステータスを取得するには、次のコマンドを使用します。

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

単一のインスタンスのステータスを取得するには、以下のコマンドを使用します。

```
aws ec2 describe-instance-status --instance-ids i-1234567890abcdef0
```

または、以下のコマンドを使用します。

- [Get-EC2InstanceState](#) (AWS Tools for Windows PowerShell)
- [DescribeInstanceState](#) (Amazon EC2 クエリ API)

ステータスチェックが失敗したインスタンスがある場合は、「[ステータスチェックに失敗したインスタンスのトラブルシューティング \(p. 1146\)](#)」を参照してください。

## インスタンスステータスの報告

ステータスが `impaired` と表示されていないインスタンスで問題が生じている場合や、障害のあるインスタンスで発生している問題に関する追加の詳細を AWS に送る場合は、フィードバックを送信することができます。

Amazon では、報告されたフィードバックを使用して、複数のお客様に影響する可能性のある問題を識別していますが、個々のアカウントの問題には返答しておりません。フィードバックをご提供いただいても、現在インスタンスに関して表示されているステータスチェックの結果は変わりません。

### コンソールを使用したステータスフィードバックの報告

インスタンスステータスを報告するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択して、[Status Checks] タブを選択し、[Submit feedback] を選択します。
4. [Report Instance Status] フォームに入力し、[Submit] を選択します。

### コマンドラインを使用したステータスフィードバックのレポート

障害のあるインスタンスのステータスに関するフィードバックを送信するには、次の `report-instance-status` (AWS CLI) コマンドを使用します。

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --reason-codes code
```

または、以下のコマンドを使用します。

- [Send-EC2InstanceState](#) (AWS Tools for Windows PowerShell)
- [ReportInstanceState](#) (Amazon EC2 クエリ API)

## ステータスチェックアラームの作成と編集

ステータスチェックメトリクス (p. 649) を使用して、インスタンスのステータスチェックに失敗したときに通知されるように CloudWatch アラームを作成することができます。

### コンソールを使用したステータスチェックアラームの作成

次の手順に従って、E メールで通知を送信するか、ステータスチェックに失敗したときにインスタンスを停止、終了、または回復するアラームを設定します。

ステータスチェックアラームを作成するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。

3. インスタンスを選択して、[Status Checks] タブを選択し、[Create Status Check Alarm] を選択します。
4. [Send a notification to] を選択します。既存の SNS トピックを選択し、[create topic] を選択して新しいトピックを作成します。新しいトピックを作成する場合、[With these recipients] に自分のメールアドレスを入力します。追加の受信者がいる場合はそのアドレスを、カンマで区切って入力します。
5. (オプション) [Take the action] を選択し、実行するアクションを選択します。
6. [Whenever] で、通知を受けるステータスチェックを選択します。

前のステップで [Recover this instance] を選択した場合、[Status Check Failed (System)] を選択します。

7. [For at least] ボックスで、評価する期間数を設定し、[consecutive periods] で、アラームをトリガーして E メールを送信するまでの評価の間隔を選択します。
8. (オプション) [Name of alarm] で、デフォルト名をアラームの別の名前に置き換えます。
9. [Create Alarm] を選択します。

#### Important

受信者のリストにメールアドレスを追加したか、トピックを新規作成した場合、Amazon SNS から追加した各メールアドレスにサブスクリプションの確認メールメッセージが送信されます。各受信者は、そのメッセージに記載されているリンクを選択してサブスクリプションを確認する必要があります。アラート通知は確認されたアドレスにのみ送信されます。

インスタンスステータスのアラームを変更する必要がある場合は、そのアラームを編集できます。

コンソールを使用してステータスチェックアラームを編集するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択して、[Actions]、[CloudWatch Monitoring]、および [Add/Edit Alarms] を選択します。
4. [Alarm Details] ダイアログボックスで、アラームの名前を選択します。
5. [Edit Alarm] ダイアログボックスで、希望する変更を行い、[Save] を選択します。

## AWS CLI を使用したステータスチェックアラームの作成

次の例では、インスタンスが少なくとも 2 つの連続する期間内にインスタンスチェックまたはシステムステータスチェックに失敗した場合、アラームが SNS トピックに通知 `arn:aws:sns:us-west-2:111122223333:my-sns-topic` を発行します。使用する CloudWatch メトリクスは `StatusCheckFailed` です。

AWS CLI を使用してステータスチェックアラームを作成するには

1. 既存の SNS トピックを選択するか、新しいキーペアを作成することができます。詳細については、『AWS Command Line Interface ユーザーガイド』の「[Amazon SNS での AWS CLI の使用](#)」を参照してください。
2. Amazon EC2 の使用可能な Amazon CloudWatch メトリクスを表示するには、`list-metrics` コマンドを使用します。

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

3. アラームを作成するには、次の `put-metric-alarm` コマンドを使用します。

```
aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-i-1234567890abcdef0 --metric-name StatusCheckFailed --namespace AWS/EC2 --
```

```
statistic Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 --unit Count --period 300 --evaluation-periods 2 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --alarm-actions arn:aws:sns:us-west-2:111122223333:my-sns-topic
```

期間は Amazon CloudWatch メトリクスが収集される期間(秒)です。この例では、60 秒に 5 分を乗算した 300 を使用します。評価期間は、メトリクスの値がしきい値と比較されなければならない連続した期間の数です。この例では 2 を使用します。アラームアクションは、このアラームがトリガーされたときに実行するアクションです。この例では、Amazon SNS を使用してメールを送信するようにアラームを設定します。

## インスタンスの予定されたイベント

AWS は、再起動、停止/開始、またはリタイアなど、インスタンスのイベントを予定できます。これらのイベントは頻繁には発生しません。インスタンスのいずれかが予定されたイベントの影響を受ける場合、予定されたイベントの前に AWS アカウントに関する連絡先情報を E メールアドレスに E メールが AWS から送信されます。この E メールは、開始日と終了日などのイベントの詳細を提供します。イベントによっては、イベントのタイミングを管理するアクションを実行することができます。

予定されたイベントに通知を受け取ることができるようにアカウントの連絡先情報を更新するには、「[アカウント設定](#)」ページを参照してください。

### コンテンツ

- [予定されたイベントのタイプ \(p. 633\)](#)
- [予定されたイベントの表示 \(p. 633\)](#)
- [停止またはリタイアが予定されているインスタンスの操作 \(p. 637\)](#)
- [再起動が予定されているインスタンスの操作 \(p. 638\)](#)
- [メンテナンスが予定されているインスタンスの操作 \(p. 641\)](#)

## 予定されたイベントのタイプ

Amazon EC2 は、インスタンスのイベントとして以下のタイプをサポートしています。イベントはスケジュールされた時間に発生します。

- **インスタンスの停止:** スケジュールされた時刻になると、インスタンスは停止します。再度起動すると、新しいホストに移行されます。Amazon EBS によってバックアップされるインスタンスにのみ適用されます。
- **Instance retirement (インスタンスのリタイア):** スケジュールされた時刻に、インスタンスは、Amazon EBS によってバックアップされると停止し、インスタンスストアによってバックアップされると削除されます。
- **インスタンスの再起動:** スケジュールされた時刻になると、インスタンスは再起動されます。
- **システムの再起動:** スケジュールされた時刻になると、インスタンスのホストは再起動されます。
- **[System maintenance]:** スケジュールされた時刻になると、インスタンスは、ネットワークメンテナンスまたは電源のメンテナンスの影響を一時的に受ける場合があります。

## 予定されたイベントの表示

予定されたイベントの通知を E メールで受信することに加え、以下のいずれかの方法を使用して予定されたイベントを確認できます。

## 新しいコンソール

コンソールを使用してインスタンスに予定されたイベントを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. スケジュールされたイベントは、次の画面で表示できます。
  - ナビゲーションペインの [Events] を選択します。イベントに関連付けられたリソースがすべて表示されます。リソース ID、リソースタイプ、アベイラビリティゾーン、イベントステータス、またはイベントタイプでフィルタリングできます。

The screenshot shows the 'Events (103)' page in the AWS EC2 console. At the top, there are three filters: 'Resource type: instance' (selected), 'Event status: Scheduled', and 'Event type: instance-stop'. Below the filters is a table header with columns: Resource ID, Event status, and Event type. A single event is listed: 'i-02c48fffba61cd16f' with status 'Scheduled' and type 'instance-stop'.

- または、ナビゲーションペインで [EC2 Dashboard] を選択します。イベントに関連付けられているすべてのリソースが、[Scheduled Events] に表示されます。

The screenshot shows the 'Scheduled events' section of the EC2 Dashboard. It displays information for the 'US East (N. Virginia)' region:

- 7 instance(s) have scheduled events
- 1 volume(s) are impaired

- 一部のイベントは影響を受けるリソースにも表示されます。たとえば、ナビゲーションペインの [Instances] を選択して、インスタンスを選択します。インスタンスに関連付けられたインスタンス停止またはインスタンスリタイアイベントがある場合、そのイベントが下のペインに表示されます。

Retiring: This instance is scheduled for retirement after February 12, 2020 at 12:00:00 AM UTC+2. ⓘ

## 古いコンソール

コンソールを使用してインスタンスに予定されたイベントを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. スケジュールされたイベントは、次の画面で表示できます。
  - ナビゲーションペインの [Events] を選択します。イベントに関連付けられたリソースがすべて表示されます。リソースタイプ、または特定のイベントのタイプでフィルタリングできます。リソースを選択すると、詳細を表示できます。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
予定されているイベント

Filter: All resource types ▾ All event types ▾ Ongoing and scheduled ▾			
Resource Name	Resource Type	Resource Id	Event Type
my-instance	instance	i-c3870335	instance-stop

Event: i-c3870335

Availability Zone us-west-2a  
Event type instance-stop  
Event status Scheduled  
Description The instance is running on degraded hardware  
Start time May 22, 2015 at 5:00:00 PM UTC-7  
End time

- または、ナビゲーションペインで [EC2 Dashboard] を選択します。イベントに関連付けられているすべてのリソースが、[Scheduled Events] に表示されます。

#### Scheduled Events

US West (Oregon):

1 instances have scheduled events

- 一部のイベントは影響を受けるリソースにも表示されます。たとえば、ナビゲーションペインの [Instances] を選択して、インスタンスを選択します。インスタンスに関連付けられたインスタンス停止またはインスタンスリタイアイベントがある場合、そのイベントが下のペインに表示されます。

**A** Retiring: This instance is scheduled for retirement after May 22, 2015 at 5:00:00 PM UTC-7. i

## AWS CLI

AWS CLI を使用してインスタンスに予定されたイベントを表示するには

`describe-instance-status` コマンドを使用します。

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0 --query  
"InstanceStatuses[].[Events]"
```

以下の出力例は、再起動イベントを示しています。

```
[  
  "Events": [  
    {  
      "InstanceEventId": "instance-event-0d59937288b749b32",  
      "Code": "system-reboot",  
      "Description": "The instance is scheduled for a reboot",  
      "NotAfter": "2019-03-15T22:00:00.000Z",  
      "NotBefore": "2019-03-14T20:00:00.000Z",  
      "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"  
    }  
  ]  
]
```

インスタンスのリタイアイベントを示す出力例を次に示します。

```
[
```

```
"Events": [
  {
    "InstanceEventId": "instance-event-0e439355b779n26",
    "Code": "instance-stop",
    "Description": "The instance is running on degraded hardware",
    "NotBefore": "2015-05-23T00:00:00.000Z"
  }
]
```

## PowerShell

AWS Tools for Windows PowerShell を使用してインスタンスに予定されたイベントを表示するには次の [Get-EC2InstanceState](#) コマンドを使用します。

```
PS C:\> (Get-EC2InstanceState -InstanceId i-1234567890abcdef0).Events
```

インスタンスのリタイアイベントを示す出力例を次に示します。

```
Code      : instance-stop
Description : The instance is running on degraded hardware
NotBefore : 5/23/2015 12:00:00 AM
```

## Instance metadata

インスタンスマタデータを使用してインスタンスに予定されたイベントを表示するには

インスタンスのアクティブなメンテナンスイベントに関する情報は、インスタンスマタデータサービスバージョン 2 または インスタンスマタデータサービスバージョン 1 を使用して[インスタンスマタデータ](#) (p. 593)から取得できます。

### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/scheduled
```

以下は、予定されたシステムの再起動イベントに関する情報を JSON 形式で出力した例です。

```
[
  {
    "NotBefore" : "21 Jan 2019 09:00:43 GMT",
    "Code" : "system-reboot",
    "Description" : "scheduled reboot",
    "EventId" : "instance-event-0d59937288b749b32",
    "NotAfter" : "21 Jan 2019 09:17:23 GMT",
    "State" : "active"
  }
]
```

インスタンスマタデータを使用して、インスタンスの完了またはキャンセルされたイベントのイベント履歴を表示するには

インスタンスの完了済みまたはキャンセル済みイベントに関する情報は、インスタンスマタデータサービスバージョン 2 または インスタンスマタデータサービスバージョン 1 を使用して [インスタンスマタデータ \(p. 593\)](#) から取得できます。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/events/maintenance/history
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/events/maintenance/history
```

以下は、取り消されたシステム再起動イベントおよび完了したシステム再起動イベントに関する情報を JSON 形式で出力した例です。

```
[  
 {  
     "NotBefore" : "21 Jan 2019 09:00:43 GMT",  
     "Code" : "system-reboot",  
     "Description" : "[Canceled] scheduled reboot",  
     "EventId" : "instance-event-0d59937288b749b32",  
     "NotAfter" : "21 Jan 2019 09:17:23 GMT",  
     "State" : "canceled"  
 },  
 {  
     "NotBefore" : "29 Jan 2019 09:00:43 GMT",  
     "Code" : "system-reboot",  
     "Description" : "[Completed] scheduled reboot",  
     "EventId" : "instance-event-0d59937288b749b32",  
     "NotAfter" : "29 Jan 2019 09:17:23 GMT",  
     "State" : "completed"  
 }  
 ]
```

## 停止またはリタイアが予定されているインスタンスの操作

AWS は、インスタンスの基盤となるホストの回復不能な障害を検出すると、インスタンスのルートデバイスのタイプに応じて、インスタンスの停止または削除を予定します。ルートデバイスが EBS ボリュームの場合、インスタンスが停止するように予定されます。ルートデバイスがインスタンスストアボリュームの場合、インスタンスは終了するように予定されます。詳細については、「[インスタンスのリタイア \(p. 543\)](#)」を参照してください。

#### Important

インスタンスストアボリュームに格納されているデータはいずれも、インスタンスが停止または終了されると失われます。これには、EBS ボリュームをルートデバイスとするインスタンスにアタッチされたインスタンスストアボリュームも含まれます。インスタンスが停止または終了される前に、後で必要となるインスタンスストアボリュームからデータを必ず保存しておきます。

#### Amazon EBS によりバックアップされたインスタンスのアクション

インスタンスが予定どおりに停止されるのを待機できます。または、インスタンスを自分で停止および起動して、新しいホストに移行することもできます。インスタンスが停止したときにインスタンス設定を変

更する方法に加えて、インスタンスの停止についての詳細は、「[インスタンスの停止と起動 \(p. 529\)](#)」を参照してください。

スケジュールされたインスタンスの停止イベントに対応した、即時の停止と開始を自動化することができます。詳細については、AWS Health ユーザーガイドの「[EC2 インスタンスのアクションの自動化](#)」を参照してください。

#### インスタンスストアによりバックアップされたインスタンスのアクション

最新の AMI から代替インスタンスを起動し、インスタンスの削除を予定する前に必要なすべてのデータを代替インスタンスに移行することをお勧めします。その後、元のインスタンスを終了するか、予定どおりに終了されるのを待機することができます。

## 再起動が予定されているインスタンスの操作

AWS は、更新のインストールや基盤となるホストのメンテナンスなどのタスクを実行する必要があるとき、インスタンスまたは基盤となるホストの再起動を予定できます。都合に合わせて指定する日付と時刻にインスタンスが再起動するように、[ほとんどの再起動イベントを再スケジュール \(p. 639\)](#)できます。

リンクされた [EC2-Classic インスタンス \(p. 815\)](#) を停止した場合、インスタンスは VPC から自動的にリンクが解除され、VPC セキュリティグループはインスタンスとの関連付けが失われます。インスタンスを再起動した後、インスタンスを VPC に再びリンクできます。

### 再起動イベントタイプを表示する

次のいずれかの方法を使用して、再起動イベントがインスタンスの再起動またはシステムの再起動であるかを確認できます。

#### 新しいコンソール

コンソールを使用して予定された再起動イベントのタイプを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Events] を選択します。
3. フィルターリストから [リソースタイプ: インスタンス] を選択します。
4. インスタンスごとに、[イベントタイプ] 列の値を表示します。値は system-reboot または instance-reboot のいずれかです。

#### 古いコンソール

コンソールを使用して予定された再起動イベントのタイプを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Events] を選択します。
3. フィルターリストから [インスタンスリソース] を選択します。
4. 各インスタンスで、[イベントタイプ] 列の値を表示します。値は system-reboot または instance-reboot のいずれかです。

#### AWS CLI

AWS CLI を使用して予定された再起動イベントのタイプを表示するには

[describe-instance-status](#) コマンドを使用します。

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

スケジュールされた再起動イベントでは、Code の値は system-reboot あるいは instance-reboot です。次の出力例は system-reboot イベントを示しています。

```
[  
  "Events": [  
    {  
      "InstanceEventId": "instance-event-0d59937288b749b32",  
      "Code": "system-reboot",  
      "Description": "The instance is scheduled for a reboot",  
      "NotAfter": "2019-03-14T22:00:00.000Z",  
      "NotBefore": "2019-03-14T20:00:00.000Z",  
      "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"  
    }  
  ]  
]
```

### インスタンスの再起動のアクション

予定されたメンテナンスウィンドウ内のインスタンスの再起動まで待機することも、都合に合わせた日付と時刻にインスタンスの再起動を[再スケジュール \(p. 639\)](#)することも、または都合のよい時間にインスタンスを手動で[再起動 \(p. 542\)](#)することもできます。

インスタンスが再起動されると、予定されたイベントがクリアになり、このイベントの説明が更新されます。基になるホストに対する保留中のメンテナンスが完了し、インスタンスが完全に起動したら、インスタンスの使用を再開できます。

### システムの再起動のアクション

システムを自分で再起動することはできません。予定されたメンテナンスウィンドウ中におけるシステムの再起動まで待機することも、都合に合わせた日付と時刻でシステムの再起動を[再スケジュール \(p. 639\)](#)することもできます。システムの再起動は通常数分で完了します。システムの再起動後、インスタンスの IP アドレスと DNS 名、およびローカルインスタンスストアボリュームのデータは保持されます。システムの再起動が完了すると、インスタンスに予定されているイベントはクリアされ、インスタンスのソフトウェアが正常に動作していることを確認できます。

または、インスタンスのメンテナンス時間を変更する必要があり、システムの再起動を再スケジュールできない場合は、Amazon EBS-backed インスタンスを停止して再起動すると、新しいホストに移行できます。ただし、ローカルインスタンスストアボリュームのデータは保持されません。また、スケジュールされたシステム再起動イベントに対応した、インスタンスの即時の停止と開始を自動化することができます。詳細については、AWS Health ユーザーガイドの「[EC2 インスタンスのアクションの自動化](#)」を参照してください。Instance Store-Backed インスタンスでシステムの再起動を再スケジュールできない場合、最新の AMI から代替インスタンスを起動し、予定されたメンテナンス期間より前に必要なデータをすべて代替インスタンスに移行した後、元のインスタンスを削除できます。

## 再起動イベントの再スケジュール

再起動イベントを再スケジュールして、適切な特定の日時にインスタンスを再起動できます。期限が設定されているイベントのみを再スケジュールできます。[再起動イベントの再スケジュールに適用される制限 \(p. 641\)](#)は他にもあります。

再起動イベントは、次のいずれかの方法で再スケジュールできます。

### 新しいコンソール

コンソールを使用して再起動イベントを再スケジュールするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインの [Events] を選択します。
3. フィルターリストから [リソースタイプ: インスタンス] を選択します。
4. 1つ以上のインスタンスを選択し、[アクション]、[Schedule event] の順に選択します。

[期限] でイベント期限を設定したイベントのみを再スケジュールできます。選択したイベントのいずれかに期限がない場合、[アクション]、[Schedule event] は無効になります。

5. [New start time] に、再起動の新しい日時を入力します。新しい日時は、[Event deadline] より前に設定する必要があります。
6. [Save] を選択します。

更新されたイベント開始時刻がコンソールに反映されるまで、1~2 分かかることがあります。

## 古いコンソール

コンソールを使用して再起動イベントを再スケジュールするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Events] を選択します。
3. フィルターリストから [インスタンスリソース] を選択します。
4. 1つ以上のインスタンスを選択したら、[アクション]、[Schedule Event (イベントのスケジュール)] を選択します。

[Event Deadline (イベント期限)] の値で示されるイベント期限があるイベントのみ、再スケジュールできます。

5. [Event start time (イベント開始時刻)] に再起動する新しい日付と時刻を入力します。新しい日時は、[Event Deadline] より前に設定する必要があります。
6. [Schedule Event (イベントのスケジュール)] を選択します。

更新されたイベント開始時刻がコンソールに反映されるまで、1~2 分かかることがあります。

## AWS CLI

AWS CLI を使用して再起動イベントを再スケジュールするには

1. NotBeforeDeadline の値で示されるイベント期限があるイベントのみ、再スケジュールできます。[describe-instance-status](#) コマンドを使用して NotBeforeDeadline パラメータ値を表示します。

```
aws ec2 describe-instance-status --instance-id i-1234567890abcdef0
```

次の出力例は、NotBeforeDeadline に値があるため再スケジュールできる system-reboot イベントを示しています。

```
[{"Events": [{"InstanceId": "i-1234567890abcdef0", "EventId": "instance-event-0d59937288b749b32", "Code": "system-reboot", "Description": "The instance is scheduled for a reboot", "NotAfter": "2019-03-14T22:00:00.000Z", "NotBefore": "2019-03-14T20:00:00.000Z", "NotBeforeDeadline": "2019-04-05T11:00:00.000Z"}]}
```

- [ ]
2. イベントを再スケジュールするには、[modify-instance-event-start-time](#) コマンドを使用します。not-before パラメータを使用して新しいイベント開始時刻を指定します。新しいイベント開始時刻は、NotBeforeDeadline より前にする必要があります。

```
aws ec2 modify-instance-event-start-time --instance-id i-1234567890abcdef0
--instance-event-id instance-event-0d59937288b749b32 --not-
before 2019-03-25T10:00:00.000
```

[describe-instance-status](#) コマンドが更新された not-before パラメータ値を返すまでに、1~2 分かかることがあります。

### 再起動イベントの制限

- イベント期限がある再起動イベントのみ再スケジュールできます。イベントは、イベント期限日まで再スケジュールできます。コンソールの [期限] 列と AWS CLI の NotBeforeDeadline フィールドは、イベントに期限が設定されていることを示します。
- まだ開始していない再起動イベントのみ再スケジュールできます。コンソールの [開始時刻] 列と AWS CLI の NotBefore フィールドは、イベントの開始時刻を示します。あと 5 分で開始するようにスケジュールされている再起動イベントは、再スケジュールできません。
- 新しいイベント開始時刻は、現在の時刻から少なくとも 60 分後にする必要があります。
- コンソールを使用して複数のイベントを再スケジュールすると、イベント期限は最も早い期限日のイベントによって決定されます。

## メンテナンスが予定されているインスタンスの操作

AWS は、インスタンスの基盤となるホストをメンテナンスする必要があるときに、インスタンスのメンテナンスを予定します。2 種類のメンテナンスイベントがあります。1 つはネットワークメンテナンスで、もう 1 つは電源のメンテナンスです。

ネットワークメンテナンス中は、短い期間、予定されたインスタンスのネットワーク接続が切断されます。メンテナンスが終了すると、インスタンスとの通常のネットワーク接続が回復します。

電源のメンテナンス中は、短い期間、予定されたインスタンスはオフラインになり、その後再起動されます。再起動されると、インスタンスの設定内容はすべて維持されます。

インスタンスが再起動したら（通常、数分かかります）、アプリケーションが正常に動作していることを確認します。この時点では、インスタンスにスケジュールされたイベントは残っていません。残っている場合は、スケジュールされたイベントの説明の先頭に [Completed] と表示されます。インスタンスのステータス説明が更新するのに、最大で 1 時間ほどかかる場合があります。完了したメンテナンスイベントは、最長で 1 週間、Amazon EC2 コンソールのダッシュボードに表示されます。

### Amazon EBS によりバックアップされたインスタンスのアクション

メンテナンスが予定どおりに実行されるのを待機できます。または、インスタンスを停止および起動して、新しいホストに移行することもできます。インスタンスが停止したときにインスタンス設定を変更する方法に加えて、インスタンスの停止についての詳細は、「[インスタンスの停止と起動 \(p. 529\)](#)」を参照してください。

スケジュールされたメンテナンスイベントに対応した、即時の停止と開始を自動化することができます。詳細については、AWS Health ユーザーガイドの「[EC2 インスタンスのアクションの自動化](#)」を参照してください。

### インスタンスストアによりバックアップされたインスタンスのアクション

メンテナンスが予定どおりに実行されるのを待機できます。または、予定されたメンテナンス期間中に通常の運用を維持する場合、最新の AMI から代替インスタンスを起動し、予定されたメンテナンス期間より前に必要なデータをすべて代替インスタンスに移行した後、元のインスタンスを終了できます。

## CloudWatch を使用したインスタンスのモニタリング

Amazon CloudWatch を使用してインスタンスをモニタリングすることで、Amazon EC2 から未加工データを収集し、リアルタイムに近い読み取り可能なメトリクスに加工することができます。これらの統計は 15 か月間記録されるため、履歴情報にアクセスしてウェブアプリケーションやサービスの動作をより的確に把握できます。

デフォルトでは、Amazon EC2 は 5 分ごとにメトリクスデータを CloudWatch に送信します。1 分ごとにインスタンスのメトリクスデータを CloudWatch に送信するには、インスタンスで詳細モニタリングを有効にできます。詳細については、「[インスタンスの詳細モニタリングの有効化または無効化 \(p. 642\)](#)」を参照してください。

Amazon EC2 コンソールには、Amazon CloudWatch の未加工データに基づいて一連のグラフが表示されます。必要に応じて、コンソールのグラフではなく Amazon CloudWatch からインスタンスのデータを取得することもできます。

Amazon CloudWatch の詳細については、「[Amazon CloudWatch ユーザーガイド](#)」を参照してください。

### コンテンツ

- [インスタンスの詳細モニタリングの有効化または無効化 \(p. 642\)](#)
- [インスタンスの利用可能な CloudWatch メトリクスのリスト表示 \(p. 644\)](#)
- [インスタンスのメトリクスの統計情報の取得 \(p. 654\)](#)
- [インスタンスのグラフメトリクス \(p. 662\)](#)
- [インスタンスの CloudWatch アラームの作成 \(p. 662\)](#)
- [インスタンスを停止、終了、再起動、または復旧するアラームを作成する \(p. 663\)](#)

## インスタンスの詳細モニタリングの有効化または無効化

デフォルトでは、インスタンスで基本モニタリングが有効になります。オプションで詳細モニタリングを有効にできます。詳細モニタリングを有効にすると、Amazon EC2 コンソールに、インスタンスの 1 分ごとのモニタリンググラフが表示されます。次の表では、インスタンスの基本モニタリングと詳細モニタリングについて説明します。

モニタリングタイプ	説明
基本	データは自動的に 5 分間無料で取得できます。
詳細	1 分間のデータを取得できます。追加料金がかかります。このレベルのデータを取得するには、インスタンスのデータ取得を明確に有効にする必要があります。詳細モニタリングを有効にしたインスタンスでは、同様のインスタンスグループの集約データを取得することもできます。

モニタリングタイプ	説明
	料金表の詳細については、「 <a href="#">Amazon CloudWatch 製品ページ</a> 」を参照してください。

## 詳細モニタリングの有効化

インスタンスが実行または停止された後で、起動時にインスタンスの詳細モニタリングを有効にできます。インスタンスで詳細モニタリングを有効にしても、そのインスタンスに接続されている EBS ボリュームのモニタリングには影響しません。詳細については、「[Amazon EBS の Amazon CloudWatch メトリクス \(p. 1060\)](#)」を参照してください。

既存のインスタンスの詳細モニタリングを有効にするには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択して、[Actions]、[CloudWatch Monitoring]、[Enable Detailed Monitoring] を選択します。
4. [詳細モニタリングを有効化] ダイアログボックスで、[Yes, Enable] を選択します。
5. [Close] を選択します。

インスタンスの起動時に詳細モニタリングを有効にするには (コンソール)

AWS マネジメントコンソールを使用してインスタンスを起動する場合は、[Configure Instance Details] ページの [Monitoring] チェックボックスをオンにします。

既存のインスタンスの詳細モニタリングを有効にするには (AWS CLI)

次の `monitor-instances` コマンドを使用して、指定したインスタンスの詳細モニタリングを有効にします。

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

インスタンスの起動時に詳細モニタリングを有効にするには (AWS CLI)

`run-instances` コマンドを `--monitoring` フラグとともに使用して詳細モニタリングを有効にします。

```
aws ec2 run-instances --image-id ami-09092360 --monitoring Enabled=true...
```

## 詳細モニタリングの無効化

インスタンスが実行または停止された後で、起動時にインスタンスの詳細モニタリングを無効にできます。

詳細モニタリングを無効にするには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択して、[Actions]、[CloudWatch Monitoring]、[Disable Detailed Monitoring] を選択します。
4. [詳細モニタリングを無効化] ダイアログボックスで、[Yes, Disable] を選択します。
5. [Close] を選択します。

詳細モニタリングを無効にするには (AWS CLI)

次の `unmonitor-instances` コマンドを使用して、指定したインスタンスの詳細モニタリングを無効にします。

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

## インスタンスの利用可能な CloudWatch メトリクスのリスト表示

Amazon EC2 はメトリクスを Amazon CloudWatch に送信します。AWS マネジメントコンソール、AWS CLI、または API を使用して、Amazon EC2 が CloudWatch に送信するメトリクスをリスト表示できます。デフォルトで、各データポイントではインスタンスのアクティビティの開始後 5 分間が対象となります。詳細モニタリングを有効にした場合、各データポイントは開始後 1 分間のアクティビティを対象とします。

これらのメトリクスの統計の取得については、「[インスタンスのメトリクスの統計情報の取得 \(p. 654\)](#)」を参照してください。

### コンテンツ

- [インスタンスマトリクス \(p. 644\)](#)
- [CPU クレジットメトリクス \(p. 646\)](#)
- [Nitro ベースのインスタンスの Amazon EBS メトリクス \(p. 648\)](#)
- [ステータスチェックメトリクス \(p. 649\)](#)
- [トラフィックミラーリングのメトリクス \(p. 650\)](#)
- [Amazon EC2 メトリクスディメンション \(p. 650\)](#)
- [Amazon EC2 使用状況メトリクス \(p. 651\)](#)
- [コンソールを使用してメトリクスをリスト表示する \(p. 652\)](#)
- [AWS CLI を使用してメトリクスをリスト表示する \(p. 653\)](#)

## インスタンスマトリクス

AWS/EC2 名前空間には、次のインスタンスマトリクスが含まれます。

メトリクス	説明
CPUUtilization	<p>割り当てられた EC2 コンピュートユニットのうち、現在インスタンス上で使用しているものの比率。このメトリクスによって、選択したインスタンスでアプリケーションを実行するのに必要な処理能力を特定できます。</p> <p>インスタンスタイプによっては、インスタンスがフルプロセッサコアに割り当てられていない場合に、オペレーティングシステムのツールが CloudWatch よりも低い比率を示す場合があります。</p> <p>単位: パーセント</p>
DiskReadOps	<p>指定された期間にインスタンスで利用できるすべてのインスタンストアボリュームでの、完了した読み取り操作。</p> <p>その期間の 1 秒あたりの I/O 操作回数 (IOPS) の平均を算出するには、その期間の操作回数の合計をその期間の秒数で割ります。</p> <p>インスタンストアボリュームがない場合は、値が 0 であるか、メトリクスがレポートされません。</p>

メトリクス	説明
	単位: Count
DiskWriteOps	<p>指定された期間にインスタンスで利用できるすべてのインスタンストアボリュームへの、完了した書き込み操作。</p> <p>その期間の 1 秒あたりの I/O 操作回数 (IOPS) の平均を算出するには、その期間の操作回数の合計をその期間の秒数で割ります。</p> <p>インスタンストアボリュームがない場合は、値が 0 であるか、メトリクスがレポートされません。</p> <p>単位: Count</p>
DiskReadBytes	<p>インスタンスで利用できるすべてのインスタンストアボリュームから読み取られたバイト数。</p> <p>このメトリクスを使用すると、このインスタンスのハードディスクからアプリケーションが読み取るデータの量がわかります。これを利用すると、アプリケーションの速度がわかります。</p> <p>報告された数は、期間中に受信されたバイト数です。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算してバイト/秒を求めることができます。詳細 (1 分) モニタリングを使用している場合は、この数を 60 で除算します。</p> <p>インスタンストアボリュームがない場合は、値が 0 であるか、メトリクスがレポートされません。</p> <p>単位: バイト</p>
DiskWriteBytes	<p>インスタンスで利用できるすべてのインスタンストアボリュームに書き込まれたバイト数。</p> <p>このメトリクスを使用すると、このインスタンスのハードディスクにアプリケーションが書き込むデータの量がわかります。これを利用すると、アプリケーションの速度がわかります。</p> <p>報告された数は、期間中に受信されたバイト数です。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算してバイト/秒を求めることができます。詳細 (1 分) モニタリングを使用している場合は、この数を 60 で除算します。</p> <p>インスタンストアボリュームがない場合は、値が 0 であるか、メトリクスがレポートされません。</p> <p>単位: バイト</p>
NetworkIn	<p>すべてのネットワークインターフェイスでの、このインスタンスによって受信されたバイトの数。このメトリクスは、1 つのインスタンスへの受信ネットワークフィックの量を表しています。</p> <p>報告された数は、期間中に受信されたバイト数です。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算してバイト/秒を求めることができます。詳細 (1 分) モニタリングを使用している場合は、この数を 60 で除算します。</p> <p>単位: バイト</p>

メトリクス	説明
NetworkOut	<p>すべてのネットワークインターフェイスでの、このインスタンスから送信されたバイトの数。このメトリクスは、1つのインスタンスからの送信ネットワークトラフィックの量を表しています。</p> <p>報告された数は、期間中に送信されたバイト数です。基本(5分)モニタリングを使用している場合、この数を300で除算してバイト/秒を求めることができます。詳細(1分)モニタリングを使用している場合は、この数を60で除算します。</p> <p>単位: バイト</p>
NetworkPacketsIn	<p>すべてのネットワークインターフェイスでの、このインスタンスによって受信されたパケットの数。このメトリクスは、受信トラフィックのボリュームを单一インスタンスでのパケット数として識別します。このメトリクスは基本モニタリング専用です。</p> <p>単位: Count</p> <p>統計: Minimum、Maximum、Average</p>
NetworkPacketsOut	<p>すべてのネットワークインターフェイスでの、このインスタンスから送信されたパケットの数。このメトリクスは、送信トラフィックのボリュームを单一インスタンスでのパケット数として識別します。このメトリクスは基本モニタリング専用です。</p> <p>単位: Count</p> <p>統計: Minimum、Maximum、Average</p>
MetadataNoToken	<p>トークンを使用しないメソッドを使用してインスタンスマタデータサービスに正常にアクセスした回数。</p> <p>このメトリクスにより、トークンを使用しないインスタンスマタデータサービスバージョン1を使用してインスタンスマタデータにアクセスするプロセスがあるかどうかがわかります。すべてのリクエストがトークン支援のセッション(インスタンスマタデータサービスバージョン2)を使用している場合、値は0になります。詳細については、「<a href="#">インスタンスマタデータサービスバージョン2使用への移行(p. 596)</a>」を参照してください。</p> <p>単位: カウント</p>

## CPU クレジットメトリクス

AWS/EC2名前空間は、バーストパフォーマンスインスタンス (p. 199) の以下の CPU クレジットメトリクスを含みます。

メトリクス	説明
CPUTCreditUsage	CPU 使用率に関してインスタンスで消費される CPU クレジットの数。1つの CPU クレジットは、1個の vCPU が 100% の使用率で1分間実行されること、または、vCPU、使用率、時間の同等の組み合わせ(たとえば、1個の vCPU が 50% の使用率で2分間実行されるか、2個の vCPU が 25% の使用率で2分間実行される)に相当します。

メトリクス	説明
	<p>CPU クレジットメトリクスは、5 分間隔でのみ利用可能です。5 分を超える期間を指定する場合は、Sum 統計の代わりに Average 統計を使用します。</p> <p>単位: クレジット (vCPU 分)</p>
CPUCreditBalance	<p>インスタンスが起動または開始後に蓄積した獲得 CPU クレジットの数。T2 スタンダードの場合、CPUCreditBalance には蓄積された起動クレジットの数も含まれます。</p> <p>クレジットは、獲得後にクレジット残高に蓄積され、消費されるとクレジット残高から削除されます。クレジットバランスには、インスタンスサイズによって決まる上限があります。制限に到達すると、獲得された新しいクレジットはすべて破棄されます。T2 スタンダードの場合、起動クレジットは制限に対してカウントされません。</p> <p>CPUCreditBalance のクレジットは、インスタンスがそのベースライン CPU 使用率を超えてバーストするために消費できます。</p> <p>インスタンスが実行中の場合、CPUCreditBalance のクレジットは期限切れになりません。T3 または T3a インスタンスが停止すると、CPUCreditBalance 値は 7 日間保持されます。その後、蓄積されたすべてのクレジットが失われます。T2 インスタンスが停止すると、CPUCreditBalance 値は保持されず、蓄積されたすべてのクレジットが失われます。</p> <p>CPU クレジットメトリクスは、5 分間隔でのみ利用可能です。</p> <p>単位: クレジット (vCPU 分)</p>
CPUSurplusCreditBalance	<p>CPUCreditBalance 値がゼロの場合に unlimited インスタンスによって消費された余剰クレジットの数。</p> <p>CPUSurplusCreditBalance 値は獲得した CPU クレジットによって支払われます。余剰クレジットの数が、24 時間にインスタンスが獲得できるクレジットの最大数を超えている場合、最大数を超えて消費された余剰クレジットに対しては料金が発生します。</p> <p>CPU クレジットメトリクスは、5 分間隔でのみ利用可能です。</p> <p>単位: クレジット (vCPU 分)</p>

メトリクス	説明
CPUSurplusCreditsCharged	<p>獲得 CPU クレジットにより支払われないために追加料金が発生した、消費された余剰クレジットの数。</p> <p>消費された余剰クレジットは、以下のいずれかの状況に当てはまる場合が発生します。</p> <ul style="list-style-type: none"> <li>消費された余剰クレジットが、インスタンスが 24 時間に獲得できる最大クレジット数を超えており、最大数を越えて消費された余剰クレジットは、時間の最後に課金されます。</li> <li>インスタンスが停止または終了した。</li> <li>インスタンスは <code>unlimited</code> から <code>standard</code> に切り替わります。</li> </ul> <p>CPU クレジットメトリクスは、5 分間隔でのみ利用可能です。</p> <p>単位: クレジット (vCPU 分)</p>

## Nitro ベースのインスタンスの Amazon EBS メトリクス

AWS/EC2 名前空間には、ペアメタルインスタンスではない Nitro ベースのインスタンス用の次の Amazon EBS メトリクスが含まれます。Nitro ベースのインスタンスタイプのリストについては、「[Nitro ベースのインスタンス \(p. 187\)](#)」を参照してください。

Nitro ベースのインスタンスのメトリクス値は常に整数 (0 と正の整数) で、Xen ベースのインスタンスの値は小数をサポートしています。したがって、Nitro ベースインスタンスでインスタンスの CPU 使用率が低い場合は、0 に切り下げられて表示される場合があります。

メトリクス	説明
EBSReadOps	<p>指定された期間にインスタンスに接続されたすべての Amazon EBS ボリュームからの、完了した読み込みオペレーション。</p> <p>その期間の 1 秒あたりの読み込み I/O 操作回数 (読み込み IOPS) の平均を算出するには、その期間の操作回数の合計をその期間の秒数で割ります。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算して読み込み IOPS を計算することができます。詳細 (1 分) モニタリングを使用している場合は、この数を 60 で除算します。</p> <p>単位: 個</p>
EBSWriteOps	<p>指定された期間にインスタンスに接続されたすべての EBS ボリュームからの、完了した書き込み操作。</p> <p>その期間の 1 秒あたりの書き込み I/O 操作回数 (書き込み IOPS) の平均を算出するには、その期間の操作回数の合計をその期間の秒数で割ります。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算して書き込み IOPS を計算することができます。詳細 (1 分) モニタリングを使用している場合は、この数を 60 で除算します。</p>

メトリクス	説明
	単位: 個
EBSReadBytes	<p>指定した期間内にインスタンスに接続されたすべての EBS ボリュームから読み取られたバイト数。</p> <p>報告された数は、期間中に読み取られたバイト数です。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算して読み込みバイト/秒を求めることができます。詳細 (1 分) モニタリングを使用している場合は、この数を 60 で除算します。</p> <p>単位: バイト</p>
EBSWriteBytes	<p>指定した期間内にインスタンスに接続されたすべての EBS ボリュームに書き込まれたバイト数。</p> <p>報告された数は、期間中に書き込まれたバイト数です。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算して書き込みバイト/秒求めることができます。詳細 (1 分) モニタリングを使用している場合は、この数を 60 で除算します。</p> <p>単位: バイト</p>
EBSIOBalance%	<p>小さいインスタンスサイズにのみ使用できます。バーストバケットの I/O 残りクレジットの割合に関する情報を提供します。このメトリクスは基本モニタリング専用です。</p> <p>Sum 統計は、このメトリクスに該当しません。</p> <p>単位: パーセント</p>
EBSByteBalance%	<p>小さいインスタンスサイズにのみ使用できます。バーストバケットのスループット残りクレジットの割合に関する情報を提供します。このメトリクスは基本モニタリング専用です。</p> <p>Sum 統計は、このメトリクスに該当しません。</p> <p>単位: パーセント</p>

EBS ボリューム用に提供されるメトリクスの詳細については、「[Amazon EBS のメトリクス \(p. 1060\)](#)」を参照してください。スポットフリートに提供されるメトリクスの詳細については、「[スポットフリートの CloudWatch メトリクス \(p. 371\)](#)」を参照してください。

## ステータスチェックメトリクス

AWS/EC2 名前空間には、次のステータスチェックメトリクスが含まれます。デフォルトでは、ステータスチェックメトリクスは無料で 1 分の頻度で利用できます。新しく起動したインスタンスの場合、ステータスチェックメトリクスデータは、インスタンスが初期化状態を完了した後でのみ使用できます（インスタンスが実行中の状態になってから数分以内）。EC2 のステータスチェックの詳細については、[インスタンスのステータスチェック](#)を参照してください。

メトリクス	説明
StatusCheckFailed	<p>インスタンスが過去 1 分間にインスタンスのステータスチェックとシステムステータスチェックの両方に合格したかどうかを報告します。</p> <p>このメトリクスは 0 (合格) または 1 (失敗) となります。</p> <p>デフォルトでは、このメトリクスは無料で 1 分の頻度で利用できます。</p> <p>単位: Count</p>
StatusCheckFailed_Instance	<p>最近 1 分間にインスタンスが インスタンスステータスチェックに成功したかどうかを報告します。</p> <p>このメトリクスは 0 (合格) または 1 (失敗) となります。</p> <p>デフォルトでは、このメトリクスは無料で 1 分の頻度で利用できます。</p> <p>単位: Count</p>
StatusCheckFailed_System	<p>最近 1 分間にインスタンスが システムステータスチェックに成功したかどうかを報告します。</p> <p>このメトリクスは 0 (合格) または 1 (失敗) となります。</p> <p>デフォルトでは、このメトリクスは無料で 1 分の頻度で利用できます。</p> <p>単位: カウント</p>

## トラフィックミラーリングのメトリクス

AWS/EC2 名前空間には、ミラートラフィックのメトリクスが含まれます。詳細については、Amazon VPC トラフィックミラーリングガイドの「[Amazon CloudWatch によるミラートラフィックのモニタリング](#)」を参照してください。

## Amazon EC2 メトリクスディメンション

次のディメンションを使用して、前のテーブルに示したメトリクスを絞り込むことができます。

ディメンション	説明
AutoScalingGroupName	このディメンションを指定すると、リクエストしたデータがファイルタリングされて、指定したキャパシティーグループ内のインスタンスのものだけになります。Auto Scaling グループは、Auto Scaling を使用する場合に定義するインスタンスのコレクションです。このディメンションを Amazon EC2 のメトリクスに対して使用できるのは、インスタンスが Auto Scaling グループ内にあるときに限られます。詳細モニタリングまたは基本モニタリングが有効になっているインスタンスに対して使用できます。
ImageId	このディメンションを指定すると、リクエストしたデータがファイルタリングされて、この Amazon EC2 Amazon Machine Image (AMI)

ディメンション	説明
	を実行しているインスタンスのものだけになります。詳細モニタリングが有効になっているインスタンスに対して使用できます。
InstanceId	このディメンションを指定すると、リクエストしたデータがフィルタリングされて、指定のインスタンスのものだけになります。これを利用すると、どのインスタンスからのデータをモニタリングするかを指定できます。
InstanceType	このディメンションを指定すると、リクエストしたデータがフィルタリングされて、指定のインスタンスタイプで実行されているインスタンスのものだけになります。これを利用すると、実行されているインスタンスのタイプでデータを分類することができます。たとえば、m1.small インスタンスと m1.large インスタンスのデータを比較して、アプリケーションに対するビジネス価値はどちらが上かを判断します。詳細モニタリングが有効になっているインスタンスに対して使用できます。

## Amazon EC2 使用状況メトリクス

CloudWatch 使用状況メトリクスを使用して、アカウントのリソースの使用状況を把握できます。これらのメトリクスを使用して、CloudWatch グラフやダッシュボードで現在のサービスの使用状況を可視化できます。

Amazon EC2 使用状況メトリクスは、AWS のサービスクォータに対応しています。使用量がサービスクォータに近づいたときに警告するアラームを設定することもできます。サービスクォータと CloudWatch の統合の詳細については、「[サービスクォータの統合と使用状況のメトリクス](#)」を参照してください。

Amazon EC2 は、AWS/Usage 名前空間に以下のメトリクスを公開します。

メトリクス	説明
ResourceCount	アカウントで実行されている指定されたリソースの数。リソースは、メトリクスに関連付けられたディメンションによって定義されます。  このメトリクスで最も役に立つ統計は MAXIMUM です。これは、1 分間の期間中に使用されるリソースの最大数を表します。

次のディメンションは、Amazon EC2 によって発行される使用状況メトリクスを絞り込むために使用されます。

ディメンション	説明
Service	リソースを含む AWS のサービスの名前。Amazon EC2 使用状況メトリクスの場合、このディメンションの値は EC2 です。
Type	レポートされるエンティティのタイプ。現在、Amazon EC2 使用状況メトリクスの有効な値は Resource のみです。
Resource	実行中のリソースのタイプ。現在、Amazon EC2 使用状況メトリクスの有効な値は vCPU のみです。これは、実行中のインスタンスに関する情報を返します。

ディメンション	説明
Class	<p>追跡されるリソースのクラス。Resource ディメンションの値として vCPU を使用する Amazon EC2 使用状況メトリクスの場合、有効な値は、Standard/OnDemand、F/OnDemand、G/OnDemand、Inf/OnDemand、P/OnDemand、および X/OnDemand です。</p> <p>このディメンションの値は、メトリクスによって報告されるインスタンスタイプの最初の文字を定義します。たとえば、Standard/OnDemand は、A、C、D、H、I、M、R、T、Z で始まるタイプのすべての実行中のインスタンスに関する情報を返し、G/OnDemand は G で始まるタイプのすべてのインスタンスに関する情報を返します。</p>

## コンソールを使用してメトリクスをリスト表示する

メトリクスはまず名前空間ごとにグループ化され、次に各名前空間内の種々のディメンションの組み合わせごとにグループ化されます。たとえば、Amazon EC2 で提供されるすべてのメトリクスを表示させることもできれば、インスタンス ID、インスタンスタイプ、イメージ (AMI) ID、Auto Scaling グループでグループ化された EC2 メトリクスを表示することもできます。

利用可能なメトリクスをカテゴリー別に表示するには (コンソール)

1. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
2. ナビゲーションペインで メトリクスを選択します。
3. EC2 のメトリクスの名前空間を選択します。

The screenshot shows the CloudWatch Metrics console interface. At the top, there are three tabs: 'All metrics' (highlighted in orange), 'Graphed metrics', and 'Graph options'. Below the tabs is a search bar with placeholder text 'Search for any metric, dimension or resource id'. The main area is titled '722 Metrics' and contains seven boxes, each representing a service and its metric count:

サービス	メトリクス数
EBS	117 Metrics
EC2	316 Metrics
EFS	7 Metrics
ELB	210 Metrics
ElasticBeanstalk	8 Metrics
RDS	60 Metrics
S3	4 Metrics

4. メトリクスのディメンション (たとえば、インスタンス別メトリクス) を選択します。

The screenshot shows the AWS CloudWatch Metrics console with the following interface elements:

- Top navigation bar: All metrics, Graphed metrics, Graph options.
- Breadcrumbs: All > EC2.
- Search bar: Search for any metric, dimension or resource id.
- Main content area: 103 Metrics
- Category cards:
  - By Auto Scaling Group: 28 Metrics
  - By Image (AMI) Id: 7 Metrics
  - Per-Instance Metrics: 54 Metrics
  - Aggregated by Instance Type: 7 Metrics
  - Across All Instances: 7 Metrics

5. メトリクスを並べ替えるには、列見出しを使用します。メトリクスをグラフ表示するには、メトリクスの横にあるチェックボックスを選択します。リソースでフィルタするには、リソース ID を選択し、[Add to search] を選択します。メトリクスでフィルタするには、メトリクスの名前を選択し、[Add to search] を選択します。

The screenshot shows the AWS CloudWatch Metrics console with the following interface elements:

- Top navigation bar: All metrics, Graphed metrics, Graph options.
- Breadcrumbs: All > EC2 > Per-Instance Metrics.
- Search bar: Search for any metric, dimension or resource id.
- Main content area: A table listing Per-Instance Metrics for instance i-abbc12a7.
- Table columns: Instance Name (192), InstanceId, Metric Name.
- Table rows:
  - my-instance, i-abbc12a7, CPUUtilization
  - my-instance, i-abbc12a7, DiskReadBytes
  - my-instance, i-abbc12a7, DiskReadOps
  - my-instance, i-abbc12a7, DiskWriteBytes
  - my-instance, i-abbc12a7, DiskWriteOps
  - my-instance, i-abbc12a7, NetworkIn
  - my-instance, i-abbc12a7, NetworkOut
  - my-instance, i-abbc12a7, NetworkPacketsIn
  - my-instance, i-abbc12a7, NetworkPacketsOut
- A context menu is open over the first row (my-instance, i-abbc12a7, CPUUtilization). The menu items are:
  - Add to search
  - Search for this only
  - Add to graph
  - Graph this metric only
  - Graph all search results
  - Jump to resource

## AWS CLI を使用してメトリクスをリスト表示する

`list-metrics` コマンドを使用して、インスタンスの CloudWatch メトリクスをリスト表示します。

Amazon EC2 の利用可能なすべてのメトリクスを表示するには (AWS CLI)

次の例では、Amazon EC2 のすべてのメトリクスを表示する目的で AWS/EC2 名前空間を指定します。

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

出力例を次に示します。

```
{  
    "Metrics": [  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ],  
            "MetricName": "NetworkOut"  
        },  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ],  
            "MetricName": "CPUUtilization"  
        },  
        {  
            "Namespace": "AWS/EC2",  
            "Dimensions": [  
                {  
                    "Name": "InstanceId",  
                    "Value": "i-1234567890abcdef0"  
                }  
            ],  
            "MetricName": "NetworkIn"  
        },  
        ...  
    ]  
}
```

インスタンスで利用可能なすべてのメトリクスをリスト表示するには (AWS CLI)

次の例では、指定のインスタンスの結果だけを表示する目的で AWS/EC2 名前空間と InstanceId ディメンションを指定します。

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions  
    Name=InstanceId,Value=i-1234567890abcdef0
```

すべてのインスタンス間でメトリクスをリスト表示するには (AWS CLI)

次の例では、指定のメトリクスの結果だけを表示する目的で AWS/EC2 名前空間とメトリクス名を指定します。

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

## インスタンスのメトリクスの統計情報の取得

インスタンスの CloudWatch メトリクスの統計情報を取得できます。

### コンテンツ

- [統計の概要 \(p. 655\)](#)

- 特定の インスタンスの統計を取得する (p. 655)
- インスタンス全体の統計の集約 (p. 658)
- Auto Scaling グループ別に統計を集約する (p. 660)
- AMI 別に統計を集計する (p. 660)

## 統計の概要

統計は、指定した期間のメトリクスデータの集計です。CloudWatch は、カスタムデータまたは AWS の他のサービスから CloudWatch に提供されたメトリクスデータポイントを基に、統計を提供します。集約は、指定した期間内に、名前空間、メトリクス名、ディメンション、データポイントの測定単位を用いて行われます。次の表は利用可能な統計を説明しています。

統計	説明
Minimum	指定された期間に認められた最小値です。この値を用いて、アプリケーションの低ボリュームのアクティビティを判断できます。
Maximum	指定された期間に認められた最大値です。この値を用いて、アプリケーションの高ボリュームのアクティビティを判断できます。
Sum	該当するメトリクスで加算されたすべての合計値です。この統計は、メトリクスの合計ボリュームを判断するのに役立ちます。
Average	指定した期間の Sum/SampleCount の値です。この統計を Minimum および Maximum と比較することで、メトリクスの全容、および平均使用量がどれくらい Minimum と Maximum に近いかを判断できます。この比較は、必要に応じていつリソースを増減させるべきかを知るのに役立ちます。
SampleCount	統計計算で使用するデータポイントのカウント (数) です。
pNN.NN	指定されたパーセンタイルの値。小数点以下最大 2 衔を使用して、任意のパーセンタイルを指定できます (p95.45 など)。

## 特定の インスタンスの統計を取得する

次の例では、AWS マネジメントコンソール または AWS CLI を使用して、特定の EC2 インスタンスの最大 CPU 使用率を決定することができます。

### 要件

- インスタンスの ID が必要です。インスタンス ID は、AWS マネジメントコンソール コンソールまたは `describe-instances` コマンドを使って取得します。
- デフォルトでは、基本モニタリングが有効化されていますが、詳細モニタリングを有効化することもできます。詳細については、「[インスタンスの詳細モニタリングの有効化または無効化 \(p. 642\)](#)」を参照してください。

### 特定のインスタンスの CPU 使用率を表示するには (コンソール)

- <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
- ナビゲーションペインで メトリクスを選択します。
- EC2 のメトリクスの名前空間を選択します。

The screenshot shows the AWS Metrics Explorer interface. At the top, there are three tabs: "All metrics" (highlighted in orange), "Graphed metrics", and "Graph options". Below the tabs is a search bar with the placeholder text "Search for any metric, dimension or resource id". The main area displays "722 Metrics" categorized by service:

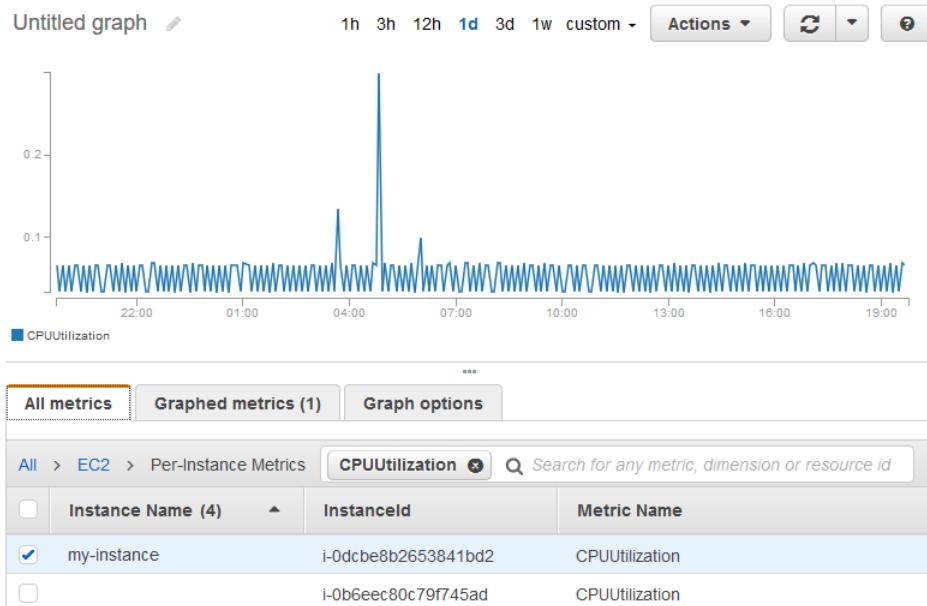
- EBS: 117 Metrics
- EC2: 316 Metrics
- EFS: 7 Metrics
- ELB: 210 Metrics
- ElasticBeanstalk: 8 Metrics
- RDS: 60 Metrics
- S3: 4 Metrics

4. インスタンス別メトリクスのディメンションを選択します。

The screenshot shows the AWS Metrics Explorer interface for the EC2 service. At the top, it shows "All > EC2" and a search bar. The main area displays "103 Metrics" categorized by dimension:

- By Auto Scaling Group: 28 Metrics
- By Image (AMI) Id: 7 Metrics
- Per-Instance Metrics: 54 Metrics
- Aggregated by Instance Type: 7 Metrics
- Across All Instances: 7 Metrics

5. 検索フィールドに **CPUUtilization** と入力して Enter キーを押します。特定のインスタンスの行を選択します。すると、そのインスタンスの [CPUUtilization] メトリクスのグラフが表示されます。グラフに名前を付けるには、鉛筆アイコンを選択します。時間範囲を変更するには、事前定義済みの値を選択するか、[custom] を選択します。



6. メトリクスの統計または期間を変更するには、[Graphed metrics] タブを選択します。列見出しまでは個々の値を選択し、次に異なる値を選択します。

Label	Namespace	Dimensions	Metric Name	Statistic	Period
CPUUtilization	AWS/EC2	Dimensions (1)	CPUUtilization	Average	1 Minute

特定のインスタンスの CPU 使用率を取得するには (AWS CLI)

次の [get-metric-statistics](#) コマンドを使用すると、期間と時間間隔を指定して、特定のインスタンスの [CPUUtilization] メトリクスを取得できます。

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization --period 3600 \
--statistics Maximum --dimensions Name=InstanceId,Value=i-1234567890abcdef0 \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00
```

出力例を次に示します。それぞれの値は、単一の EC2 インスタンスの最大 CPU 使用率を表しています。

```
{
  "Datapoints": [
    {
      "Timestamp": "2016-10-19T00:18:00Z",
      "Maximum": 0.33000000000000002,
      "Unit": "Percent"
    }
  ]
}
```

```
        "Timestamp": "2016-10-19T03:18:00Z",
        "Maximum": 99.67000000000002,
        "Unit": "Percent"
    },
    {
        "Timestamp": "2016-10-19T07:18:00Z",
        "Maximum": 0.3400000000000002,
        "Unit": "Percent"
    },
    {
        "Timestamp": "2016-10-19T12:18:00Z",
        "Maximum": 0.3400000000000002,
        "Unit": "Percent"
    },
    ...
],
"Label": "CPUUtilization"
}
```

## インスタンス全体の統計の集約

詳細モニタリングが有効になっているインスタンスの集約された統計を使用することができます。基本モニタリングを使用するインスタンスは集約されません。加えて、Amazon CloudWatch は複数のリージョンにまたがってデータを集約することはありません。そのため、メトリクスはリージョン間で完全に分離されています。複数のインスタンスにわたって集計された統計情報を取得するには、1 分間隔でデータが提供される詳細モニタリングを事前に有効にしておく必要があります(追加料金がかかります)。

この例では、EC2 インスタンスの平均 CPU 使用率を取得するために詳細モニタリングを使用する方法について示します。ディメンションを指定していないため、CloudWatch は、AWS/EC2 名前空間にある全ディメンションの統計を返します。

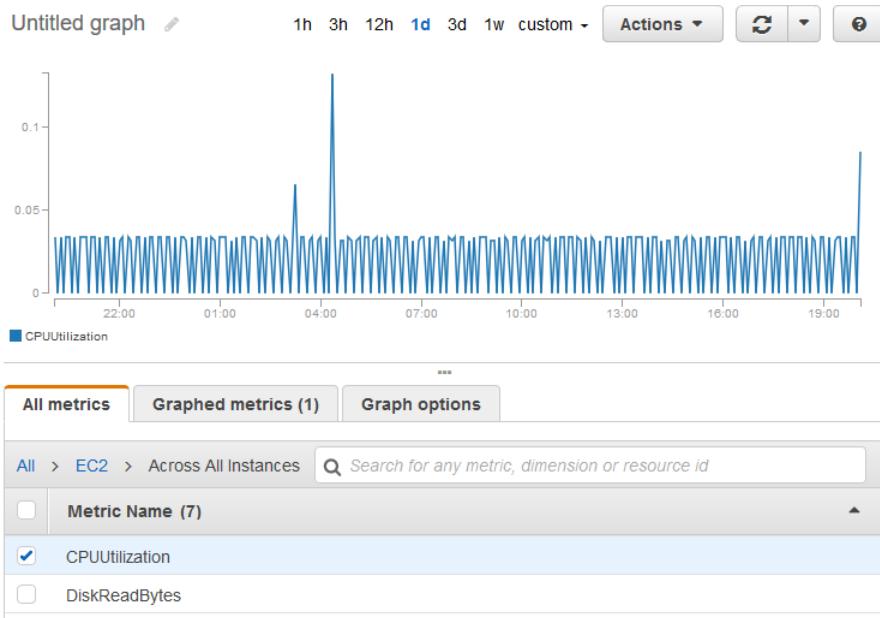
### Important

AWS 名前空間にあるすべてのディメンションを取得するこの手法は、Amazon CloudWatch にパブリッシュするカスタム名前空間では機能しません。カスタム名前空間の場合、データポイントを含む統計を取得するには、そのデータポイントに関連付けられたディメンション式をすべて指定する必要があります。

### インスタンスの平均 CPU 使用率を表示するには(コンソール)

1. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
2. ナビゲーションペインで メトリクスを選択します。
3. [EC2] 名前空間を選択して、[Across All Instances] を選択します。
4. [CPUUtilization] を含む行を選択します。すべての EC2 インスタンスのメトリクスがグラフとして表示されます。グラフに名前を付けるには、鉛筆アイコンを選択します。時間範囲を変更するには、事前定義済みの値を選択するか、[custom] を選択します。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
メトリクスの統計情報を取得する



5. メトリクスの統計または期間を変更するには、[Graphed metrics] タブを選択します。列見出しまでは個々の値を選択し、次に異なる値を選択します。

複数のインスタンスの平均 CPU 使用率を取得するには (AWS CLI)

次のように `get-metric-statistics` コマンドを使用し、インスタンス全体の平均 [CPUUtilization] メトリクスを取得します。

```
aws cloudwatch get-metric-statistics --namespace AWS/Ec2 --metric-name CPUUtilization \
--period 3600 --statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00
```

出力例を次に示します。

```
{
  "Datapoints": [
    {
      "SampleCount": 238.0,
      "Timestamp": "2016-10-12T07:18:00Z",
      "Average": 0.038235294117647062,
      "Unit": "Percent"
    },
    {
      "SampleCount": 240.0,
      "Timestamp": "2016-10-12T09:18:00Z",
      "Average": 0.16670833333333332,
      "Unit": "Percent"
    },
    {
      "SampleCount": 238.0,
      "Timestamp": "2016-10-11T23:18:00Z",
      "Average": 0.041596638655462197,
      "Unit": "Percent"
    },
    ...
  ],
  "Label": "CPUUtilization"
```

}

## Auto Scaling グループ別に統計を集約する

Auto Scaling グループ内で EC2 インスタンスの統計を集計することができます。Amazon CloudWatch は、複数のリージョンをまたがってデータを集計することはできません。メトリクスは、リージョン間で完全に独立しています。

この例では、1 つの Auto Scaling グループについて、ディスクに書き込まれた総バイト数を取得する方法を説明します。合計は、指定された Auto Scaling グループのすべての EC2 インスタンスで、24 時間おきに 1 分間にに対して算出されます。

Auto Scaling グループ内のインスタンスの DiskWriteBytes を表示するには (コンソール)

1. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
2. ナビゲーションペインで メトリクスを選択します。
3. [EC2] 名前空間を選択し、次に [By Auto Scaling Group] を選択します。
4. [DiskWriteBytes] メトリクスの行と特定の Auto Scaling グループを選択します。すると、その Auto Scaling グループ内にあるインスタンスのメトリクスがグラフとして表示されます。グラフに名前を付けるには、鉛筆アイコンを選択します。時間範囲を変更するには、事前定義済みの値を選択するか、[custom] を選択します。
5. メトリクスの統計または期間を変更するには、[Graphed metrics] タブを選択します。列見出しありまたは個々の値を選択し、次に異なる値を選択します。

Auto Scaling グループ内のインスタンスの DiskWriteBytes を表示するには (AWS CLI)

以下のように `get-metric-statistics` コマンドを使用します。

```
aws cloudwatch get-metric-statistics --namespace AWS/ElasticComputeCloud --metric-name DiskWriteBytes --period 360 \
--statistics "Sum" "SampleCount" --dimensions Name=AutoScalingGroupName,Value=my-asg --
start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00
```

出力例を次に示します。

```
{
  "Datapoints": [
    {
      "SampleCount": 18.0,
      "Timestamp": "2016-10-19T21:36:00Z",
      "Sum": 0.0,
      "Unit": "Bytes"
    },
    {
      "SampleCount": 5.0,
      "Timestamp": "2016-10-19T21:42:00Z",
      "Sum": 0.0,
      "Unit": "Bytes"
    }
  ],
  "Label": "DiskWriteBytes"
}
```

## AMI 別に統計を集計する

統計の集計は、詳細モニタリングが有効化されているインスタンスに対して行うことができます。その場合、基本モニタリングを使用するインスタンスは含まれません。Amazon CloudWatch は、複数のリージョン間で統計情報を集計するため、複数のリージョンをまたがる AMI では、統計情報を集計することができません。

ジョンをまたがってデータを集計することはできません。メトリクスは、リージョン間で完全に独立しています。

複数のインスタンスにわたって集計された統計情報を取得するには、1 分間隔でデータが提供される詳細モニタリングを事前に有効にしておく必要があります(追加料金がかかります)。詳細については、「[インスタンスの詳細モニタリングの有効化または無効化 \(p. 642\)](#)」を参照してください。

この例では、特定の Amazon Machine Image (AMI) を使用するすべてのインスタンスの平均 CPU 使用率を特定する方法を説明します。平均値は、1 日間、60 秒間隔の平均値です。

AMI 別の平均 CPU 使用率を表示するには(コンソール)

1. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
2. ナビゲーションペインで メトリクスを選択します。
3. [EC2] 名前空間を選択し、次に [By Image (AMI) Id] を選択します。
4. [CPUUtilization] メトリクスの行と特定の AMI を選択します。すると、その AMI のメトリクスがグラフとして表示されます。グラフに名前を付けるには、鉛筆アイコンを選択します。時間範囲を変更するには、事前定義済みの値を選択するか、[custom] を選択します。
5. メトリクスの統計または期間を変更するには、[Graphed metrics] タブを選択します。列見出しまだ個々の値を選択し、次に異なる値を選択します。

特定のイメージ ID の平均 CPU 使用率を取得するには(AWS CLI)

以下のように `get-metric-statistics` コマンドを使用します。

```
aws cloudwatch get-metric-statistics --namespace AWS/Ec2 --metric-name CPUUtilization --period 3600 \
--statistics Average --dimensions Name=ImageId,Value=ami-3c47a355 --start-time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00
```

出力例を次に示します。それぞれの値は、指定した AMI を実行する EC2 インスタンスの平均 CPU 使用率(%)を表します。

```
{
  "Datapoints": [
    {
      "Timestamp": "2016-10-10T07:00:00Z",
      "Average": 0.04100000000000009,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-10T14:00:00Z",
      "Average": 0.079579831932773085,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2016-10-10T06:00:00Z",
      "Average": 0.03600000000000011,
      "Unit": "Percent"
    },
    ...
  ],
  "Label": "CPUUtilization"
}
```

## インスタンスのグラフメトリクス

インスタンスを起動した後、Amazon EC2 コンソールを開いて、[Monitoring] タブでインスタンスのモニタリンググラフを表示できます。各グラフは、利用可能な Amazon EC2 メトリクスのいずれかに基づいています。

以下のグラフが利用可能です。

- 平均 CPU 使用率 (パーセント)
- 平均ディスク読み込み (バイト)
- 平均ディスク書き込み (バイト)
- 最大ネットワーク受信 (バイト)
- 最大ネットワーク送信 (Bytes)
- 要約ディスク読み取り操作 (カウント)
- 要約ディスク書き込み操作 (カウント)
- 要約ステータス (任意)
- 要約ステータスインスタンス (カウント)
- 要約ステータスシステム (カウント)

グラフに表示されるメトリクスおよびデータの詳細については、「[インスタンスの利用可能な CloudWatch メトリクスのリスト表示 \(p. 644\)](#)」を参照してください。

CloudWatch コンソールを使用したメトリクスのグラフ化

CloudWatch コンソールを使用して、Amazon EC2 や他の AWS サービスによって生成されたメトリクスデータをグラフ化することができます。詳細については、『Amazon CloudWatch ユーザーガイド』の「[メトリクスをグラフ化する](#)」を参照してください。

## インスタンスの CloudWatch アラームの作成

インスタンスの CloudWatch メトリクスをモニタリングする CloudWatch アラームを作成できます。CloudWatch は、指定したしきい値にメトリクスが達すると、自動的に通知を送信します。CloudWatch アラームは、Amazon EC2 コンソールを使用するか、CloudWatch コンソールに用意されている高度なオプションを使用して作成できます。

CloudWatch コンソールを使用してアラームを作成するには

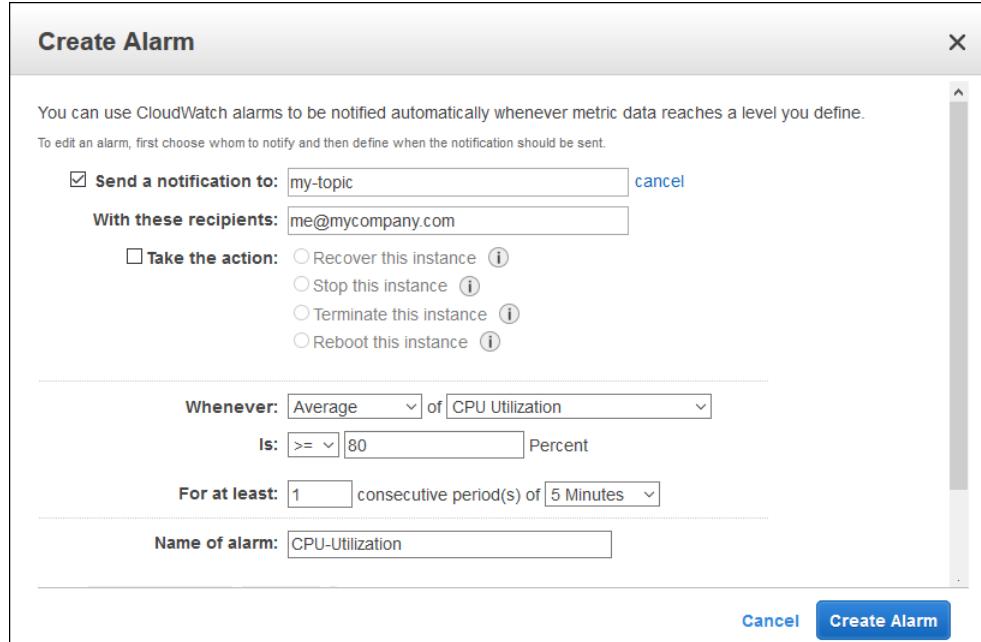
例については、「[Amazon CloudWatch アラームの作成](#)」(Amazon CloudWatch ユーザーガイド) を参照してください。

Amazon EC2 コンソールを使用してアラームを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択します。
4. [Monitoring] タブで、[Create Alarm] を選択します。
5. [アラームの作成] ダイアログボックスで、次の操作を行います。
  - a. [create topic (トピックの作成)] を選択します。[Send a notification to] に、SNS トピックの名前を入力します。[With these recipients] に、通知の受取先となる 1 つ以上の E メールアドレスを入力します。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
インスタンスを停止、終了、再起動、  
または復旧するアラームを作成する

- b. ポリシーのメトリクスおよび条件を指定します。たとえば、[Whenever] はデフォルト設定のままにすることができます (CPU 使用率の平均)。[Is] で、>= を選択して [80] パーセントを入力します。[For at least] では、[1] の連続した期間に 5 Minutes を入力します。
- c. [Create Alarm] を選択します。



## インスタンスを停止、終了、再起動、または復旧するアラームを作成する

Amazon CloudWatch アラームアクションを使用して、インスタンスを自動的に停止、終了、再起動、または復旧するアラームを作成できます。停止または終了アクションを使用すると、今後インスタンスを実行する必要がなくなったときにコストを節約できます。再起動アクションを使用すると、これらのインスタンスを自動的に再起動でき、復旧アクションを使用すると、システムで障害が発生した場合に新しいハードウェアで復旧できます。

サービスにリンクされたロール `AWSLambdaRoleForCloudWatchEvents` を使用すると、AWS がお客様に代わってアラームアクションを実行できます。AWS マネジメントコンソール、IAM CLI、または IAM API で初めてアラームを作成する場合は、サービスにリンクされたロールが CloudWatch によって作成されます。

自動的にインスタンスを停止または終了するシナリオはいくつもあります。たとえば、バッチ給与計算処理ジョブまたは科学計算タスクを専用に行うインスタンスを使用している場合が挙げられます。これらのインスタンスは一定期間動作して仕事を完了します。このようなインスタンスは、アイドル状態（課金されている状態）にせずに、停止または終了するとコスト削減につながります。停止アラームアクションと終了アラームアクションの主な違いとして、停止したインスタンスは、後で再実行が必要な場合に再起動しやすいことと、同じインスタンス ID およびルートボリュームを維持できることがあります。しかし、終了したインスタンスを再起動することはできません。代わりに新しいインスタンスを開始する必要があります。

停止、終了、再起動、復旧の各アクションは、Amazon EC2 インスタンスマトリクスごとに設定されている任意のアラームに追加できます。これには、Amazon CloudWatch によって (AWS/EC2 名前空間で) 提供される基本モニタリングや詳細モニタリングのメトリクスが含まれます。また、InstanceId ディメン

ションを含む任意のカスタムメトリクスも(その値が実行中の有効な Amazon EC2 インスタンスを参照する場合に限り)含まれます。

### コンソールのサポート

Amazon EC2 コンソールまたは CloudWatch コンソールを使用してアラームを作成できます。このドキュメントの手順では、Amazon EC2 コンソールを使用します。CloudWatch コンソールを使用する手順については、『Amazon CloudWatch ユーザーガイド』の「[インスタンスを停止、終了、再起動、または復旧するアラームを作成する](#)」を参照してください。

### アクセス許可

AWS Identity and Access Management (IAM) ユーザーの場合、アラームを作成または変更するには次のアクセス権限が必要です。

- `iam:CreateServiceLinkedRole`、`iam:GetPolicy`、`iam:GetPolicyVersion`、および `iam:GetRole` – Amazon EC2 アクションでのすべてのアラーム用
- `ec2:DescribeInstanceStatus` と `ec2:DescribeInstances` – Amazon EC2 インスタンスステータスマトリクスに対するすべてのアラーム用。
- `ec2:StopInstances` – 停止アクションを含むアラーム用。
- `ec2:TerminateInstances` – 終了アクションを含むアラーム用。
- 復旧アクションを含むアラームに特別なアクセス許可は不要です。

Amazon CloudWatch に対する読み取り/書き込みのアクセス権限があり、Amazon EC2 に対するアクセス権限がない場合、アラームは作成できますが、Amazon EC2 インスタンスで停止または終了アクションが実行されません。ただし、関連付けられている Amazon EC2 API の使用許可が後で付与される場合、以前に作成したアラームアクションは実行されるようになります。IAM アクセス許可の詳細については、『IAM ユーザーガイド』の「[アクセス許可とポリシー](#)」を参照してください。

### コンテンツ

- [Amazon CloudWatch アラームへの停止アクションの追加 \(p. 664\)](#)
- [Amazon CloudWatch アラームへの終了アクションの追加 \(p. 665\)](#)
- [Amazon CloudWatch アラームへの再起動アクションの追加 \(p. 666\)](#)
- [Amazon CloudWatch アラームへの復旧アクションの追加 \(p. 667\)](#)
- [Amazon CloudWatch コンソールを使用してアラームとアクションの履歴を確認する \(p. 668\)](#)
- [Amazon CloudWatch のアラームアクションのシナリオ \(p. 669\)](#)

## Amazon CloudWatch アラームへの停止アクションの追加

一定のしきい値に達したときに Amazon EC2 インスタンスを停止するアラームを作成できます。たとえば、開発またはテスト用のインスタンスを実行したまま、終了するのを忘れることがたまにあります。平均 CPU 利用率が 24 時間 10% 未満である場合に、インスタンスがアイドル状態で使用されていないという信号を発してトリガーするアラームを作成できます。しきい値、持続時間、期間をニーズに合わせて調整し、アラームがトリガーされたときにメールを受信するよう Amazon Simple Notification Service (Amazon SNS) 通知を追加できます。

Amazon EBS ボリュームをルートデバイスとして使用するインスタンスは停止または終了できますが、インスタンスストアをルートデバイスとして使用するインスタンスでは終了のみ行えます。

アイドル状態のインスタンスを停止させるアラームを作成するには (Amazon EC2 コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

- 
2. ナビゲーションペインで、[インスタンス] を選択します。
  3. インスタンスを選択します。[Monitoring] タブで、[Create Alarm] を選択します。
  4. [アラームの作成] ダイアログボックスで、次の操作を行います。
    - a. アラームがトリガされたときに E メールが届くようにするには、[通知の送信先] で既存の Amazon SNS トピックを選択するか、または [トピックを作成] を選択して新しいトピックを作成します。

トピックを新規作成するには、[Send a notification to] にトピック名を入力し、[With these recipients] に受信者のメールアドレスを入力します（カンマ区切り）。アラームの作成後、サブスクライブの確認メールが届きます。このトピックの通知を受け取れるようになるには、このメールを確認する必要があります。
    - b. [アクションを実行]、[このインスタンスを停止する] を選択します。
    - c. [Whenever] で使用する統計を選択してから、メトリクスを選択します。この例では、[Average] と [CPU Utilization] を選択します。
    - d. [状況] で、メトリクスのしきい値を定義します。この例では、[10] パーセントを入力します。
    - e. [最低期間] で、アラームの評価期間を指定します。この例では、1 時間の期間で 24 期間連続と入力しています。
    - f. アラーム名を変更するには、[Name of alarm] に新しい名前を入力します。アラーム名には ASCII 文字のみを使用する必要があります。

アラーム名を入力しない場合は、Amazon CloudWatch によってアラーム名が自動的に作成されます。

#### Note

アラーム設定は、アラームを作成する前に実際の要件に基づいて調整することも、アラーム作成後に編集することもできます。これにはメトリクス、しきい値、持続時間、アクション、通知設定などがあります。ただし、アラームの作成後のアラーム名の編集はできません。

- g. [Create Alarm] を選択します。

## Amazon CloudWatch アラームへの終了アクションの追加

(インスタンスで終了保護が有効になっていない限り)、一定のしきい値に達したときに EC2 インスタンスを自動的に終了させるアラームを作成することができます。たとえば、インスタンスが仕事を終え、再びそのインスタンスを使用する必要がない場合は、インスタンスを削除することをお勧めします。後でインスタンスを使用する可能性がある場合は、インスタンスを削除するのではなく、停止するほうが良いでしょう。インスタンスの削除保護の有効化および無効化の詳細については、『Linux インスタンス用 Amazon EC2 ユーザーガイド』の「[インスタンスの削除保護の有効化](#)」を参照してください。

アイドル状態のインスタンスを終了するアラームを作成するには (Amazon EC2 コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択します。[Monitoring] タブで、[Create Alarm] を選択します。
4. [アラームの作成] ダイアログボックスで、次の操作を行います。
  - a. アラームがトリガされたときに E メールが届くようにするには、[通知の送信先] で既存の Amazon SNS トピックを選択するか、または [トピックを作成] を選択して新しいトピックを作成します。

トピックを新規作成するには、[Send a notification to] にトピック名を入力し、[With these recipients] に受信者のメールアドレスを入力します（カンマ区切り）。アラームの作成後、サブス

クライアントの確認メールが届きます。このトピックの通知を受け取れるようになるには、このメールを確認する必要があります。

- b. [アクションを実行]、[このインスタンスの終了]を選択します。
- c. [Whenever]で統計を選択し、メトリクスを選択します。この例では、[Average]と[CPU Utilization]を選択します。
- d. [状況]で、メトリクスのしきい値を定義します。この例では、[10] パーセントを入力します。
- e. [最低期間]で、アラームの評価期間を指定します。この例では、1 時間の期間で 24 期間連続と入力しています。
- f. アラーム名を変更するには、[Name of alarm]に新しい名前を入力します。アラーム名には ASCII 文字のみを使用する必要があります。

アラーム名を入力しない場合は、Amazon CloudWatch によってアラーム名が自動的に作成されます。

#### Note

アラーム設定は、アラームを作成する前に実際の要件に基づいて調整することも、アラーム作成後に編集することもできます。これにはメトリクス、しきい値、持続時間、アクション、通知設定などがあります。ただし、アラームの作成後のアラーム名の編集はできません。

- g. [Create Alarm]を選択します。

## Amazon CloudWatch アラームへの再起動アクションの追加

Amazon EC2 インスタンスをモニタリングし、自動的に再起動する Amazon CloudWatch アラームを作成できます。再起動アラームアクションは、インスタンスのヘルスチェックが失敗した場合に推奨されます（システムのヘルスチェックが失敗した場合には、復旧アラームアクションが推奨されます）。インスタンスの再起動は、オペレーティングシステムの再起動と同等です。ほとんどの場合、インスタンスの再起動には数分しかかかりません。インスタンスを再起動すると、インスタンスは同じホスト上で保持されるため、インスタンスのパブリックドメイン名、プライベート IP アドレス、およびインスタンスストアボリューム上のすべてのデータは保持されます。

インスタンスを再起動しても、インスタンスの停止と再起動とは異なり、新しいインスタンスの課金(秒単位、最低 1 分間分)は開始されません。詳細については、『Linux インスタンス用 Amazon EC2 ユーザーガイド』の「[Reboot Your Instance](#)」を参照してください。

#### Important

再起動と復旧アクション間で不具合が発生するのを回避するには、再起動アラームと復旧アラームを同じ評価期間に設定するのを避けます。再起動アラームを各 1 分間の 3 回の評価期間に設定することをお勧めします。詳細については、『Amazon CloudWatch ユーザーガイド』の「[アラームを評価する](#)」を参照してください。

インスタンスを再起動するアラームを作成するには(Amazon EC2 コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス]を選択します。
3. インスタンスを選択します。[Monitoring] タブで、[Create Alarm]を選択します。
4. [アラームの作成]ダイアログボックスで、次の操作を行います。
  - a. アラームがトリガされたときに E メールが届くようにするには、[通知の送信先]で既存の Amazon SNS トピックを選択するか、または [トピックを作成]を選択して新しいトピックを作成します。

トピックを新規作成するには、[Send a notification to]にトピック名を入力し、[With these recipients]に受信者のメールアドレスを入力します(カンマ区切り)。アラームの作成後、サブス

ライブの確認メールが届きます。このトピックの通知を受け取れるようになるには、このメールを確認する必要があります。

- b. [アクションを実行]、[このインスタンスの再起動] を選択します。
- c. [次の時] で、[ステータスチェックに失敗 (インスタンス)] を選択します。
- d. [最低期間] で、アラームの評価期間を指定します。この例では、1 分の期間で 3 期間連続と入力しています。
- e. アラーム名を変更するには、[Name of alarm] に新しい名前を入力します。アラーム名には ASCII 文字のみを使用する必要があります。

アラーム名を入力しない場合は、Amazon CloudWatch によってアラーム名が自動的に作成されます。

- f. [Create Alarm] を選択します。

## Amazon CloudWatch アラームへの復旧アクションの追加

Amazon EC2 インスタンスをモニタリングする Amazon CloudWatch アラームを作成できます。下層のハードウェア障害または修復に AWS を必要とする問題によりインスタンスが正常に機能しなくなった場合に、自動的にインスタンスを復旧できます。終了したインスタンスは復旧できません。復旧されたインスタンスは、インスタンス ID、プライベート IP アドレス、Elastic IP アドレス、すべてのインスタンスマターダーを含め、元のインスタンスと同じです。

CloudWatch では、復旧アクションをサポートしていないインスタンスにあるアラームに、復旧アクションを追加することはできません。

`StatusCheckFailed_System` アラームがトリガーされ、復旧アクションが開始されると、アラームを作成し、復旧アクションに関連付けたときに選択した Amazon SNS トピックによって通知されます。インスタンスを復旧する際、インスタンスを再起動するときにインスタンスは移行され、メモリ内にあるデータは失われます。プロセスが完了すると、情報はアラームに設定された SNS トピックに発行されます。この SNS トピックをサブスクリーピングしているすべてのユーザーは、復旧処理のステータスと、それ以降の手順を含むメールの通知を受け取ります。インスタンスが復旧した時点でインスタンスが再起動されたことがわかります。

復旧アクションは、`StatusCheckFailed_System` でのみ使用できます。`StatusCheckFailed_Instance` では使用できません。

以下の問題が発生すると、システムステータスのチェックに失敗する可能性があります。

- ネットワーク接続の喪失
- システム電源の喪失
- 物理ホストのソフトウェアの問題
- ネットワーク到達可能性に影響する、物理ホスト上のハードウェアの問題

復旧アクションは、次のような特性を持つインスタンスでのみサポートされています。

- 次のインスタンスタイプのいずれかを使用している:  
A1、C3、C4、C5、C5n、Inf1、M3、M4、M5、M5a、M5n、P3、R3、R4、R5、R5a、R5n、T2、T3、T3a、X1、または X1e
- `default` または `dedicated` インスタンスのテナント属性を使用している
- EBS ボリュームのみを使用します (インスタンストアボリュームは設定しないでください)。詳細については、「['Recover this instance' is disabled](#)」を参照してください。

インスタンスにパブリック IP アドレスが割り当てられている場合、復旧後にパブリック IP アドレスが維持されます。

---

## Important

再起動と復旧アクション間で不具合が発生するのを回避するには、再起動アラームと復旧アラームを同じ評価期間に設定するのを避けます。復旧アラームを各 1 分間の 2 回の評価期間に設定することをお勧めします。詳細については、『Amazon CloudWatch ユーザーガイド』の「[アラームを評価する](#)」を参照してください。

インスタンスを復旧するアラームを作成するには (Amazon EC2 コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択します。[Monitoring] タブで、[Create Alarm] を選択します。
4. [アラームの作成] ダイアログボックスで、次の操作を行います。
  - a. アラームがトリガされたときに E メールが届くようにするには、[通知の送信先] で既存の Amazon SNS トピックを選択するか、または [トピックを作成] を選択して新しいトピックを作成します。

トピックを新規作成するには、[Send a notification to] にトピック名を入力し、[With these recipients] に受信者のメールアドレスを入力します (カンマ区切り)。アラームの作成後、サブスクライブの確認メールが届きます。このトピックの E メールを受け取れるようになるには、このメールを確認する必要があります。

### Note

- 今後、アラームがトリガーされたときにメール通知を受信するためには、指定された SNS トピックをサブスクライブする必要があります。
  - AWS アカウントルートユーザーは、SNS トピックが指定されていない場合でも、自動インスタンス復旧アクションが発生すると、常に E メール通知を受信します。
  - AWS アカウントルートユーザーは、指定した SNS トピックをサブスクライブしていない場合でも、自動インスタンス復旧アクションが発生すると、常に E メール通知を受信します。
- b. [アクションを実行]、[このインスタンスの復元] を選択します。
  - c. [次の時] で、[ステータスチェックに失敗 (システム)] を選択します。
  - d. [最低期間] で、アラームの評価期間を指定します。この例では、1 分の期間で 2 期間連続と入力しています。
  - e. アラーム名を変更するには、[Name of alarm] に新しい名前を入力します。アラーム名には ASCII 文字のみを使用する必要があります。
- アラーム名を入力しない場合は、Amazon CloudWatch によってアラーム名が自動的に作成されます。
- f. [Create Alarm] を選択します。

## Amazon CloudWatch コンソールを使用してアラームとアクションの履歴を確認する

Amazon CloudWatch コンソールで、アラームとアクションの履歴を見ることができます。Amazon CloudWatch は、過去 2 週間分のアラームとアクションの履歴を保管します。

トリガーされたアラームとアクションを表示するには (CloudWatch コンソール)

1. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
2. ナビゲーションペインで、[Alarms] を選択します。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
インスタンスを停止、終了、再起動、  
または復旧するアラームを作成する

- 
3. アラームを選択します。
  4. [Details] タブには、直近の状態遷移、および時間とメトリクス値が表示されます。
  5. 直近の履歴のエントリを表示するには、[History] タブを選択します。

## Amazon CloudWatch のアラームアクションのシナリオ

Amazon EC2 (Amazon EC2) コンソールを使用して、一定の条件が満たされたときにインスタンスを停止または終了させるアラームアクションを作成することができます。アラームアクションが設定する以下のコンソールページの画面キャプチャー内に、設定の順番を付けました。また、アクションを適切に作成できるよう、次のシナリオの設定にも順番を付けました。

**Create Alarm**

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define. To edit an alarm, first choose whom to notify and then define when the notification should be sent.

Send a notification to:  [create topic](#)

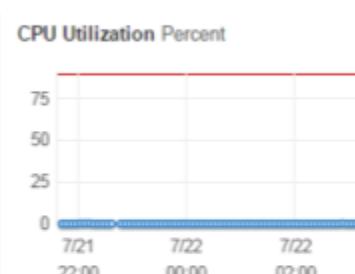
Take the action:  Recover this instance [i](#)  
 Stop this instance [i](#)  
 Terminate this instance [i](#)  
 Reboot this instance [i](#)

Whenever:  of   
Is:  Percent

For at least:  consecutive period(s) of

Name of alarm:

CPU Utilization Percent



Cancel

### シナリオ 1: アイドル状態の開発インスタンスおよびテストインスタンスを停止する

ソフトウェアの開発またはテストに使用するインスタンスが 1 時間以上アイドル状態である場合に停止するアラームを作成します。

設定	値
1	停止
2	最大
3	CPUUtilization
4	<=
5	10%
6	60 分

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
インスタンスを停止、終了、再起動、  
または復旧するアラームを作成する

設定	値
7	1

## シナリオ 2: アイドル状態のインスタンスを停止する

インスタンスが 24 時間アイドル状態である場合、インスタンスを停止し、メールを送信するアラームを作成します。

設定	値
1	停止および E メール
2	平均
3	CPUUtilization
4	<=
5	5%
6	60 分
7	24

## シナリオ 3: トラフィック量が異常に多いウェブサーバーについて E メールを送信する

インスタンスの 1 日当たりのアウトバウンドネットワークトラフィックが 10 GB を超える場合にメールを送信するアラームを作成します。

設定	値
1	メール
2	合計
3	NetworkOut
4	>
5	10 GB
6	1 日
7	1

## シナリオ 4: 異常な高トラフィック状態のウェブサーバーを停止する

アウトバウンドトラフィックが 1 時間当たり 1 GB を超えた場合にインスタンスを停止し、テキストメッセージ (SMS) を送信するアラームを作成します。

設定	値
1	Stop and send SMS

設定	値
2	合計
3	NetworkOut
4	>
5	1 GB
6	1 時間
7	1

## シナリオ 5: メモリリークが発生しているインスタンスを停止する

トラブルシューティングに使えるアプリケーションログを取得できるよう、メモリ使用率が 90% 以上になった場合にインスタンスを停止するアラームを作成します。

**Note**

MemoryUtilization メトリクスはカスタムメトリクスです。MemoryUtilization メトリクスを使用するには、Linux インスタンスの Perl スクリプトをインストールする必要があります。詳細については、「[Amazon EC2 Linux インスタンスのメモリとディスクのメトリクスのモニタリング](#)」を参照してください。

設定	値
1	停止
2	最大
3	MemoryUtilization
4	>=
5	90%
6	1 分
7	1

## シナリオ 6: 障害のあるインスタンスを停止する

3 回連続で状態チェック (5 分間隔で実施) が不合格のインスタンスを停止するアラームを作成します。

設定	値
1	停止
2	平均
3	StatusCheckFailed_System
4	>=
5	1

設定	値
6	15 分
7	1

### シナリオ 7: バッチ処理ジョブの完了時にインスタンスを削除する

バッチジョブを実行するインスタンスが結果データを送信しなくなったときに、そのインスタンスを削除するアラームを作成します。

設定	値
1	終了
2	最大
3	NetworkOut
4	<=
5	100,000 bytes
6	5 分
7	1

## CloudWatch イベントによる Amazon EC2 の自動化

Amazon CloudWatch Events を使用すると、AWS サービスを自動化して、アプリケーションの可用性の問題やリソースの変更などのシステムイベントに自動的に対応できます。AWS サービスからのイベントは、ほぼリアルタイムに CloudWatch イベントに提供されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。自動的にトリガーできるオペレーションには、以下が含まれます。

- AWS Lambda 関数の呼び出し
- Amazon EC2 Run Command の呼び出し
- Amazon Kinesis Data Streams へのイベントの中継
- AWS Step Functions ステートマシンのアクティブ化
- Amazon SNS トピックまたは Amazon SQS キューの通知

Amazon EC2 で CloudWatch イベントを使用する例をいくつか以下に示します。

- 新しい Amazon EC2 インスタンスが起動するたびに Lambda 関数をアクティブ化する。
- Amazon EBS ボリュームの作成時または変更時に Amazon SNS トピックを通知する。
- AWS の別のサービスで特定のイベントが発生するたびに、Amazon EC2 Run Command を使用して 1 つ以上の Amazon EC2 インスタンスにコマンドを送信する。

詳細については、「[Amazon CloudWatch Events ユーザーガイド](#)」を参照してください。

# Amazon EC2 Linux インスタンスのメモリとディスクのメトリクスのモニタリング

Amazon CloudWatch を使用して、EC2 インスタンスのオペレーティングシステムからメトリクスおよびログを収集できます。

## CloudWatch エージェント

CloudWatch エージェントを使用して、Amazon EC2 インスタンスとオンプレミスサーバーからシステムメトリクスとログファイルの両方を収集できます。エージェントでは Windows Server と Linux の両方がサポートされ、CPUあたりのコアのようなサブリソースメトリクスなど、収集するメトリクスを選択できます。メトリクスおよびログは、モニタリングスクリプトを使用せずに、エージェントを使用して収集することをお勧めします。詳細については、『Amazon CloudWatch ユーザーガイド』の「[CloudWatch エージェントを使用して Amazon EC2 インスタンスとオンプレミスサーバーからメトリクスを収集する](#)」を参照してください。

## CloudWatch モニタリングスクリプト

### Important

CloudWatch エージェントを使用してメトリクスおよびログを収集することをお勧めします。古いモニタリングスクリプトを使用して Linux インスタンスから情報を収集しているお客様には、モニタリングスクリプトに関する情報が提供されます。

モニタリングスクリプトは、Amazon CloudWatch のカスタムメトリクスを作成して利用する方法を示しています。これらの Perl スクリプトのサンプルは、Linux インスタンスのメモリ、スワップ、およびディスクスペースの使用状況メトリクスをレポートする、完全に機能する例で構成されます。

Amazon CloudWatch 標準のカスタムメトリクスの利用料金が、これらのスクリプトの使用に適用されます。詳細については、「[Amazon CloudWatch 料金表ページ](#)」を参照してください。

### コンテンツ

- サポートされているシステム (p. 673)
- 必要なアクセス許可 (p. 674)
- 必要なパッケージのインストール (p. 674)
- モニタリングスクリプトをインストールする (p. 675)
- mon-put-instance-data.pl (p. 676)
- mon-get-instance-stats.pl (p. 679)
- コンソールでのカスタムメトリクスの表示 (p. 680)
- トラブルシューティング (p. 680)

## サポートされているシステム

モニタリングスクリプトは、次のシステムを使用してインスタンスでテストされました。

- Amazon Linux 2
- Amazon Linux AMI 2014.09.2 以降
- Red Hat Enterprise Linux 6.9 および 7.4
- SUSE Linux Enterprise Server 12
- Ubuntu Server 14.04 および 16.04

## 必要なアクセス許可

IAM ロールをインスタンスに関連付けて、次のアクションを呼び出すアクセス許可がスクリプトにあることを確認します。

- cloudwatch:PutMetricData
- cloudwatch:GetMetricStatistics
- cloudwatch>ListMetrics
- ec2:DescribeTags

詳細については、「[IAM ロールの使用 \(p. 891\)](#)」を参照してください。

## 必要なパッケージのインストール

Linux の一部のバージョンでは、モニタリングスクリプトを使用する前に、Perl モジュールをインストールする必要があります。

Amazon Linux 2 および Amazon Linux AMI に必要なパッケージをインストールするには

- インスタンスにログオンします。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。
- コマンドプロンプトで以下のようにパッケージをインストールします。

```
sudo yum install -y perl-Switch perl-Datetime perl-Sys-Syslog perl-LWP-Protocol-https  
perl-Digest-SHA.x86_64
```

Ubuntu に必要なパッケージをインストールするには

- インスタンスにログオンします。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。
- コマンドプロンプトで以下のようにパッケージをインストールします。

```
sudo apt-get update  
sudo apt-get install unzip  
sudo apt-get install libwww-perl libdatetime-perl
```

Red Hat Enterprise Linux 7 に必要なパッケージをインストールするには

- インスタンスにログオンします。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。
- コマンドプロンプトで以下のようにパッケージをインストールします。

```
sudo yum install perl-Switch perl-Datetime perl-Sys-Syslog perl-LWP-Protocol-https  
perl-Digest-SHA --enablerepo="rhui-REGION-rhel-server-optional" -y  
sudo yum install zip unzip
```

Red Hat Enterprise Linux 6.9 に必要なパッケージをインストールするには

- インスタンスにログオンします。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。

2. コマンドプロンプトで以下のようにパッケージをインストールします。

```
sudo yum install perl-Datetime perl-CPAN perl-Net-SSLeay perl-IO-Socket-SSL perl-Digest-SHA gcc -y  
sudo yum install zip unzip
```

3. 昇格されたユーザーとして CPAN を実行します。

```
sudo cpan
```

次のプロンプトが表示されるまで、各プロンプトで Enter キーを押します。

```
cpan[1]>
```

4. CPAN プロンプトで、次の各コマンドを実行します。1 つのコマンドを実行してインストールを実行し、CPAN プロンプトに戻ったら次のコマンドを実行します。次の処理に進むことを求めるプロンプトが表示されたら、Enter キーを押します。

```
cpan[1]> install YAML  
cpan[2]> install LWP::Protocol::https  
cpan[3]> install Sys::Syslog  
cpan[4]> install Switch
```

SUSE に必要なパッケージをインストールするには

1. インスタンスにログオンします。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。
2. SUSE Linux Enterprise Server 12 が実行されているサーバーでは、perl-Switch パッケージのダウンロードが必要な場合があります。次のコマンドを使用して、このパッケージをダウンロードおよびインストールできます。

```
wget http://download.opensuse.org/repositories/devel:/languages:/perl/SLE_12_SP3/noarch/perl-Switch-2.17-32.1.noarch.rpm  
sudo rpm -i perl-Switch-2.17-32.1.noarch.rpm
```

3. 次のように、必要なパッケージをインストールします。

```
sudo zypper install perl-Switch perl-Datetime  
sudo zypper install -y "perl(LWP::Protocol::https)"
```

## モニタリングスクリプトをインストールする

以下の手順では、EC2 Linux インスタンスで CloudWatch Monitoring Scripts のダウンロード、解凍、構成を行う方法について示します。

モニタリングスクリプトのダウンロード、インストール、設定を行うには

1. コマンドプロンプトで、モニタリングスクリプトを保存するフォルダに移動し、次のコマンドを実行してモニタリングスクリプトをダウンロードします。

```
curl https://aws-cloudwatch.s3.amazonaws.com/downloads/  
CloudWatchMonitoringScripts-1.2.2.zip -O
```

2. ダウンロードしたモニタリングスクリプトをインストールするには、以下のコマンドを実行します。

```
unzip CloudWatchMonitoringScripts-1.2.2.zip && \
rm CloudWatchMonitoringScripts-1.2.2.zip && \
cd aws-scripts-mon
```

モニタリングスクリプトのパッケージに、以下のファイルが含まれます。

- CloudWatchClient.pm – 共通 Perl モジュール。これを使って、他のスクリプトから簡単に Amazon CloudWatch を呼び出すことができます。
- mon-put-instance-data.pl – Amazon EC2 インスタンス（メモリ、スワップ、ディスクスペースの使用状況）のシステムメトリクスを収集し、Amazon CloudWatch に送信します。
- mon-get-instance-stats.pl – Amazon CloudWatch に問い合わせて、このスクリプトが実行される EC2 インスタンスの最新の使用状況統計を表示します。
- awscreds.template – アクセスキー ID とシークレットアクセスキーを保存する AWS 認証情報のファイルテンプレートです。
- LICENSE.txt – Apache 2.0 のライセンスを含むテキストファイルです。
- NOTICE.txt – 著作権情報です。

## mon-put-instance-data.pl

このスクリプトは、現行システムにあるメモリ、スワップ、ディスクスペースの使用状況のデータを収集します。その後、Amazon CloudWatch へのリモート呼び出しを行って、収集したデータをカスタムメトリクスとしてレポートします。

### オプション

名前	説明
--mem-util	MemoryUtilization メトリクスをパーセント（%）単位で収集し、送信します。このメトリクスには、使用されているアプリケーションとオペレーティングシステムによって割り当てられたメモリがカウントされるほか、--mem-used-incl-cache-buff オプションを指定した場合は、使用されているキャッシュとバッファメモリもカウントされます。
--mem-used	メガバイト（MB）単位でレポートされる MemoryUsed メトリクスを収集し、送信します。このメトリクスには、使用されているアプリケーションとオペレーティングシステムによって割り当てられたメモリがカウントされるほか、--mem-used-incl-cache-buff オプションを指定した場合は、使用されているキャッシュとバッファメモリもカウントされます。
--mem-used-incl-cache-buff	このオプションを含めると、キャッシュおよびバッファに現在使用されているメモリは、--mem-util、--mem-used、--mem-avail のメトリクスがレポートされるときに、"used" としてカウントされます。
--mem-avail	メガバイト（MB）単位でレポートされる MemoryAvailable メトリクスを収集し、送信します。このメトリクスには、使用されているアプリケーションとオペレーティングシステムによって割り当てられたメモリがカウントされるほか、--mem-used-incl-cache-buff オプションを指定した場合は、使用されているキャッシュとバッファメモリもカウントされます。

名前	説明
--swap-util	パーセント(%) 単位でレポートされる SwapUtilization メトリクスを収集し、送信します。
--swap-used	メガバイト(MB) 単位でレポートされる SwapUsed メトリクスを収集し、送信します。
--disk-path=PATH	<p>レポートするディスクを選択します。</p> <p>PATH では、マウントポイント、またはレポートが必要なファイルシステムのマウントポイントにあるファイルを指定できます。複数のディスクを選択するには、それぞれに対して --disk-path=PATH を指定します。</p> <p>/ および /home にマウントされたファイルシステムのディスクを選択するには、パラメータを使用します。</p> <p>--disk-path=/ --disk-path=/home</p>
--disk-space-util	<p>選択したディスクについて、DiskSpaceUtilization メトリクスを収集し送信します。メトリクスはパーセンテージでレポートされます。</p> <p>このスクリプトによって計算されたディスクの使用状況メトリクスは、df -k -l コマンドによって計算された値とは異なることに注意してください。df -k -l の値のほうが有用であると判断した場合は、スクリプトのほうの計算値を変更できます。</p>
--disk-space-used	<p>選択したディスクについて、DiskSpaceUsed メトリクスを収集し送信します。メトリクスは、デフォルトにより、ギガバイトでレポートされます。</p> <p>Linux オペレーティングシステムには予約ディスクスペースがあるため、使用済みディスクスペースと使用可能なディスクスペースを合計しても正確なディスクスペースの合計にならないことがあります。</p>
--disk-space-avail	<p>選択したディスクについて、DiskSpaceAvailable メトリクスを収集し送信します。メトリクスはギガバイトでレポートされます。</p> <p>Linux オペレーティングシステムには予約ディスクスペースがあるため、使用済みディスクスペースと使用可能なディスクスペースを合計しても正確なディスクスペースの合計にならないことがあります。</p>
--memory-units=UNITS	メモリ使用量をレポートする単位を指定します。指定がない場合、メモリはメガバイト(MB)でレポートされます。UNITS は、バイト(B)、キロバイト(KB)、メガバイト(MB)、ギガバイト(GB)のいずれかになります。
--disk-space-units=UNITS	ディスクスペース使用量をレポートする単位を指定します。指定がない場合、ディスクスペースはギガバイト(GB)でレポートされます。UNITS は、バイト(B)、キロバイト(KB)、メガバイト(MB)、ギガバイト(GB)のいずれかになります。
--aws-credential-file=PATH	<p>AWS 認証情報を持っているファイルの場所を提供します。</p> <p>このパラメータは、--aws-access-key-id および --aws-secret-key パラメータと一緒にには使用できません。</p>

名前	説明
--aws-access-key-id=VALUE	発信者を識別するために使用する AWS アクセスキー ID を指定します。--aws-secret-key オプションと一緒に使用する必要があります。このオプションを --aws-credential-file パラメータと一緒に使用しないでください。
--aws-secret-key=VALUE	CloudWatch へのリクエストの署名に使用する AWS シークレット アクセスキーを指定します。--aws-access-key-id オプションと一緒に使用する必要があります。このオプションを --aws-credential-file パラメータと一緒に使用しないでください。
--aws-iam-role=VALUE	AWS 認証情報を提供するために使用する IAM ロールを指定します。値 =VALUE が必要です。認証情報が指定されていない場合、EC2 インスタンスに関連付けられたデフォルトの IAM ロールが適用されます。使用できる IAM ロールは 1 つのみです。IAM ロールが検出されない場合、または 2 つ以上の IAM ロールが検出された場合、スクリプトはエラーを返します。  このオプションを --aws-credential-file、--aws-access-key-id、または --aws-secret-key パラメータと併せて使用しないでください。
--aggregated[=only]	インスタンスタイプ、AMI ID、リージョン全体の集約されたメトリクスを追加します。値 =only はオプションです。指定した場合、スクリプトは集約されたメトリクスのみをレポートします。
--auto-scaling[=only]	Auto Scaling グループの集約されたメトリクスを追加します。値 =only はオプションです。指定すると、スクリプトは Auto Scaling メトリクスのみをレポートします。スクリプトを使って IAM アカウントまたはロールに関連付けられている <a href="#">IAM ポリシー</a> には、EC2 アクション <a href="#">DescribeTags</a> を呼び出すアクセス許可が必要になります。
--verify	メトリクスを収集するスクリプトのテストランを実行したり、完全な HTTP リクエストを用意したりしますが、実際に CloudWatch を呼び出してデータをレポートすることはありません。このオプションで、認証情報が提供されていることも確認できます。冗長モードで実行すると、このオプションは CloudWatch に送信するメトリクスを出力します。
--from-cron	cron からスクリプトを呼び出す際はこのオプションを使用します。このオプションを使用すると、すべての診断出力が抑えられますが、エラーメッセージがユーザー アカウントのローカルシステムログに送信されます。
--verbose	スクリプトの実行内容の詳細を表示します。
--help	使用状況の情報を表示します。
--version	スクリプトのバージョン番号を表示します。

#### 例

次の例では、IAM ロールまたは `awscreds.conf` ファイルを指定していることを前提としています。それ以外の場合は、これらのコマンドで `--aws-access-key-id` および `--aws-secret-key` パラメータを使用して認証情報を指定する必要があります。

次の例では、CloudWatch にデータを送信せずに簡単なテストを実行します。

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

以下の例では、使用可能なメモリメトリクスをすべて収集し、CloudWatch に送信して、使用されているキャッシュとバッファメモリをカウントします。

```
./mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --mem-used --mem-avail
```

以下の例では、Auto Scaling グループの集約メトリクスを収集し、個々のインスタンスマトリクスをレポートすることなく Amazon CloudWatch に送信します。

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --auto-scaling=only
```

以下の例では、インスタンスタイプ、AMI ID、リージョンの集約されたメトリクスを収集し、個々のインスタンスマトリクスをレポートすることなく Amazon CloudWatch に送信します。

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail --aggregated=only
```

CloudWatch にレポートされたメトリクスの cron スケジュールを設定するには、crontab -e コマンドを使用して crontab の編集を開始します。5 分ごとにメモリとディスクスペースの使用状況を CloudWatch にレポートするには、以下のコマンドを追加します。

```
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl --mem-used-incl-cache-buff --mem-util --disk-space-util --disk-path=/ --from-cron
```

スクリプトにエラーが発生した場合、スクリプトのシステムログにエラーメッセージが書き込まれます。

## mon-get-instance-stats.pl

このスクリプトは、直近の時間数を用いて、指定された時間間隔内で、メモリ、スワップ、ディスクスペースメトリクスの統計について CloudWatch に問い合わせます。このデータは、このスクリプトが実行される Amazon EC2 インスタンスに関するものです。

### オプション

名前	説明
--recent-hours=N	レポートする直近の時間数を N で表記して指定します。ここで N は整数です。
--aws-credential-file=PATH	AWS 認証情報を持っているファイルの場所を提供します。
--aws-access-key-id=VALUE	発信者を識別するために使用する AWS アクセスキー ID を指定します。--aws-secret-key オプションと一緒に使用する必要があります。このオプションを --aws-credential-file オプションと併せて使用しないでください。
--aws-secret-key=VALUE	CloudWatch へのリクエストの署名に使用する AWS シークレットアクセスキーを指定します。--aws-access-key-id オプションと一緒に使用する必要があります。このオプションを --aws-credential-file オプションと併せて使用しないでください。
--aws-iam-role=VALUE	AWS 認証情報を提供するために使用する IAM ロールを指定します。値 =VALUE が必要です。認証情報が指定されていない場合、EC2 インスタンスに関連付けられたデフォルトの IAM ロールが適用されます。使用できる IAM ロールは 1 つのみです。IAM ロール

名前	説明
	が検出されない場合、または 2 つ以上の IAM ロールが検出された場合、スクリプトはエラーを返します。 このオプションを <code>--aws-credential-file</code> 、 <code>--aws-access-key-id</code> 、または <code>--aws-secret-key</code> パラメータと併せて使用しないでください。
<code>--verify</code>	スクリプトのテストを実行します。このオプションで、認証情報が提供されていることも確認できます。
<code>--verbose</code>	スクリプトの実行内容の詳細を表示します。
<code>--help</code>	使用状況の情報を表示します。
<code>--version</code>	スクリプトのバージョン番号を表示します。

#### 例

過去 12 時間の利用統計情報を取得するには、次のコマンドを実行します。

```
./mon-get-instance-stats.pl --recent-hours=12
```

以下に、応答の例を示します。

```
Instance metric statistics for the last 12 hours.

CPU Utilization
    Average: 1.06%, Minimum: 0.00%, Maximum: 15.22%

Memory Utilization
    Average: 6.84%, Minimum: 6.82%, Maximum: 6.89%

Swap Utilization
    Average: N/A, Minimum: N/A, Maximum: N/A

Disk Space Utilization on /dev/xvda1 mounted as /
    Average: 9.69%, Minimum: 9.69%, Maximum: 9.69%
```

## コンソールでのカスタムメトリクスの表示

正常に `mon-put-instance-data.pl` スクリプトを実行すると、Amazon CloudWatch コンソールでカスタムメトリクスを確認できます。

カスタムメトリクスを表示するには

- 前述のとおりに `mon-put-instance-data.pl` を実行します。
- <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
- [View Metrics] を選択します。
- [Viewing] では、スクリプトによって投入されたカスタムメトリクスが System/Linux というプレフィックス付きで表示されます。

## トラブルシューティング

CloudWatchClient.pm モジュールは、インスタンスのメタデータをローカルでキャッシュします。モニタリングスクリプトを実行しているインスタンスから AMI を作成すると、キャッシング TTL (デフォルト: 6 時

間、Auto Scaling グループでは 24 時間) 以内にこの AMI から起動したすべてのインスタンスは、元のインスタンスのインスタンス ID を使用してメトリクスを出力します。キャッシング TTL 期間が経過した後は、スクリプトは新しいデータを取得し、モニタリングスクリプトは現在のインスタンスのインスタンス ID を使用します。これをすぐに修正するには、次のコマンドを使用してキャッシングされたデータを削除します。

```
rm /var/tmp/aws-mon/instance-id
```

## AWS CloudTrail による Amazon EC2 および Amazon EBS の API コールのログ記録

Amazon EC2 および Amazon EBS は AWS CloudTrail と統合されています。このサービスは、Amazon EC2 および Amazon EBS 内でユーザーやロール、または AWS のサービスによって実行されたアクションを記録するサービスです。CloudTrail は、コンソールからのコールや、API オペレーションへのコード呼び出しを含む、Amazon EC2 および Amazon EBS のすべての API コールをイベントとしてキャプチャします。証跡を作成する場合は、Amazon EC2 や Amazon EBS のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [Event history (イベント履歴)] で最新のイベントを表示できます。CloudTrail によって収集された情報を使用して、リクエストの作成元の IP アドレス、リクエストの実行者、リクエストの実行日時などの詳細を調べて、Amazon EC2 および Amazon EBS に対してどのようなリクエストが行われたかを判断できます。

CloudTrail の詳細については、「[AWS CloudTrail User Guide](#)」を参照してください。

## CloudTrail での Amazon EC2 と Amazon EBS に関する情報

CloudTrail は、アカウント作成時に AWS アカウントで有効になります。Amazon EC2 および Amazon EBS でアクティビティが発生すると、そのアクティビティは [Event history (イベント履歴)] に AWS の他のサービスのイベントと共に CloudTrail イベントとして記録されます。最近のイベントは、AWS アカウントで表示、検索、ダウンロードできます。詳細については、「[CloudTrail イベント履歴でのイベントの表示](#)」を参照してください。

Amazon EC2 または Amazon EBS のイベントなど、AWS アカウントのイベントの継続的な記録用に、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべてのリージョンに適用されます。証跡では、AWS パーティションのすべてのリージョンからのイベントがログに記録され、指定した Amazon S3 バケットにログファイルが配信されます。さらに、より詳細な分析と AWS ログで収集されたデータに基づいた行動のためにその他の CloudTrail サービスを設定できます。詳細については、以下のトピックを参照してください。

- [証跡を作成するための概要](#)
- [CloudTrail でサポートされるサービスと統合](#)
- [CloudTrail の Amazon SNS 通知の設定](#)
- 「[複数のリージョンから CloudTrail ログファイルを受け取る](#)」と「[複数のアカウントから CloudTrail ログファイルを受け取る](#)」

Amazon EC2 および Amazon EBS のアクションはすべて CloudTrail によって記録されます。また、これらのアクションは「[Amazon EC2 API Reference](#)」で説明されています。たとえば、[RunInstances](#)、[DescribeInstances](#)、または [CreateImage](#) アクションへの呼び出しにより、CloudTrail ログファイルのエントリが生成されます。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。この ID 情報は以下のことを確認するのに役立ちます。

- リクエストが、ルートと IAM ユーザー認証情報のどちらを使用して送信されたか。
- リクエストが、ロールとフェデレーティッドユーザーのどちらの一時的なセキュリティ認証情報を使用して送信されたか。
- リクエストが、別の AWS サービスによって送信されたかどうか。

詳細については、「[CloudTrail userIdentity 要素](#)」を参照してください。

## Amazon EC2 および Amazon EBS のログファイルエントリの概要

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できる設定です。CloudTrail ログファイルには、1つ以上のログエントリが含まれます。イベントは任意の送信元からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次のログファイルレコードは、ユーザーがインスタンスを終了したことを示しています。

```
{  
    "Records": [  
        {  
            "eventVersion": "1.03",  
            "userIdentity": {  
                "type": "Root",  
                "principalId": "123456789012",  
                "arn": "arn:aws:iam::123456789012:root",  
                "accountId": "123456789012",  
                "accessKeyId": "AKIAIOSFODNN7EXAMPLE",  
                "userName": "user"  
            },  
            "eventTime": "2016-05-20T08:27:45Z",  
            "eventSource": "ec2.amazonaws.com",  
            "eventName": "TerminateInstances",  
            "awsRegion": "us-west-2",  
            "sourceIPAddress": "198.51.100.1",  
            "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7botocore/1.4.1",  
            "requestParameters": {  
                "instancesSet": {  
                    "items": [  
                        {  
                            "instanceId": "i-1a2b3c4d"  
                        }  
                    ]  
                }  
            },  
            "responseElements": {  
                "instancesSet": {  
                    "items": [  
                        {  
                            "instanceId": "i-1a2b3c4d",  
                            "currentState": {  
                                "code": 32,  
                                "name": "shutting-down"  
                            },  
                            "previousState": {  
                                "code": 16,  
                                "name": "running"  
                            }  
                        }  
                    ]  
                }  
            }  
        }  
    ]  
}
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EC2 Instance Connect を介し  
て接続するユーザーを監査する

```
        },
        "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
        "eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
        "eventType": "AwsApiCall",
        "recipientAccountId": "123456789012"
    }
}
```

## EC2 Instance Connect を介して接続するユーザーを AWS CloudTrail で監査する

EC2 Instance Connect を介してインスタンスに接続するユーザーを AWS CloudTrail で監査します。

AWS CloudTrail コンソールを使用して EC2 Instance Connect 経由で SSH アクティビティを監査するには

1. AWS CloudTrail コンソール (<https://console.aws.amazon.com/cloudtrail/>) を開きます。
2. 正しいリージョンを使用していることを確認します。
3. ナビゲーションペインで [Event history (イベント履歴)] を選択します。
4. [Filter (フィルター)] で、[Event source (イベントソース)]、[ec2-instance-connect.amazonaws.com] の順に選択します。
5. (オプション) [Time range (時間範囲)] で、時間範囲を選択します。
6. [Refresh events (イベントの更新)] アイコンを選択します。
7. **SendSSHPublicKey** API コールに対応するイベントがページに表示されます。矢印を使用してイベントを開します。ユーザー名、SSH 接続を行うために使用した AWS アクセスキー、ソース IP アドレスなどの詳細が表示されます。
8. すべてのイベント情報を JSON 形式で表示するには、[View event (イベントの表示)] を選択します。[requestParameters] フィールドに、SSH 接続を行うために使用されたターゲットインスタンス ID、OS ユーザー名、およびパブリックキーが表示されます。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDEFGGONGNOMOOOCB6XYTQEXAMPLE",
        "arn": "arn:aws:iam::1234567890120:user/IAM-friendly-name",
        "accountId": "123456789012",
        "accessKeyId": "ABCDEFGHIJKLMNO01234567890EXAMPLE",
        "userName": "IAM-friendly-name",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-09-21T21:37:58Z"
            }
        }
    },
    "eventTime": "2018-09-21T21:38:00Z",
    "eventSource": "ec2-instance-connect.amazonaws.com",
    "eventName": "SendSSHPublicKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "aws-cli/1.15.61 Python/2.7.10 Darwin/16.7.0 botocore/1.10.60",
    "requestParameters": {
        "instanceId": "i-0123456789EXAMPLE",
        "osUser": "ec2-user",
        "SSHKey": {
            "publicKey": "ssh-rsa ABCDEFGHIJKLMNOP01234567890EXAMPLE"
        }
    }
}
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EC2 Instance Connect を介し  
て接続するユーザーを監査する

```
"responseElements": null,  
"requestID": "1a2s3d4f-bde6-11e8-a892-f7ec64543add",  
"eventID": "1a2w3d4r5-a88f-4e28-b3bf-30161f75be34",  
"eventType": "AwsApiCall",  
"recipientAccountId": "0987654321"  
}
```

CloudTrail イベントを S3 バケット内に収集するように AWS アカウントを設定している場合は、ロググラムで情報をダウンロードして監査できます。詳細については、AWS CloudTrail User Guide の「[CloudTrail ログファイルの取得と表示](#)」を参照してください。

# Amazon EC2におけるネットワーク

Amazon EC2は、以下のようなネットワーク機能を備えています。

## 特徴

- [Amazon EC2 インスタンスの IP アドレッシング \(p. 685\)](#)
- [自分の IP アドレスを使用する \(BYOIP\) \(p. 701\)](#)
- [Elastic IP アドレス \(p. 705\)](#)
- [Elastic Network Interface \(p. 713\)](#)
- [Linux の拡張ネットワーキング \(p. 737\)](#)
- [Elastic Fabric Adapter \(p. 763\)](#)
- [プレイスメントグループ \(p. 791\)](#)
- [EC2 インスタンスの最大ネットワーク送信単位 \(MTU\) \(p. 801\)](#)
- [Virtual Private Cloud \(p. 804\)](#)
- [EC2-Classic \(p. 804\)](#)

## Amazon EC2 インスタンスの IP アドレッシング

Amazon EC2 と Amazon VPC は、IPv4 と IPv6 の両方のアドレス設定プロトコルをサポートします。デフォルトでは、Amazon EC2 と Amazon VPC は IPv4 アドレス設定プロトコルを使用します。この動作を無効にすることはできません。VPC の作成時には IPv4 CIDR ブロック (プライベート IPv4 アドレスの範囲) を指定する必要があります。必要に応じて、IPv6 CIDR ブロックを VPC とサブネットに割り当て、そのブロックからサブネットのインスタンスに IPv6 アドレスを割り当てるることができます。IPv6 アドレスはインターネットから到達できます。VPC での IPv6 の詳細については、『Amazon VPC ユーザーガイド』の「[VPC での IP アドレス指定](#)」を参照してください。

## コンテンツ

- [プライベート IPv4 アドレスと内部 DNS ホスト名 \(p. 685\)](#)
- [パブリック IPv4 アドレスと外部 DNS ホスト名 \(p. 686\)](#)
- [Elastic IP アドレス \(IPv4\) \(p. 687\)](#)
- [Amazon DNS サーバー \(p. 687\)](#)
- [IPv6 アドレス \(p. 687\)](#)
- [インスタンスの IP アドレスの使用 \(p. 688\)](#)
- [複数の IP アドレス \(p. 693\)](#)

## プライベート IPv4 アドレスと内部 DNS ホスト名

プライベート IPv4 アドレスは、インターネットから到達できない IP アドレスです。プライベート IPv4 アドレスは、同じネットワーク内のインスタンス間の通信に使用できます。プライベート IPv4 アドレスの標準および仕様については、[RFC 1918](#) を参照してください。DHCP を使用してインスタンスにプライベート IPv4 アドレスが割り当てられます。

### Note

RFC 1918 に指定されているプライベート IPv4 アドレスの範囲に含まれない、パブリックにルーティングできる CIDR ブロックを持つ VPC を作成できます。ただし、このドキュメントでプラ

イベート IPv4 アドレス（または「プライベート IP アドレス」）と言う場合は、VPC の IPv4 CIDR 範囲に含まれる IP アドレスを指します。

インスタンスを起動すると、そのインスタンスのプライマリプライベート IPv4 アドレスが割り当てられます。また、各インスタンスには、プライマリプライベート IPv4 アドレスに解決される内部 DNS ホスト名 (ip-10-251-50-12.ec2.internal など) が割り当てられます。同じ VPC 内のインスタンス間の通信に内部 DNS ホスト名を使用できますが、VPC の外部で内部 DNS ホスト名を解決することはできません。

インスタンスはサブネットの IPv4 アドレス範囲からプライマリプライベート IP アドレスを受け取ります。詳細については、『Amazon VPC ユーザーガイド』の「[VPC とサブネットのサイズ設定](#)」を参照してください。プライマリプライベート IP アドレスを指定しないでインスタンスを起動すると、サブネットの IPv4 範囲内で使用可能な IP アドレスが自動的に選択されます。各インスタンスには、プライマリプライベート IPv4 アドレスが割り当てられたデフォルトのネットワークインターフェイス (eth0) があります。追加のプライベート IPv4 アドレス（セカンダリプライベート IPv4 アドレス）も指定できます。プライマリプライベート IP アドレスとは異なり、セカンダリプライベート IP アドレスは、別のインスタンスに割り当て直すことができます。詳細については、「[複数の IP アドレス \(p. 693\)](#)」を参照してください。

プライベート IPv4 アドレスは、プライマリアドレスまたはセカンダリアドレスを問わず、インスタンスを停止して再起動してもネットワークインターフェイスに関連付けられたままになり、インスタンスを終了すると解放されます。

## パブリック IPv4 アドレスと外部 DNS ホスト名

パブリック IP アドレスは、インターネットから到達可能な IPv4 アドレスです。インスタンスとインターネット間で通信するには、パブリック アドレスを使用できます。

パブリック IP アドレスを受け取る各インスタンスには、外部 DNS ホスト名 (ec2-203-0-113-25.compute-1.amazonaws.com など) が割り当てられます。外部 DNS ホスト名を、VPC の外部からインスタンスのパブリック IP アドレスに解決し、VPC の内部からインスタンスのプライベート IPv4 アドレスに解決します。パブリック IP アドレスは、ネットワークアドレス変換 (NAT) によって、プライマリプライベート IP アドレスにマッピングされます。詳細については、「[RFC 1631: The IP Network Address Translator \(NAT\)](#)」を参照してください。

デフォルトの VPC でインスタンスを起動すると、デフォルトでパブリック IP アドレスが割り当てられます。デフォルト以外の VPC でインスタンスを起動するとき、サブネットには、そのサブネットで起動するインスタンスがパブリック IPv4 アドレスプールからパブリック IP アドレスを受け取るかどうかを決定する属性があります。デフォルトでは、デフォルト以外のサブネットで起動されたインスタンスにパブリック IP アドレスを割り当てません。

インスタンスがパブリック IP アドレスを割り当てられるかどうかを制御するには、以下の方法を使用します。

- サブネットのパブリック IP アドレス属性を変更する。詳細については、『Amazon VPC ユーザーガイド』の「[サブネットの IPv4 アドレス指定属性の変更](#)」を参照してください。
- 起動時にパブリック IP アドレス機能を有効または無効にする。これにより、サブネットのパブリック IP アドレス属性がオーバーライドされます。詳細については、「[インスタンス起動時のパブリック IPv4 アドレスの割り当て \(p. 691\)](#)」を参照してください。

パブリック IP アドレスは、Amazon のパブリック IPv4 アドレスプールからインスタンスに割り当てられ、お客様の AWS アカウントには関連付けられません。パブリック IP アドレスをインスタンスから割り当て解除すると、そのパブリック IPv4 アドレスはパブリック IP アドレスプールに戻され、再利用することはできません。

手動でパブリック IP アドレスをインスタンスに関連付けること、また、手動でインスタンスから割り当て解除することはできません。場合によって、パブリック IP アドレスはインスタンスから解放されたり、新しいインスタンスに割り当てられたりします。

- インスタンスのパブリック IP アドレスが停止または終了すると、インスタンスのパブリック IP アドレスが解放されます。停止していたインスタンスが再起動されると、そのインスタンスには新しいパブリック IP アドレスが送信されます。
- Elastic IP アドレスをこれに関連付けると、インスタンスのパブリック IP アドレスが解放されまします。Elastic IP アドレスをインスタンスから割り当て解除すると、そのインスタンスには新しいパブリック IP アドレスが送信されます。
- VPC 内のインスタンスのパブリック IP アドレスが既に解放されている場合には、複数のネットワークインターフェイスがインスタンスにアタッチされていると、インスタンスに新しいパブリック IP アドレスは送信されません。
- インスタンスのパブリック IP アドレスが解放され、Elastic IP アドレスに関連付けられたセカンダリプライベート IP アドレスがある場合、インスタンスは新しいパブリック IP アドレスを受信しません。

必要に応じて、インスタンスに関連付けおよびインスタンスから関連付けできる永続的なパブリック IP アドレスが必要な場合は、Elastic IP アドレスを使用します。

動的 DNS を使用して既存の DNS 名を新しいインスタンスのパブリック IP アドレスにマッピングした場合、その IP アドレスがインターネット内に伝達されるまでに最大 24 時間かかることがあります。その結果、新しいインスタンスはトラフィックを受信せず、終了したインスタンスがリクエストの受信を継続することができます。この問題を解決するには、Elastic IP アドレスを使用します。独自の Elastic IP アドレスを割り当てて、それをインスタンスに関連付けることができます。詳細については、「[Elastic IP アドレス \(p. 705\)](#)」を参照してください。

インスタンスに Elastic IP アドレスを割り当てるとき、DNS ホスト名が有効な場合、インスタンスは IPv4 DNS ホスト名を受け取ります。詳細については、「Amazon VPC ユーザーガイド」の「[VPC での DNS の使用](#)」を参照してください。

#### Note

インスタンスがパブリック NAT IP アドレスを使用して他のインスタンスにアクセスする場合、アクセス先のインスタンスが同じリージョンにあるかどうかによって、リージョンデータ転送またはインターネットデータ転送に対して課金されます。

## Elastic IP アドレス (IPv4)

Elastic IP アドレスは、アカウントに割り当てることができるパブリック IPv4 アドレスです。必要に応じて、インスタンスに関連付けることができます。これは、解放するように選択しない限り、インスタンスに割り当てられたままです。Elastic IP アドレスとその使用方法の詳細については、「[Elastic IP アドレス \(p. 705\)](#)」を参照してください。

IPv6 に対する Elastic IP アドレスはサポートされていません。

## Amazon DNS サーバー

Amazon は、Amazon が提供する IPv4 DNS ホスト名を解決する DNS サーバーを IPv4 アドレスに提供します。Amazon DNS サーバーは VPC ネットワークの範囲に 2 をプラスしたアドレスにあります。詳細については、「Amazon VPC ユーザーガイド」の「[Amazon DNS サーバー](#)」を参照してください。

## IPv6 アドレス

必要に応じて、IPv6 CIDR ブロックを VPC と関連付けることができます。また、IPv6 CIDR ブロックをサブネットと関連付けることができます。VPC の IPv6 CIDR ブロックは、Amazon の IPv6 アドレスのプールから自動的に割り当てられます。独自にアドレス範囲を選択することはできません。詳細については、「Amazon VPC ユーザーガイド」の次のトピックを参照してください。

- [IPv6 用の VPC とサブネットのサイズ設定](#)

- IPv6 CIDR ブロックと VPC の関連付け
- IPv6 CIDR ブロックとサブネットの関連付け

IPv6 アドレスはグローバルに一意であるため、インターネット経由で到達可能です。IPv6 CIDR ブロックが VPC およびサブネットと関連付けられていて、以下のいずれかに該当する場合、インスタンスには IPv6 アドレスが割り当てられます。

- 起動時にサブネットからインスタンスに IPv6 アドレスが自動的に割り当てられるように設定されている。詳細については、「[サブネットの IPv6 アドレス指定属性の変更](#)」を参照してください。
- 起動時に IPv6 アドレスをインスタンスに割り当てる。
- 起動後に IPv6 アドレスをインスタンスのプライマリネットワークインターフェイスに割り当てる。
- 起動後に IPv6 アドレスを同じサブネットのネットワークインターフェイスに割り当て、そのネットワークインターフェイスをインスタンスにアタッチする。

起動時にインスタンスに IPv6 アドレスが割り当てられると、そのアドレスはインスタンスのプライマリネットワークインターフェイス (eth0) と関連付けられます。IPv6 アドレスとネットワークインターフェイスの関連付けは解除できます。インスタンスの IPv6 DNS ホスト名はサポートされていません。

IPv6 アドレスは、インスタンスの停止および開始時には保持され、インスタンスの終了時に解放されます。IPv6 アドレスは、別のネットワークインターフェイスに割り当てられている間は再割り当てできません。最初に割り当てを解除する必要があります。—

追加の IPv6 アドレスをインスタンスに割り当てるには、インスタンスにアタッチされたネットワークインターフェイスにアドレスを割り当てます。ネットワークインターフェイスに割り当てることができる IPv6 アドレスの数と、インスタンスにアタッチできるネットワークインターフェイスの数は、インスタンスタイプごとに異なります。詳細については、「[各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数 \(p. 714\)](#)」を参照してください。

## インスタンスの IP アドレスの使用

インスタンスに割り当てられた IP アドレスを表示し、起動時にパブリック IPv4 アドレスまたは IPv6 アドレスをインスタンスに割り当てるすることができます。

### コンテンツ

- [パブリック IP アドレス、プライベート IP アドレス、Elastic IP アドレスの確認 \(p. 688\)](#)
- [IPv6 アドレスの確認 \(p. 690\)](#)
- [インスタンス起動時のパブリック IPv4 アドレスの割り当て \(p. 691\)](#)
- [インスタンスへの IPv6 アドレスの割り当て \(p. 692\)](#)
- [インスタンスからの IPv6 アドレスの割り当て解除 \(p. 692\)](#)

## パブリック IP アドレス、プライベート IP アドレス、Elastic IP アドレスの確認

Amazon EC2 コンソールを使用して、インスタンスのプライベート IPv4 アドレス、パブリック IPv4 アドレス、および Elastic IP アドレスを確認できます。また、インスタンスマタデータを使用して、インスタンス内からインスタンスのパブリック IPv4 アドレスとプライベート IPv4 アドレスを確認することもできます。詳細については、「[インスタンスマタデータとユーザーデータ \(p. 593\)](#)」を参照してください。

コンソールを使用してインスタンスのプライベート IPv4 アドレスを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択します。詳細ペインで、[Private IPs] フィールドからプライベート IPv4 アドレスを取得し、[Private DNS] フィールドから内部 DNS ホスト名を取得します。
4. インスタンスに接続されているネットワークインターフェイスに割り当てられた 1 つ以上のセカンダリプライベート IPv4 アドレスがある場合は、[Secondary private IPs] フィールドからこれらの IP アドレスを取得します。
5. または、ナビゲーションペインで [Network Interfaces] を選択し、インスタンスに関連付けられているネットワークインターフェイスを選択します。
6. [Primary private IPv4 IP] フィールドからプライマリプライベート IP アドレス、および [Private DNS (IPv4)] フィールドから内部 DNS ホスト名を取得します。
7. ネットワークインターフェイスにセカンダリプライベート IP アドレスを割り当てる場合は、[Secondary private IPv4 IPs] フィールドからこれらの IP アドレスを取得します。

コンソールを使用してインスタンスのパブリック IPv4 アドレスを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択します。詳細ペインで、[IPv4 Public IP] フィールドからパブリック IP アドレスを、[Public DNS (IPv4)] フィールドから外部 DNS ホスト名を取得します。
4. 1 つ以上の Elastic IP アドレスがインスタンスに関連付けられている場合は、Elastic IP アドレスを [Elastic IPs] フィールドから取得します。

Note

インスタンスにパブリック IPv4 アドレスがなくても、インスタンスのネットワークインターフェイスに Elastic IP アドレスが関連付けられている場合、[IPv4 Public IP] フィールドに Elastic IP アドレスが表示されます。

5. または、ナビゲーションペインで [Network Interfaces] を選択し、インスタンスに関連付けられているネットワークインターフェイスを選択します。
6. [IPv4 Public IP] フィールドからパブリック IP アドレスを取得します。アスタリスク (\*) は、プライマリプライベート IPv4 アドレスにマッピングされているパブリック IPv4 アドレスまたは Elastic IP アドレスを示します。

Note

パブリック IPv4 アドレスは、コンソールのネットワークインターフェイスのプロパティとして表示されますが、NAT によってプライマリプライベート IPv4 アドレスにマッピングされます。したがって、インスタンスのネットワークインターフェイスのプロパティを、たとえば `ifconfig` (Linux) または `ipconfig` (Windows) を通して調べてみると、パブリック IPv4 アドレスは表示されていません。インスタンス内からインスタンスのパブリック IPv4 アドレスを確認するには、インスタンスのメタデータを使用できます。

インスタンスのメタデータを使用してインスタンスの IPv4 アドレスを確認するには

1. インスタンスに接続します。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。
2. プライベート IP アドレスにアクセスするには、次のコマンドを使用します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-ipv4
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4
```

3. パブリック IP アドレスにアクセスするには、次のコマンドを使用します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-ipv4
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-ipv4
```

インスタンスに Elastic IP アドレスが関連付けられている場合、返される値は Elastic IP アドレスの値です。

## IPv6 アドレスの確認

Amazon EC2 コンソールを使用してインスタンスの IPv6 アドレスを確認できます。

コンソールを使用してインスタンスの IPv6 アドレスを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択します。詳細ペインで、[IPv6 IPs] から IPv6 アドレスを取得します。

インスタンスのメタデータを使用してインスタンスの IPv6 アドレスを確認するには

1. インスタンスに接続します。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。
2. 次のコマンドを使用して IPv6 アドレスを表示します (<http://169.254.169.254/latest/meta-data/network/interfaces/macs/> から MAC アドレスを取得できます)。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/mac-address/ipv6s
```

## インスタンス起動時のパブリック IPv4 アドレスの割り当て

各サブネットに、そのサブネット内で起動されるインスタンスにパブリック IP アドレスが割り当てられるかどうかを決定する属性があります。デフォルトでは、デフォルト以外のサブネットではこの属性が `false` に設定されており、デフォルトのサブネットではこの属性が `true` に設定されています。インスタンスを起動する場合、パブリック IPv4 アドレス指定機能を使用してインスタンスにパブリック IPv4 アドレスを割り当てるかどうかを制御することもできます。サブネットの IP アドレス指定属性のデフォルトの動作をオーバーライドできます。パブリック IPv4 アドレスは、Amazon のパブリック IPv4 アドレスプールから割り当てられ、デバイスインデックス `eth0` を持つネットワークインターフェイスに割り当てられます。この機能は、インスタンス起動時の特定の条件により異なります。

### Important

起動後に、インスタンスからパブリック IP アドレスの割り当てを手動で解除することはできません。ただし、特定の場合に、アドレスが自動的に解放され、その後再利用できなくなります。詳細については、「[パブリック IPv4 アドレスと外部 DNS ホスト名 \(p. 686\)](#)」を参照してください。お客様の意志で関連付けたり関連付けを解除したりできる永続的なパブリック IP アドレスを必要とする場合は、起動してからインスタンスに Elastic IP アドレスを割り当てます。詳細については、「[Elastic IP アドレス \(p. 705\)](#)」を参照してください。

インスタンス起動時にパブリック IP アドレス機能にアクセスするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. AMI およびインスタンスタイプを選択し、[Next: Configure Instance Details] を選択します。
4. [Configure Instance Details] ページの [Network] で VPC を選択します。[Auto-assign Public IP] リストが表示されます。[Enable] または [Disable] を選択して、サブネットのデフォルトの設定をオーバーライドします。

### Important

複数のネットワークインターフェイスを指定した場合、パブリック IP アドレスを自動割り当てすることはできません。さらに、`eth0` のように既存のネットワークインターフェイスを指定すると、パブリック IP の自動割り当て機能を使用してサブネット設定をオーバーライドすることはできません。

5. ウィザードの後続ページに表示されるステップにしたがって、インスタンスのセットアップを最後まで実行します。ウィザード設定オプションの詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。最終ページの [Review Instance Launch] で、設定内容を確認します。[Launch] を選択してキーペアを選択し、インスタンスを起動します。
6. [Instances] ページで、新しいインスタンスを選択し、そのパブリック IP アドレスを、詳細ペインの [IPv4 Public IP] フィールドで確認します。

パブリック IP アドレス機能は起動時にのみ使用できます。ただし、起動時にパブリック IP アドレスをインスタンスに割り当てるかどうかにかかわらず、起動後に Elastic IP アドレスをインスタンスに関連付けることができます。詳細については、「[Elastic IP アドレス \(p. 705\)](#)」を参照してください。サブネットのパブリック IPv4 アドレス指定動作を変更することもできます。詳細については、「[サブネットの IPv4 アドレス指定属性の変更](#)」を参照してください。

コマンドラインを使用してパブリック IP アドレス指定機能を有効または無効にするには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- `run-instances` コマンド (AWS CLI) で `--associate-public-ip-address` または `--no-associate-public-ip-address` オプションを使用します。
- `New-EC2Instance` コマンド (AWS Tools for Windows PowerShell) で `-AssociatePublicIp` パラメータを使用します。

## インスタンスへの IPv6 アドレスの割り当て

VPC とサブネットに IPv6 CIDR ブロックが関連付けられている場合は、起動時または起動後に IPv6 アドレスをインスタンスに割り当てるすることができます。IPv6 アドレスは、サブネットの IPv6 アドレス範囲から割り当てられ、eth0 のデバイスインデックスを持つネットワークインターフェイスに割り当てられます。

IPv6 は、現行世代のすべてのインスタンスタイプと、旧世代の C3、R3、I2 のインスタンスタイプでサポートされています。

起動時に IPv6 アドレスをインスタンスに割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. IPv6 をサポートする AMI およびインスタンスタイプを選択し、[Next: Configure Instance Details] を選択します。
3. [Configure Instance Details] ページで、[Network] から VPC を選択し、[Subnet] からサブネットを選択します。[Auto-assign IPv6 IP] で、[Enable] を選択します。
4. ウィザードの残りの手順に従ってインスタンスを起動します。

別の方として、起動後に IPv6 アドレスをインスタンスに割り当てるすることもできます。

起動後に IPv6 アドレスをインスタンスに割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[アクション]、[ネットワーキング]、[IP アドレスの管理] の順に選択します。
4. [IPv6 Addresses] で、[Assign new IP] を選択します。サブネットの範囲から IPv6 アドレスを指定するか、[Auto-assign] を使って IPv6 アドレスを自動的に選択することができます。
5. [Save] を選択します。

### Note

Amazon Linux 2016.09.0 以降または Windows Server 2008 R2 以降を使用してインスタンスを起動した場合、インスタンスは IPv6 用に設定されるため、インスタンスで IPv6 アドレスを認識するための追加のステップは不要です。古い AMI からインスタンスを起動した場合は、必要に応じてインスタンスを手動で設定します。詳細については、「[インスタンスでの IPv6 の設定](#)」(Amazon VPC ユーザーガイド) を参照してください。

コマンドラインを使用して IPv6 アドレスを割り当てるには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `run-instances` コマンド (AWS CLI) で `--ipv6-addresses` オプションを使用する
- `New-EC2Instance` コマンド (AWS Tools for Windows PowerShell) で `-NetworkInterface` の `Ipv6Addresses` プロパティを使用する
- `assign-ipv6-addresses` (AWS CLI)
- `Register-EC2Ipv6AddressList` (AWS Tools for Windows PowerShell)

## インスタンスからの IPv6 アドレスの割り当て解除

Amazon EC2 コンソールを使用してインスタンスから IPv6 アドレスを割り当て解除できます。

インスタンスから IPv6 アドレスを割り当て解除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[アクション]、[ネットワーキング]、[IP アドレスの管理] の順に選択します。
4. [IPv6 Addresses] で、割り当て解除する IPv6 アドレスに対して [割り当て解除] を選択します。
5. [Yes, Update] を選択します。

コマンドラインを使用して IPv6 アドレスを割り当て解除するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `unassign-ipv6-addresses` (AWS CLI)
- `Unregister-EC2Ipv6AddressList` (AWS Tools for Windows PowerShell)

## 複数の IP アドレス

インスタンスに複数のプライベート IPv4 および IPv6 アドレスを指定できます。インスタンスに指定できるネットワークインターフェイスとプライベート IPv4 および IPv6 アドレスの数は、インスタンスタイプによって異なります。詳細については、「[各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数 \(p. 714\)](#)」を参照してください。

次のような場合、複数の IP アドレスを VPC 内のインスタンスに割り当てるに便利です。

- 1 つのサーバーで複数の SSL 証明書を使用し、各インターフェイスに各 IP アドレスに割り当てることで、1 つのサーバーで複数のウェブサイトをホストする。
- 各ネットワークインターフェイス用に複数の IP アドレスを持つネットワークアプライアンス (ファイアウォールやロードバランサーなど) を運用する。
- インスタンスでエラーが発生した場合に、セカンダリ IP アドレスをスタンバイインスタンスに再割り当てるにによって、内部トラフィックをスタンバイインスタンスにリダイレクトする。

### コンテンツ

- [複数の IP アドレスを使用する方法 \(p. 693\)](#)
- [複数の IPv4 アドレスの使用 \(p. 694\)](#)
- [複数の IPv6 アドレスの使用 \(p. 698\)](#)

## 複数の IP アドレスを使用する方法

次の一覧は、ネットワークインターフェイスで複数の IP アドレスを使用する方法の説明です。

- セカンダリプライベート IPv4 アドレスをネットワークインターフェイスに割り当てるすることができます。ネットワークインターフェイスをこのインスタンスにアタッチする必要はありません。
- IPv6 CIDR ブロックが関連付けられているサブネット内のネットワークインターフェイスに複数の IPv6 アドレスを割り当てるすることができます。
- ネットワークインターフェイスのサブネットの IPv4 CIDR ブロック範囲からセカンダリ IPv4 アドレスを選択する必要があります。
- ネットワークインターフェイスのサブネットの IPv6 CIDR ブロック範囲から IPv6 アドレスを選択する必要があります。

- セキュリティグループを関連付けるのは、個々の IP アドレスではなく、ネットワークインターフェイスです。そのため、ネットワークインターフェイスで指定した各 IP アドレスは、そのネットワークインターフェイスのセキュリティグループの対象です。
- 複数の IP アドレスは、実行中または停止したインスタンスにアタッチされたネットワークインターフェイスに割り当てたり、割り当て解除したりできます。
- ネットワークインターフェイスに割り当てられているセカンダリプライベート IPv4 アドレスは、明示的に許可された場合、別のネットワークインターフェイスに割り当て直すことができます。
- IPv6 アドレスは、最初に既存のネットワークインターフェイスから割り当て解除しない限り、別のネットワークインターフェイスに再割り当てすることはできません。
- コマンドラインツールまたは API を使用して複数の IP アドレスをネットワークインターフェイスに割り当てるときに、いずれかの IP アドレスを割り当てることができない場合、オペレーション全体が失敗します。
- プライマリプライベート IPv4 アドレス、セカンダリプライベート IPv4 アドレス、Elastic IP アドレス、および IPv6 アドレスは、セカンダリネットワークインターフェイスをインスタンスからデタッチしたり、インスタンスにアタッチしても、セカンダリネットワークインターフェイスへの割り当ては維持します。
- プライマリネットワークインターフェイスをインスタンスからデタッチすることはできませんが、プライマリネットワークインターフェイスのセカンダリプライベート IPv4 アドレスを別のネットワークインターフェイスに再割り当てすることはできます。

次の一覧は、Elastic IP アドレスで複数の IP アドレスを使用する方法の説明です (IPv4 のみ)。

- 各プライベート IPv4 アドレスを関連付けることができる Elastic IP アドレスは 1 つであり、逆に各 Elastic IP アドレスを関連付けることができるプライベート IPv4 アドレスは 1 つです。
- セカンダリプライベート IPv4 アドレスを別のインターフェイスに再割り当てる場合、セカンダリプライベート IPv4 アドレスと Elastic IP アドレスの関連付けは維持されます。
- セカンダリプライベート IPv4 アドレスとインターフェイスの割り当てを解除すると、関連付けられた Elastic IP アドレスとセカンダリプライベート IPv4 アドレスとの関連付けは自動的に解除されます。

## 複数の IPv4 アドレスの使用

セカンダリプライベート IPv4 アドレスは、インスタンスに割り当てたり、Elastic IPv4 アドレスと関連付けたり、割り当て解除したりできます。

### コンテンツ

- [セカンダリプライベート IPv4 アドレスを割り当てる \(p. 694\)](#)
- [セカンダリプライベート IPv4 アドレスを認識するようにインスタンスのオペレーティングシステムを設定する \(p. 696\)](#)
- [Elastic IP アドレスをセカンダリプライベート IPv4 アドレスに割り当てる \(p. 697\)](#)
- [セカンダリプライベート IPv4 アドレスを確認する \(p. 697\)](#)
- [セカンダリプライベート IPv4 アドレスを割り当てる解除する \(p. 697\)](#)

## セカンダリプライベート IPv4 アドレスを割り当てる

セカンダリプライベート IPv4 アドレスは、インスタンスの起動時または起動後に、インスタンスのネットワークインターフェイスに割り当てるすることができます。このセクションでは、次の手順を紹介します。

- [インスタンスの起動時にセカンダリプライベート IPv4 アドレスを割り当てるには \(p. 695\)](#)
- [コマンドラインを使用して起動時にセカンダリ IPv4 アドレスを割り当てるには \(p. 695\)](#)
- [セカンダリプライベート IPv4 アドレスをネットワークインターフェイスに割り当てるには \(p. 696\)](#)

- コマンドラインを使用して既存のインスタンスにセカンダリプライベート IPv4 を割り当てるには (p. 696)

インスタンスの起動時にセカンダリプライベート IPv4 アドレスを割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. AMI を選択し、次にインスタンスタイプを選択して、[Next: Configure Instance Details] を選択します。
4. [Configure Instance Details] ページで、[Network] から VPC を選択し、[Subnet] からサブネットを選択します。
5. [Network Interfaces] セクションで、次の手順を実行し、[Next: Add Storage] を選択します。
  - 別のネットワークインターフェイスを追加するには、[Add Device] を選択します。コンソールでは、インスタンス起動時のネットワークインターフェイスを最大 2 つ指定できます。インスタンスを起動したら、ナビゲーションペインで [Network Interfaces] を選択し、ネットワークインターフェイスを追加します。アタッチできるネットワークインターフェイスの合計数はインスタンスタイプによって異なります。詳細については、「[各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数 \(p. 714\)](#)」を参照してください。

Important

2 つ目のネットワークインターフェイスを追加すると、システムは、パブリック IPv4 アドレスを自動的に割り当てることができなくなります。プライマリネットワークインターフェイス (eth0) に Elastic IP アドレスを割り当てない限り、IPv4 経由でインスタンスに接続することはできません。起動ウィザードを完了した後は、Elastic IP アドレスを割り当ることができます。詳細については、「[Elastic IP アドレスの操作 \(p. 706\)](#)」を参照してください。

- ネットワークインターフェイスごとに、[Secondary IP addresses] の下にある [Add IP] を選択し、サブネットの範囲に含まれるプライベート IP アドレスを入力するか、デフォルトの Auto-assign のままにしてアドレスを自動的に選択します。
6. 次の [Add Storage] ページで、AMI によって指定されるボリューム (ルートデバイスピリュームなど) 以外にインスタンスにアタッチするボリュームを指定し、[Next: Add Tags] を選択します。
7. [Add Tags] ページで、ユーザーフレンドリーな名前などを使ってインスタンスのタグを指定し、[Next: Configure Security Group] を選択します。
8. [Configure Security Group] ページで、既存のセキュリティグループを選択するか、新しいグループを作成します。[Review and Launch] を選択します。
9. [Review Instance Launch] ページで、設定内容を確認します。[Launch] を選択して、キーペアを選択し、インスタンスを起動します。Amazon EC2 を初めて使用する場合、これまでにキーペアを作成したことがなければ、ウィザードによってキーペアを作成するよう求めるメッセージが表示されます。

Important

セカンダリプライベート IP アドレスをネットワークインターフェイスに追加した後、インスタンスに接続して、インスタンス自体でセカンダリプライベート IP アドレスを設定する必要があります。詳細については、「[セカンダリプライベート IPv4 アドレスを認識するようにインスタンスのオペレーティングシステムを設定する \(p. 696\)](#)」を参照してください。

コマンドラインを使用して起動時にセカンダリ IPv4 アドレスを割り当てるには

- 次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。
- `run-instances` コマンド (AWS CLI) の `--secondary-private-ip-addresses` オプション

- `-NetworkInterface` を定義し、[New-EC2Instance](#) コマンド (AWS Tools for Windows PowerShell) に `PrivateIpAddresses` パラメータを指定します。

セカンダリプライベート IPv4 アドレスをネットワークインターフェイスに割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択し、インスタンスにアタッチされているネットワークインターフェイスを選択します。
3. [Actions]、[Manage IP Addresses] の順に選択します。
4. [IPv4 Addresses] で、[Assign new IP] を選択します。
5. インスタンスのサブネットの範囲に含まれる特定の IPv4 アドレスを入力するか、フィールドを空のままにして IP アドレスを自動的に選択します。
6. (省略可能) セカンダリプライベート IP アドレスがすでに別のネットワークインターフェイスに割り当てられている場合、[Allow reassignment] を選択して、セカンダリプライベート IP アドレスを割り当て直すことができます。
7. [Yes, Update] を選択します。

または、インスタンスにセカンダリプライベート IPv4 アドレスを割り当てるすることができます。ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。次に、[Actions] を選択し、[Networking]、[Manage IP Addresses] の順に選択します。上記のステップに従って、同じ情報を設定できます。IP アドレスは、インスタンスのプライマリネットワークインターフェイス (eth0) に割り当てられます。

コマンドラインを使用して既存のインスタンスにセカンダリプライベート IPv4 を割り当てるには

- 次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。
  - `assign-private-ip-addresses` (AWS CLI)
  - `Register-EC2PrivateIpAddress` (AWS Tools for Windows PowerShell)

## セカンダリプライベート IPv4 アドレスを認識するようにインスタンスのオペレーティングシステムを設定する

セカンダリプライベート IPv4 アドレスをインスタンスに割り当てたら、セカンダリプライベート IP アドレスを認識するようにインスタンスのオペレーティングシステムを設定する必要があります。

- Amazon Linux を使用している場合、`ec2-net-utils` パッケージがこの処理を自動実行します。このパッケージは、インスタンスの実行中にアタッチされる追加のネットワークインターフェイスを設定し、DHCP リースの更新中にセカンダリ IPv4 アドレスを更新して、関連するルーティングルールを更新します。コマンド `sudo service network restart` を使用して即座にインターフェースの一覧を更新し、`ip addr li` を使用することで最新の一覧を表示することができます。ネットワーク構成を手動で構成する必要がある場合、`ec2-net-utils` パッケージを削除できます。詳細については、「[ec2-net-utils を使用したネットワークインターフェイスの設定 \(p. 726\)](#)」を参照してください。
- 別の Linux ディストリビューションを使用している場合、Linux ディストリビューションのドキュメントを参照してください。追加のネットワークインターフェイスとセカンダリ IPv4 アドレスの設定に関する情報が記載されています。同じサブネットのインスタンスに複数のインターフェイスがある場合、非対称のルーティングに対処する方法については、ルーティングルールの使用に関する情報を検索してください。

Windows インスタンスの設定については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows インスタンスのセカンダリプライベート IP アドレスの設定](#)」を参照してください。

## Elastic IP アドレスをセカンダリプライベート IPv4 アドレスに割り当てる

Elastic IP アドレスをセカンダリプライベート IPv4 アドレスに関連付けるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. [Actions] を選択し、次に [Associate address] を選択します。
4. [Network interface] でネットワークインターフェイスを選択し、次に [Private IP] リストからセカンダリ IP アドレスを選択します。
5. [Associate] を選択します。

コマンドラインを使用して Elastic IP アドレスにセカンダリプライベート IPv4 アドレスを関連付けるには

- 次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。
  - `associate-address` (AWS CLI)
  - `Register-EC2Address` (AWS Tools for Windows PowerShell)

## セカンダリプライベート IPv4 アドレスを確認する

ネットワークインターフェイスに割り当てられたプライベート IPv4 アドレスを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. 確認するプライベート IP アドレスがあるネットワークインターフェイスを選択します。
4. 詳細ペインの [Details] タブで、[Primary private IPv4 IP] フィールドと [Secondary private IPv4 IPs] フィールドに表示されている、ネットワークインターフェイスに割り当てられているプライマリプライベート IPv4 アドレスとセカンダリプライベート IPv4 アドレスを確認します。

インスタンスに割り当てられたプライベート IPv4 アドレスを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. 確認するプライベート IPv4 アドレスがあるインスタンスを選択します。
4. 詳細ペインの [Description] タブで、ネットワークインターフェイス経由でインスタンスに割り当てられているプライマリプライベート IPv4 アドレスとセカンダリプライベート IPv4 アドレスの [Private IPs] フィールドと [Secondary Private IPs] フィールドを確認します。

## セカンダリプライベート IPv4 アドレスを割り当て解除する

セカンダリプライベート IPv4 アドレスが不要になった場合、インスタンスやネットワークインターフェイスから割り当て解除できます。セカンダリプライベート IPv4 アドレスをネットワークインターフェイスから割り当て解除した場合、Elastic IP アドレス (存在する場合) の関連付けも解除されます。

インスタンスからセカンダリプライベート IPv4 アドレスを割り当て解除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[Actions]、[Networking]、[Manage IP Addresses] の順に選択します。

4. [IPv4 Addresses] で、割り当て解除する IPv4 アドレスに対して [Unassign] を選択します。
5. [Yes, Update] を選択します。

ネットワークインターフェイスからセカンダリプライベート IPv4 アドレスを割り当て解除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェースを選択し、[Actions]、[Manage IP Addresses] の順に選択します。
4. [IPv4 Addresses] で、割り当て解除する IPv4 アドレスに対して [Unassign] を選択します。
5. [Yes, Update] を選択します。

コマンドラインを使用してセカンダリプライベート IPv4 アドレスを割り当て解除するには

- 次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。
  - `unassign-private-ip-addresses` (AWS CLI)
  - `Unregister-EC2PrivateIpAddress` (AWS Tools for Windows PowerShell)

## 複数の IPv6 アドレスの使用

インスタンスに複数の IPv6 アドレスを割り当て、インスタンスに割り当てられている IPv6 アドレスを表示したり、インスタンスから IPv6 アドレスを割り当て解除したりできます。

### コンテンツ

- [複数の IPv6 アドレスを割り当てる \(p. 698\)](#)
- [IPv6 アドレスを確認する \(p. 700\)](#)
- [IPv6 アドレスの割り当て解除 \(p. 700\)](#)

## 複数の IPv6 アドレスを割り当てる

起動時または起動後のインスタンスに 1 つ以上の IPv6 アドレスを割り当てるすることができます。IPv6 アドレスをインスタンスに割り当てるには、インスタンスを起動した VPC およびサブネットに IPv6 CIDR ブロックが関連付けられている必要があります。詳細については、『Amazon VPC ユーザーガイド』の「[VPC とサブネット](#)」を参照してください。

起動時に複数の IPv6 アドレスを割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ダッシュボードから、[Launch Instance] を選択します。
3. AMI を選択し、次にインスタンスタイプを選択して、[Next: Configure Instance Details] を選択します。IPv6 をサポートするインスタンスタイプを必ず選択します。詳細については、[「インスタンスタイプ \(p. 183\)」](#) を参照してください。
4. [Configure Instance Details] ページで、[Network] リストから VPC を選択し、[Subnet] リストからサブネットを選択します。
5. [Network Interfaces] セクションで、次の手順を実行し、[Next: Add Storage] を選択します。
  - IPv6 アドレスをプライマリネットワークインターフェイス (eth0) に割り当てるには、[IPv6 IPs]、[Add IP] の順に選択します。セカンダリ IPv6 アドレスを追加するには、再度 [Add IP] 選択します。サブネットの範囲から IPv6 アドレスを入力するか、デフォルトの [Auto-assign] を使用してサブネットから自動的に IPv6 アドレスを選択することができます。

- [Add Device] を選択して別のネットワークインターフェイスを追加し、上記のステップを繰り返してそのネットワークインターフェイスに 1 つ以上の IPv6 アドレスを追加します。コンソールでは、インスタンス起動時のネットワークインターフェイスを最大 2 つ指定できます。インスタンスを起動したら、ナビゲーションペインで [Network Interfaces] を選択し、ネットワークインターフェイスを追加します。アタッチできるネットワークインターフェイスの合計数はインスタンスタイプによって異なります。詳細については、「[各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数 \(p. 714\)](#)」を参照してください。
6. ボリュームをアタッチしてインスタンスにタグを付けるには、ウィザードの以下のステップに従ってください。
  7. [Configure Security Group] ページで、既存のセキュリティグループを選択するか、新しいグループを作成します。IPv6 経由でインスタンスに到達可能にする場合は、IPv6 アドレスからのアクセスを許可するルールがセキュリティグループにあることを確認します。詳細については、「[セキュリティグループのルールのリファレンス \(p. 919\)](#)」を参照してください。[Review and Launch] を選択します。
  8. [Review Instance Launch] ページで、設定内容を確認します。[Launch] を選択して、キーペアを選択し、インスタンスを起動します。Amazon EC2 を初めて使用する場合、これまでにキーペアを作成したことがなければ、ウィザードによってキーペアを作成するよう求めるメッセージが表示されます。

Amazon EC2 コンソールの [インスタンス] 画面を使用して、既存のインスタンスに複数の IPv6 アドレスを割り当てることができます。IPv6 アドレスは、インスタンスのプライマリネットワークインターフェイス (eth0) に割り当てられます。IPv6 アドレスをインスタンスに割り当てるには、IPv6 アドレスが別のインスタンスやネットワークインターフェイスにまだ割り当てられていないことを確認します。

#### 複数の IPv6 アドレスを既存のインスタンスに割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[アクション]、[ネットワーキング]、[IP アドレスの管理] の順に選択します。
4. [IPv6 Addresses] で、追加する IPv6 アドレスごとに [Assign new IP] を選択します。サブネットの範囲から IPv6 アドレスを指定するか、[Auto-assign] を使って IPv6 アドレスを自動的に選択することができます。
5. [Yes, Update] を選択します。

また、既存のネットワークインターフェイスに複数の IPv6 アドレスを割り当てるすることができます。そのネットワークインターフェイスは、IPv6 CIDR ブロックが関連付けられているサブネットで作成されている必要があります。特定の IPv6 アドレスをネットワークインターフェイスに割り当てるには、その IPv6 アドレスが別のネットワークインターフェイスにまだ割り当てられていないことを確認します。

#### 複数の IPv6 アドレスをネットワークインターフェイスに割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェースを選択し、[Actions]、[Manage IP Addresses] の順に選択します。
4. [IPv6 Addresses] で、追加する IPv6 アドレスごとに [Assign new IP] を選択します。サブネットの範囲から IPv6 アドレスを指定するか、[Auto-assign] を使って IPv6 アドレスを自動的に選択することができます。
5. [Yes, Update] を選択します。

#### CLI の概要

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- 起動時に IPv6 アドレスを割り当てる:
  - [run-instances](#) コマンド (AWS CLI) で、`--ipv6-addresses` または `--ipv6-address-count` オプションを使用する
  - `-NetworkInterface` を定義し、[New-EC2Instance](#) コマンド (AWS Tools for Windows PowerShell) で、`Ipv6Addresses` パラメータまたは `I_pv6AddressCount` パラメータを指定する
- IPv6 アドレスをネットワークインターフェイスに割り当てる:
  - [assign-ipv6-addresses](#) (AWS CLI)
  - [Register-EC2Ipv6AddressList](#) (AWS Tools for Windows PowerShell)

## IPv6 アドレスを確認する

インスタンスまたはネットワークインターフェイスの IPv6 アドレスを確認できます。

インスタンスに割り当てられた IPv6 アドレスを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択します。詳細ペインで、[IPv6 IPs] フィールドを確認します。

ネットワークインターフェイスに割り当てられた IPv6 アドレスを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択します。詳細ペインで、[IPv6 IPs] フィールドを確認します。

## CLI の概要

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- インスタンスの IPv6 アドレスを確認する場合
  - [describe-instances](#) (AWS CLI)
  - [Get-EC2Instance](#) (AWS Tools for Windows PowerShell)
- ネットワークインターフェイスの IPv6 アドレスを確認する場合
  - [describe-network-interfaces](#) (AWS CLI)
  - [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

## IPv6 アドレスの割り当て解除

インスタンスのプライマリネットワークインターフェイスから IPv6 アドレスを割り当て解除できます。また、ネットワークインターフェイスから IPv6 アドレスを割り当て解除できます。

インスタンスから IPv6 アドレスを割り当て解除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[アクション]、[ネットワーキング]、[IP アドレスの管理] の順に選択します。
4. [IPv6 Addresses] で、割り当て解除する IPv6 アドレスに対して [割り当て解除] を選択します。
5. [Yes, Update] を選択します。

## ネットワークインターフェイスから IPv6 アドレスを割り当て解除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェースを選択し、[Actions]、[Manage IP Addresses] の順に選択します。
4. [IPv6 Addresses] で、割り当て解除する IPv6 アドレスに対して [割り当て解除] を選択します。
5. [Save] を選択します。

## CLI の概要

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `unassign-ipv6-addresses` (AWS CLI)
- `Unregister-EC2Ipv6AddressList` (AWS Tools for Windows PowerShell)

# 自分の IP アドレスを使用する (BYOIP)

すべての公開 IPv4 アドレスの範囲の一部またはすべてをオンプレミスのネットワークから AWS アカウントに導入できます。引き続きアドレス範囲を所有できますが、AWS はこれをインターネット上でアドバタイズします。アドレス範囲を AWS に設定すると、そのアドレス範囲はアドレスプールとしてアカウントに表示されます。アドレスプールから Elastic IP アドレスを作成し、EC2 インスタンス、NAT ゲートウェイ、Network Load Balancer などの AWS リソースで使用することができます。

### Important

BYOIP は、一部のリージョンでは使用できません。サポートされているリージョンのリストについては、「[FAQ for Bring Your Own IP](#)」を参照してください。

## 要件

- アドレス範囲は、American Registry for Internet Numbers (ARIN)、Réseaux IP Européens Network Coordination Centre (RIPE) または Asia-Pacific Network Information Centre (APNIC) といった地域インターネットレジストリ (RIR、Regional internet registry) に登録する必要があります。アドレス範囲は、事業体または機関エンティティについて登録を受ける必要があり、個人については登録を受けられない場合があります。
- 指定できる最も具体的なアドレス範囲は /24 です。
- 各アドレス範囲は、一度に 1 つのリージョンで使用できます。
- AWS アカウントには、リージョンあたり 5 つのアドレス範囲を登録できます。
- IP アドレス範囲内のアドレスには、消去履歴が含まれている必要があります。弊社は、IP アドレス範囲に評価が低いまたは悪意のある挙動に関連付けられている IP アドレスが含まれている場合、当該範囲の評価を調査したり、当該範囲を拒否する権利を留保したりすることがあります。
- 使用する IP アドレスは自分が所有している必要があります。つまり、以下のものがサポートされます。
  - ARIN - "Direct Allocation" および "Direct Assignment" ネットワークタイプ
  - RIPE - "ALLOCATED PA"、"LEGACY"、および "ASSIGNED PI" 割り当てステータス
  - APNIC - "ALLOCATED PORTABLE" および "ASSIGNED PORTABLE" 割当てステータス

## AWS アカウントにアドレス範囲を持ち込むための準備

お客様ご自身だけが、アドレス範囲をご自身の AWS アカウントに登録できるようにするために、お客様は、Amazon による当該アドレス範囲の公開を認める必要があります。また、署名付き認可メッセージにより、ご自身が当該アドレス範囲の所有者であるという証拠も提出する必要があります。

Route Origin Authorization (ROA) は、利用している RIR を介して作成できる、経路広告に関する電子署名付き証明書です。これには、アドレス範囲、そのアドレス範囲を公開することを許可された自律システム番号 (ASN)、および有効期限が含まれています。ROA は Amazon が特定の AS 番号のアドレス範囲を公開することを承認します。ただし、その AWS アカウントに対して、アドレス範囲を AWS に持ち込むことを承認するには、アドレス範囲について Registry Data Access Protocol (RDAP) の注釈で自己署名付きの X509 証明書を発行する必要があります。証明書にはパブリックキーが含まれており、AWS はこれを使用してあなたが提供する認証コンテキスト署名を確認します。プライベートキーを安全に管理し、これを使用して認証コンテキストメッセージを署名する必要があります。

これらのタスクのコマンドは、Linux でサポートされています。Windows では、[Windows Subsystem for Linux](#) を使用して、Linux コマンドを実行できます。

### タスク

- [ROA オブジェクトを作成する \(p. 702\)](#)
- [自己署名の X509 証明書を作成する \(p. 702\)](#)
- [署名付き認可メッセージを作成する \(p. 703\)](#)

## ROA オブジェクトを作成する

Amazon ASN 16509 および 14618 を承認してアドレス範囲を公開するための ROA オブジェクトと、現在そのアドレス範囲を公開することが承認されている ASN を作成します。持ち込む最小プレフィックスのサイズに最大長を設定する必要があります (たとえば、/24)。ROA が Amazon で使用できるようになるまで最大 24 時間かかる場合があります。詳細については、以下を参照してください。

- ARIN — [ROA のリクエスト数](#)
- RIPE — [ROA の管理](#)
- APNIC — 経路管理

## 自己署名の X509 証明書を作成する

次の手順を使用して、自己署名 X509 証明書を作成し、RIR の RDAP レコードに追加します。openssl コマンドには、OpenSSL バージョン 1.0.2 以降が必要です。

自己署名 X509 証明書を作成し、RDAP レコードに追加するには

1. 以下に示すように RSA 2048 ビットのキーペアを生成します。

```
openssl genrsa -out private.key 2048
```

2. 次のコマンドを使用してキーペアからパブリック X509 証明書を作成します。この例では、証明書は 365 日で期限切れになり、それ以降は信頼されません。したがって、有効期限は適切に設定してください。情報の入力を求められたら、デフォルト値をそのまま使用します。

```
openssl req -new -x509 -key private.key -days 365 | tr -d "\n" > publickey.cer
```

3. X509 証明書を使用して RIR の RDAP レコードを更新します。必ず証明書から-----BEGIN CERTIFICATE-----および-----END CERTIFICATE-----をコピーしてください。前のステップで tr-d "\n" コマンドを使用して改行文字を削除していない場合は、ここで削除してあることを確認します。証明書を表示するには、以下のコマンドを実行します。

```
cat publickey.cer
```

ARIN の場合は、アドレス範囲について [Public Comments] セクションに証明書を追加します。

RIPE の場合は、アドレス範囲について新しい "descr" フィールドとして証明書を追加します。

APNIC の場合は、パブリックキーを電子メールで[helpdesk@apnic.net](mailto:helpdesk@apnic.net)に送信し、手動で "remarks" フィールドに追加します。APNIC の IP アドレスに関する正規連絡先に電子メールを送信します。

## 署名付き認可メッセージを作成する

署名付き承認メッセージの形式は以下のとおりですが、日付はメッセージの有効期限になります。

```
1|aws|account|cidr|YYYYMMDD|SHA256|RSAPSS
```

まず、プレーンテキストの認証メッセージを作成し、次のように text\_message という名前の変数に保存します。サンプルのアカウント番号、アドレス範囲、および有効期限を独自の値に置き換えます。

```
text_message="1|aws|123456789012|198.51.100.0/24|20191201|SHA256|RSAPSS"
```

次に、作成したキーペアを使用して text\_message 内の承認メッセージに署名し、次のように、 signed\_message と命名された変数内に格納します。

```
signed_message=$(echo $text_message | tr -d "\n" | openssl dgst -sha256 -sigopt rsa_padding_mode:pss -sigopt rsa_pss_saltlen:-1 -sign private.key -keyform PEM | openssl base64 | tr -- '+=' '/' '-_-' | tr -d "\n")
```

## AWS で使用するためのアドレス範囲のプロビジョニング

AWS で使用するアドレス範囲をプロビジョニングする場合は、当該範囲の所有者であることを証明し、Amazon による当該範囲の公開を承認します。また、署名済みの認可メッセージを使用して、アドレス範囲を所有していることを確認します。このメッセージには、X509 証明書で RDAP レコードを更新するときに使用した自己署名 X509 キーペアで署名されます。

アドレス範囲をプロビジョニングするには、次の `provision-byoip-cidr` コマンドを使用します。サンプルのアドレス範囲を独自のアドレス範囲に置き換えます。--cidr-authorization-context オプションでは、以前に作成した変数を使用します。ROA メッセージではありません。

```
aws ec2 provision-byoip-cidr --cidr address-range --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

アドレス範囲のプロビジョニングは非同期オペレーションであるため、呼び出しはすぐに戻りますが、アドレスの範囲は、そのステータスが pending-provision から provisioned に変わるまで使用できません。プロビジョニングプロセスの完了までには最大で 3 週間かかることがあります。プロビジョニングしたアドレス範囲のステータスを監視するには、以下の `describe-byoip-cidrs` コマンドを使用します。

```
aws ec2 describe-byoip-cidrs --max-results 5
```

アドレスプールから Elastic IP アドレスを作成するには、[allocate-address](#) コマンドを使用します。--public-ipv4-pool オプションを使用して、[describe-byoip-cidrs](#) が返すアドレスプールの ID を指定したり、--address オプションを使用して、プロビジョニングしたアドレス範囲からのアドレスを指定したりすることができます。

## AWS からアドレス範囲の公開

アドレス範囲をプロビジョニングすると、公開することができるようになります。プロビジョニングした正確なアドレス範囲をアドバタイズする必要があります。プロビジョニングしたアドレス範囲の一部のみアドバタイズすることはできません。

アドレス範囲は、AWS から公開する前に、他の場所からの公開を停止することをお勧めします。他の場所から IP アドレス範囲を公開し続ける場合、当社では、その IP アドレス範囲を信頼してサポートしたり、問題をトラブルシューティングすることができなくなります。具体的には、そのアドレス範囲へのトラブルが当社のネットワークに入るのを保証できません。

ダウンタイムを最小限に抑えるには、アドレス範囲が公開される前にご使用のアドレスプールからアドレスを使用するように AWS リソースを設定してから、同時に現在の場所からの公開を停止して、AWS からの公開を開始します。アドレスプールからの Elastic IP アドレスの割り当ての詳細については、「[Elastic IP アドレスの割り当て \(p. 706\)](#)」を参照してください。

アドレス範囲を公開するには、以下の[advertise-byoip-cidr](#) コマンドを使用します。

```
aws ec2 advertise-byoip-cidr --cidr address-range
```

### Important

アドレス範囲が毎回異なる場合でも、[advertise-byoip-cidr](#) コマンドは 10 秒ごとに最大 1 回しか実行できません。

アドレス範囲の公開を停止するには、以下の[withdraw-byoip-cidr](#) コマンドを使用します。

```
aws ec2 withdraw-byoip-cidr --cidr address-range
```

### Important

アドレス範囲が毎回異なる場合でも、[withdraw-byoip-cidr](#) コマンドは 10 秒ごとに最大 1 回しか実行できません。

## アドレス範囲のプロビジョニング解除

AWS によるアドレス範囲の使用を停止するには、アドレスプールから割り当てられている Elastic IP アドレスを解放してアドレス範囲の公開を停止した後に、当該範囲のプロビジョニングを解除します。

各 Elastic IP アドレスを解放するには、以下の[release-address](#) コマンドを使用します。

```
aws ec2 release-address --allocation-id eipalloc-12345678
```

アドレス範囲の公開を停止するには、以下の[withdraw-byoip-cidr](#) コマンドを使用します。

```
aws ec2 withdraw-byoip-cidr --cidr address-range
```

アドレス範囲のプロビジョニングを解除するには、以下の[deprovision-byoip-cidr](#)コマンドを使用します。

```
aws ec2 deprovision-byoip-cidr --cidr address-range
```

## Elastic IP アドレス

Elastic IP アドレスは、動的なクラウドコンピューティングのために設計された静的 IPv4 アドレスです。Elastic IP アドレスは、AWS アカウントに関連付けられます。Elastic IP アドレスを使用すると、アドレスをアカウント内の別のインスタンスに迅速に再マップして、インスタンスやソフトウェアのエラーを隠すことができます。

Elastic IP アドレスは、インターネットからアクセス可能なパブリック IPv4 アドレスです。インスタンスにパブリック IPv4 アドレスがない場合は、Elastic IP アドレスをインスタンスに関連付けて、インターネットとの通信を有効にできます。たとえば、これにより、ローカルコンピュータからインスタンスに接続できます。

現在、IPv6 に対する Elastic IP アドレスはサポートされていません。

### コンテンツ

- [Elastic IP アドレスの基本 \(p. 705\)](#)
- [Elastic IP アドレスの操作 \(p. 706\)](#)
- [電子メールアプリケーションでの逆引き DNS の使用 \(p. 712\)](#)
- [Elastic IP アドレスの制限 \(p. 712\)](#)

## Elastic IP アドレスの基本

Elastic IP アドレスの基本的な特徴を次に示します。

- Elastic IP アドレスを使用するには、まずアカウントに 1 つ割り当ててから、それをインスタンスまたはネットワークインターフェイスに関連付けます。
- Elastic IP アドレスをインスタンスと関連付けると、インスタンスのプライマリネットワークインターフェイスとも関連付けられます。Elastic IP アドレスをインスタンスにアタッチされたネットワークインターフェイスと関連付けると、インスタンスとも関連付けられます。
- Elastic IP アドレスをインスタンスまたはそのプライマリネットワークインターフェイスに関連付けると、インスタンスのパブリック IPv4 アドレス(既に割り当てられていた場合)が Amazon のパブリック IPv4 アドレスのプールに戻されます。パブリック IPv4 アドレスを再利用することはできず、パブリック IPv4 アドレスを Elastic IP アドレスに変換することはできません。詳細については、「[パブリック IPv4 アドレスと外部 DNS ホスト名 \(p. 686\)](#)」を参照してください。
- リソースから Elastic IP アドレスの関連付けを解除し、別のリソースと関連付けることができます。インスタンスへの開かれた接続は、Elastic IP アドレスの関連付けを解除し、別のインスタンスに再割り当てした後でも、しばらくの間は継続して機能します。再割り当てした Elastic IP アドレスを使用して、これらの接続をもう一度開くことをお勧めします。
- 関連付けが解除された Elastic IP アドレスは、明示的に解放するまでアカウントに割り当てられたままです。
- Elastic IP アドレスを効率的に使用するため、Elastic IP アドレスが実行中のインスタンスに関連付けられていない場合や、停止しているインスタンスやアタッチされていないネットワークインターフェイスに関連付けられている場合は、時間毎に小額の料金が請求されます。インスタンスを実行しているときは、インスタンスに関連付けられた 1 つの Elastic IP アドレスに対して料金は発生しませんが、インスタンスに関連付けられた追加の Elastic IP アドレスがある場合、その追加分に対しては料金が発生します。詳細については、[Amazon EC2 料金表](#)を参照してください。

- Elastic IP アドレスは、ネットワーク境界グループ別に専用になっています。
- パブリック IPv4 アドレスが前回割り当てられたインスタンスに Elastic IP アドレスを関連付けると、インスタンスのパブリック DNS ホスト名は、Elastic IP アドレスに一致するように変更されます。
- パブリック DNS ホスト名を解決すると、インスタンスのパブリック IPv4 アドレスまたは Elastic IP アドレス（インスタンスのネットワークの外部の場合）、およびインスタンスのプライベート IPv4 アドレス（インスタンスのネットワーク内からの場合）となります。
- AWS アカウントに持ち込んだ IP アドレスプールから Elastic IP アドレスを割り当てた場合、Elastic IP アドレス制限にカウントされません。
- Elastic IP アドレスを割り当てるとき、Elastic IP アドレスをネットワーク境界グループに関連付けることができます。このグループで CIDR ブロックをアドバタイズします。ネットワーク境界グループを設定すると、CIDR ブロックがこのグループに制限されます。ネットワーク境界グループを指定しない場合は、リージョン（us-west-2 など）のすべてのアベイラビリティーゾーンを含む境界グループが自動的に設定されます。

## Elastic IP アドレスの操作

以下のセクションでは、Elastic IP アドレスの使用方法について説明します。

### タスク

- [Elastic IP アドレスの割り当て \(p. 706\)](#)
- [Elastic IP アドレスの説明 \(p. 707\)](#)
- [Elastic IP アドレスのタグ付け \(p. 708\)](#)
- [Elastic IP アドレスを実行中のインスタンスまたはネットワークインターフェイスに関連付ける \(p. 709\)](#)
- [Elastic IP アドレスの関連付け解除 \(p. 710\)](#)
- [Elastic IP アドレスを解放する \(p. 711\)](#)
- [Elastic IP アドレスの復旧 \(p. 711\)](#)

## Elastic IP アドレスの割り当て

Elastic IP アドレスは、Amazon のパブリック IPv4 アドレスのプールまたは AWS アカウントに持ち込んだカスタム IP アドレスプールから割り当てることができます。AWS アカウントへの独自の IP アドレス範囲の持ち込みの詳細については、「[自分の IP アドレスを使用する \(BYOIP\) \(p. 701\)](#)」を参照してください。

以下のいずれかの方法を使用して、Elastic IP アドレスを割り当てることができます。

### 新しいコンソール

Elastic IP アドレスを割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. [Allocate Elastic IP address] を選択します。
4. [スコープ] には、使用するスコープに応じて [VPC] または [EC2-Classic] を選択します。
5. (VPC スコープのみ) [Public IPv4 address pool (パブリック IPv4 アドレスプール)] で、以下のいずれかを選択します。
  - [Amazon's pool of IP addresses (Amazon の IP アドレスのプール)] — Amazon の IP アドレスのプールから IPv4 アドレスを割り当てる場合。

- [My pool of public IPv4 addresses (パブリック IPv4 アドレスのプール)] — AWS アカウントに持ち込んだ IP アドレスプールから IPv4 アドレスを割り当てる場合。IP アドレスプールがない場合、このオプションは無効になります。
6. [Allocate] を選択します。

#### 古いコンソール

Elastic IP アドレスを割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. [Allocate new address] を選択します。
4. [IPv4 address pool] で [Amazon pool] を選択します。
5. [Allocate] を選択し、確認画面を閉じます。

#### AWS CLI

Elastic IP アドレスを割り当てるには

`allocate-address` AWS CLI コマンドを使用します。

#### PowerShell

Elastic IP アドレスを割り当てるには

`New-EC2Address` AWS Tools for Windows PowerShell コマンドを使用します。

## Elastic IP アドレスの説明

以下のいずれかの方法を使用して、Elastic IP アドレスの情報を取得できます。

#### 新しいコンソール

Elastic IP アドレスの情報を取得するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. 表示する Elastic IP アドレスを選択してから、[Actions (アクション)]、[View details (詳細の表示)] の順に選択します。

#### 古いコンソール

Elastic IP アドレスの情報を取得するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. リソース属性リストからフィルタを選択して検索を開始します。単一の検索に複数のフィルタを使用できます。

#### AWS CLI

Elastic IP アドレスの情報を取得するには

[describe-addresses](#) AWS CLI コマンドを使用します。

PowerShell

Elastic IP アドレスの情報を取得するには

[Get-EC2Address](#) AWS Tools for Windows PowerShell コマンドを使用します。

## Elastic IP アドレスのタグ付け

Elastic IP アドレスにカスタムタグを割り当てて、目的、所有者、環境など、さまざまな方法で分類できます。これにより、割り当てたカスタムタグに基づいて特定の Elastic IP アドレスをすばやく見つけることができるようになります。

タグ付けできる Elastic IP アドレスは、VPC スコープ内のものだけです。

Note

Elastic IP アドレスタグを使用したコスト配分の追跡はサポートされていません。

以下のいずれかの方法を使用して、Elastic IP アドレスにタグ付けできます。

新しいコンソール

Elastic IP アドレスにタグを適用するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. タグ付けする Elastic IP アドレスを選択してから、[Actions (アクション)]、[View details (詳細の表示)] の順に選択します。
4. [Tags (タグ)] タブで、[Manage tags (タグの管理)] を選択します。
5. タグのキーと値のペアを指定します。
6. (オプション) [タグの追加] を選択して、タグを追加します。
7. [Save] を選択します。

古いコンソール

Elastic IP アドレスにタグを適用するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. タグを付ける Elastic IP アドレスを選択し、[Tags] を選択します。
4. [Add/Edit Tags] を選択します。
5. [Add/Edit Tags] ダイアログボックスで、[Create Tag] を選択してから、各タグのキーと値を指定します。
6. (オプション) [Create Tag] を選択して、Elastic IP アドレスにさらにタグを追加します。
7. [Save] を選択します。

AWS CLI

Elastic IP アドレスにタグを適用するには

[create-tags](#) AWS CLI コマンドを使用します。

```
aws ec2 create-tags --resources eipalloc-12345678 --tags Key=Owner,Value=TeamA
```

PowerShell

Elastic IP アドレスにタグを適用するには

[New-EC2Tag](#) AWS Tools for Windows PowerShell コマンドを使用します。

New-EC2Tag コマンドには、Elastic IP アドレスのタグに使用するキーと値のペアを指定する Tag パラメータが必要です。以下のコマンドでは、Tag パラメータを作成します。

```
PS C:\> $tag = New-Object Amazon.EC2.Model.Tag
PS C:\> $tag.Key = "Owner"
PS C:\> $tag.Value = "TeamA"
```

```
PS C:\> New-EC2Tag -Resource eipalloc-12345678 -Tag $tag
```

## Elastic IP アドレスを実行中のインスタンスまたはネットワークインターフェイスに関連付ける

Elastic IP アドレスをインスタンスに関連付けてインターネットとの通信を有効にする場合、インスタンスがパブリックサブネットに属していることも確認する必要があります。詳細については、『Amazon VPC ユーザーガイド』の「インターネットゲートウェイ」を参照してください。

以下のいずれかの方法を使用して、Elastic IP アドレスをインスタンスまたはネットワークインターフェイスに関連付けることができます。

新しいコンソール

Elastic IP アドレスをインスタンスに関連付けるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. 関連付ける Elastic IP アドレスを選択してから、[Actions (アクション)]、[Associate Elastic IP address (Elastic IP アドレスの関連付け)] の順に選択します。
4. [リソースタイプ] で、[Instance (インスタンス)] を選択します。
5. たとえば、Elastic IP アドレスを関連付けるインスタンスを選択します。テキストを入力して特定のインスタンスを検索することもできます。
6. (オプション) [プライベート IP アドレス] で、Elastic IP アドレスを関連付けるプライベート IP アドレスを指定します。
7. [Associate] を選択します。

Elastic IP アドレスとネットワークインターフェイスを関連付けるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. 関連付ける Elastic IP アドレスを選択してから、[Actions (アクション)]、[Associate Elastic IP address (Elastic IP アドレスの関連付け)] の順に選択します。
4. [リソースタイプ] で、[ネットワークインターフェイス] を選択します。

5. [ネットワークインターフェイス] で、Elastic IP アドレスを関連付けるネットワークインターフェイスを選択します。テキストを入力して、特定のネットワークインターフェイスを検索することができます。
6. (オプション) [プライベート IP アドレス] で、Elastic IP アドレスを関連付けるプライベート IP アドレスを指定します。
7. [Associate] を選択します。

#### 古いコンソール

Elastic IP アドレスをインスタンスに関連付けるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. Elastic IP アドレスを選択し、[Actions]、[Associate address] の順に選択します。
4. [Instance] からインスタンスを選択し、次に [Associate] を選択します。

#### AWS CLI

Elastic IP アドレスを関連付けるには

`associate-address` AWS CLI コマンドを使用します。

#### PowerShell

Elastic IP アドレスを関連付けるには

`Register-EC2Address` AWS Tools for Windows PowerShell コマンドを使用します。

## Elastic IP アドレスの関連付け解除

インスタンスまたはネットワークインターフェイスから Elastic IP アドレスの関連付けをいつでも解除できます。Elastic IP アドレスの関連付けを解除した後、そのアドレスを別のリソースに再度関連付けることができます。

以下のいずれかの方法を使用して、Elastic IP アドレスの関連付けを解除できます。

#### 新しいコンソール

Elastic IP アドレスの関連付けを解除して再度関連付けするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. 関連付けを解除する Elastic IP アドレスを選択してから、[Actions (アクション)]、[Elastic IP アドレスの関連付けの解除] の順に選択します。
4. [関連付け解除] を選択します。

#### 古いコンソール

Elastic IP アドレスの関連付けを解除して再度関連付けするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. Elastic IP アドレスを選択し、[Actions]、[Disassociate address] の順に選択します。

4. [Disassociate address] を選択します。

#### AWS CLI

Elastic IP アドレスの関連付けを解除するには

[disassociate-address](#) AWS CLI コマンドを使用します。

#### PowerShell

Elastic IP アドレスの関連付けを解除するには

[Unregister-EC2Address](#) AWS Tools for Windows PowerShell コマンドを使用します。

## Elastic IP アドレスを解放する

Elastic IP アドレスが不要になった場合は、以下のいずれかの方法を使用して解放することをお勧めします。解放するアドレスは、インスタンスに関連付けられていない必要があります。

#### 新しいコンソール

Elastic IP アドレスを解放するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. 解放する Elastic IP アドレスを選択してから、[Actions (アクション)]、[Release Elastic IP addresses (Elastic IP アドレスの解放)] の順に選択します。
4. [Release (解放)] を選択します。

#### 古いコンソール

Elastic IP アドレスを解放するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. Elastic IP アドレスを選択し、[Actions]、[Release addresses] の順に選択します。プロンプトが表示されたら、[Release] を選択します。

#### AWS CLI

Elastic IP アドレスを解放するには

[release-address](#) AWS CLI コマンドを使用します。

#### PowerShell

Elastic IP アドレスを解放するには

[Remove-EC2Address](#) AWS Tools for Windows PowerShell コマンドを使用します。

## Elastic IP アドレスの復旧

Elastic IP アドレスを解放した場合でも、復元できる可能性があります。以下のルールが適用されます。

- Elastic IP アドレスが別の AWS アカウントに割り当てられている場合や Elastic IP アドレスの制限を超える場合は、Elastic IP アドレスを復元できません。

- Elastic IP アドレスに関連付けられたタグを復旧することはできません。
- Elastic IP アドレスは、Amazon EC2 API コンソールまたはコマンドラインツールでのみ復元できます。

#### AWS CLI

Elastic IP アドレスを復元するには

以下のように、`--address` パラメータを指定した [allocate-address](#) AWS CLI コマンドを使用して、IP アドレスを指定します。

```
aws ec2 allocate-address --domain vpc --address 203.0.113.3
```

#### PowerShell

Elastic IP アドレスを復元するには

以下のように、`-Address` パラメータを指定した [New-EC2Address](#) AWS Tools for Windows PowerShell コマンドを使用して、IP アドレスを指定します。

```
PS C:\> New-EC2Address -Address 203.0.113.3 -Domain vpc -Region us-east-1
```

## 電子メールアプリケーションでの逆引き DNS の使用

インスタンスから第三者に電子メールを送信する場合、1つ以上の Elastic IP アドレスをプロビジョニングし、AWS に提供しておくことをお勧めします。AWS は、ISP およびインターネットアンチスパム組織と協力して、これらのアドレスから送信された E メールにスパムのフラグが付く可能性を減らしています。

また、E メールの送信に使用される Elastic IP アドレスに静的逆引き DNS レコードを割り当てるとき、アンチスパム組織により E メールにスパムのフラグが付くことを避けられることがあります。逆引き DNS レコードを作成できるようになる前に、その Elastic IP アドレスを参照する、対応するフォワード DNS レコード（レコードタイプ A）が存在している必要があることにご注意ください。

逆引き DNS レコードが Elastic IP アドレスに関連付けられている場合、その Elastic IP アドレスはアカウントにロックされ、レコードが削除されるまでアカウントから解放することはできません。

E メール送信制限を解除したり、Elastic IP アドレスを提供して DNS レコードを予約するには、「[E メール送信制限解除申請](#)」ページを参照してください。

## Elastic IP アドレスの制限

デフォルトでは、すべての AWS アカウントでリージョンあたり 5 つの Elastic IP アドレスまでに制限されています。これは、パブリック (IPv4) インターネットアドレスが数に限りのあるパブリックリソースであるためです。インスタンスに障害が発生した場合にアドレスを他のインスタンスに再マップする機能のために主に Elastic IP アドレスを使用し、他のすべてのノード間通信には DNS ホスト名を使用することをお勧めします。

使用中の Elastic IP アドレスの数を確認するには、<https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開き、ナビゲーションペインから [Elastic IP] を選択します。

Elastic IP アドレスの現在のアカウント制限を確認するには、次のいずれかを実行します。

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開き、ナビゲーションペインから [制限] を選択し、検索フィールドに「IP」と入力します。

- <https://console.aws.amazon.com/servicequotas/> で サービスクォータ コンソールを開き、検索フィールドに「Amazon EC2」と入力して [Amazon Elastic Compute Cloud (Amazon EC2)] を選択します。検索フィールドに IP と入力します。

ご利用のアーキテクチャで追加の Elastic IP アドレスが必要な場合、クォータの引き上げを サービス クォータ コンソールから直接リクエストできます。

## Elastic Network Interface

Elastic Network Interface (このドキュメントではネットワークインターフェイスと呼びます) は、仮想ネットワークカードを表す VPC 内の論理ネットワーキングコンポーネントです。

ネットワークインターフェイスには以下の属性を含めることができます。

- VPC の IPv4 アドレス範囲からのプライマリプライベート IPv4 アドレス
- VPC の IPv4 アドレス範囲からの 1 つ以上のセカンダリプライベート IPv4 アドレス
- プライベート IPv4 アドレスごとに 1 つの Elastic IP アドレス (IPv4)
- 1 つのパブリック IPv4 アドレス
- 1 つ以上の IPv6 アドレス
- 1 つ以上のセキュリティグループ
- MAC アドレス
- 送信元/送信先チェックフラグ
- 説明

アカウントで独自のネットワークインターフェイスを作成して設定し、VPC 内のインスタンスにアタッチできます。アカウントでは、AWS のサービスで作成および管理されるリクエスタマネージド型のネットワークインターフェイスも使用できます。これらを通じて他のリソースやサービスを利用できます。これらは、ユーザーが直接管理できないネットワークインターフェイスです。詳細については、「リクエスタマネージド型のネットワークインターフェイス (p. 736)」を参照してください。

すべてのネットワークインターフェイスには、eni- で始まるリソース識別子があります。

### Important

「Elastic Network Interface」という用語は、「ENI」と短縮される場合があります。これは Elastic Network Adapter (ENA) とは異なります。ENA は、一部のインスタンスタイプでネットワークパフォーマンスを最適化するためのカスタムインターフェイスです。詳細については、「Linux の拡張ネットワーキング (p. 737)」を参照してください。

### コンテンツ

- ネットワークインターフェイスの基本 (p. 713)
- 各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数 (p. 714)
- ネットワークインターフェイスのシナリオ (p. 724)
- ネットワークインターフェイスの設定に関するベストプラクティス (p. 726)
- ネットワークインターフェイスでの作業 (p. 727)
- リクエスタマネージド型のネットワークインターフェイス (p. 736)

## ネットワークインターフェイスの基本

ネットワークインターフェイスを作成したり、インスタンスにアタッチしたり、インスタンスからデタッチしたり、別のインスタンスにアタッチしたりできます。ネットワークインターフェイスをインスタンス

---

にアタッチしたり、インスタンスからデタッチして別のインスタンスに再アタッチしたりするときには、ネットワークインターフェイスの属性が保持されます。インスタンス間でネットワークインターフェイスを移動すると、ネットワークトラフィックは新しいインスタンスにリダイレクトされます。

ネットワークインターフェイスの属性を変更することもできます。たとえば、そのセキュリティグループを変更したり、IP アドレスを管理したりできます。

VPC の各インスタンスには、プライマリネットワークインターフェイス (eth0) と呼ばれるデフォルトのネットワークインターフェイスがあります。プライマリネットワークインターフェイスをインスタンスからデタッチすることはできません。追加のネットワークインターフェイスを作成し、アタッチできます。使用できるネットワークインターフェイスの最大数はインスタンスタイプによって異なります。詳細については、「[各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数 \(p. 714\)](#)」を参照してください。

#### ネットワークインターフェイスのパブリック IPv4 アドレス

VPC では、すべてのサブネットに、そのサブネットで作成されるネットワークインターフェイス (結果的にそのサブネットで起動されるインスタンス) にパブリック IPv4 アドレスを割り当てるかどうかを決定する、変更可能な属性があります。詳細については、『Amazon VPC ユーザーガイド』の「[サブネットのパブリック IP アドレス動作](#)」を参照してください。パブリック IPv4 アドレスは Amazon のパブリック IPv4 アドレスのプールから割り当てられます。インスタンスを起動すると、作成されたプライマリネットワークインターフェイス (eth0) に IP アドレスが割り当てられます。

ネットワークインターフェイスを作成すると、サブネットからパブリック IPv4 アドレス指定属性を継承します。後でサブネットのパブリック IPv4 アドレス指定属性を変更しても、ネットワークインターフェイスでは作成時に有効だった設定が保持されます。インスタンスを起動し、eth0 に既存のネットワークインターフェイスを指定する場合は、パブリック IPv4 アドレス指定属性はネットワークインターフェイスによって決定されます。

詳細については、「[パブリック IPv4 アドレスと外部 DNS ホスト名 \(p. 686\)](#)」を参照してください。

#### ネットワークインターフェイスの IPv6 アドレス

IPv6 CIDR ブロックを VPC とサブネットに関連付け、サブネットの範囲から 1 つ以上の IPv6 アドレスをネットワークインターフェイスに割り当てることができます。

すべてのサブネットには、そのサブネットで作成されるネットワークインターフェイス (結果的にそのサブネットで起動されるインスタンス) にサブネットの範囲から IPv6 アドレスを自動的に割り当てるかどうかを決定する、変更可能な属性があります。詳細については、『Amazon VPC ユーザーガイド』の「[サブネットのパブリック IP アドレス動作](#)」を参照してください。インスタンスを起動すると、作成されたプライマリネットワークインターフェイス (eth0) に IPv6 アドレスが割り当てられます。

詳細については、「[IPv6 アドレス \(p. 687\)](#)」を参照してください。

#### IP トラフィックのモニタリング

ネットワークインターフェイスで VPC フローログを有効にして、ネットワークインターフェイスとの間で行き来する IP トラフィックに関する情報をキャプチャできます。フローログを作成すると、そのデータを Amazon CloudWatch Logs で表示し、取得できます。詳細については、『Amazon VPC ユーザーガイド』の「[VPC フローログ](#)」を参照してください。

## 各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数

以下の表に示しているのは、各インスタンスタイプのネットワークインターフェイスの最大数と、ネットワークインターフェイスあたりのプライベート IPv4 アドレスと IPv6 アドレスの最大数です。ネットワークインターフェイスあたりの IPv6 アドレスとプライベート IPv4 アドレスの制限は異なります。すべてのインスタンスタイプで IPv6 アドレス指定がサポートされているわけではありません。ネットワークイ

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数

---

シナリオ、複数のプライベート IPv4 アドレス、IPv6 アドレスは、VPC で実行されているインスタンスにのみ使用できます。IPv6 アドレスは公開され、インターネットに到達できます。詳細については、「[複数の IP アドレス \(p. 693\)](#)」を参照してください。VPC での IPv6 の詳細については、『Amazon VPC ユーザーガイド』の「[VPC での IP アドレス指定](#)」を参照してください。

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
a1.medium	2	4	4
a1.large	3	10	10
a1.xlarge	4	15	15
a1.2xlarge	4	15	15
a1.4xlarge	8	30	30
c1.medium	2	6	IPv6 はサポートされていません
c1.xlarge	4	15	IPv6 はサポートされていません
c3.large	3	10	10
c3.xlarge	4	15	15
c3.2xlarge	4	15	15
c3.4xlarge	8	30	30
c3.8xlarge	8	30	30
c4.large	3	10	10
c4.xlarge	4	15	15
c4.2xlarge	4	15	15
c4.4xlarge	8	30	30
c4.8xlarge	8	30	30
c5.large	3	10	10
c5.xlarge	4	15	15
c5.2xlarge	4	15	15
c5.4xlarge	8	30	30
c5.9xlarge	8	30	30
c5.12xlarge	8	30	30
c5.18xlarge	15	50	50
c5.24xlarge	15	50	50
c5.metal	15	50	50

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークイ  
 ンターフェイスあたりの IP アドレス数

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
c5d.large	3	10	10
c5d.xlarge	4	15	15
c5d.2xlarge	4	15	15
c5d.4xlarge	8	30	30
c5d.9xlarge	8	30	30
c5d.12xlarge	8	30	30
c5d.18xlarge	15	50	50
c5d.24xlarge	15	50	50
c5d.metal	15	50	50
c5n.large	3	10	10
c5n.xlarge	4	15	15
c5n.2xlarge	4	15	15
c5n.4xlarge	8	30	30
c5n.9xlarge	8	30	30
c5n.18xlarge	15	50	50
c5n.metal	15	50	50
cc2.8xlarge	8	30	IPv6 はサポートされていません
cr1.8xlarge	8	30	IPv6 はサポートされていません
d2.xlarge	4	15	15
d2.2xlarge	4	15	15
d2.4xlarge	8	30	30
d2.8xlarge	8	30	30
f1.2xlarge	4	15	15
f1.4xlarge	8	30	30
f1.16xlarge	8	50	50
g2.2xlarge	4	15	IPv6 はサポートされていません
g2.8xlarge	8	30	IPv6 はサポートされていません
g3s.xlarge	4	15	15

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークイ  
 ンターフェイスあたりの IP アドレス数

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
g3.4xlarge	8	30	30
g3.8xlarge	8	30	30
g3.16xlarge	15	50	50
g4dn.xlarge	3	10	10
g4dn.2xlarge	3	10	10
g4dn.4xlarge	3	10	10
g4dn.8xlarge	4	15	15
g4dn.12xlarge	8	30	30
g4dn.16xlarge	4	15	15
h1.2xlarge	4	15	15
h1.4xlarge	8	30	30
h1.8xlarge	8	30	30
h1.16xlarge	15	50	50
hs1.8xlarge	8	30	IPv6 はサポートされていません
i2.xlarge	4	15	15
i2.2xlarge	4	15	15
i2.4xlarge	8	30	30
i2.8xlarge	8	30	30
i3.large	3	10	10
i3.xlarge	4	15	15
i3.2xlarge	4	15	15
i3.4xlarge	8	30	30
i3.8xlarge	8	30	30
i3.16xlarge	15	50	50
i3.metal	15	50	50
i3en.large	3	10	10
i3en.xlarge	4	15	15
i3en.2xlarge	4	15	15
i3en.3xlarge	4	15	15

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークイ  
 ンターフェイスあたりの IP アドレス数

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
i3en.6xlarge	8	30	30
i3en.12xlarge	8	30	30
i3en.24xlarge	15	50	50
i3en.metal	15	50	50
m1.small	2	4	IPv6 はサポートされていません
m1.medium	2	6	IPv6 はサポートされていません
m1.large	3	10	IPv6 はサポートされていません
m1.xlarge	4	15	IPv6 はサポートされていません
m2.xlarge	4	15	IPv6 はサポートされていません
m2.2xlarge	4	30	IPv6 はサポートされていません
m2.4xlarge	8	30	IPv6 はサポートされていません
m3.medium	2	6	IPv6 はサポートされていません
m3.large	3	10	IPv6 はサポートされていません
m3.xlarge	4	15	IPv6 はサポートされていません
m3.2xlarge	4	30	IPv6 はサポートされていません
m4.large	2	10	10
m4.xlarge	4	15	15
m4.2xlarge	4	15	15
m4.4xlarge	8	30	30
m4.10xlarge	8	30	30
m4.16xlarge	8	30	30
m5.large	3	10	10
m5.xlarge	4	15	15
m5.2xlarge	4	15	15

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークイ  
 ンターフェイスあたりの IP アドレス数

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
m5.4xlarge	8	30	30
m5.8xlarge	8	30	30
m5.12xlarge	8	30	30
m5.16xlarge	15	50	50
m5.24xlarge	15	50	50
m5.metal	15	50	50
m5a.large	3	10	10
m5a.xlarge	4	15	15
m5a.2xlarge	4	15	15
m5a.4xlarge	8	30	30
m5a.8xlarge	8	30	30
m5a.12xlarge	8	30	30
m5a.16xlarge	15	50	50
m5a.24xlarge	15	50	50
m5ad.large	3	10	10
m5ad.xlarge	4	15	15
m5ad.2xlarge	4	15	15
m5ad.4xlarge	8	30	30
m5ad.8xlarge	8	30	30
m5ad.12xlarge	8	30	30
m5ad.16xlarge	15	50	50
m5ad.24xlarge	15	50	50
m5d.large	3	10	10
m5d.xlarge	4	15	15
m5d.2xlarge	4	15	15
m5d.4xlarge	8	30	30
m5d.8xlarge	8	30	30
m5d.12xlarge	8	30	30
m5d.16xlarge	15	50	50
m5d.24xlarge	15	50	50

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークイ  
 ンターフェイスあたりの IP アドレス数

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
m5d.metal	15	50	50
m5dn.large	3	10	10
m5dn.xlarge	4	15	15
m5dn.2xlarge	4	15	15
m5dn.4xlarge	8	30	30
m5dn.8xlarge	8	30	30
m5dn.12xlarge	8	30	30
m5dn.16xlarge	15	50	50
m5dn.24xlarge	15	50	50
m5n.large	3	10	10
m5n.xlarge	4	15	15
m5n.2xlarge	4	15	15
m5n.4xlarge	8	30	30
m5n.8xlarge	8	30	30
m5n.12xlarge	8	30	30
m5n.16xlarge	15	50	50
m5n.24xlarge	15	50	50
p2.xlarge	4	15	15
p2.8xlarge	8	30	30
p2.16xlarge	8	30	30
p3.2xlarge	4	15	15
p3.8xlarge	8	30	30
p3.16xlarge	8	30	30
p3dn.24xlarge	15	50	50
r3.large	3	10	10
r3.xlarge	4	15	15
r3.2xlarge	4	15	15
r3.4xlarge	8	30	30
r3.8xlarge	8	30	30
r4.large	3	10	10

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークイ  
 ンターフェイスあたりの IP アドレス数

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
r4.xlarge	4	15	15
r4.2xlarge	4	15	15
r4.4xlarge	8	30	30
r4.8xlarge	8	30	30
r4.16xlarge	15	50	50
r5.large	3	10	10
r5.xlarge	4	15	15
r5.2xlarge	4	15	15
r5.4xlarge	8	30	30
r5.8xlarge	8	30	30
r5.12xlarge	8	30	30
r5.16xlarge	15	50	50
r5.24xlarge	15	50	50
r5.metal	15	50	50
r5a.large	3	10	10
r5a.xlarge	4	15	15
r5a.2xlarge	4	15	15
r5a.4xlarge	8	30	30
r5a.8xlarge	8	30	30
r5a.12xlarge	8	30	30
r5a.16xlarge	15	50	50
r5a.24xlarge	15	50	50
r5ad.large	3	10	10
r5ad.xlarge	4	15	15
r5ad.2xlarge	4	15	15
r5ad.4xlarge	8	30	30
r5ad.8xlarge	8	30	30
r5ad.12xlarge	8	30	30
r5ad.16xlarge	15	50	50
r5ad.24xlarge	15	50	50

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークイ  
 ンターフェイスあたりの IP アドレス数

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
r5d.large	3	10	10
r5d.xlarge	4	15	15
r5d.2xlarge	4	15	15
r5d.4xlarge	8	30	30
r5d.8xlarge	8	30	30
r5d.12xlarge	8	30	30
r5d.16xlarge	15	50	50
r5d.24xlarge	15	50	50
r5d.metal	15	50	50
r5dn.large	3	10	10
r5dn.xlarge	4	15	15
r5dn.2xlarge	4	15	15
r5dn.4xlarge	8	30	30
r5dn.8xlarge	8	30	30
r5dn.12xlarge	8	30	30
r5dn.16xlarge	15	50	50
r5dn.24xlarge	15	50	50
r5n.large	3	10	10
r5n.xlarge	4	15	15
r5n.2xlarge	4	15	15
r5n.4xlarge	8	30	30
r5n.8xlarge	8	30	30
r5n.12xlarge	8	30	30
r5n.16xlarge	15	50	50
r5n.24xlarge	15	50	50
t1.micro	2	2	IPv6 はサポートされていません
t2.nano	2	2	2
t2.micro	2	2	2
t2.small	3	4	4

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 各インスタンスタイプのネットワークイ  
 ンターフェイスあたりの IP アドレス数

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
t2.medium	3	6	6
t2.large	3	12	12
t2.xlarge	3	15	15
t2.2xlarge	3	15	15
t3.nano	2	2	2
t3.micro	2	2	2
t3.small	3	4	4
t3.medium	3	6	6
t3.large	3	12	12
t3.xlarge	4	15	15
t3.2xlarge	4	15	15
t3a.nano	2	2	2
t3a.micro	2	2	2
t3a.small	2	4	4
t3a.medium	3	6	6
t3a.large	3	12	12
t3a.xlarge	4	15	15
t3a.2xlarge	4	15	15
u-6tb1.metal	5	30	30
u-9tb1.metal	5	30	30
u-12tb1.metal	5	30	30
u-18tb1.metal	15	50	50
u-24tb1.metal	15	50	50
x1.16xlarge	8	30	30
x1.32xlarge	8	30	30
x1e.xlarge	3	10	10
x1e.2xlarge	4	15	15
x1e.4xlarge	4	15	15
x1e.8xlarge	4	15	15
x1e.16xlarge	8	30	30

インスタンスタイプ	ネットワークインターフェイスの最大数	インターフェイスあたりのプライベート IPv4 アドレス	インターフェイスあたりの IPv6 アドレス
x1e.32xlarge	8	30	30
z1d.large	3	10	10
z1d.xlarge	4	15	15
z1d.2xlarge	4	15	15
z1d.3xlarge	8	30	30
z1d.6xlarge	8	30	30
z1d.12xlarge	15	50	50
z1d.metal	15	50	50

## ネットワークインターフェイスのシナリオ

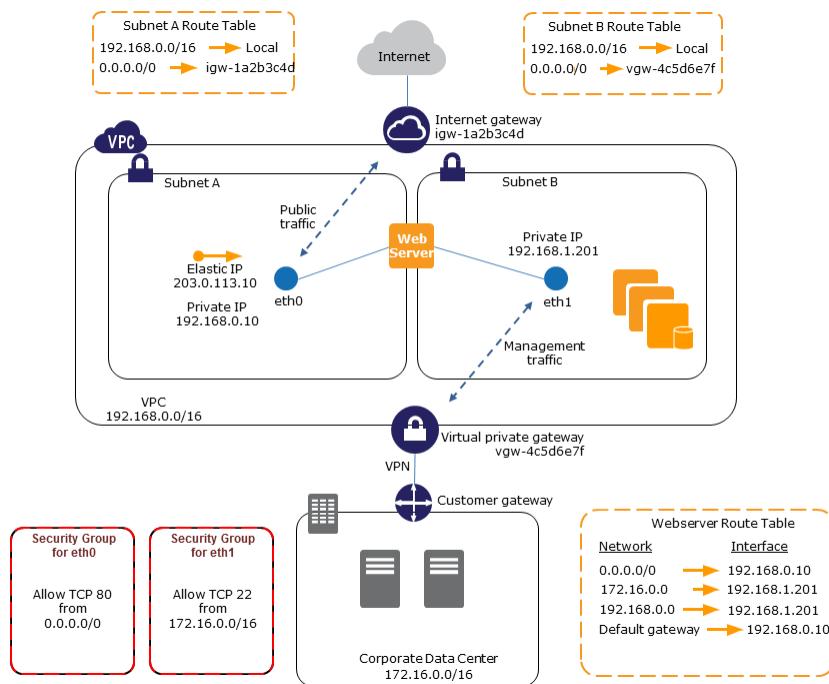
次の作業を行う場合、複数のネットワークインターフェイスをインスタンスにアタッチすると便利です。

- ・ 管理用ネットワークを作成する。
- ・ VPC 内でネットワークアプライアンスやセキュリティアプライアンスを使用する。
- ・ 別個のサブネット上のワークロード/ロールを使用するデュアルホーム接続インスタンスを作成する。
- ・ 低予算で可用性の高いソリューションを作成する。

## 管理用ネットワークの作成

ネットワークインターフェイスを利用して管理用ネットワークを作成できます。このシナリオでは、インスタンスのプライマリネットワークインターフェイス (eth0) でパブリックトラフィックを処理し、セカンダリネットワークインターフェイス (eth1) でバックエンドの管理用トラフィックを処理します。プライマリネットワークインターフェイスはアクセス制御を強化した VPC 内の個別のサブネットに接続されます。パブリック側のインターフェイス (ロードバランサーの背後に置かれる場合とそうでない場合があります) には、インターネットからサーバーへのアクセスを許可するセキュリティグループが関連付けられます (たとえば、0.0.0.0/0 またはロードバランサーからの TCP ポート 80 および 443 を許可します)。プライベート側のインターフェイスには、VPC 内またはインターネットからの許容範囲の IP アドレス、VPC 内のプライベートサブネット、または仮想プライベートゲートウェイからの SSH アクセスのみを許可するセキュリティグループが関連付けられます。

フェイルオーバー機能が確実に動作するように、ネットワークインターフェイスの受信トラフィックに対してセカンダリプライベート IPv4 を使用することをお勧めします。インスタンスに障害が発生した場合は、インターフェイスまたはセカンダリプライベート IPv4 アドレスあるいはその両方をスタンバイ用のインスタンスに移行できます。



## VPC 内でネットワークアプライアンスとセキュリティアプライアンスを使用する

ロードバランサー、ネットワークアドレス変換 (NAT) サーバー、プロキシサーバーなど、ネットワークアプライアンスやセキュリティアプライアンスの中には、複数のネットワークインターフェイスを使用した構成が優先されるものがあります。セカンダリネットワークインターフェイスを作成して、これらのタイプのアプリケーションを実行する VPC 内のインスタンスにアタッチし、専用のパブリック IP アドレスとプライベート IP アドレス、セキュリティグループ、およびソース/デスティネーションチェックを使用して追加のインターフェイスを構成することができます。

## 別個のサブネット上のワークロード/ロールを使用するデュアルホーム接続インスタンスを作成する

アプリケーションサーバーが存在するミッドティアネットワークに接続する Web サーバーのそれぞれにネットワークインターフェイスを置くことができます。このアプリケーションサーバーは、データベースサーバーが存在するバックエンドネットワーク（サブネット）にデュアルホーム接続することもできます。デュアルホーム接続されたインスタンスを介してネットワークパケットをルーティングする代わりに、デュアルホーム接続された各インスタンスは、フロントエンドでリクエストを受信して処理し、バックエンドとの接続を開始して、バックエンドネットワーク上のサーバーにリクエストを送信します。

## 低予算で可用性の高いソリューションを構築する

特定の機能にサービスを提供しているインスタンスのいずれかが機能しなくなった場合は、そのネットワークインターフェイスを同じ役割で構成された交換用またはホットスタンバイ用のインスタンスにアタッチすることで、サービスを迅速に回復できます。たとえば、データベースインスタンスや NAT インスタンスなどの重要なサービスに対するプライマリまたはセカンダリのネットワークインターフェイスとしてネットワークインターフェイスを使用することができます。そのインスタンスが機能しなくなった場合、お客様（通常はお客様に代わって実行されるコード）がネットワークインターフェイスをホットスタンバイ用のインスタンスにアタッチすることができます。インターフェイスでは、プライベート IP アドレ

ス、Elastic IP アドレス、および MAC アドレスがそのまま維持されるため、交換用のインスタンスにネットワークインターフェイスを接続するとすぐに、ネットワークトラフィックはスタンバイ用のインスタンスに流れ始めます。インスタンスに障害が発生してから、ネットワークインターフェイスがスタンバイ用のインスタンスにアタッチされるまで、一時的な接続断が発生しますが、VPC ルートテーブルや DNS サーバーに変更を加える必要はありません。

## ネットワークインターフェイスの設定に関するベストプラクティス

- ネットワークインターフェイスは、インスタンスの実行中、インスタンスの停止中、インスタンスの起動中にインスタンスにアタッチできます(それぞれ、ホットアタッチ、ウォームアタッチ、コールドアタッチと呼ばれています)。
- セカンダリネットワークインターフェイスは、インスタンスの実行中または停止中にデタッチできます。ただし、プライマリネットワークインターフェイス(eth0)をデタッチすることはできません。
- インスタンスが同じアベイラビリティゾーンと VPC にあるが、異なるサブネットにある場合、ネットワークインターフェースを 1 つのインスタンスから別のインスタンスに移動できます。
- CLI、API、または SDK を使用してインスタンスを起動する場合、プライマリネットワークインターフェイス(eth0) および追加のネットワークインターフェイスを指定できます。
- 複数のネットワークインターフェイスを使用して Amazon Linux または Windows Server インスタンスを起動すると、インスタンスのオペレーティングシステム上でインターフェイス、プライベート IPv4 アドレス、ルートテーブルが自動的に設定されます。
- 追加ネットワークインターフェイスをウォームアタッチまたはホットアタッチするとき、場合によっては、手動で 2 つ目のインターフェイスを起動し、プライベート IPv4 アドレスを設定し、ルートテーブルを適宜変更する必要があります。Amazon Linux または Windows Server を実行するインスタンスは、ウォームアタッチまたはホットアタッチを自動的に認識し、それらのインスタンス自体を設定します。
- デュアルホーム接続インスタンスに対するネットワーク帯域幅を増加または倍増させる方法として、別のネットワークインターフェイスをインスタンスにアタッチする機能(NIC チーミング設定など)は使用できません。
- 同じサブネットから複数のネットワークインターフェイスをインスタンスにアタッチすると、非対称ルーティングなどのネットワーク問題が発生する場合があります。可能であれば、代わりにプライマリネットワークインターフェイス上でセカンダリプライベート IPv4 アドレスを使用します。詳細については、「[セカンダリプライベート IPv4 アドレスを割り当てる \(p. 694\)](#)」を参照してください。

## ec2-net-utils を使用したネットワークインターフェイスの設定

Amazon Linux AMI には、AWS によって ec2-net-utils という追加のスクリプトがインストールされていることがあります。これらのスクリプトはオプションで、ネットワークインターフェイスの設定を自動化します。これらのスクリプトは Amazon Linux でのみ使用できます。

パッケージをまだインストールしていない場合は、以下のコマンドを使用して Amazon Linux にインストールします。インストール済みで、利用可能な更新がある場合は、更新します。

```
$ yum install ec2-net-utils
```

ec2-net-utils には、以下のコンポーネントが含まれます。

udev ルール (/etc/udev/rules.d)

実行中のインスタンスにネットワークインターフェイスがアタッチ、デタッチ、または再アタッチされたときに、そのネットワークインターフェイスを特定し、ホットプラグスクリプトが実行されることを確認します(53-ec2-network-interfaces.rules)。MAC アドレスをデバイス名にマッピングします(75-persistent-net-generator.rules を生成する 70-persistent-net.rules)。

## ホットプラグスクリプト

DHCP での使用に適したインターフェイス設定ファイルを生成します (`/etc/sysconfig/network-scripts/ifcfg-ethN`)。また、ルート設定ファイルも生成します (`/etc/sysconfig/network-scripts/route-ethN`)。

### DHCP スクリプト

ネットワークインターフェイスが新しい DHCP リースを受け取るたびに、このスクリプトがインスタンスマターダーに対し、Elastic IP アドレスを求めるクエリを実行します。これにより、各 Elastic IP アドレスごとに、そのアドレスからのアウトバンドトラフィックが正しいネットワークインターフェイスを使用するよう、ルーティングポリシーデータベースにルールが追加されます。また、各プライベート IP アドレスを、セカンダリアアドレスとしてネットワークインターフェイスに追加します。

### ec2ifup ethN

標準の `ifup` の機能を拡張します。このスクリプトが設定ファイル `ifcfg-ethN` および `route-ethN` を書き換えた後、`ifup` を実行します。

### ec2ifdown ethN

標準の `ifdown` の機能を拡張します。このスクリプトがルーティングポリシーデータベースからネットワークインターフェイスのルールをすべて削除した後、`ifdown` を実行します。

### ec2ifscan

まだ設定されていないネットワークインターフェイスを探して、それらを設定します。

このスクリプトは、`ec2-net-utils` の初期リリースでは提供されていません。

`ec2-net-utils` によって生成された設定ファイルをリストするには、以下のコマンドを使用します。

```
$ ls -l /etc/sysconfig/network-scripts/*-eth?
```

インスタンスごとのオートメーションを無効にするには、対応する `EC2SYNC=no` `ifcfg-ethN` ファイルに `EC2SYNC=yes` を追加します。たとえば、`eth1` インターフェイスの自動化を無効にするには、以下のコマンドを使用します。

```
$ sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

オートメーションを完全に無効にするには、次のコマンドを使用してパッケージを削除できます。

```
$ yum remove ec2-net-utils
```

## ネットワークインターフェイスでの作業

ネットワークインターフェイスは、Amazon EC2 コンソールまたはコマンドラインを使用して操作できます。

### コンテンツ

- ネットワークインターフェイスを作成する (p. 728)
- ネットワークインターフェイスの削除 (p. 728)
- ネットワークインターフェイスに関する詳細の表示 (p. 729)
- インスタンスの起動時にネットワークインターフェイスをアタッチする (p. 729)
- 停止したインスタンスまたは実行中のインスタンスにネットワークインターフェイスをアタッチする (p. 730)
- ネットワークインターフェイスをインスタンスからデタッチする (p. 731)
- セキュリティグループの変更 (p. 732)

- 送信元または送信先チェックの変更 (p. 733)
- Elastic IP アドレス (IPv4) の関連付け (p. 733)
- Elastic IP アドレス (IPv4) の関連付けの解除 (p. 734)
- IPv6 アドレスの割り当て (p. 734)
- IPv6 アドレスの割り当て解除 (p. 735)
- 終了動作の変更 (p. 735)
- 説明の追加または編集 (p. 735)
- タグの追加または編集 (p. 736)

## ネットワークインターフェイスを作成する

ネットワークインターフェイスはサブネットで作成できます。作成後のネットワークインターフェイスを別のサブネットに移動することはできません。ネットワークインターフェイスは、同じアベイラビリティーゾーンのインスタンスにのみアタッチできます。

コンソールを使用してネットワークインターフェイスを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. [Create Network Interface] を選択します。
4. [Description] で、記述的な名前を入力します。
5. [Subnet] で、サブネットを選択します。
6. [Private IP] (または [IPv4 Private IP]) に、プライマリプライベート IPv4 アドレスを入力します。IPv4 アドレスを指定しない場合、選択されているサブネット内で使用可能な IPv4 アドレスが自動的に選択されます。
7. ([IPv6 のみ) IPv6 CIDR ブロックが関連付けられているサブネットを選択した場合は、オプションで [IPv6 IP] フィールドに IPv6 アドレスを指定できます。
8. [Security groups] で、1 つまたは複数のセキュリティグループを選択します。
9. [Yes, Create] を選択します。

コマンドラインを使用してネットワークインターフェイスを作成するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `create-network-interface` (AWS CLI)
- `New-EC2NetworkInterface` (AWS Tools for Windows PowerShell)

## ネットワークインターフェイスの削除

ネットワークインターフェイスを削除するには、最初にそれをデタッチする必要があります。ネットワークインターフェイスを削除すると、そのインターフェイスに関連付けられているすべての属性が解放され、別のインスタンスで使用できるように、プライベート IP アドレスまたは Elastic IP アドレスが解放されます。

コンソールを使用してネットワークインターフェイスを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[Delete] を選択します。
4. [Delete Network Interface] ダイアログボックスで、[Yes, Delete] を選択します。

## コマンドラインを使用してネットワークインターフェイスを削除するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [delete-network-interface](#) (AWS CLI)
- [Remove-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

## ネットワークインターフェイスに関する詳細の表示

アカウントのすべてのネットワークインターフェイスを表示できます。

コンソールを使用してネットワークインターフェイスを記述するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択します。
4. 詳細を表示するには、[Details] を選択します。

コマンドラインを使用してネットワークインターフェイスを記述するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-network-interfaces](#) (AWS CLI)
- [Get-EC2NetworkInterface](#) (AWS Tools for Windows PowerShell)

コマンドラインを使用してネットワークインターフェイス属性を記述するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-network-interface-attribute](#) (AWS CLI)
- [Get-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

## インスタンスの起動時にネットワークインターフェイスをアタッチする

インスタンスの起動時に、既存のネットワークインターフェイスを指定するか、追加のネットワークインターフェイスをアタッチすることができます。

### Note

ネットワークインターフェイスをインスタンスにアタッチしたときにエラーが発生した場合、インスタンスは正しく起動されません。

コンソールを使用してインスタンスの起動時にネットワークインターフェイスをアタッチするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. AMI およびインスタンスタイプを選択し、[次の手順: インスタンスの詳細の設定] を選択します。

4. [Configure Instance Details] ページで、[Network] の VPC を選択し、[Subnet] のサブネットを選択します。
5. コンソールの [Network Interfaces] セクションで、インスタンスを起動する際に最大 2 つのネットワークインターフェイス(新規、既存、またはその組み合わせ)を指定することができます。また、すべての新規のネットワークインターフェイスについて、1 つのプライマリ IPv4 アドレスと、1 つ以上のセカンダリ IPv4 アドレスを入力することができます。

インスタンスの起動後に、追加のネットワークインターフェイスを追加できます。アタッチできるネットワークインターフェイスの合計数はインスタンスタイプによって異なります。詳細については、「[各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス数 \(p. 714\)](#)」を参照してください。

Note

複数のネットワークインターフェイスを指定した場合、インスタンスにパブリック IPv4 アドレスを自動的に割り当てるることはできません。

6. (IPv6 のみ) IPv6 CIDR ブロックが関連付けられているサブネットでインスタンスを起動する場合は、接続するすべてのネットワークインターフェイスに対して IPv6 アドレスを指定できます。[IPv6 IPs] で、[Add IP] を選択します。セカンダリ IPv6 アドレスを追加するには、再度 [Add IP] 選択します。サブネットの範囲から IPv6 アドレスを入力するか、デフォルトの [Auto-assign] を使用してサブネットから自動的に IPv6 アドレスを選択することができます。
7. [次の手順: ストレージの追加] を選択します。
8. [Add Storage] ページで、AMI によって指定されるボリューム(ルートデバイスピリュームなど)以外にインスタンスにアタッチするボリュームを指定し、[Next: Add Tags] を選択します。
9. [Add Tags] ページで、ユーザーフレンドリーな名前などを使ってインスタンスのタグを指定し、[Next: Configure Security Group] を選択します。
10. [Configure Security Group] ページで、セキュリティグループを選択するか、新しいグループを作成できます。[Review and Launch] を選択します。

Note

ステップ 5 で既存のネットワークインターフェイスを指定した場合、このステップで選択したオプションに関係なしに、インスタンスはそのネットワークインターフェイスのセキュリティグループと関連付けられます。

11. [Review Instance Launch] ページに、プライマリおよび追加ネットワークインターフェイスの詳細情報が表示されます。設定を確認し、[Launch] を選択して、キーペアを選択し、インスタンスを起動します。Amazon EC2 を初めて使用する場合、これまでにキーペアを作成したことがなければ、ウィザードによってキーペアを作成するよう求めるメッセージが表示されます。

コマンドラインを使用してインスタンスの起動時にネットワークインターフェイスをアタッチするには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [run-instances \(AWS CLI\)](#)
- [New-EC2Instance \(AWS Tools for Windows PowerShell\)](#)

## 停止したインスタンスまたは実行中のインスタンスにネットワークインターフェイスをアタッチする

VPC 内で停止したインスタンスまたは実行中のインスタンスにネットワークインターフェイスをアタッチできます。アタッチするには、Amazon EC2 コンソールの [インスタンス] ページまたは [ネットワークインターフェイス] ページを使用します。

#### Note

インスタンスのパブリック IPv4 アドレスが解放される場合、複数のネットワークインターフェイスがそのインスタンスにアタッチされていると、インスタンスに新しいパブリック IP アドレスは送信されません。パブリック IPv4 アドレスの動作の詳細については、「[パブリック IPv4 アドレスと外部 DNS ホスト名 \(p. 686\)](#)」を参照してください。

インスタンスページを使用してネットワークインターフェイスをインスタンスにアタッチするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. [Actions]、[Networking]、[Attach Network Interface] の順に選択します。
4. [Attach Network Interface] ダイアログボックスで、Network Interface を選択し、[Attach] を選択します。

ネットワークインターフェイスページを使用してネットワークインターフェイスをインスタンスにアタッチするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[アタッチ] を選択します。
4. [Attach Network Interface] ダイアログボックスでインスタンスを選択し、[Attach] を選択します。

コマンドラインを使用してインスタンスにネットワークインターフェイスをアタッチするには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [attach-network-interface \(AWS CLI\)](#)
- [Add-EC2NetworkInterface \(AWS Tools for Windows PowerShell\)](#)

## ネットワークインターフェイスをインスタンスからデタッチする

セカンダリネットワークインターフェイスは、Amazon EC2 コンソールの [インスタンス] ページまたは [ネットワークインターフェイス] ページを使用して、いつでもデタッチできます。

インスタンスページを使用してネットワークインターフェイスをインスタンスからデタッチするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. [Actions]、[Networking]、[Detach Network Interface] の順に選択します。
4. [Detach Network Interface] ダイアログボックスで、Network Interface を選択し、[Detach] を選択します。

ネットワークインターフェイスページを使用してネットワークインターフェイスをインスタンスからデタッチするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[デタッチ] を選択します。

- [Detach Network Interface] ダイアログボックスで [Yes, Detach] を選択します。Network Interface をインスタンスからデタッチできなかった場合は、[Force detachment] を選択し、再試行します。

Note

- [強制デタッチ] オプションは、失敗したインスタンスからネットワークインターフェイスをデタッチする最後の手段としてのみ使用してください。
- [強制デタッチ] オプションを使用してネットワークインターフェイスを切断する場合、最初にインスタンスを停止して起動しないと、インスタンスの同じインデックスに別のネットワークインターフェイスをアタッチできない場合があります。
- ネットワークインターフェイスのデタッチを強制すると、[インスタンスのメタデータ \(p. 593\)](#)が更新されない場合があります。つまり、デタッチされたネットワークインターフェイスに関連付けられた属性がその後も表示される可能性があります。インスタンスを停止および開始すると、インスタンスのメタデータが更新されます。
- ネットワークインターフェイスを EC2-Classic インスタンスから強制的にデタッチすることはできません。

コマンドラインを使用してネットワークインターフェイスをデタッチするには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [detach-network-interface \(AWS CLI\)](#)
- [Dismount-EC2NetworkInterface \(AWS Tools for Windows PowerShell\)](#)

## セキュリティグループの変更

ネットワークインターフェイスに関連付けられているセキュリティグループを変更できます。セキュリティグループを作成するとき、ネットワークインターフェイスのサブネットと同じ VPC を必ず指定します。

Note

Elastic Load Balancing などの他のサービスが所有するインターフェイスのセキュリティグループメンバーシップを変更するには、そのサービスのコンソールまたはコマンドラインインターフェイスを使用します。

コンソールを使用してネットワークインターフェイスのセキュリティグループを変更するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Network Interfaces] を選択します。
- ネットワークインターフェイスを選択し、[アクション] を選択して、[セキュリティグループの変更] を選択します。
- [Change Security Groups] ダイアログボックスで、使用するセキュリティグループを選択し、[Save] を選択します。

コマンドラインを使用してネットワークインターフェイスのセキュリティグループを変更するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [modify-network-interface-attribute \(AWS CLI\)](#)
- [Edit-EC2NetworkInterfaceAttribute \(AWS Tools for Windows PowerShell\)](#)

## 送信元または送信先チェックの変更

[Source/Destination Check] 属性により、送信元/送信先チェックがインスタンスで有効になっているかどうかが制御されます。この属性を無効にすると、インスタンスで自身にアドレス指定されていないネットワークトラフィックを処理することが可能になります。たとえば、ネットワークアドレス変換、ルーティング、ファイアウォールなどのサービスを実行するインスタンスではこの値を `disabled` に設定する必要があります。デフォルト値は `enabled` です。

コンソールを使用してネットワークインターフェイスの送信元/送信先チェックを変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[Actions] を選択して、[Change Source/Dest Check] を選択します。
4. In the dialog box, choose Enabled (if enabling) or Disabled (if disabling), and Save.

コマンドラインを使用してネットワークインターフェイスの送信元/送信先チェックを変更するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `modify-network-interface-attribute` (AWS CLI)
- `Edit-EC2NetworkInterfaceAttribute` (AWS Tools for Windows PowerShell)

## Elastic IP アドレス (IPv4) の関連付け

Elastic IP アドレス (IPv4) が与えられている場合、ネットワークインターフェイスのプライベート (IPv4) アドレスの 1 つをそれと関連付けることができます。1 つの Elastic IP アドレスと各プライベート IPv4 アドレスを関連付けることができます。

Amazon EC2 コンソールまたはコマンドラインを使用して、Elastic IP アドレスを関連付けることができます。

コンソールを使用して Elastic IP アドレスを関連付けるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[アクション] を選択して、[アドレスの関連付け] を選択します。
4. [Associate Elastic IP Address] ダイアログボックスで、[Address] リストから Elastic IP アドレスを選択します。
5. [Associate to private IP address] で、Elastic IP アドレスに関連付けるプライベート IPv4 アドレスを選択します。
6. [Allow reassociation] を選択して、指定したネットワークインターフェイスに Elastic IP アドレスを関連付け (現在、別のインスタンスまたはネットワークインターフェイスに関連付けられている場合)、[Associate Address] を選択します。

コマンドラインを使用して Elastic IP アドレスを関連付けるには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [associate-address \(AWS CLI\)](#)
- [Register-EC2Address \(AWS Tools for Windows PowerShell\)](#)

## Elastic IP アドレス (IPv4) の関連付けの解除

ネットワークインターフェイスに Elastic IP アドレス (IPv4) が関連付けられている場合、アドレスの関連付けを解除し、別のネットワークインターフェイスに関連付けるか、解放してアドレスプールに戻すことができます。ネットワークインターフェイスは特定のサブネットに固有であるため、これはネットワークインターフェイスを使用して別のサブネットや VPC 内のインスタンスに Elastic IP アドレスを関連付ける唯一の方法になります。

Amazon EC2 コンソールまたはコマンドラインを使用して、Elastic IP アドレスの関連付けを解除することができます。

コンソールを使用して Elastic IP アドレスの関連付けを解除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[Actions] を選択して、[Disassociate Address] を選択します。
4. [Disassociate IP Address] ダイアログボックスで [Yes, Disassociate] を選択します。

コマンドラインを使用して Elastic IP アドレスを別のインスタンスに関連付けるには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [disassociate-address \(AWS CLI\)](#)
- [Unregister-EC2Address \(AWS Tools for Windows PowerShell\)](#)

## IPv6 アドレスの割り当て

1 つ以上の IPv6 アドレスをネットワークインターフェイスに割り当てるすることができます。そのネットワークインターフェイスは、IPv6 CIDR ブロックが関連付けられているサブネットにあることが必要です。特定の IPv6 アドレスをネットワークインターフェイスに割り当てるには、その IPv6 アドレスが別のネットワークインターフェイスにまだ割り当てられていないことを確認します。

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Network Interfaces] を選択してから、ネットワークインターフェイスを選択します。
3. [Actions]、[Manage IP Addresses] の順に選択します。
4. [IPv6 Addresses] で、[Assign new IP] を選択します。サブネットの範囲から IPv6 アドレスを指定します。AWS で自動的にアドレスを選択するには、[Auto-assign] 値をそのままにします。
5. [Yes, Update] を選択します。

コマンドラインを使用して IPv6 アドレスをネットワークインターフェイスに割り当てるには

- 次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。
  - [assign-ipv6-addresses \(AWS CLI\)](#)
  - [Register-EC2Ipv6AddressList \(AWS Tools for Windows PowerShell\)](#)

## IPv6 アドレスの割り当て解除

Amazon EC2 コンソールを使用してネットワークインターフェイスから IPv6 アドレスを割り当て解除できます。

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Network Interfaces] を選択してから、ネットワークインターフェイスを選択します。
3. [Actions]、[Manage IP Addresses] の順に選択します。
4. [IPv6 Addresses] で、割り当て解除する IPv6 アドレスに対して [Unassign] を選択します。
5. [Yes, Update] を選択します。

コマンドラインを使用してネットワークインターフェイスから IPv6 アドレスを割り当て解除するには

- 次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。
  - `unassign-ipv6-addresses` (AWS CLI)
  - `Unregister-EC2Ipv6AddressList` (AWS Tools for Windows PowerShell)

## 終了動作の変更

インスタンスにアタッチされているネットワークインターフェイスの終了動作を設定できます。アタッチしたインスタンスの終了時に、ネットワークインターフェイスを自動的に削除するかどうかを指定できます。

Amazon EC2 コンソールまたはコマンドラインを使用して、ネットワークインターフェイスの終了時の動作を変更することができます。

コンソールを使用してネットワークインターフェイスの終了時の動作を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[アクション] を選択して、[Change Termination Behavior (終了時の動作を変更)] を選択します。
4. インスタンスの終了時にネットワークインターフェイスを削除する場合は、[Change Termination Behavior] ダイアログボックスの [Delete on termination] チェックボックスを選択します。

コマンドラインを使用してネットワークインターフェイスの終了時の動作を変更するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `modify-network-interface-attribute` (AWS CLI)
- `Edit-EC2NetworkInterfaceAttribute` (AWS Tools for Windows PowerShell)

## 説明の追加または編集

Amazon EC2 コンソールまたはコマンドラインを使用して、ネットワークインターフェイスの説明を変更することができます。

コンソールを使用してネットワークインターフェイスの説明を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[アクション] を選択して、[説明の変更] を選択します。
4. [Change Description] ダイアログボックスで Network Interface の説明を入力し、[Save] を選択します。

コマンドラインを使用してネットワークインターフェイスの説明を変更するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [modify-network-interface-attribute](#) (AWS CLI)
- [Edit-EC2NetworkInterfaceAttribute](#) (AWS Tools for Windows PowerShell)

## タグの追加または編集

タグとは、ネットワークインターフェイスに追加できるメタデータです。タグはプライベートとして扱われ、アカウントでのみ表示できます。各タグはキーとオプションの値で構成されます。タグの詳細については、[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#) を参照してください。

コンソールを使用してネットワークインターフェイスのタグを追加または編集するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択します。
4. 詳細ペインの [Tags] を選択し、[Add/Edit Tags] を選択します。
5. [Add/Edit Tags] ダイアログボックスで、作成するタグごとに [Create Tag] を選択して、キーとオプションの値を入力します。完了したら、[Save] を選択します。

コマンドラインを使用してネットワークインターフェイスのタグを追加または編集するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [create-tags](#) (AWS CLI)
- [New-EC2Tag](#) (AWS Tools for Windows PowerShell)

## リクエスタマネージド型のネットワークインターフェイス

リクエスタマネージド型のネットワークインターフェイスは、AWS のサービスによって VPC 内に作成されるネットワークインターフェイスです。このネットワークインターフェイスは、別のサービスのインスタンス (Amazon RDS インスタンスなど) を表すことができます。または、別のサービスやリソース (AWS PrivateLink サービスや Amazon ECS タスクなど) にユーザーがアクセスすることを可能にします。

リクエスタマネージド型のネットワークインターフェイスを変更またはデタッチすることはできません。ネットワークインターフェイスが表すリソースを削除すると、AWS のサービスはユーザーに代わってネットワークインターフェイスをデタッチおよび削除します。リクエスタマネージド型のネットワークインターフェイスのセキュリティグループを変更するには、そのサービスのコンソールまたはコマンドライン

ツールの使用が必要になる場合があります。詳細については、サービス固有のドキュメントを参照してください。

リクエスタマネージド型のネットワークインターフェイスにはタグを付けることができます。詳細については、「[タグの追加または編集 \(p. 736\)](#)」を参照してください。

アカウントにあるリクエスタマネージド型のネットワークインターフェイスを表示できます。

コンソールを使用してリクエスタマネージド型のネットワークインターフェイスを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、詳細ペインで以下の情報を確認します。
  - [Attachment owner]: ユーザーがネットワークインターフェイスを作成した場合は、このフィールドに AWS アカウントID が表示されます。それ以外の場合は、ネットワークインターフェイスを作成したプリンシパルやサービスのエイリアスまたは ID が表示されます。
  - [Description]: ネットワークインターフェイスの用途（「VPC エンドポイントインターフェイス」など）を説明します。

コマンドラインを使用してリクエスタマネージド型のネットワークインターフェイスを表示するには

1. AWS CLI の `describe-network-interfaces` コマンドを使用してアカウントのネットワークインターフェイスを記述します。

```
aws ec2 describe-network-interfaces
```

2. ネットワークインターフェイスが AWS の別のサービスで管理されている場合は、出力の [RequesterManaged] フィールドに `true` と表示されます。

```
{  
    "Status": "in-use",  
    ...  
    "Description": "VPC Endpoint Interface vpce-089f2123488812123",  
    "NetworkInterfaceId": "eni-c8fbc27e",  
    "VpcId": "vpc-1a2b3c4d",  
    "PrivateIpAddresses": [  
        {  
            "PrivateDnsName": "ip-10-0-2-227.ec2.internal",  
            "Primary": true,  
            "PrivateIpAddress": "10.0.2.227"  
        }  
    ],  
    "RequesterManaged": true,  
    ...  
}
```

または、Tools for Windows PowerShell の `Get-EC2NetworkInterface` コマンドを使用します。

## Linux の拡張ネットワーキング

拡張ネットワーキングでは、シングルルート I/O 仮想化 (SR-IOV) を使用して、[サポートされるインスタンスタイプ \(p. 738\)](#)における高性能ネットワーキング機能が提供されます。SR-IOV は、従来の仮想化ネットワークインターフェイスと比較し、I/O パフォーマンスが高く、CPU 利用率が低いデバイス仮想化の手法です。拡張ネットワーキングは、高い帯域幅、1 秒あたりのパケット (PPS) の高いパフォーマンス、

常に低いインスタンス間レイテンシーを実現します。拡張ネットワーキングは追加料金なしで使用できます。

#### コンテンツ

- [拡張ネットワーキングのタイプ \(p. 738\)](#)
- [インスタンスでの拡張ネットワーキングの有効化 \(p. 738\)](#)
- [Linux インスタンスにおける Elastic Network Adapter \(ENA\) を使用した拡張ネットワーキングの有効化 \(p. 738\)](#)
- [Linux インスタンスにおけるインテル 82599 VF インターフェイスを使用した拡張ネットワーキングの有効化 \(p. 751\)](#)
- [Elastic Network Adapter \(ENA\) のトラブルシューティング \(p. 757\)](#)

## 拡張ネットワーキングのタイプ<sup>®</sup>

インスタンスタイプに応じて、次のいずれかのメカニズムを使用して拡張ネットワーキングを有効にすることができます。

#### Elastic Network Adapter (ENA)

Elastic Network Adapter (ENA) は、サポート対象のインスタンスタイプに対して最大 100 Gbps のネットワーク速度をサポートします。

A1, C5, C5d, C5n, F1, G3, G4, H1, I3, I3en, Inf1, m4.16xlarge, M5, M5a, M5ad, M5d, M5dn, M5n, P2, P3, R4, R5, R5a, R5ad, R5d, R5dn, R5n, T3, T3a, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, u-24tb1.metal, X1, X1e, and z1d インスタンスでは、拡張ネットワーキングで Elastic Network Adapter を使用します。

#### Intel 82599 Virtual Function (VF) インターフェイス

Intel 82599 Virtual Function インターフェイスでは、サポートされているインスタンスタイプについて最大 10 Gbps のネットワーク速度がサポートされています。

C3、C4、D2、I2、M4 (m4.16xlarge を除く)、および R3 インスタンスでは、拡張ネットワーキングにインテル 82599 VF インターフェイスが使用されます。

各インスタンスタイプでサポートされているネットワーク速度については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

## インスタンスでの拡張ネットワーキングの有効化

ご使用のインスタンスタイプで拡張ネットワーキングに Elastic Network Adapter がサポートされている場合、「[Linux インスタンスにおける Elastic Network Adapter \(ENA\) を使用した拡張ネットワーキングの有効化 \(p. 738\)](#)」の手順に従います。

ご使用のインスタンスタイプで拡張ネットワーキングに Intel 82599 VF インターフェイスがサポートされている場合、「[Linux インスタンスにおけるインテル 82599 VF インターフェイスを使用した拡張ネットワーキングの有効化 \(p. 751\)](#)」の手順に従います。

## Linux インスタンスにおける Elastic Network Adapter (ENA) を使用した拡張ネットワーキングの有効化

Amazon EC2 は、Elastic Network Adapter (ENA) を介してネットワーキング機能を提供します。

#### コンテンツ

- [要件 \(p. 739\)](#)
- [拡張ネットワーキングが有効化されているかどうかのテスト \(p. 739\)](#)
- [Amazon Linux AMI での拡張ネットワーキングの有効化 \(p. 741\)](#)
- [Ubuntu での拡張ネットワーキングの有効化 \(p. 742\)](#)
- [Linux での拡張ネットワーキングの有効化 \(p. 744\)](#)
- [DKMS を使用した Ubuntu での拡張ネットワーキングの有効化 \(p. 746\)](#)
- [トラブルシューティング \(p. 747\)](#)
- [オペレーティングシステムの最適化 \(p. 747\)](#)

## 要件

ENA を使用した拡張ネットワーキングを準備するには、次のようにインスタンスをセットアップします。

- サポートされているインスタンスタイプは次のタイプのみです: A1, C5, C5d, C5n, F1, G3, G4, H1, I3, I3en, Inf1, m4.16xlarge, M5, M5a, M5ad, M5d, M5dn, M5n, P2, P3, R4, R5, R5a, R5ad, R5d, R5dn, R5n, T3, T3a, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, u-18tb1.metal, u-24tb1.metal, X1, X1e, and z1d。
- サポートされているバージョンの Linux カーネルとサポートされているディストリビューションを使用してインスタンスを起動します。インスタンスに対して ENA 拡張ネットワーキングが自動的に有効化されます。詳細については、[ENA Linux Kernel Driver Release Notes](#) を参照してください。
- インスタンスがインターネットに接続されていることを確認します。
- 選択した任意のコンピュータ、できればローカルのデスクトップまたはノート PC に、[AWS CLI](#) または [AWS Tools for Windows PowerShell](#) をインストールして設定します。詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。拡張ネットワーキングは、Amazon EC2 コンソールから管理することはできません。
- 保持する必要がある重要なデータがインスタンスにある場合、インスタンスから AMI を作成してそのデータをバックアップする必要があります。enaSupport 属性を有効にするとともに、カーネルおよびカーネルモジュールを更新すると、互換性のないインスタンスがレンダリングされたり、オペレーティングシステムに接続できなくなったりする可能性があります。最近のバックアップがある場合は、これが発生してもデータは保持されます。

## 拡張ネットワーキングが有効化されているかどうかのテスト

拡張ネットワーキングが既に有効になっているかどうかをテストするには、ena モジュールがインスタンスにインストールされていることと、enaSupport 属性が設定されていることを確認します。インスタンスがこれら 2 つの条件を満たしている場合は、ethtool -i ethn コマンドによって、ネットワークインターフェイスで使用されているモジュールが表示されます。

### カーネルモジュール (ena)

ena モジュールがインストールされたことを確認するには、以下の例に示されるように modinfo コマンドを使用します。

```
[ec2-user ~]$ modinfo ena
filename:      /lib/modules/4.14.33-59.37.amzn2.x86_64/kernel/drivers/amazon/net/ena/
ena.ko
version:       1.5.0g
license:        GPL
description:   Elastic Network Adapter (ENA)
author:         Amazon.com, Inc. or its affiliates
srcversion:    692C7C68B8A9001CB3F31D0
alias:          pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:          pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:          pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
```

```
alias:          pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
retpoline:      Y
intree:         Y
name:          ena
...
```

上の Amazon Linux のケースでは、ena モジュールはインストールされています。

```
ubuntu:~$ modinfo ena
ERROR: modinfo: could not find module ena
```

上の Ubuntu インスタンスでは、モジュールはインストールされていないため、まずインストールする必要があります。詳細については、「[Ubuntu での拡張ネットワーキングの有効化 \(p. 742\)](#)」を参照してください。

#### インスタンス属性 (enaSupport)

インスタンスに拡張ネットワーキングの enaSupport 属性が設定されているかどうかを確認するには、次のいずれかのコマンドを使用します。属性が設定されている場合、レスポンスは true です。

- [describe-instances](#) (AWS CLI)

```
aws ec2 describe-instances --instance-ids instance_id --query
  "Reservations[].[Instances[].[EnaSupport]]"
```

- [Get-EC2Instance](#) (Tools for Windows PowerShell)

```
(Get-EC2Instance -InstanceId instance-id).Instances.EnaSupport
```

#### イメージ属性 (enaSupport)

AMI に拡張ネットワーキングの enaSupport 属性が設定されているかどうかを確認するには、次のいずれかのコマンドを使用します。属性が設定されている場合、レスポンスは true です。

- [describe-images](#) (AWS CLI)

```
aws ec2 describe-images --image-id ami_id --query "Images[].[EnaSupport]"
```

- [Get-EC2Image](#) (Tools for Windows PowerShell)

```
(Get-EC2Image -ImageId ami_id).EnaSupport
```

#### ネットワークインターフェイスドライバー

次のコマンドを使用して、ena モジュールが特定のインターフェイスで使用されていることを確認し、確認するインターフェイス名に置き換えます。単一のインターフェイス (デフォルト) を使用している場合は、eth0 です。オペレーティングシステムで[予測可能なネットワーク名 \(p. 744\)](#)がサポートされている場合は、ens5 のような名前にすることができます。

次の例で、リストされているドライバーは vif であるため、ena モジュールはロードされません。

```
[ec2-user ~]$ ethtool -i eth0
driver: vif
version:
firmware-version:
```

```
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

この例では、ena モジュールがロードされており、最小推奨バージョンです。このインスタンスでは、拡張ネットワーキングが適切に設定されています。

```
[ec2-user ~]$ ethtool -i eth0
driver: ena
version: 1.5.0g
firmware-version:
expansion-rom-version:
bus-info: 0000:00:05.0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

## Amazon Linux AMI での拡張ネットワーキングの有効化

Amazon Linux 2 および最新バージョンの Amazon Linux AMI では、拡張ネットワーキングに必要なモジュールがインストールされており、必要な `enaSupport` 属性も設定されています。したがって、サポートされるインスタンスタイプで HVM バージョンの Amazon Linux を使用してインスタンスを起動した場合、拡張ネットワーキングは既にインスタンスで有効になっています。詳細については、「[拡張ネットワーキングが有効化されているかどうかのテスト \(p. 739\)](#)」を参照してください。

以前の Amazon Linux AMI を使用してインスタンスを起動し、まだ拡張ネットワーキングが有効になっていない場合、拡張ネットワーキングを有効にするには次の手順を実行します。

Amazon Linux AMI で拡張ネットワーキングを有効化するには

1. インスタンスに接続します。
2. インスタンスから、次のコマンドを実行して、ena を含む最新のカーネルとカーネルモジュールでインスタンスを更新します。

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

3. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを再起動します。[reboot-instances](#) (AWS CLI)、[Restart-EC2Instance](#) (AWS Tools for Windows PowerShell)。
4. インスタンスに再接続し、「[拡張ネットワーキングが有効化されているかどうかのテスト \(p. 739\)](#)」の `modinfo ena` コマンドを使用して、ena モジュールがインストールされ、最小推奨バージョンであることを確認します。
5. [EBS-backed インスタンス] ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを停止します。[stop-instances](#) (AWS CLI)、[Stop-EC2Instance](#) (AWS Tools for Windows PowerShell)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。

[Instance store-backed インスタンス] インスタンスを停止して属性を変更することはできません。代わりに、この手順に進んでください: [Amazon Linux AMI で拡張ネットワーキングを有効にするには \(Instance store-backed インスタンス\) \(p. 742\)](#)。

6. ローカルコンピュータから、次のいずれかのコマンドを使用して拡張ネットワーキングの属性を有効化します。

- [modify-instance-attribute \(AWS CLI\)](#)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute \(Tools for Windows PowerShell\)](#)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

7. (オプション) 「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」の説明に従って、インスタンスから AMI を作成します。AMI は、インスタンスから拡張ネットワーキング enaSupport 属性を継承します。このため、この AMI を使用することで、拡張ネットワーキングがデフォルトで有効になっている別のインスタンスを起動できます。
8. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを起動します。[start-instances \(AWS CLI\)](#)、[Start-EC2Instance \(AWS Tools for Windows PowerShell\)](#)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。
9. インスタンスに接続し、[拡張ネットワーキングが有効化されているかどうかのテスト \(p. 739\)](#) の ethtool -i ethn コマンドを使用して、ena モジュールがインストールされ、ネットワークインターフェイスにロードされていることを確認します。

拡張ネットワーキングを有効にした後にインスタンスに接続できない場合、「[Elastic Network Adapter \(ENA\) のトラブルシューティング \(p. 757\)](#)」を参照してください。

Amazon Linux AMI で拡張ネットワーキングを有効にするには (Instance store-backed インスタンス)

インスタンスを停止するステップまで、前の手順に従います。「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」に記述されているように、新しい AMI を作成します。AMI を登録するときに拡張ネットワーキング属性を有効にしてください。

- [register-image \(AWS CLI\)](#)

```
aws ec2 register-image --ena-support ...
```

- [Register-EC2Image \(AWS Tools for Windows PowerShell\)](#)

```
Register-EC2Image -EnaSupport $true ...
```

## Ubuntu での拡張ネットワーキングの有効化

最新の Ubuntu HVM AMI では、ENA を使用した拡張ネットワーキングに必要なモジュールがインストールされており、必要な enaSupport 属性も設定されています。したがって、サポートされるインスタンスタイプで最新の Ubuntu HVM AMI を使用してインスタンスを起動した場合、拡張ネットワーキングは既にインスタンスで有効になっています。詳細については、「[拡張ネットワーキングが有効化されているかどうかのテスト \(p. 739\)](#)」を参照してください。

以前の AMI を使用してインスタンスを起動した場合、まだ拡張ネットワーキングが有効になっていなければ、linux-aws カーネルパッケージをインストールして最新の拡張ネットワーキングドライバーを取得して、必要な属性を更新できます。

linux-aws カーネルパッケージをインストールするには (Ubuntu 16.04 以降)

Ubuntu 16.04 および 18.04 には、Ubuntu カスタムカーネル (linux-aws カーネルパッケージ) が付属しています。別のカーネルを使用する場合は、[AWS サポート](#)までお問い合わせください。

linux-aws カーネルパッケージをインストールするには (Ubuntu Trusty 14.04)

1. インスタンスに接続します。
2. パッケージキャッシュおよびパッケージを更新します。

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

#### Important

更新プロセス中に grub をインストールするよう求められた場合は、/dev/xvda のインストール先として grub を使用し、現在のバージョンの /boot/grub/menu.lst を保持することを選択します。

3. [EBS-backed インスタンス] ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを停止します。[stop-instances](#) (AWS CLI)、[Stop-EC2Instance](#) (AWS Tools for Windows PowerShell)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。

[Instance store-backed インスタンス] インスタンスを停止して属性を変更することはできません。代わりに、この手順に進んでください: [Ubuntu で拡張ネットワーキングを有効にするには \(Instance store-backed インスタンス\) \(p. 743\)](#)。

4. ローカルコンピュータから、次のいずれかのコマンドを使用して拡張ネットワーキングの属性を有効化します。
  - [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute](#) (Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

5. (オプション) 「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」の説明に従って、インスタンスから AMI を作成します。AMI は、インスタンスから拡張ネットワーキング enaSupport 属性を継承します。このため、この AMI を使用することで、拡張ネットワーキングがデフォルトで有効になっている別のインスタンスを起動できます。
6. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを起動します。[start-instances](#) (AWS CLI)、[Start-EC2Instance](#) (AWS Tools for Windows PowerShell)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。

Ubuntu で拡張ネットワーキングを有効にするには (Instance store-backed インスタンス)

インスタンスを停止するステップまで、前の手順に従います。「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」に記述されているように、新しい AMI を作成します。AMI を登録するときに拡張ネットワーキング属性を有効にしてください。

- [register-image](#) (AWS CLI)

```
aws ec2 register-image --ena-support ...
```

- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

```
Register-EC2Image -EnaSupport $true ...
```

## Linux での拡張ネットワーキングの有効化

次の手順では、SUSE Linux Enterprise Server (SLES)、Red Hat Enterprise Linux、または CentOS など、Amazon Linux AMI または Ubuntu 以外の Linux ディストリビューションで拡張ネットワーキングを有効にするための一般的なステップを示します。開始する前に、「[拡張ネットワーキングが有効化されているかどうかのテスト \(p. 739\)](#)」を参照して、インスタンスで拡張ネットワーキングがすでに有効になっているかどうかを確認します。コマンドの詳細な構文、ファイルの場所、パッケージやツールのサポートなどの詳細については、使用する Linux ディストリビューションのドキュメントを参照してください。

Linux で拡張ネットワーキングを有効化するには

1. インスタンスに接続します。
2. の GitHub からインスタンスで ena モジュールのソースコードのクローンを作成します。(SUSE SLES 12 SP2 以降にはデフォルトで ENA 2.02 が含まれているため、ENA ドライバーをダウンロードしてコンパイルする必要はありません。SLES 12 SP2 以降では、必要なドライバーバージョンを標準カーネルに追加するように義務付ける必要があります)。

```
git clone https://github.com/amzn/amzn-drivers
```

3. インスタンスで ena モジュールをコンパイルし、インストールします。これらの手順は Linux ディストリビューションによって異なります。Red Hat Enterprise Linux でのモジュールのコンパイルの詳細については、[AWS ナリッジセンターの記事](#)を参照してください。
4. sudo depmod コマンドを実行して、モジュールの依存関係を更新します。
5. 起動時に新しいモジュールがロードされるように、インスタンスの initramfs を更新します。たとえば、ディストリビューションで dracut がサポートされる場合、次のコマンドを使用できます。

```
dracut -f -v
```

6. システムがデフォルトで予測可能なネットワークインターフェイス名を使用するかどうかを確認します。systemd または udev のバージョン 197 以上を使用するシステムの場合、イーサネットデバイスの名前を変更でき、単一ネットワークインターフェイスの名前が eth0 になることは保証されません。この動作は、インスタンスに接続する際に問題の原因となる可能性があります。詳細と他の設定オプションについては、freedesktop.org ウェブサイトで「[Predictable Network Interface Names/](#)」を参照してください。
  - a. 次のコマンドを使用して、RPM ベースのシステムで systemd または udev のバージョンを確認できます。

```
rpm -qa | grep -e '^systemd-[0-9]\+\|^\udev-[0-9]\+'  
systemd-208-11.el7_0.2.x86_64
```

上記の Red Hat Enterprise Linux 7 の例では、systemd のバージョンは 208 であるため、予測可能なネットワークインターフェイス名は無効になっている必要があります。

- b. net.ifnames=0 オプションを GRUB\_CMDLINE\_LINUX の /etc/default/grub 行に追加することによって、予測可能なネットワークインターフェイス名を無効にします。

```
sudo sed -i '/^GRUB_CMDLINE_LINUX/s/"$"/ net.ifnames=0"/' /etc/default/grub
```

- c. grub の設定ファイルを再ビルドします。

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [EBS-backed インスタンス] ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを停止します。[stop-instances](#) (AWS CLI)、[Stop-EC2Instance](#) (AWS Tools for Windows PowerShell)。インスタンスを AWS OpsWorks で管理する場合、インスタン

スの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。

[Instance store-backed インスタンス] インスタンスを停止して属性を変更することはできません。代わりに、この手順に進んでください: [Linux で拡張ネットワーキングを有効にするには \(Instance store-backed インスタンス\) \(p. 745\)](#)

8. ローカルコンピュータから、次のいずれかのコマンドを使用して拡張ネットワーキングの `enaSupport` 属性を有効化します。

- [modify-instance-attribute \(AWS CLI\)](#)

```
aws ec2 modify-instance-attribute --instance-id instance_id --ena-support
```

- [Edit-EC2InstanceAttribute \(Tools for Windows PowerShell\)](#)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -EnaSupport $true
```

9. (オプション) 「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」の説明に従って、インスタンスから AMI を作成します。AMI は、インスタンスから拡張ネットワーキング `enaSupport` 属性を継承します。このため、この AMI を使用することで、拡張ネットワーキングがデフォルトで有効になっている別のインスタンスを起動できます。

**Important**

インスタンスオペレーティングシステムに `/etc/udev/rules.d/70-persistent-net.rules` が含まれている場合には、AMI を作成する前にそれを削除する必要があります。このファイルには、元のインスタンスのイーサネットアダプターの MAC アドレスが保存されています。別のインスタンスがこのファイルを使用して起動した場合、オペレーティングシステムがそのデバイスを検出できなくなり、`eth0` が失敗して、起動に関する問題が発生することがあります。このファイルは次の起動サイクルで再び生成され、AMI から起動されるインスタンスごとに独自のバージョンが作成されます。

10. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを起動します。[start-instances \(AWS CLI\)](#)、[Start-EC2Instance \(AWS Tools for Windows PowerShell\)](#)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。
11. (オプション) インスタンスに接続し、モジュールがインストールされていることを確認します。

拡張ネットワーキングを有効にした後にインスタンスに接続できない場合、「[Elastic Network Adapter \(ENA\) のトラブルシューティング \(p. 757\)](#)」を参照してください。

### Linux で拡張ネットワーキングを有効にするには (Instance store-backed インスタンス)

インスタンスを停止するステップまで、前の手順に従います。「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」に記述されているように、新しい AMI を作成します。AMI を登録するときに拡張ネットワーキング属性を有効にしてください。

- [register-image \(AWS CLI\)](#)

```
aws ec2 register-image --ena-support ...
```

- [Register-EC2Image \(AWS Tools for Windows PowerShell\)](#)

```
Register-EC2Image -EnaSupport ...
```

## DKMS を使用した Ubuntu での拡張ネットワーキングの有効化

この方法は、テストおよびフィードバックのみを目的としています。本番稼働用デプロイによる使用を目的としていません。本番稼働デプロイについては、「[Ubuntu での拡張ネットワーキングの有効化 \(p. 742\)](#)」を参照してください。

### Important

DKMS を使用すると、サブスクリプションのサポート契約が無効になります。最新の利用可能なカーネルモジュールを実行するには、代替手段として、kmod 設定を使用します。

Ubuntu で ENA を使用した拡張ネットワーキングを有効にするには (EBS-backed インスタンス)

1. [Ubuntu での拡張ネットワーキングの有効化 \(p. 742\)](#) のステップ 1 および 2 を行います。
2. build-essential パッケージをインストールしてカーネルモジュールと dkms パッケージをコンパイルし、カーネルが更新されるたびに ena モジュールが再構築されるようにします。

```
ubuntu:~$ sudo apt-get install -y build-essential dkms
```

3. の GitHub からインスタンスで ena モジュールのソースのクローンを作成します。

```
ubuntu:~$ git clone https://github.com/amzn/amzn-drivers
```

4. amzn-drivers パッケージを /usr/src/ ディレクトリに移動して、カーネルの更新のたびに DKMS がこのパッケージを見つけて構築できるようにします。ソースコードのバージョン番号 (現在のバージョン番号はリリースノートにあります) をディレクトリ名に付加します。たとえば、バージョン 1.0.0 は以下のようになります。

```
ubuntu:~$ sudo mv amzn-drivers /usr/src/amzn-drivers-1.0.0
```

5. 以下の値を使用して DKMS 設定ファイルを作成し、ena のバージョンに置き換えます。

ファイルを作成します。

```
ubuntu:~$ sudo touch /usr/src/amzn-drivers-1.0.0/dkms.conf
```

ファイルを編集し、次の値を追加します。

```
ubuntu:~$ sudo vim /usr/src/amzn-drivers-1.0.0/dkms.conf
PACKAGE_NAME="ena"
PACKAGE_VERSION="1.0.0"
CLEAN="make -C kernel/linux/ena clean"
MAKE="make -C kernel/linux/ena/ BUILD_KERNEL=${kernelver}"
BUILT_MODULE_NAME[0]="ena"
BUILT_MODULE_LOCATION="kernel/linux/ena"
DEST_MODULE_LOCATION[0]="/updates"
DEST_MODULE_NAME[0]="ena"
AUTOINSTALL="yes"
```

6. DKMS を使用して、インスタンスで ena モジュールを追加、構築、インストールします。

DKMS にモジュールを追加します。

```
ubuntu:~$ sudo dkms add -m amzn-drivers -v 1.0.0
```

dkms コマンドを使用してモジュールを構築します。

```
ubuntu:~$ sudo dkms build -m amzn-drivers -v 1.0.0
```

dkms を使用してモジュールをインストールします。

```
ubuntu:~$ sudo dkms install -m amzn-drivers -v 1.0.0
```

7. 起動時に正しいモジュールがロードされるように、initramfs を再ビルトします。

```
ubuntu:~$ sudo update-initramfs -u -k all
```

8. [拡張ネットワーキングが有効化されているかどうかのテスト \(p. 739\)](#) から modinfo ena コマンドを使用して、ena モジュールがインストールされていることを確認します。

```
ubuntu:~$ modinfo ena
filename:      /lib/modules/3.13.0-74-generic/updates/dkms/ena.ko
version:       1.0.0
license:        GPL
description:   Elastic Network Adapter (ENA)
author:        Amazon.com, Inc. or its affiliates
srcversion:    9693C876C54CA64AE48F0CA
alias:         pci:v00001D0Fd0000EC21sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd0000EC20sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd00001EC2sv*sd*bc*sc*i*
alias:         pci:v00001D0Fd00000EC2sv*sd*bc*sc*i*
depends:
vermagic:     3.13.0-74-generic SMP mod_unload modversions
parm:          debug:Debug level (0=none,...,16=all) (int)
parm:          push_mode:Descriptor / header push mode
              (0=automatic,1=disable,3=enable)
              0 - Automatically choose according to device capability (default)
              1 - Don't push anything to device memory
              3 - Push descriptors and header buffer to device memory (int)
parm:          enable_wd:Enable keepalive watchdog (0=disable,1=enable,default=1)
              (int)
parm:          enable_missing_tx_detection:Enable missing Tx completions. (default=1)
              (int)
parm:          numa_node_override_array:Numa node override map
              (array of int)
parm:          numa_node_override:Enable/Disable numa node override (0=disable)
              (int)
```

9. Ubuntu での拡張ネットワーキングの有効化 ([p. 742](#)) のステップ 3 に進みます。

## トラブルシューティング

ENA アダプターのトラブルシューティングの詳細については、「[Elastic Network Adapter \(ENA\) のトラブルシューティング \(p. 757\)](#)」を参照してください。

## オペレーティングシステムの最適化

ネットワーキングが拡張されたインスタンスで最大のネットワークパフォーマンスを実現するには、デフォルトのオペレーティングシステムの設定を変更する必要がある場合があります。高いネットワークパフォーマンスを必要とするアプリケーションには、次の設定変更をお勧めします。

これらのオペレーティングシステムの最適化に加えて、ネットワークトラフィックの最大送信単位 (MTU) も考慮し、ワークロードとネットワークアーキテクチャーに応じて調整する必要があります。詳細については、「[EC2 インスタンスの最大ネットワーク送信単位 \(MTU\) \(p. 801\)](#)」を参照してください。

AWS では、99.9 パーセンタイルで 50us のクラスター プレイスマートグループで起動されたインスタンスと 200us のテール レイテンシーの間のラウンドトリップ レイテンシーを定期的に測定しています。アプリケーションで一貫して低レイテンシーが必要な場合、固定パフォーマンスの Nitro ベース インスタンスで最新バージョンの ENA ドライバーを使用することをお勧めします。

これらの手順は Amazon Linux 2 および Amazon Linux AMI に関する説明です。ただし、カーネルバージョン 3.9 以降のその他の Linux ディストリビューションにも使用できる可能性があります。詳細については、システム固有のドキュメントを参照してください。

拡張ネットワーキング用に Amazon Linux インスタンスを最適化するには

1. インスタンスのクロックソースを確認します。

```
cat /sys/devices/system/clocksource/clocksource0/current_clocksource
```

2. クロックソースが `xen` である場合、次のサブステップを実行します。それ以外の場合は [Step 3 \(p. 748\)](#) に進みます。

- a. GRUB 設定を編集し、カーネル起動オプションに `clocksource=tsc` と `xen_nopvspin=1` を追加します。
  - Amazon Linux 2 については、次に示すように `/etc/default/grub` ファイルを編集し、これらのオプションを `GRUB_CMDLINE_LINUX_DEFAULT` 行に追加します。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1 clocksource=tsc" GRUB_TIMEOUT=0
```

- Amazon Linux AMI については、次に示すように `/boot/grub/grub.conf` ファイルを編集し、これらのオプションを `kernel` 行に追加します。

```
kernel /boot/vmlinuz-4.14.62-65.117.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1 clocksource=tsc
```

- b. (Amazon Linux 2 のみ) GRUB 設定ファイルを再構築して、以下の変更を取得します。

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. インスタンスタイプが [EC2 インスタンスタイプのプロセッサのステート制御 \(p. 561\)](#) でサポート対象として表示されている場合は、低レイテンシーのシステムパフォーマンスを確保するために、システムがより深い C ステートを使用しないようにします。詳細については、「[深い C ステートの制限による高パフォーマンスと低レイテンシー \(p. 563\)](#)」を参照してください。

- a. GRUB 設定を編集し、カーネル起動オプションに `intel_idle.max_cstate=1` を追加します。
  - Amazon Linux 2 については、次に示すように `/etc/default/grub` ファイルを編集し、このオプションを `GRUB_CMDLINE_LINUX_DEFAULT` 行に追加します。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1 clocksource=tsc intel_idle.max_cstate=1" GRUB_TIMEOUT=0
```

- Amazon Linux AMI については、次に示すように `/boot/grub/grub.conf` ファイルを編集し、このオプションを `kernel` 行に追加します。

```
kernel /boot/vmlinuz-4.14.62-65.117.amzn1.x86_64 root=LABEL=/ console=tty1  
console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295 xen_nopvspin=1  
clocksource=tsc intel_idle.max_cstate=1
```

- b. (Amazon Linux 2 のみ) GRUB 設定ファイルを再構築して、以下の変更を取得します。

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

4. 高いパケットバッファの割り当てを維持するのに十分な予約済みカーネルメモリがあることを確認します（デフォルト値が小さすぎる可能性があります）。
- 任意のテキストエディタで（root として、または sudo を使用して）/etc/sysctl.conf ファイルを開きます。
  - インスタンスタイプに対応する予約済みカーネルメモリの値（KB 単位）で vm.min\_free\_kbytes 行をファイルに追加します。簡単に言うと、この値を使用可能なシステムメモリの 1~3% の範囲に設定し、アプリケーションのニーズに合わせてこの値を増減して調整します。

```
vm.min_free_kbytes = 1048576
```

- c. この設定を次のコマンドを使って適用します。

```
sudo sysctl -p
```

- d. 次のコマンドを使用して、設定が適用されたことを確認します。

```
sudo sysctl -a 2>&1 | grep min_free_kbytes
```

5. インスタンスを再起動して、新しい設定をロードします。

```
sudo reboot
```

6. (オプション) パケットの受信割り込みが異なる CPU に関連付けられ、すべてが同じ NUMA ノードに属するように手動で分散します。ただし、irqbalancer はグローバルに無効になっているため、この方法は慎重に使用してください。

#### Note

このステップでの設定変更は、再起動後には無効になります。

- a. smp\_affinity.sh というファイルを作成して、次のコードブロックを貼り付けます。

```
#!/bin/sh  
service irqbalance stop  
affinity_values=(00000001 00000002 00000004 00000008 00000010 00000020 00000040  
00000080)  
irqs=$(grep eth /proc/interrupts|awk '{print $1}'|cut -d : -f 1)  
irqLen=${#irqs[@]}  
for (( i=0; i<${irqLen}; i++ ));  
do  
    echo ${affinity_values[$i]} > /proc/irq/${irqs[$i]}/smp_affinity;  
    echo "IRQ ${irqs[$i]} =" $(cat /proc/irq/${irqs[$i]}/smp_affinity);  
done
```

- b. 次のコマンドを使用してスクリプトを実行します。

```
sudo bash ./smp_affinity.sh
```

7. (オプション) 受信 IRQ を処理する vCPU が過負荷になっている場合、またはアプリケーションネットワーク処理が CPU で要求している場合、受信パケットステアリング (RPS) によりネットワーク処理の一部を他のコアにオフロードすることができます。NUMA ノード間ロックを避けるため、RPS に使用するコアが同じ NUMA ノードに属していることを確認します。たとえば、パケット処理にコア 8~15 を使用する場合は、次のコマンドを使用します。

Note

このステップでの設定変更は、再起動後には無効になります。

```
for i in `seq 0 7`; do echo $(printf "0000,00000000,00000000,00000000,0000ff00") | sudo tee /sys/class/net/eth0/queues/rx-$i/rps_cpus; done
```

8. (オプション) 可能であれば、すべての処理を同じ NUMA ノードに保持します。

- a. numactl をインストールします。

```
sudo yum install -y numactl
```

- b. ネットワーク処理プログラムを実行する場合は、単一の NUMA ノードにバインドします。たとえば、次のコマンドでは、シェルスクリプト run.sh を NUMA ノード 0 にバインドします。

```
numactl --cpunodebind=0 --membind=0 run.sh
```

- c. ハイペースレッディングを有効化している場合、CPU コアごとに 1 つのハードウェアスレッドのみを使用するようにアプリケーションを設定できます。
  - lscpu コマンドを使用して、どの CPU コアが NUMA ノードにマップするかを表示できます。

```
lscpu | grep NUMA
```

出力:

```
NUMA node(s):      2
NUMA node0 CPU(s): 0-15,32-47
NUMA node1 CPU(s): 16-31,48-63
```

- 次のコマンドを使用して、どのハードウェアスレッドが物理的 CPU に属しているかを表示できます。

```
cat /sys/devices/system/cpu/cpu0/topology/thread_siblings_list
```

出力:

```
0,32
```

この例では、スレッド 0 と 32 が CPU 0 にマップされます。

- スレッド 32~47 (実際には、0~15 同様、同じ CPU のハードウェアスレッド) での実行を避けるには、次のコマンドを使用します。

```
numactl --physcpubind=+0-15 --membind=0 ./run.sh
```

- 異なるクラスのトラフィックには、複数の Elastic Network Interface を使用します。たとえば、バックエンドデータベースを使用するウェブサーバーを実行している場合、ウェブサーバーのフロントエンドに 1 つの Elastic Network Interface を使用し、データベース接続にもう 1 つを使用します。

## Linux インスタンスにおけるインテル 82599 VF インターフェイスを使用した拡張ネットワーキングの有効化

Amazon EC2 は Intel 82599 VF インターフェイスを通じて拡張ネットワーキング機能を提供しますが、この機能では Intel ixgbevf ドライバーを使用します。

### コンテンツ

- 要件 (p. 751)
- 拡張ネットワーキングが有効化されているかどうかのテスト (p. 752)
- Amazon Linux での拡張ネットワーキングの有効化 (p. 753)
- Ubuntu での拡張ネットワーキングの有効化 (p. 754)
- 他の Linux ディストリビューションでの拡張ネットワーキングの有効化 (p. 755)
- 接続性の問題のトラブルシューティング (p. 757)

## 要件

Intel 82599 VF インターフェイスを使用した拡張ネットワーキングを準備するには、次のようにインスタンスをセットアップします。

- 次のサポートされているインスタンスタイプから選択します: C3、C4、D2、I2、M4 (m4.16xlarge を除く)、および R3。
- Linux カーネルバージョン 2.6.32 以降を使用して、HVM AMI からインスタンスを起動します。最新の Amazon Linux HVM AMI では、拡張ネットワーキングに必要なモジュールがインストールされており、必要な属性も設定されています。したがって、拡張ネットワーキングがサポートされている、Amazon EBS-Backed インスタンスを最新の Amazon Linux HVM AMI を使用して起動した場合は、拡張ネットワーキングが既に有効化されています。

### Warning

拡張ネットワーキングは、HVM インスタンスでのみサポートされています。PV インスタンスで拡張ネットワーキングを有効にすると、このインスタンスに到達できなくなります。また、適切なモジュールまたはモジュールバージョンを使用せずにこの属性を設定すると、インスタンスにアクセスできなくなる場合があります。

- インスタンスがインターネットに接続されていることを確認します。
- 選択した任意のコンピュータ、できればローカルのデスクトップまたはノート PC に、AWS CLI または AWS Tools for Windows PowerShell をインストールして設定します。詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。拡張ネットワーキングは、Amazon EC2 コンソールから管理することはできません。
- 保持する必要がある重要なデータがインスタンスにある場合、インスタンスから AMI を作成してそのデータをバックアップする必要があります。srivNetSupport 属性を有効になるとともに、カーネルおよびカーネルモジュールを更新すると、互換性のないインスタンスがレンダリングされたり、オペレーティングシステムに接続できなくなったりする可能性があります。最近のバックアップがある場合は、これが発生してもデータは保持されます。

## 拡張ネットワーキングが有効化されているかどうかのテスト

`ixgbevf` モジュールがインスタンスにインストールされており、`sriovNetSupport` 属性が設定されている場合は、Intel 82599 VF インターフェイスを使用した拡張ネットワーキングが既に有効になっています。

### インスタンス属性 (`sriovNetSupport`)

インスタンスに拡張ネットワーキングの `sriovNetSupport` 属性が設定されているかどうかを確認するには、次のいずれかのコマンドを使用します。

- [describe-instance-attribute](#) (AWS CLI)

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

- [Get-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
Get-EC2InstanceAttribute -InstanceId instance_id -Attribute sriovNetSupport
```

属性が設定されていない場合、`SriovNetSupport` は空です。属性が設定されている場合、以下の出力例に示すように、値は `simple` です。

```
"SriovNetSupport": {  
    "Value": "simple"  
},
```

### イメージ属性 (`sriovNetSupport`)

AMI に拡張ネットワーキングの `sriovNetSupport` 属性がすでに設定されているかどうかを確認するには、次のいずれかのコマンドを使用します。

- [describe-images](#) (AWS CLI)

```
aws ec2 describe-images --image-id ami_id --query "Images[].[SriovNetSupport]"
```

- [Get-EC2Image](#) (AWS Tools for Windows PowerShell)

```
(Get-EC2Image -ImageId ami_id).SriovNetSupport
```

属性が設定されていない場合、`SriovNetSupport` は空です。属性が設定されている場合、値は `simple` です。

### ネットワークインターフェイスドライバー

次のコマンドを使用して、モジュールが特定のインターフェイスで使用されていることを確認し、確認するインターフェイス名に置き換えます。単一のインターフェイス(デフォルト)を使用している場合は、`eth0` です。オペレーティングシステムで[予測可能なネットワーク名 \(p. 755\)](#)がサポートされている場合は、`ens5` のような名前にすることができます。

次の例で、リストされているドライバーは `vif` であるため、`ixgbevf` モジュールはロードされていません。

```
[ec2-user ~]$ ethtool -i eth0  
driver: vif  
version:  
firmware-version:
```

```
bus-info: vif-0
supports-statistics: yes
supports-test: no
supports-eeprom-access: no
supports-register-dump: no
supports-priv-flags: no
```

この例では、`ixgbevf` モジュールがロードされます。このインスタンスでは、拡張ネットワーキングが適切に設定されています。

```
[ec2-user ~]$ ethtool -i eth0
driver: ixgbevf
version: 4.0.3
firmware-version: N/A
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

## Amazon Linux での拡張ネットワーキングの有効化

最新の Amazon Linux HVM AMI では、拡張ネットワーキングに必要な `ixgbevf` モジュールがインストールされており、必要な `sriovNetSupport` 属性も設定されています。したがって、最新の Amazon Linux HVM AMI を使用してインスタンスタイプを起動した場合は、拡張ネットワーキングが既にインスタンスに対して有効になっています。詳細については、「[拡張ネットワーキングが有効化されているかどうかのテスト \(p. 752\)](#)」を参照してください。

以前の Amazon Linux AMI を使用してインスタンスを起動し、まだ拡張ネットワーキングが有効になっていない場合、拡張ネットワーキングを有効にするには次の手順を実行します。

### Warning

拡張ネットワーキング属性は、いったん有効にすると無効にする方法はありません。

### 拡張ネットワーキングを有効にするには

1. インスタンスに接続します。
2. インスタンスから、次のコマンドを実行して、`ixgbevf` を含む最新のカーネルとカーネルモジュールでインスタンスを更新します。

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

3. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを再起動します。[reboot-instances \(AWS CLI\)](#)、[Restart-EC2Instance \(AWS Tools for Windows PowerShell\)](#)。
4. インスタンスに再接続し、「[拡張ネットワーキングが有効化されているかどうかのテスト \(p. 752\)](#)」の `modinfo ixgbevf` コマンドを使用して、`ixgbevf` モジュールがインストールされ、最小推奨バージョンであることを確認します。
5. [EBS-backed インスタンス] ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを停止します。[stop-instances \(AWS CLI\)](#)、[Stop-EC2Instance \(AWS Tools for Windows PowerShell\)](#)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。

[Instance store-backed インスタンス] インスタンスを停止して属性を変更することはできません。代わりに、この手順に進んでください：[拡張ネットワーキングを有効にするには \(Instance store-backed インスタンス\) \(p. 754\)](#)。

6. ローカルコンピュータから、次のいずれかのコマンドを使用して拡張ネットワーキングの属性を有効化します。

- [modify-instance-attribute](#) (AWS CLI)

```
aws ec2 modify-instance-attribute --instance-id instance_id --srivnet-support simple
```

- [Edit-EC2InstanceAttribute](#) (AWS Tools for Windows PowerShell)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```

7. (オプション) 「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」の説明に従って、インスタンスから AMI を作成します。AMI は、インスタンスから拡張ネットワーキング属性を継承します。このため、この AMI を使用することで、拡張ネットワーキングがデフォルトで有効になっている別のインスタンスを起動できます。
8. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを起動します。[start-instances](#) (AWS CLI)、[Start-EC2Instance](#) (AWS Tools for Windows PowerShell)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。
9. インスタンスに接続し、[拡張ネットワーキングが有効化されているかどうかのテスト \(p. 752\)](#) の ethtool -i ethn コマンドを使用して、ixgbevf モジュールがインストールされ、ネットワークインターフェイスにロードされていることを確認します。

拡張ネットワーキングを有効にするには (Instance store-backed インスタンス)

インスタンスを停止するステップまで、前の手順に従います。「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」に記述されているように、新しい AMI を作成します。AMI を登録するときに拡張ネットワーキング属性を有効にしてください。

- [register-image](#) (AWS CLI)

```
aws ec2 register-image --srivnet-support simple ...
```

- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

## Ubuntu での拡張ネットワーキングの有効化

開始する前に、インスタンスで[拡張ネットワーキングがすでに有効になっているかどうかを確認 \(p. 752\)](#)します。

クイックスタート Ubuntu HVM AMI には、拡張ネットワーキングに必要なドライバーが搭載されています。ixgbevf 2.16.4 より前のバージョンを使用している場合は、linux-aws カーネルパッケージをインストールして最新の拡張ネットワーキングドライバーを取得できます。

以下の手順は、Ubuntu インスタンスで ixgbevf モジュールをコンパイルするための一般的なステップを示しています。

linux-aws カーネルパッケージをインストールするには

1. インスタンスに接続します。
2. パッケージキャッシュおよびパッケージを更新します。

```
ubuntu:~$ sudo apt-get update && sudo apt-get upgrade -y linux-aws
```

#### Important

更新プロセス中に grub をインストールするよう求められた場合は、grub のインストール先として /dev/xvda を使用し、現在のバージョンの /boot/grub/menu.lst を保持することを選択します。

## 他の Linux ディストリビューションでの拡張ネットワーキングの有効化

開始する前に、インスタンスで [拡張ネットワーキングがすでに有効になっているかどうかを確認 \(p. 752\)](#) します。最新のクイックスタート HVM AMI には、拡張ネットワーキングに必要なドライバーが含まれているため、追加ステップを実行する必要はありません。

次の手順では、Amazon Linux または Ubuntu 以外の Linux ディストリビューションで Intel 82599 VF インターフェイスを使用した拡張ネットワーキングを有効にする必要がある場合の一般的なステップを説明します。コマンドの詳細な構文、ファイルの場所、パッケージやツールのサポートなどの詳細については、使用する Linux ディストリビューションのドキュメントを参照してください。

Linux で拡張ネットワーキングを有効化するには

1. インスタンスに接続します。
2. Sourceforge (<https://sourceforge.net/projects/e1000/files/ixgbefv%20stable/>) からインスタンスに ixgbefv モジュールのソースをダウンロードします。  
ixgbefv の 2.16.4 より前のバージョン (バージョン 2.14.2 を含む) は、一部の Linux ディストリビューション (特定のバージョンの Ubuntu など) では適切にビルドされません。
3. インスタンスで ixgbefv モジュールをコンパイルし、インストールします。

#### Warning

現在のカーネルに ixgbefv モジュールをコンパイルし、新しいカーネルをドライバを再構築しないで更新すると、システムは次回の再起動の際にディストリビューション固有の ixgbefv モジュールに戻る場合があります。これにより、ディストリビューション固有のバージョンが拡張ネットワーキングと互換性がない場合に、システムに接続できなくなります。

4. sudo depmod コマンドを実行して、モジュールの依存関係を更新します。
5. 起動時に新しいモジュールがロードされるように、インスタンスの initramfs を更新します。
6. システムがデフォルトで予測可能なネットワークインターフェイス名を使用するかどうかを確認します。systemd または udev のバージョン 197 以上を使用するシステムの場合、イーサネットデバイスの名前を変更でき、単一ネットワークインターフェイスの名前が eth0 になることは保証されません。この動作は、インスタンスに接続する際に問題の原因となる可能性があります。詳細と他の設定オプションについては、freedesktop.org ウェブサイトで「[Predictable Network Interface Names/](#)」を参照してください。
  - a. 次のコマンドを使用して、RPM ベースのシステムで systemd または udev のバージョンを確認できます。

```
[ec2-user ~]$ rpm -qa | grep -e '^systemd-[0-9]+\+|udev-[0-9]+\+'  
systemd-208-11.el7_0.2.x86_64
```

上記の Red Hat Enterprise Linux 7 の例では、systemd のバージョンは 208 であるため、予測可能なネットワークインターフェイス名は無効になっている必要があります。

- b. `net.ifnames=0` オプションを `GRUB_CMDLINE_LINUX` の `/etc/default/grub` 行に追加することによって、予測可能なネットワークインターフェイス名を無効にします。

```
[ec2-user ~]$ sudo sed -i '/^GRUB_CMDLINE_LINUX/s/"$/ net.ifnames=0"/' /etc/default/grub
```

- c. `grub` の設定ファイルを再ビルトします。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

7. [EBS-backed インスタンス] ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを停止します。[stop-instances \(AWS CLI\)](#)、[Stop-EC2Instance \(AWS Tools for Windows PowerShell\)](#)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。

[Instance store-backed インスタンス] インスタンスを停止して属性を変更することはできません。代わりに、この手順に進んでください: [拡張ネットワーキングを有効にするには \(Instance store-backed インスタンス\) \(p. 757\)](#)。

8. ローカルコンピュータから、次のいずれかのコマンドを使用して拡張ネットワーキングの属性を有効化します。

- [modify-instance-attribute \(AWS CLI\)](#)

```
aws ec2 modify-instance-attribute --instance-id instance_id --srivnet-support simple
```

- [Edit-EC2InstanceAttribute \(AWS Tools for Windows PowerShell\)](#)

```
Edit-EC2InstanceAttribute -InstanceId instance_id -SriovNetSupport "simple"
```

9. (オプション) 「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」の説明に従って、インスタンスから AMI を作成します。AMI は、インスタンスから拡張ネットワーキング属性を継承します。このため、この AMI を使用することで、拡張ネットワーキングがデフォルトで有効になっている別のインスタンスを起動できます。

#### Important

インスタンスオペレーティングシステムに `/etc/udev/rules.d/70-persistent-net.rules` が含まれている場合には、AMI を作成する前にそれを削除する必要があります。このファイルには、元のインスタンスのイーサネットアダプターの MAC アドレスが保存されています。別のインスタンスがこのファイルを使用して起動した場合、オペレーティングシステムがそのデバイスを検出できなくなり、`eth0` が失敗して、起動に関する問題が発生することがあります。このファイルは次の起動サイクルで再び生成され、AMI から起動されるインスタンスごとに独自のバージョンが作成されます。

10. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを起動します。[start-instances \(AWS CLI\)](#)、[Start-EC2Instance \(AWS Tools for Windows PowerShell\)](#)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。
11. (オプション) インスタンスに接続し、モジュールがインストールされていることを確認します。

#### 拡張ネットワーキングを有効にするには (Instance store-backed- インスタンス)

インスタンスを停止するステップまで、前の手順に従います。「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」に記述されているように、新しい AMI を作成します。AMI を登録するときに拡張ネットワーキング属性を有効にしてください。

- register-image (AWS CLI)

```
aws ec2 register-image --srivnet-support simple ...
```

- Register-EC2Image (AWS Tools for Windows PowerShell)

```
Register-EC2Image -SriovNetSupport "simple" ...
```

## 接続性の問題のトラブルシューティング

拡張ネットワーキングを有効化しているときに接続が失われると、`ixgbevf` モジュールとカーネルの互換性が保たれない可能性があります。この場合、インスタンスの Linux ディストリビューションに含まれる `ixgbevf` モジュールのバージョンをインストールしてみます。

PV インスタンスまたは AMI で拡張ネットワーキングを有効にすると、お使いのインスタンスに到達できなくなります。

詳細については、「[EC2 で拡張ネットワーキングを有効化および設定する方法](#)」を参照してください。

## Elastic Network Adapter (ENA) のトラブルシューティング

Elastic Network Adapter (ENA) は、オペレーティングシステムのヘルスを向上し、予期しないハードウェア動作や障害による長期的な停止の可能性を減らすように設計されています。ENA アーキテクチャでは、デバイスやドライバーの障害がシステムに対して可能な限り透過的に保持されます。このトピックでは、ENA のトラブルシューティングについて説明します。

インスタンスに接続できない場合は、まず「[接続性の問題のトラブルシューティング \(p. 757\)](#)」セクションを参照してください。

インスタンスに接続できる場合、このトピックの以降のセクションに記載されている障害検出/復旧メカニズムを使用して診断情報を収集することができます。

### コンテンツ

- [接続性の問題のトラブルシューティング \(p. 757\)](#)
- [キープアライブメカニズム \(p. 758\)](#)
- [読み取りタイムアウトの登録 \(p. 759\)](#)
- [統計 \(p. 760\)](#)
- [syslog のドライバーログ \(p. 762\)](#)

## 接続性の問題のトラブルシューティング

拡張ネットワーキングを有効化しているときに接続が失われると、`ena` モジュールとインスタンスの現在実行中のカーネルの互換性が保たれない可能性があります。これは、特定のカーネルバージョンのモジュールをインストール (`dkms` を使用しないか、不適切な設定の `dkms.conf` ファイルを使用) したため、インスタンスカーネルが更新された場合に発生します。起動時にロードされるインスタンスカーネルにより、`ena` モジュールが正しくインストールされない場合、インスタンスがネットワークアダプタを認識せず、インスタンスが到達不可能になります。

PV インスタンスまたは AMI で拡張ネットワーキングを有効にすると、お使いのインスタンスにも到達できなくなります。

ENA を使用して拡張ネットワーキングを有効した後インスタンスが到達不可能になった場合、インスタンスの `enaSupport` 属性を無効にすると、ストックネットワークアダプタにフォールバックできます。

## ENA を使用して拡張ネットワーキングを無効にするには (EBS-backed インスタンス)

1. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを停止します。[stop-instances](#) (AWS CLI)、[Stop-EC2Instance](#) (AWS Tools for Windows PowerShell)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。

### Important

instance store-backed インスタンスを使用している場合、インスタンスを停止することはできません。代わりに、「[ENA を使用して拡張ネットワーキングを無効にするには \(Instance store-backed インスタンス\) \(p. 758\)](#)」に進みます。

2. ローカルコンピュータから、次のコマンドを使用して拡張ネットワーキングの属性を無効化します。
  - [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --no-ena-support
```

3. ローカルコンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを起動します。[start-instances](#) (AWS CLI)、[Start-EC2Instance](#) (AWS Tools for Windows PowerShell)。インスタンスを AWS OpsWorks で管理する場合、インスタンスの状態が同期し続けるために、AWS OpsWorks コンソールでインスタンスを停止する必要があります。
4. (オプション) インスタンスに接続し、「ena」のステップに従って、現在のカーネルバージョンを使用して [Linux インスタンスにおける Elastic Network Adapter \(ENA\) を使用した拡張ネットワーキングの有効化 \(p. 738\)](#) モジュールの再インストールを試みます。

## ENA を使用して拡張ネットワーキングを無効にするには (Instance store-backed インスタンス)

インスタンスが instance store-backed インスタンスの場合、「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」の説明に従って新しい AMI を作成します。AMI を登録するときに、必ず拡張ネットワーキング enaSupport 属性を無効化してください。

- [register-image](#) (AWS CLI)
- ```
$ aws ec2 register-image --no-ena-support ...
```
- [Register-EC2Image](#) (AWS Tools for Windows PowerShell)

```
C:\> Register-EC2Image -EnaSupport $false ...
```

## キープアライブメカニズム

ENA デバイスは、キープアライブイベントを一定の速度 (通常は 1 秒に 1 回) で送信します。ENA ドライバーは、これらのキープアライブメッセージの存在を確認するウォッチドッグメカニズムを実装します。メッセージが存在する場合、ウォッチドッグが再実装されます。存在しない場合、ドライバーはデバイスで障害が発生したと判断し、次の処理を行います。

- 現在の統計を syslog にダンプする
- ENA デバイスをリセットする
- ENA のドライバー状態をリセットする

上記のリセット手順を実行すると、トラフィックが短時間失われる可能性がありますが (TCP 接続は回復可能です)、ユーザーに影響は及びません。

ENA デバイスは、キープアライブ通知を送信しないことによりデバイスリセット手順を間接的にリクエストすることができます。たとえば、ENA デバイスが回復不可能な設定をロードした後に不明な状態になった場合などです。

リセット手順の例を以下に示します。

```
[18509.800135] ena 0000:00:07.0 eth1: Keep alive watchdog timeout. // The watchdog process initiates a reset
[18509.815244] ena 0000:00:07.0 eth1: Trigger reset is on
[18509.825589] ena 0000:00:07.0 eth1: tx_timeout: 0 // The driver logs the current statistics
[18509.834253] ena 0000:00:07.0 eth1: io_suspend: 0
[18509.842674] ena 0000:00:07.0 eth1: io_resume: 0
[18509.850275] ena 0000:00:07.0 eth1: wd_expired: 1
[18509.857855] ena 0000:00:07.0 eth1: interface_up: 1
[18509.865415] ena 0000:00:07.0 eth1: interface_down: 0
[18509.873468] ena 0000:00:07.0 eth1: admin_q_pause: 0
[18509.881075] ena 0000:00:07.0 eth1: queue_0_tx_cnt: 0
[18509.888629] ena 0000:00:07.0 eth1: queue_0_tx_bytes: 0
[18509.895286] ena 0000:00:07.0 eth1: queue_0_tx_queue_stop: 0
.....
[18511.280972] ena 0000:00:07.0 eth1: free uncompleted tx skb qid 3 idx 0x7 // At the end of the down process, the driver discards incomplete packets.
[18511.420112] [ENA_COM: ena_com_validate_version] ena device version: 0.10 //The driver begins its up process
[18511.420119] [ENA_COM: ena_com_validate_version] ena controller version: 0.0.1 implementation version 1
[18511.420127] [ENA_COM: ena_com_admin_init] ena_defs : Version:[b9692e8] Build date [Wed Apr 6 09:54:21 IDT 2016]
[18512.252108] ena 0000:00:07.0: Device watchdog is Enabled
[18512.674877] ena 0000:00:07.0: irq 46 for MSI/MSI-X
[18512.674933] ena 0000:00:07.0: irq 47 for MSI/MSI-X
[18512.674990] ena 0000:00:07.0: irq 48 for MSI/MSI-X
[18512.675037] ena 0000:00:07.0: irq 49 for MSI/MSI-X
[18512.675085] ena 0000:00:07.0: irq 50 for MSI/MSI-X
[18512.675141] ena 0000:00:07.0: irq 51 for MSI/MSI-X
[18512.675188] ena 0000:00:07.0: irq 52 for MSI/MSI-X
[18512.675233] ena 0000:00:07.0: irq 53 for MSI/MSI-X
[18512.675279] ena 0000:00:07.0: irq 54 for MSI/MSI-X
[18512.772641] [ENA_COM: ena_com_set_hash_function] Feature 10 isn't supported
[18512.772647] [ENA_COM: ena_com_set_hash_ctrl] Feature 18 isn't supported
[18512.775945] ena 0000:00:07.0: Device reset completed successfully // The reset process is complete
```

## 読み取りタイムアウトの登録

ENA アーキテクチャでは、Memory Mapped I/O (MMIO) の読み取りオペレーションの限定的に使用することが推奨されます。MMIO レジスタには、初期化手順中のみ ENA デバイスドライバーがアクセスします。

ドライバーログ (dmesg 出力にあります) が読み取りオペレーションの失敗を示している場合、互換性のないドライバーまたは適切にコンパイルされていないドライバー、ビジー状態のハードウェアドライバー、ハードウェア障害が原因の可能性があります。

読み取りオペレーションの失敗を示すログエントリが断続的に発生する場合は、問題とみなさないでください。この場合はドライバーによって再試行されます。ただし、読み取りの失敗を含むログエントリが連続して発生する場合は、ドライバーまたはハードウェアの問題を示しています。

タイムアウトによる読み取りオペレーション失敗を示すドライバーログエントリの例を以下に示します。

```
[ 47.113698] [ENA_COM: ena_com_reg_bar_read32] reading reg failed for timeout. expected: req id[1] offset[88] actual: req id[57006] offset[0]
```

```
[ 47.333715] [ENACOM: ena_com_reg_bar_read32] reading reg failed for timeout. expected:  
req id[2] offset[8] actual: req id[57007] offset[0]  
[ 47.346221] [ENACOM: ena_com_dev_reset] Reg read32 timeout occurred
```

## 統計

ネットワークパフォーマンスが不十分な場合やレイテンシーの問題がある場合、デバイス統計情報を取得して調査する必要があります。これらの統計は、以下に示すように ethtool を使用して取得できます。

```
[ec2-user ~]$ ethtool -S ethN  
NIC statistics:  
    tx_timeout: 0  
    io_suspend: 0  
    io_resume: 0  
    wd_expired: 0  
    interface_up: 1  
    interface_down: 0  
    admin_q_pause: 0  
    queue_0_tx_cnt: 4329  
    queue_0_tx_bytes: 1075749  
    queue_0_tx_queue_stop: 0  
...
```

次のコマンド出力パラメータの説明を以下に示します。

`tx_timeout: N`

Netdev ウオッチドッグがアクティブになった回数。

`io_suspend: N`

サポートされていません。この値は、常に 0 にする必要があります。

`io_resume: N`

サポートされていません。この値は、常に 0 にする必要があります。

`wd_expired: N`

ドライバーが直近 3 秒以内にキープアライブイベントを受け取らなかつた回数。

`interface_up: N`

ENA インターフェイスが起動された回数。

`interface_down: N`

ENA インターフェイスが停止された回数。

`admin_q_pause: N`

管理者キューが不安定な状態です。この値は、常に 0 にする必要があります。

`queue_N_tx_cnt: N`

キュー N の送信されたパケットの数。

`queue_N_tx_bytes: N`

キュー N の送信バイト数。

`queue_N_tx_queue_stop: N`

キュー N がいっぱいになって停止された回数。

`queue_N_tx_queue_wakeup: N`

停止後にキュー N が再開された回数。

queue\_N\_tx\_dma\_mapping\_err: N

直接メモリアクセスエラーの数。この値が 0 の場合は、システムリソースが低いことを示しています。

queue\_N\_tx\_napi\_comp: N

napi ハンドラがキュー N の napi\_complete を呼び出した回数。

queue\_N\_tx\_poll: N

napi ハンドラがキュー N にスケジュールされた回数。

queue\_N\_tx\_doorbells: N

キュー N の送信ドアベルの数。

queue\_N\_tx\_linearize: N

キュー N に SKB 線形化が試行された回数。

queue\_N\_tx\_linearize\_failed: N

キュー N の SKB 線形化が失敗した回数。

queue\_N\_tx\_prepare\_ctx\_err: N

キュー N の ena\_com\_prepare\_tx が失敗した回数。この値は、常に 0 になる必要があります。そうでない場合はドライバーログを参照してください。

queue\_N\_tx\_missing\_tx\_comp: codeN

キュー N の未処理のパケットの数。この値は、常に 0 にする必要があります。

queue\_N\_tx\_bad\_req\_id: N

キュー N の無効な req\_id。有効な req\_id は 0、マイナス queue\_size、マイナス 1 です。

queue\_N\_rx\_cnt: N

キュー N の受信されたパケットの数。

queue\_N\_rx\_bytes: N

キュー N の受信バイト数。

queue\_N\_rx\_refil\_partial: N

ドライバーが rx キューの空いている部分にキュー N のバッファーを補充できなかった回数。この値が 0 でない場合、メモリリソースが低いことを示しています。

queue\_N\_rx\_bad\_csum: N

rx キューに、キュー N の不良なチェックサムがあった回数(rx チェックサムオフロードがサポートされている場合のみ)。

queue\_N\_rx\_page\_alloc\_fail: N

キュー N のページ割り当てに失敗した回数。この値が 0 でない場合、メモリリソースが低いことを示しています。

queue\_N\_rx\_skb\_alloc\_fail: N

キュー N の SKB 割り当てに失敗した回数。この値が 0 でない場合、システムリソースが低いことを示しています。

queue\_N\_rx\_dma\_mapping\_err: N

直接メモリアクセスエラーの数。この値が 0 の場合は、システムリソースが低いことを示しています。

queue\_ **N**\_rx\_bad\_desc\_num: **N**

パケットあたりのバッファーが多すぎます。この値が 0 でない場合、バッファーの使用量が非常に少ないことを示しています。

queue\_ **N**\_rx\_small\_copy\_len\_pkt: **N**

最適化: パケットがこのしきい値 (sysfs により設定) より小さい場合、新しいページの割り当てを避けるため、パケットはスタックに直接コピーされます。

ena\_admin\_q\_aborted\_cmd: **N**

中断された管理コマンドの数。これは、通常自動リカバリ手順中に発生します。

ena\_admin\_q\_submitted\_cmd: **N**

管理者キューのドアベルの数。

ena\_admin\_q\_completed\_cmd: **N**

管理者キューの完了数。

ena\_admin\_q\_out\_of\_space: **N**

ドライバーが新しい管理コマンドの送信を試みたが、キューがいっぱいであった回数。

ena\_admin\_q\_no\_completion: **N**

ドライバーが管理コマンドの完了を取得しなかった回数。

## syslog のドライバーエラーログ

ENA ドライバーは、システム起動時にログメッセージを syslog に書き込みます。問題が発生した場合、これらのログを調べてエラーを探すことができます。システム起動時に ENA ドライバーにより syslog に記録される情報の例と、特定のメッセージの注釈の一部を以下に示します。

```
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.416939] [ENA_COM: ena_com_validate_version]
ena device version: 0.10
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  478.420915] [ENA_COM: ena_com_validate_version]
ena controller version: 0.0.1 implementation version 1
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.256831] ena 0000:00:03.0: Device watchdog is
Enabled
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.672947] ena 0000:00:03.0: creating 8 io
queues. queue size: 1024
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.680885] [ENA_COM:
ena_com_init_interrupt_moderation] Feature 20 isn't supported // Interrupt moderation is
not supported by the device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.691609] [ENA_COM: ena_com_get_feature_ex]
Feature 10 isn't supported // RSS HASH function configuration is not supported by the
device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.694583] [ENA_COM: ena_com_get_feature_ex]
Feature 18 isn't supported // RSS HASH input source configuration is not supported by the
device
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.697433] [ENA_COM:
ena_com_set_host_attributes] Set host attribute isn't supported
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.701064] ena 0000:00:03.0 (unnamed
net_device) (uninitialized): Cannot set host attributes
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  479.704917] ena 0000:00:03.0: Elastic Network
Adapter (ENA) found at mem f3000000, mac addr 02:8a:3c:1e:13:b5 Queues 8
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  480.805037] EXT4-fs (xvda1): re-mounted. Opts:
(null)
Jun  3 22:37:46 ip-172-31-2-186 kernel: [  481.025842] NET: Registered protocol family 10
```

無視可能なエラー

システムのエラーログに記録される可能性がある以下のエラーは、Elastic Network Adapter では無視できます。

#### ホスト属性の設定がサポートされない

ホスト属性は、このデバイスではサポートされていません。

rx キューのバッファの割り当てに失敗した

これは復元可能なエラーであり、エラーがスローされたときにメモリプレッシャーの問題が発生した可能性があることを示します。

機能 **X** はサポートされていない

言及されている機能は、Elastic Network Adapter ではサポートされていません。**X** に指定できる値は、以下のとおりです。

- **10:** RSS ハッシュ関数設定は、このデバイスではサポートされていません。
- **12:** RSS 間接テーブル設定は、このデバイスではサポートされていません。
- **18:** RSS ハッシュ入力設定は、このデバイスではサポートされていません。
- **20:** 割り込みモデレーションは、このデバイスではサポートされていません。
- **27:** Elastic Network Adapter ドライバーは、snmpd からのイーサネット機能のポーリングをサポートしていません。

AENQ の設定に失敗した

Elastic Network Adapter では、AENQ 設定がサポートされていません。

サポートされていない AENQ のイベントを設定しようとしている

このエラーは、Elastic Network Adapter によりサポートされていない AENQ イベントグループを設定しようとしましたことを示しています。

## Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) は、ハイパフォーマンスコンピューティング (HPC) と機械学習アプリケーションを高速化するために Amazon EC2 インスタンスにアタッチできるネットワークデバイスです。EFA では、AWS クラウドが提供するスケーラビリティ、柔軟性、伸縮性により、オンプレミスの HPC クラスターのアプリケーションパフォーマンスを実現できます。

EFA では、クラウドベースの HPC システムで従来使用されていた TCP トランスポートよりも低く、一貫性の高いレイテンシーを提供し、高いスループットが得られます。HPC と機械学習アプリケーションのスケーリングに不可欠なインスタンス間通信のパフォーマンスが向上します。既存の AWS ネットワークインターフェーストラクチャで動作するように最適化されており、アプリケーション要件に応じてスケーリングすることができます。

EFA は、Libfabric 1.9.0 と統合されており、HPC アプリケーション向けに Open MPI 4.0.2 および Intel MPI 2019 Update 6、機械学習アプリケーション向けに Nvidia Collective Communications Library (NCCL) をサポートしています。

#### Note

EFA の OS バイパス機能は、Windows インスタンスではサポートされていません。EFA を Windows インスタンスにアタッチした場合、インスタンスは、Elastic Network Adapter として動作し、EFA 機能は追加されません。

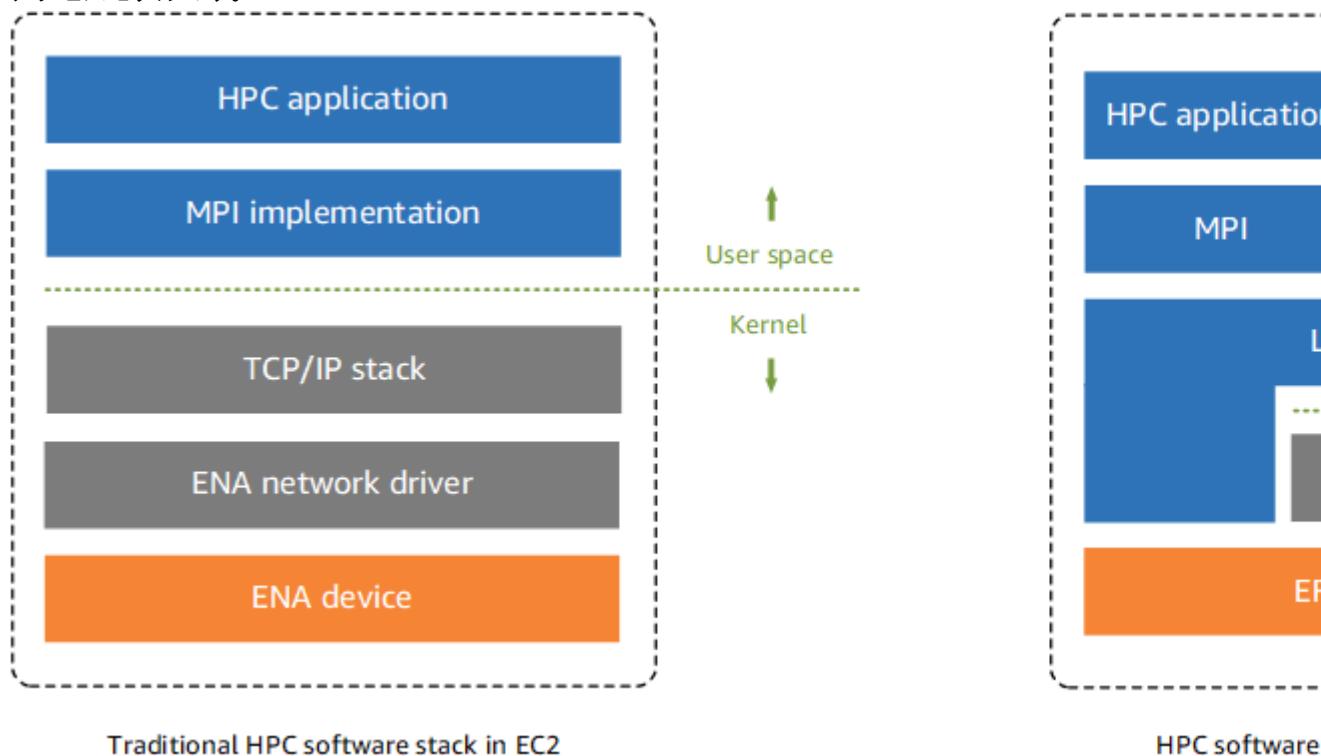
#### コンテンツ

- [EFA の基本 \(p. 764\)](#)
- [サポートされたインターフェイスとライブラリ \(p. 765\)](#)
- [サポートされるインスタンスタイプ \(p. 765\)](#)

- サポート対象の AMI (p. 765)
- EFA の制限事項 (p. 765)
- EFA および MPI の開始方法 (p. 765)
- EFA および NCCL の開始方法 (p. 772)
- EFA の使用 (p. 788)
- EFA のモニタリング (p. 791)

## EFA の基本

EFA は、機能が追加された Elastic Network Adapter (ENA) です。ENA のすべての機能に OS バイパス機能が追加されています。OS バイパスは、HPC と機械学習アプリケーションがネットワークインターフェイスハードウェアと直接通信して、レイテンシーが低く、信頼性の高い転送機能を実現できるようにするアクセスモデルです。



従来、HPC アプリケーションは、Message Passing Interface (MPI) を使用してシステムのネットワーク転送と通信していました。AWS クラウドでは、アプリケーションが MPI と通信することを意味します。MPI はオペレーティングシステムの TCP/IP スタックと ENA デバイスドライバーを使用して、インスタンス間のネットワーク通信を行います。

EFA の場合、HPC アプリケーションは MPI または NCCL を使用して Libfabric API と連携します。Libfabric API はオペレーティングシステムのカーネルをバイパスし、EFA デバイスと直接通信してパケットをネットワークに送ります。これにより、オーバーヘッドが削減され、HPC アプリケーションを効率的に実行できるようになります。

### Note

Libfabric は、OpenFabrics Interface (OFI) フレームワークのコアコンポーネントで、OFI のユーザースペース API を定義およびエクスポートします。詳細については、「[Libfabric OpenFabrics](#)」ウェブサイトを参照してください。

## EFAs と ENA の違い

Elastic Network Adapters (ENA) は、VPC ネットワークのサポートに必要な従来の IP ネットワーキング機能を提供します。EFAs は、ENA と同じ従来のすべての IP ネットワーキング機能に加えて、OS バイパス機能をサポートしています。OS バイパスにより、HPC と機械学習アプリケーションはオペレーティングシステムのカーネルをバイパスして EFA デバイスと直接通信できます。

## サポートされたインターフェイスとライブラリ

EFA は、以下のインターフェイスとライブラリをサポートしています。

- Open MPI 4.0.2
- Intel MPI 2019 Update 6
- NVIDIA Collective Communications Library (NCCL) 2.4.2 以降

## サポートされるインスタンスタイプ

次のインスタンスタイプは、EFAs: c5n.18xlarge, c5n.metal, i3en.24xlarge, i3en.metal, inf1.24xlarge, m5dn.24xlarge, m5n.24xlarge, r5dn.24xlarge, r5n.24xlarge, and p3dn.24xlarge をサポートします。

## サポート対象の AMI

次の AMI は、EFAs (Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04) をサポートします。

## EFA の制限事項

EFA には次の制約事項があります。

- インスタンスごとにアタッチできる EFA は 1 つのみです。
- EFA OS バイパストラフィックは、1 つのサブネットに制限されています。つまり、EFA トラフィックをサブネット間で送信することはできません。EFA の通常の IP トラフィックは、サブネット間で送信することができます。
- EFA OS バイパストラフィックは、ルーティングできません。EFA の通常の IP トラフィックは、引き続きルーティングできます。
- EFA は、セキュリティグループ自体との間のインバウンドおよびアウトバウンドのトラフィックをすべて許可するセキュリティグループのメンバーである必要があります。

## EFA および MPI の開始方法

本チュートリアルは、EFA と HPC ワークロードの MPI 対応インスタンスクラスターの起動に役立ちます。本チュートリアルでは、次の手順を実行します。

### コンテンツ

- [ステップ 1: EFA 対応のセキュリティグループを準備する \(p. 766\)](#)
- [ステップ 2: 一時インスタンスを起動する \(p. 766\)](#)
- [ステップ 3: EFA ソフトウェアをインストールする \(p. 767\)](#)
- [ステップ 4: Ptrace 保護を無効にする \(p. 768\)](#)
- [ステップ 5: \(オプション\) インテル MPI をインストールする \(p. 769\)](#)

- ステップ 6: HPC アプリケーションをインストールする (p. 770)
- ステップ 7: EFA 対応の AMI を作成する (p. 770)
- ステップ 8: クラスター プレイスマント グループで EFA 対応のインスタンスを起動する (p. 770)
- ステップ 9: 一時インスタンスを終了する (p. 771)
- ステップ 10: パスワードレス SSH を有効にする (p. 772)

## ステップ 1: EFA 対応のセキュリティ グループを準備する

EFA には、セキュリティ グループ自体とのインバウンドおよびアウトバウンドのトラフィックをすべて許可するセキュリティ グループが必要です。

EFA 対応のセキュリティ グループを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [セキュリティ グループ] を選択し、[セキュリティ グループの作成] を選択します。
3. [セキュリティ グループの作成] ウィンドウで以下を行います。
  - a. [セキュリティ グループ名] に、セキュリティ グループの分かりやすい名前 (例: EFA-enabled security group) を入力します。
  - b. (オプション) [説明] に、セキュリティ グループの簡単な説明を入力します。
  - c. [VPC] で、EFA 対応のインスタンスを起動する VPC を選択します。
  - d. [作成] を選択します。
4. 作成したセキュリティ グループを選択し、[説明] タブで [グループ ID] をコピーします。
5. [インバウンド] タブおよび [アウトバウンド] タブで、次の手順を実行します。
  - a. [Edit] を選択します。
  - b. [タイプ] で、[すべてのトラフィック] を選択します。
  - c. [ソース] で [カスタム] を選択します。
  - d. コピーしたセキュリティ グループ ID をフィールドに貼り付けます。
  - e. [Save] を選択します。

## ステップ 2: 一時インスタンスを起動する

EFA ソフトウェアコンポーネントのインストールおよび設定に使用する一時インスタンスを起動します。このインスタンスを使用して、EFA 対応のインスタンスを起動する EFA 対応の AMI を作成します。

一時インスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. [AMI の選択] ページで、サポートされている AMI (Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04) のいずれかを選択します。
4. [インスタンスタイプの選択] ページで、サポートされているインスタンスタイプ (c5n.18xlarge, c5n.metal, i3en.24xlarge, i3en.metal, inf1.24xlarge, m5dn.24xlarge, m5n.24xlarge, r5dn.24xlarge, r5n.24xlarge, and p3dn.24xlarge) のいずれかを選択し、[次の手順: インスタンスの詳細の設定] を選択します。
5. [Configure Instance Details] ページで以下の操作を実行します。
  - a. [Elastic Fabric Adapter] で、[有効化] を選択します。

- b. [ネットワークインターフェイス] セクションの [eth0] で、[新しいネットワークインターフェイス] を選択します。
- c. [次の手順: ストレージの追加] を選択します。
6. [Add Storage (ストレージの追加)] ページで、AMI で指定されたボリュームに加えてインスタンスにアタッチするボリューム (例: ルートデバイスボリューム) を指定します。次に、[次の手順: タグの追加] を選択します。
7. [タグの追加] ページで、一時インスタンスの識別に使用するタグを指定し、[Next: Configure Security Group (次へ: セキュリティグループの設定)] を選択します。
8. [Configure Security Group (セキュリティグループの設定)] ページの [Assign a security group (セキュリティグループの割り当て)] で、[Select an existing security group (既存のセキュリティグループの選択)] を選択し、ステップ 1 で作成したセキュリティグループを選択します。
9. [インスタンス作成の確認] ページで設定を確認し、[起動] を選択してキーペアを選択し、インスタンスを起動します。

## ステップ 3: EFA ソフトウェアをインストールする

EFA をサポートするのに必要な EFA 対応のカーネル、EFA ドライバ、Libfabric、Open MPI スタックを一時インスタンスにインストールします。

この手順は、EFA で Open MPI と Intel MPI のどちらを使用するかによって異なります。

EFA ソフトウェアをインストールするには

1. ステップ 2 で起動したインスタンスに接続します。詳細については、[Linux インスタンスへの接続 \(p. 505\)](#) を参照してください。
2. すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。このプロセスには数分かかることがあります。
  - Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7

```
$ sudo yum update -y
```

- Ubuntu 16.04 および Ubuntu 18.04

```
$ sudo apt-get update -y
```

```
$ sudo apt-get upgrade -y
```

3. EFA ソフトウェアのインストールファイルをダウンロードします。次のコマンドを使用して、安定している最新バージョンをダウンロードします。

```
$ curl -O https://s3-us-west-2.amazonaws.com/aws-efa-installer/aws-efa-installer-1.8.3.tar.gz
```

前述のコマンドのバージョン番号を `latest` に置き換えることで最新バージョンを取得することもできます。

4. ソフトウェアのインストールファイルは、圧縮された `.tar.gz` ファイルにパッケージ化されています。圧縮された `.tar.gz` ファイルからファイルを展開し、展開されたディレクトリに移動します。

```
$ tar -xf aws-efa-installer-1.8.3.tar.gz
```

```
$ cd aws-efa-installer
```

5. EFA ソフトウェアをインストールします。

- EFA で Open MPI を使用する場合は、Libfabric および Open MPI と共に EFA ソフトウェアをインストールする必要があります。また、「ステップ 4: Intel MPI のインストール」はスキップする必要があります。

Libfabric および Open MPI と共に EFA ソフトウェアをインストールするには、次のコマンドを実行します。

```
$ sudo ./efa_installer.sh -y
```

Libfabric は、/opt/amazon/efa ディレクトリにインストールされているのに対し、Open MPI は /opt/amazon/openmpi ディレクトリにインストールされています。

- EFA で Intel MPI のみを使用する場合は、Libfabric および Open MPI を使用せずに EFA ソフトウェアをインストールできます。この場合、Intel MPI は埋め込まれている Libfabric を使用します。これを選択した場合は、「ステップ 4: Intel MPI のインストール」を完了する必要があります。

Libfabric および Open MPI を使用せずに EFA ソフトウェアをインストールするには、次のコマンドを実行します。

```
$ sudo ./efa_installer.sh -y --minimal
```

6. インスタンスからログアウトしてからログインし直します。

7. EFA ソフトウェアコンポーネントが正常にインストールされたことを確認します。

```
$ fi_info -p efa
```

コマンドによって、Libfabric の EFA インターフェイスに関する情報が返ります。以下の例は、コマンド出力を示しています。

```
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-rdm
    version: 2.0
    type: FI_EP_RDM
    protocol: FI_PROTO_EFA
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgram
    version: 2.0
    type: FI_EP_DGRAM
    protocol: FI_PROTO_EFA
provider: efa;ofi_rxnd
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgram
    version: 1.0
    type: FI_EP_RDM
    protocol: FI_PROTO_RXD
```

## ステップ 4: Ptrace 保護を無効にする

HPC アプリケーションのパフォーマンスを向上させるために、Libfabric は、プロセスが同じインスタンスで実行されている場合、プロセス間通信にインスタンスのローカルメモリを使用します。

共有メモリ機能では、ptrace 保護ではサポートされない Cross-Memory Attach (CMA) が使用されます。Ubuntu など、ptrace 保護がデフォルトで有効になっている Linux ディストリビューションを使用して

いる場合は、無効にする必要があります。Linux ディストリビューションで ptrace 保護がデフォルトで有効になつてない場合は、このステップをスキップします。

ptrace 保護を無効にするには

次のいずれかを行ってください。

- テストのために ptrace 保護を一時的に無効にするには、次のコマンドを実行します。

```
$ sudo sysctl -w kernel.yama.ptrace_scope=0
```

- ptrace 保護を完全に無効にするには、kernel.yama.ptrace\_scope = 0 を /etc/sysctl.d/10-ptrace.conf に追加してインスタンスを再起動します。

## ステップ 5: (オプション) インテル MPI をインストールする

### Important

Open MPI を使用する場合は、このステップをスキップしてください。このステップは、Intel MPI を使用する場合にのみ実行します。

Intel MPI を使用するには、追加のインストールと環境変数設定が必要です。

### 前提条件

以下のステップは、sudo アクセス許可を持つユーザーが実行してください。

Intel MPI をインストールするには

1. Intel MPI のインストールファイルをダウンロードするには、[Intel Developer Zone ウェブサイト](#) を参照してください。

インストールファイルのダウンロードには登録が必要です。登録を済ませたら以下を行います。

- a. [製品] については [Intel MPI Library for Linux] を選択します。
  - b. [Version (バージョン)] で、[2019 Update 6]、[Full Product (フル製品)] の順に選択します。
2. インストールファイルは、圧縮された.tar.gz ファイルにパッケージ化されています。圧縮された.tar.gz ファイルからファイルを展開し、展開されたディレクトリに移動します。

```
$ tar -xf file_name.tgz
```

```
$ cd directory_name
```

3. お好みのテキストエディタを使用して silent.cfg を開きます。行 10 において、ACCEPT\_EULA=decline を ACCEPT\_EULA=accept に変更します。変更内容を保存してファイルを閉じます。
4. インストールスクリプトを実行します。

```
$ sudo ./install.sh -s silent.cfg
```

Intel MPI は、デフォルトでは /opt/intel/impi/ ディレクトリにインストールされます。

5. Intel MPI 環境変数がインスタンスの起動ごとに設定されるように、これらの環境変数を対応するシェル起動スクリプトに追加します。使用するシェルに応じて、以下のいずれかを実行します。
  - bash の場合は、次の環境変数を /home/**username**/.bashrc および /home/**username**/.bash\_profile に追加します。

```
source /opt/intel/compilers_and_libraries/linux/mpi/intel64/bin/mpivars.sh
```

- csh と tcsh の場合は、次の環境変数を `/home/username/.cshrc` に追加します。

```
source /opt/intel/compilers_and_libraries/linux/mpi/intel64/bin/mpivars.csh
```

6. インスタンスからログアウトしてからログインし直します。
7. 次のコマンドを実行して、Intel MPI が正しくインストールされていることを確認します。

```
$ which mpicc
```

返されたパスに `/opt/intel/` サブディレクトリが含まれていることを確認します。

#### Note

Intel MPI が不要になった場合は、シェル起動スクリプトから環境変数を削除してください。

## ステップ 6: HPC アプリケーションをインストールする

HPC アプリケーションを一時インスタンスにインストールします。インストール手順は、特定の HPC アプリケーションによって異なります。Linux インスタンスへのソフトウェアのインストールの詳細については、「[Linux インスタンスでのソフトウェアの管理](#)」を参照してください。

#### Note

インストール手順については、HPC アプリケーションのドキュメントの参照が必要になる場合があります。

## ステップ 7: EFA 対応の AMI を作成する

必要なソフトウェアコンポーネントをインストールしたら、EFA 対応のインスタンスの起動に再利用できる AMI を作成します。

一時インスタンスから AMI を作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. ステップ 1 で作成したインスタンスを作成し、[アクション]、[イメージ]、[イメージの作成] の順に選択します。
4. [イメージの作成] ウィンドウで、以下の操作を行います。
  - a. [イメージ名] に、AMI の分かりやすい名前を入力します。
  - b. (オプション) [イメージの説明] に、AMI の簡単な説明を入力します。
  - c. [イメージの説明]、[Close (閉じる)] の順に選択します。
5. ナビゲーションペインで [AMI] を選択します。
6. リストで作成した AMI を探します。ステータスが `pending` から `available` になるまで待ってから、次のステップに進みます。

## ステップ 8: クラスター プレイスマント グループで EFA 対応のインスタンスを起動する

ステップ 6 で作成した EFA 対応の AMI、および ステップ 1 で作成した EFA 対応のセキュリティ グループを使用して、EFA 対応のインスタンスをクラスターのプレイスマント グループに起動します。

#### Note

EFA 対応のインスタンスをクラスターのプレイスメントグループに起動することは絶対的な要件ではありません。ただし、EFA 対応インスタンスは、1 つのアベイラビリティーボーン内の低レイテンシーグループに起動されるため、クラスター プレイイスメントグループで実行することをお勧めします。

EFA 対応のインスタンスをクラスターのプレイスメントグループに起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. [AMI の選択] ページで、[マイ AMI] を選択し、ステップ 6 で作成した AMI を探して、[選択] を選択します。
4. [インスタンスタイプの選択] ページで、サポートされているインスタンスタイプ (c5n.18xlarge, c5n.metal, i3en.24xlarge, i3en.metal, inf1.24xlarge, m5dn.24xlarge, m5n.24xlarge, r5dn.24xlarge, r5n.24xlarge, and p3dn.24xlarge) のいずれかを選択し、[次の手順: インスタンスの詳細の設定] を選択します。
5. [Configure Instance Details] ページで以下の操作を実行します。
  - a. [インスタンス数] に、起動する EFA 対応のインスタンスの数を入力します。
  - b. [ネットワーク] および [サブネット] で、インスタンスを起動する VPC およびサブネットを選択します。
  - c. [プレイスメントグループ] で、[インスタンスをプレイスメントグループに追加します] チェックボックスをオンにします。
  - d. [プレイスメントグループ名] で、[新しいプレイスメントグループに追加します] チェックボックスをオンにし、プレイスメントグループの分かりやすい名前を入力して、[プレイスメントグループ戦略] で [クラスター] を選択します。
  - e. [EFA] で、[有効化] を選択します。
  - f. [ネットワークインターフェイス] セクションの [eth0] で、[新しいネットワークインターフェイス] を選択します。必要に応じて、プライマリ IPv4 アドレスと 1 つ以上のセカンダリ IPv4 アドレスを指定できます。関連付けられている IPv6 CIDR ブロックを持つサブネットにインスタンスを起動する場合は、必要に応じて、プライマリ IPv6 アドレスと 1 つ以上のセカンダリ IPv6 アドレスを指定することができます。
  - g. [次の手順: ストレージの追加] を選択します。
6. [ストレージの追加] ページで、AMI で指定されたボリュームに加えてインスタンスにアタッチするボリューム (例: ルートデバイスのボリューム) を指定し、[Next: Add Tags (次へ: タグの追加)] を選択します。
7. [Add Tags] ページで、ユーザーフレンドリーな名前などを使ってインスタンスのタグを指定し、[Next: Configure Security Group] を選択します。
8. [Configure Security Group (セキュリティ グループの設定)] ページの [Assign a security group (セキュリティ グループの割り当て)] で、[Select an existing security group (既存のセキュリティ グループの選択)] を選択し、ステップ 1 で作成したセキュリティ グループを選択します。
9. [Review and Launch] を選択します。
10. [インスタンス作成の確認] ページで設定を確認し、[起動] を選択してキーペアを選択し、インスタンスを起動します。

## ステップ 9: 一時インスタンスを終了する

この時点では、ステップ 1 で起動した一時インスタンスは不要になります。このインスタンスに対する料金が請求されないように、インスタンスを終了することができます。

### ステップ 7: 一時インスタンスを終了する

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. ステップ 1 で作成した一時インスタンスを選択し、[アクション]、[インスタンスの状態]、[終了]、[Yes, Terminate (はい、終了する)] の順に選択します。

## ステップ 10: パスワードレス SSH を有効にする

アプリケーションを有効にして、クラスターのすべてのインスタンス間で実行するには、リーダーノードからメンバーノードに対してパスワードレス SSH アクセスを有効にする必要があります。リーダーノードは、アプリケーションの実行元となるインスタンスです。クラスターのその他のインスタンスは、メンバーノードになります。

クラスターのインスタンス間でパスワードレス SSH を有効にするには

1. クラスターの 1 つのインスタンスをリーダーノードとして選択し、そのインスタンスに接続します。
2. リーダーノードで `strictHostKeyChecking` を無効にし、`ForwardAgent` を有効にします。適切なテキストエディタを使用して `~/.ssh/config` を開き、以下を追加します。

```
Host *
  ForwardAgent yes
Host *
  StrictHostKeyChecking no
```

3. RSA キーペアを生成します。

```
$ ssh-keygen -t rsa -N "" -f /home/ubuntu/.ssh/id_rsa
```

`$HOME/.ssh/` ディレクトリでキーペアが作成されます。

4. リーダーノードのプライベートキーの許可を変更します。

```
$ chmod 600 ~/.ssh/id_rsa
```

5. 適切なテキストエディタを使用して `~/.ssh/id_rsa.pub` を開き、キーをコピーします。
6. クラスターの各メンバーノードで、以下を実行します。
  - a. インスタンスに接続します。
  - b. 適切なテキストエディタを使用して `~/.ssh/authorized_keys` を開き、前にコピーしたパブリックキーを追加します。
7. パスワードレス SSH が予期したとおりに機能しているかテストするため、リーダーノードに接続し、次のコマンドを実行します。

```
$ ssh member_node_private_ip
```

キーまたはパスワードを求められることなく、メンバーノードに接続できるはずです。

## EFA および NCCL の開始方法

Nvidia Collective Communications Library (NCCL) は、単一のノードまたは複数のノードの複数の GPU のための集合的な標準コミュニケーションルーチンのライブラリです。NCCL は、各種の機械学習のワー

クロードをサポートするために、EFA、Libfabric、MPI と共に使用できます。詳細については、[NCCL のウェブサイト](#)を参照してください。

#### Note

- EFA を持つ NCCL は、p3dn.24xlarge インスタンスのみを使用してサポートされています。
- NCCL EFA 以降のみが 2.4.2 でサポートされています。

以下のチュートリアルは、機械学習のワークロードの EFA と NCCL 対応のインスタンスクラスターの起動に役立ちます。

- [基本 AMI の使用 \(p. 773\)](#)
- [AWS Deep Learning AMI の使用 \(p. 783\)](#)

## 基本 AMI の使用

次の手順は、基本 AMI (Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04) のいずれかを開始するのに役立ちます。

### 目次

- [ステップ 1: EFA 対応のセキュリティグループを準備する \(p. 773\)](#)
- [ステップ 2: 一時インスタンスを起動する \(p. 774\)](#)
- [ステップ 3: EFA ソフトウェアをインストールする \(p. 774\)](#)
- [ステップ 4: Nvidia GPU ドライバと Nvidia CUDA ツールキットをインストールする \(p. 776\)](#)
- [ステップ 5: NCCL をインストールする \(p. 777\)](#)
- [ステップ 6: aws-ofi-nccl プラグインをインストールする \(p. 778\)](#)
- [ステップ 7: NCCL テストをインストールする \(p. 779\)](#)
- [ステップ 8: EFA と NCCL 設定をテストする \(p. 779\)](#)
- [ステップ 9: 機械学習アプリケーションをインストールする \(p. 781\)](#)
- [ステップ 10: EFA と NCCL 対応 AMI を作成する \(p. 781\)](#)
- [ステップ 11: 一時インスタンスを終了する \(p. 781\)](#)
- [ステップ 12: クラスターのプレイスメントグループで EFA と NCCL 対応のインスタンスを起動する \(p. 781\)](#)
- [ステップ 13: パスワードレス SSH を有効にする \(p. 782\)](#)

## ステップ 1: EFA 対応のセキュリティグループを準備する

EFA には、セキュリティグループ自体とのインバウンドおよびアウトバウンドのトラフィックをすべて許可するセキュリティグループが必要です。

EFA 対応のセキュリティグループを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [セキュリティグループ] を選択し、[セキュリティグループの作成] を選択します。
3. [セキュリティグループの作成] ウィンドウで以下を行います。
  - a. [セキュリティグループ名] に、セキュリティグループの分かりやすい名前 (例: EFA-enabled security group) を入力します。

- b. (オプション) [説明] に、セキュリティグループの簡単な説明を入力します。
  - c. [VPC] で、EFA 対応のインスタンスを起動する VPC を選択します。
  - d. [作成] を選択します。
4. 作成したセキュリティグループを選択し、[説明] タブで [グループ ID] をコピーします。
  5. [インバウンド] タブおよび [アウトバウンド] タブで、次の手順を実行します。
    - a. [Edit] を選択します。
    - b. [タイプ] で、[すべてのトラフィック] を選択します。
    - c. [ソース] で [カスタム] を選択します。
    - d. コピーしたセキュリティグループ ID をフィールドに貼り付けます。
    - e. [Save] を選択します。

## ステップ 2: 一時インスタンスを起動する

EFA ソフトウェアコンポーネントのインストールおよび設定に使用する一時インスタンスを起動します。このインスタンスを使用して、EFA 対応のインスタンスを起動する EFA 対応の AMI を作成します。

### 一時インスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. [AMI の選択] ページで、AMI (Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04) のいずれかを選択します。
4. [インスタンスタイプの選択] ページで p3dn.24xlarge を選択してから、[次の手順: インスタンスの詳細の設定] を選択します。
5. [Configure Instance Details] ページで以下の操作を実行します。
  - a. [Elastic Fabric Adapter] で、[有効化] を選択します。
  - b. [ネットワークインターフェイス] セクションの [eth0] で、[新しいネットワークインターフェイス] を選択します。
  - c. [次の手順: ストレージの追加] を選択します。
6. [ストレージの追加] ページで、AMI で指定されたボリューム (ルートデバイスピリュームなど) に加えてインスタンスにアタッチするボリュームを指定します。次に、[次の手順: タグの追加] を選択します。
7. [タグの追加] ページで、一時インスタンスの識別に使用するタグを指定し、[Next: Configure Security Group (次へ: セキュリティグループの設定)] を選択します。
8. [セキュリティグループの設定] ページの [セキュリティグループの割り当て] で、[Select an existing security group (既存のセキュリティグループの選択)] を選択します。次に、ステップ 1 で作成したセキュリティグループを選択します。
9. [インスタンス作成の確認] ページで設定を確認し、[起動] を選択してキーペアを選択し、インスタンスを起動します。

## ステップ 3: EFA ソフトウェアをインストールする

EFA をサポートするのに必要な EFA 対応のカーネル、EFA ドライバ、Libfabric、Open MPI スタックを一時インスタンスにインストールします。

### EFA ソフトウェアをインストールするには

1. ステップ 2 で起動したインスタンスに接続します。詳細については、[Linux インスタンスへの接続 \(p. 505\)](#) を参照してください。

- すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。このプロセスには数分かかることがあります。

- Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7

```
$ sudo yum update -y
```

- Ubuntu 16.04 および Ubuntu 18.04

```
$ sudo apt-get update -y
```

```
$ sudo apt-get upgrade -y
```

- EFA ソフトウェアのインストールファイルをダウンロードします。次のコマンドを使用して、安定している最新バージョンをダウンロードします。

```
$ curl -O https://s3-us-west-2.amazonaws.com/aws-efa-installer/aws-efa-installer-1.8.3.tar.gz
```

前述のコマンドのバージョン番号を `latest` に置き換えることで最新バージョンを取得することもできます。

- ソフトウェアのインストールファイルは、圧縮された `.tar.gz` ファイルにパッケージ化されています。圧縮された `.tar.gz` ファイルからファイルを展開し、展開されたディレクトリに移動します。

```
$ tar -xf aws-efa-installer-1.8.3.tar.gz
```

```
$ cd aws-efa-installer
```

- EFA ソフトウェアのインストールスクリプトを実行します。

```
$ sudo ./efa_installer.sh -y
```

Libfabric は、`/opt/amazon/efa` ディレクトリにインストールされているのに対し、Open MPI は `/opt/amazon/openmpi` ディレクトリにインストールされています。

- インスタンスからログアウトしてからログインし直します。
- EFA ソフトウェアコンポーネントが正常にインストールされたことを確認します。

```
$ fi_info -p efa
```

コマンドによって、Libfabric の EFA インターフェイスに関する情報が返ります。以下の例は、コマンド出力を示しています。

```
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-rdm
    version: 2.0
    type: FI_EP_RDM
    protocol: FI_PROTO_EFA
provider: efa
    fabric: EFA-fe80::94:3dff:fe89:1b70
    domain: efa_0-dgram
    version: 2.0
    type: FI_EP_DGRAM
    protocol: FI_PROTO_EFA
```

```
provider: efa;ofi_rxd
  fabric: EFA-fe80::94:3dff:fe89:1b70
  domain: efa_0-dgrm
  version: 1.0
  type: FI_EP_RDM
  protocol: FI_PROTO_RXD
```

## ステップ 4: Nvidia GPU ドライバと Nvidia CUDA ツールキットをインストールする

Nvidia GPU ドライバと Nvidia CUDA ツールキットをインストールするには

1. Nvidia GPU ドライバと Nvidia CUDA ツールキットをインストールするために必要なユーティリティをインストールします。

- Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7

```
$ sudo yum groupinstall 'Development Tools' -y
```

- Ubuntu 16.04 および Ubuntu 18.04

```
$ sudo apt-get install build-essential -y
```

2. Nvidia GPU ドライバを使用するには、まず、nouveau オープンソースドライバを無効にする必要があります。

- a. 現在実行しているカーネルのバージョン用の gcc コンパイラおよびカーネルヘッダーパッケージをインストールします。

- Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7

```
$ sudo yum install -y gcc kernel-devel-$(uname -r)
```

- Ubuntu 16.04 および Ubuntu 18.04

```
$ sudo apt-get install -y gcc make linux-headers-$(uname -r)
```

- b. /etc/modprobe.d/blacklist.conf ブラックリストファイルに nouveau を追加します。

```
$ cat << EOF | sudo tee --append /etc/modprobe.d/blacklist.conf
blacklist vga16fb
blacklist nouveau
blacklist rivafb
blacklist nvidiafb
blacklist rivatv
EOF
```

- c. 任意のテキストエディタを使用して /etc/default/grub ファイルを開き、以下を追加します。

```
$ GRUB_CMDLINE_LINUX="rdblacklist=nouveau"
```

- d. Grub 設定を再構築します。

- Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Ubuntu 16.04 および Ubuntu 18.04

```
$ sudo update-grub
```

- インスタンスを再起動して、そのインスタンスに再接続します。
- Nvidia CUDA ツールキットインストーラーをダウンロードします。

```
$ wget http://developer.download.nvidia.com/compute/cuda/10.1/Prod/local_installers/cuda_10.1.243_418.87.00_linux.run
```

- Nvidia CUDA ツールキットインストーラーを実行します。

```
$ sudo sh cuda_10.1.243_418.87.00_linux.run
```

ライセンス契約を承諾するよう求められたら、accept を入力して、Enter キーを押します。

- CUDA インストーラーメニューで、すべての項目が選択されていることを確認し、Install (インストール) をハイライトして Enter キーを押します。
- シェル起動スクリプトに以下のステートメントを追加し、CUDA パスがインスタンスの起動時に毎回設定されることを確認します。

```
export PATH=/usr/local/cuda-10.1/bin:/usr/local/cuda-10.1/NsightCompute-2019.1${PATH:+:$PATH}  
export LD_LIBRARY_PATH=/usr/local/cuda-10.1/lib64${LD_LIBRARY_PATH:+:$LD_LIBRARY_PATH}
```

- bash シェルには、/home/*username*/.bashrc と /home/*username*/.bash\_profile にステートメントを追加します。
  - tcsh シェルには、/home/*username*/.cshrc にステートメントを追加します。
- 以下のコマンドを実行して、Nvidia GPU ドライバが機能することを確認します。

```
$ nvidia-smi -q | head
```

このコマンドは、Nvidia GPU、Nvidia GPU ドライバ、Nvidia CUDA ツールキットの情報を返します。

## ステップ 5: NCCL をインストールする

NCCL をインストールします。NCCL に関する詳細は、「[NCCL repository](#)」を参照してください。

### 前提条件

- NCCL には Nvidia CUDA 7.0 以降が必要です。最新のバージョンのインストールに関する詳細は、Nvidia ウェブサイトの「[CUDA Toolkit 10.1 Update 2 Download](#)」を参照してください。

### NCCL をインストールするには

- ホームディレクトリに移動します。

```
$ cd $HOME
```

- 公式の NCCL リポジトリをインスタンスにクローンし、ローカルのクローンされたリポジトリに移動します。

```
$ git clone https://github.com/NVIDIA/nccl.git
```

```
$ cd nccl
```

- NCCL を構築およびインストールし、CUDA インストールディレクトリを指定します。以下のコマンドは、CUDA がデフォルトのディレクトリにインストールされていることを前提としています。

```
$ make -j src.build
```

## ステップ 6: aws-ofi-nccl プラグインをインストールする

aws-ofi-nccl プラグインは、NCCL の接続目的のトランSPORT API を、Llibfabric の接続がなく信頼性の高いインターフェイスにマップします。これにより、NCCL ベースのアプリケーションの実行中に、Libfabric をネットワークプロバイダーとして使用できます。aws-ofi-nccl プラグインに関する詳細は、[aws-ofi-nccl リポジトリ](#)を参照してください。

aws-ofi-nccl プラグインをインストールするには

- ホームディレクトリに移動します。

```
$ cd $HOME
```

- aws-ofi-nccl プラグインをインストールするために必要なユーティリティをインストールします。必要なユーティリティをインストールするには、以下のコマンドを実行します。

- Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7

```
$ sudo yum install libudev-devel -y
```

- Ubuntu 16.04 および Ubuntu 18.04

```
$ sudo apt-get install libudev-dev libtool autoconf -y
```

- 公式の AWS aws-ofi-nccl リポジトリの aws ブランチをインスタンスにクローンし、ローカルのクローンされたリポジトリに移動します。

```
$ git clone https://github.com/aws/aws-nccl.git -b aws
```

```
$ cd aws-ofi-nccl
```

- configure スクリプトを生成するには、autogen.sh スクリプトを実行します。

```
$ ./autogen.sh
```

- make ファイルを生成するには、configure スクリプトを実行し、MPI、Libfabric、NCCL、CUDA インストールディレクトリを指定します。

```
$ ./configure --with-mpi=/opt/amazon/openmpi --with-libfabric=/opt/amazon/efa --with-nccl=$HOME/nccl/build --with-cuda=/usr/local/cuda-10.1
```

- aws-ofi-nccl プラグインをインストールします。

```
$ sudo make
```

```
$ sudo make install
```

## ステップ 7: NCCL テストをインストールする

NCCL テストをインストールします。NCCL テストでは、NCCL が適切にインストールされていることを確認し、想定どおりに機能していることを確認できます。NCCL テストに関する詳細は、「[nccl-tests リポジトリ](#)」を参照してください。

NCCL テストをインストールするには

1. ホームディレクトリに移動します。

```
$ cd $HOME
```

2. 公式の nccl-tests リポジトリをインスタンスにクローンし、ローカルのクローンされたリポジトリに移動します。

```
$ git clone https://github.com/NVIDIA/nccl-tests.git
```

```
$ cd nccl-tests
```

3. Libfabric ディレクトリを `LD_LIBRARY_PATH` 変数に追加します。

- Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib64:$LD_LIBRARY_PATH
```

- Ubuntu 16.04 および Ubuntu 18.04

```
$ export LD_LIBRARY_PATH=/opt/amazon/efa/lib:$LD_LIBRARY_PATH
```

4. (Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7 のみ) デフォルトでは、make ファイルは `mpi_home`/lib ディレクトリで必要なライブラリを検索します。ただし、Open MPI が EFA でインストールされている場合、ライブラリは `mpi_home`/lib64 にあります。make ファイルのパスを更新するには、以下のコマンドを実行します。

```
$ sed -i s/'NVLDFLAGS += -L$(MPI_HOME)\lib -lmpi'/'NVLDFLAGS += -L$(MPI_HOME)\lib64 - lmpi' / src/Makefile
```

5. NCCL テストをインストールし、MPI、NCCL、CUDA インストールディレクトリを指定します。

```
$ make MPI=1 MPI_HOME=/opt/amazon/openmpi NCCL_HOME=$HOME/nccl/build CUDA_HOME=/usr/local/cuda-10.1
```

## ステップ 8: EFA と NCCL 設定をテストする

テストを実行し、EFA と NCCL に一時インスタンスが適切に設定されていることを確認します。

## EFA と NCCL 設定をテストするには

1. テストを実行するホストを指定するホストファイルを作成します。以下のコマンドは、インスタンス自体へのリファレンスを含む `my-hosts` と呼ばれるホストファイルを作成します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

2. テストを実行し、ホストファイル (`--hostfile`) と使用する GPU の数 (`-n`) を指定します。以下のコマンドは、インスタンス自体の 8 つの GPU で `all_reduce_perf` テストを実行し、以下の環境変数を指定します。

- `FI_PROVIDER="efa"`— ファブリックインターフェイスプロバイダーを指定します。これは、"efa" に指定する必要があります。
- `FI_EFA_TX_MIN_CREDITS=64`— 送信者が受信者からリクエストする送信クレジットの最小の数を指定します。64 は、EFA を使用する NCCL ジョブに推奨される値です。この値は、256 MB を超えるメッセージ転送にのみ増加することができます。
- `NCCL_DEBUG=INFO`— 詳細なデバッグ出力を表示します。また、テストの開始時に NCCL バージョンのみをプリントするために `VERSION` を指定したり、エラーメッセージのみを受信するために `WARN` を指定したりすることもできます。
- `NCCL_TREE_THRESHOLD=0`— テストのツリーアルゴリズムを無効にします。

NCCL テスト引数に関する詳細は、公式の `nccl-tests` リポジトリの [NCCL Tests README](#) を参照してください。

- Amazon Linux、Amazon Linux 2、RHEL 7.6/7.7、CentOS 7

```
$ /opt/amazon/openmpi/bin/mpirun \
-x FI_PROVIDER="efa" \
-x FI_EFA_TX_MIN_CREDITS=64 \
-x LD_LIBRARY_PATH=$HOME/nccl/build/lib:/usr/local/cuda-10.1/lib64:/opt/amazon/efa/lib64:/opt/amazon/openmpi/lib64:$LD_LIBRARY_PATH \
-x NCCL_DEBUG=INFO \
-x NCCL_TREE_THRESHOLD=0 \
--hostfile my-hosts -n 8 -N 8 \
--mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
$HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

- Ubuntu 16.04 および Ubuntu 18.04

```
$ /opt/amazon/openmpi/bin/mpirun \
-x FI_PROVIDER="efa" \
-x FI_EFA_TX_MIN_CREDITS=64 \
-x LD_LIBRARY_PATH=$HOME/nccl/build/lib:/usr/local/cuda-10.1/lib64:/opt/amazon/efa/lib:/opt/amazon/openmpi/lib:$LD_LIBRARY_PATH \
-x NCCL_DEBUG=INFO \
-x NCCL_TREE_THRESHOLD=0 \
--hostfile my-hosts -n 8 -N 8 \
--mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
```

```
$HOME/nccl-tests/build/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

## ステップ 9: 機械学習アプリケーションをインストールする

一時的なインスタンスに機械学習アプリケーションをインストールします。インストール手順は、機械学習アプリケーションによって異なります。Linux インスタンスへのソフトウェアのインストールの詳細については、「[Linux インスタンスでのソフトウェアの管理](#)」を参照してください。

### Note

インストール手順については、機械学習アプリケーションのドキュメントの参照が必要になる場合があります。

## ステップ 10: EFA と NCCL 対応 AMI を作成する

必要なソフトウェアコンポーネントをインストールしたら、EFA 対応のインスタンスの起動に再利用できる AMI を作成します。

一時インスタンスから AMI を作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. ステップ 1 で作成したインスタンスを作成し、[アクション]、[イメージ]、[イメージの作成] の順に選択します。
4. [イメージの作成] ウィンドウで、以下の操作を行います。
  - a. [イメージ名] に、AMI の分かりやすい名前を入力します。
  - b. (オプション) [イメージの説明] に、AMI の簡単な説明を入力します。
  - c. [イメージの説明]、[Close (閉じる)] の順に選択します。
5. ナビゲーションペインで [AMI] を選択します。
6. リストで作成した AMI を探します。ステータスが pending から available になるまで待ってから、次のステップに進みます。

## ステップ 11: 一時インスタンスを終了する

この時点で、ステップ 1 で起動した一時インスタンスは不要になります。このインスタンスに対する料金が請求されないように、インスタンスを終了することができます。

ステップ 7: 一時インスタンスを終了する

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. ステップ 1 で作成した一時インスタンスを選択し、[アクション]、[インスタンスの状態]、[終了]、[Yes, Terminate (はい、終了する)] の順に選択します。

## ステップ 12: クラスターのプレイスメントグループで EFA と NCCL 対応のインスタンスを起動する

EFA 対応の AMI、および EFA 対応のセキュリティグループを使用して、EFA および NCCL 対応のインスタンスをクラスター プレイスマント グループに起動します。

EFA および NCCL 対応のインスタンスをクラスター プレイスマント グループに起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. [インスタンスの作成] を選択します。
3. [AMI の選択] ページで、[自分の AMI] を選択し、以前に作成した AMI を探して、[選択] を選択します。
4. [Choose an Instance Type (インスタンスタイプを選択)] ページで [p3dn.24xlarge] を選択してから、[Next: Configure Instance Details (次へ：インスタンス詳細の設定)] を選択します。
5. [Configure Instance Details] ページで以下の操作を実行します。
  - a. [インスタンス数] に、起動する EFA および NCCL 対応のインスタンスの数を入力します。
  - b. [ネットワーク] および [サブネット] で、インスタンスを起動する VPC およびサブネットを選択します。
  - c. [プレイスメントグループ] で、[インスタンスをプレイスメントグループに追加します] チェックボックスをオンにします。
  - d. [プレイスメントグループ名] で、[新しいプレイスメントグループに追加します] チェックボックスをオンにし、プレイスメントグループの分かりやすい名前を入力します。次に、[プレイスメントグループ戦略] で [クラスター] を選択します。
  - e. [EFA] で、[有効化] を選択します。
  - f. [ネットワークインターフェイス] セクションの [eth0] で、[新しいネットワークインターフェイス] を選択します。必要に応じて、プライマリ IPv4 アドレスと 1 つ以上のセカンダリ IPv4 アドレスを指定できます。関連付けられている IPv6 CIDR ブロックを持つサブネットにインスタンスを起動する場合は、必要に応じて、プライマリ IPv6 アドレスと 1 つ以上のセカンダリ IPv6 アドレスを指定することができます。
  - g. [次の手順: ストレージの追加] を選択します。
6. [Add Storage (ストレージの追加)] ページで、AMI で指定されたボリュームに加えてインスタンスにアタッチするボリューム（例: ルートデバイスボリューム）を指定します。次に、[Next: Add Tags (次へ: タグの追加)] を選択します。
7. [Add Tags] ページで、ユーザーフレンドリーな名前などを使ってインスタンスのタグを指定し、[Next: Configure Security Group] を選択します。
8. [セキュリティグループの設定] ページの [セキュリティグループの割り当て] で、[既存のセキュリティグループの選択] を選択し、前に作成したセキュリティグループを選択します。
9. [Review and Launch] を選択します。
10. [インスタンス作成の確認] ページで設定を確認し、[起動] を選択してキーペアを選択し、インスタンスを起動します。

## ステップ 13: パスワードレス SSH を有効にする

アプリケーションを有効にして、クラスターのすべてのインスタンス間で実行するには、リーダーノードからメンバーノードに対してパスワードレス SSH アクセスを有効にする必要があります。リーダーノードは、アプリケーションの実行元となるインスタンスです。クラスターのその他のインスタンスは、メンバーノードになります。

クラスターのインスタンス間でパスワードレス SSH を有効にするには

1. クラスターの 1 つのインスタンスをリーダーノードとして選択し、そのインスタンスに接続します。
2. リーダーノードで `strictHostKeyChecking` を無効にし、`ForwardAgent` を有効にします。適切なテキストエディタを使用して `~/.ssh/config` を開き、以下を追加します。

```
Host *
  ForwardAgent yes
Host *
  StrictHostKeyChecking no
```

3. RSA キーペアを生成します。

```
$ ssh-keygen -t rsa -N "" -f /home/ubuntu/.ssh/id_rsa
```

\$HOME/.ssh/ ディレクトリでキーペアが作成されます。

4. リーダーノードのプライベートキーの許可を変更します。

```
$ chmod 600 ~/.ssh/id_rsa
```

5. 適切なテキストエディタを使用して ~/.ssh/id\_rsa.pub を開き、キーをコピーします。

6. クラスターの各メンバーノードで、以下を実行します。

- a. インスタンスに接続します。

- b. 適切なテキストエディタを使用して ~/.ssh/authorized\_keys を開き、前にコピーしたパブリックキーを追加します。

7. パスワードレス SSH が予期したとおりに機能しているかテストするため、リーダーノードに接続し、次のコマンドを実行します。

```
$ ssh member_node_private_ip
```

キーまたはパスワードを求められることなく、メンバーノードに接続できるはずです。

## AWS Deep Learning AMI の使用

以下の手順は、以下の AWS Deep Learning AMI のいずれかを開始するのに役立ちます。

- Deep Learning AMI Version AMI (Amazon Linux 2) バージョン 25.0 以降
- Deep Learning AMI Version AMI (Amazon Linux) バージョン 25.0 以降
- Deep Learning AMI Version AMI (Ubuntu 18.04) バージョン 25.0 以降
- Deep Learning AMI Version AMI (Ubuntu 16.04) バージョン 25.0 以降

詳細については、「[AWS Deep Learning AMI ユーザーガイド](#)」を参照してください。

### コンテンツ

- [ステップ 1: EFA 対応のセキュリティグループを準備する \(p. 783\)](#)
- [ステップ 2: 一時インスタンスを起動する \(p. 784\)](#)
- [ステップ 3: EFA と NCCL 設定をテストする \(p. 785\)](#)
- [ステップ 4: 機械学習アプリケーションをインストールする \(p. 785\)](#)
- [ステップ 5: EFA と NCCL 対応 AMI を作成する \(p. 786\)](#)
- [ステップ 6: 一時インスタンスを終了する \(p. 786\)](#)
- [ステップ 7: クラスターのプレイスメントグループで EFA と NCCL 対応のインスタンスを起動する \(p. 786\)](#)
- [Step 8: パスワードレス SSH を有効にする \(p. 787\)](#)

### ステップ 1: EFA 対応のセキュリティグループを準備する

EFA には、セキュリティグループ自体とのインバウンドおよびアウトバウンドのトラフィックをすべて許可するセキュリティグループが必要です。

### EFA 対応のセキュリティグループを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [セキュリティグループ] を選択し、[セキュリティグループの作成] を選択します。
3. [セキュリティグループの作成] ウィンドウで以下を行います。
  - a. [セキュリティグループ名] に、セキュリティグループの分かりやすい名前 (例: EFA-enabled security group) を入力します。
  - b. (オプション) [説明] に、セキュリティグループの簡単な説明を入力します。
  - c. [VPC] で、EFA 対応のインスタンスを起動する VPC を選択します。
  - d. [作成] を選択します。
4. 作成したセキュリティグループを選択し、[説明] タブで [グループ ID] をコピーします。
5. [インバウンド] タブおよび [アウトバウンド] タブで、次の手順を実行します。
  - a. [Edit] を選択します。
  - b. [タイプ] で、[すべてのトラフィック] を選択します。
  - c. [ソース] で [カスタム] を選択します。
  - d. コピーしたセキュリティグループ ID をフィールドに貼り付けます。
  - e. [Save] を選択します。

### ステップ 2: 一時インスタンスを起動する

EFA ソフトウェアコンポーネントのインストールおよび設定に使用する一時インスタンスを起動します。このインスタンスを使用して、EFA 対応のインスタンスを起動する EFA 対応の AMI を作成します。

#### 一時インスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. [AMI を選択] ページで、サポートされた AWS Deep Learning AMI バージョン 25.0 以降を選択します。
4. [インスタンスタイプの選択] ページで p3dn.24xlarge を選択してから、[次の手順: インスタンスの詳細の設定] を選択します。
5. [Configure Instance Details] ページで以下の操作を実行します。
  - a. [Elastic Fabric Adapter] で、[有効化] を選択します。
  - b. [ネットワークインターフェイス] セクションの [eth0] で、[新しいネットワークインターフェイス] を選択します。
  - c. [次の手順: ストレージの追加] を選択します。
6. [ストレージの追加] ページで、AMI で指定されたボリューム (ルートデバイスピリュームなど) に加えてインスタンスにアタッチするボリュームを指定します。次に、[次の手順: タグの追加] を選択します。
7. [タグの追加] ページで、一時インスタンスの識別に使用するタグを指定し、[Next: Configure Security Group (次へ: セキュリティグループの設定)] を選択します。
8. [セキュリティグループの設定] ページの [セキュリティグループの割り当て] で、[Select an existing security group (既存のセキュリティグループの選択)] を選択します。次に、ステップ 1 で作成したセキュリティグループを選択します。
9. [インスタンス作成の確認] ページで設定を確認し、[起動] を選択してキーペアを選択し、インスタンスを起動します。

## ステップ 3: EFA と NCCL 設定をテストする

テストを実行し、EFA と NCCL に一時インスタンスが適切に設定されていることを確認します。

EFA と NCCL 設定をテストするには

1. テストを実行するホストを指定するホストファイルを作成します。以下のコマンドは、インスタンス自体へのリファレンスを含む `my-hosts` と呼ばれるホストファイルを作成します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> my-hosts
```

2. テストを実行し、ホストファイル (`--hostfile`) と使用する GPU の数 (`-n`) を指定します。以下のコマンドは、インスタンス自体の 8 つの GPU で `all_reduce_perf` テストを実行し、以下の環境変数を指定します。

- `FI_PROVIDER="efa"`— フアブリックインターフェイスプロバイダーを指定します。これは、"efa" に指定する必要があります。
- `FI_EFA_TX_MIN_CREDITS=64`— 送信者が受信者からリクエストする送信クレジットの最小の数を指定します。64 は、EFA を使用する NCCL ジョブに推奨される値です。この値は、256 MB を超えるメッセージ転送にのみ増加することができます。
- `NCCL_DEBUG=INFO`— 詳細なデバッグ出力を表示します。また、テストの開始時に NCCL バージョンのみをプリントするために `VERSION` を指定したり、エラーメッセージのみを受信するために `WARN` を指定したりすることもできます。
- `NCCL_TREE_THRESHOLD=0`— テストのツリーアルゴリズムを無効にします。

NCCL テスト引数に関する詳細は、公式の `nccl-tests` リポジトリの [NCCL Tests README](#) を参照してください。

```
/opt/amazon/openmpi/bin/mpirun \
-x FI_PROVIDER="efa" \
-x FI_EFA_TX_MIN_CREDITS=64 \
-x NCCL_DEBUG=INFO \
-x NCCL_TREE_THRESHOLD=0 \
--hostfile my-hosts -n 8 -N 8 \
--mca btl tcp,self --mca btl_tcp_if_exclude lo,docker0 --bind-to none \
$HOME/src/bin/efa-tests/efa-cuda-10.0/all_reduce_perf -b 8 -e 1G -f 2 -g 1 -c 1 -n 100
```

## ステップ 4: 機械学習アプリケーションをインストールする

一時的なインスタンスに機械学習アプリケーションをインストールします。インストール手順は、機械学習アプリケーションによって異なります。Linux インスタンスへのソフトウェアのインストールの詳細については、「[Linux インスタンスでのソフトウェアの管理](#)」を参照してください。

Note

インストール手順については、機械学習アプリケーションのドキュメントの参照が必要になる場合があります。

## ステップ 5: EFA と NCCL 対応 AMI を作成する

必要なソフトウェアコンポーネントをインストールしたら、EFA 対応のインスタンスの起動に再利用できる AMI を作成します。

一時インスタンスから AMI を作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. ステップ 1 で作成したインスタンスを作成し、[アクション]、[イメージ]、[イメージの作成] の順に選択します。
4. [イメージの作成] ウィンドウで、以下の操作を行います。
  - a. [イメージ名] に、AMI の分かりやすい名前を入力します。
  - b. (オプション) [イメージの説明] に、AMI の簡単な説明を入力します。
  - c. [イメージの説明]、[Close (閉じる)] の順に選択します。
5. ナビゲーションペインで [AMI] を選択します。
6. リストで作成した AMI を探します。ステータスが pending から available になるまで待ってから、次のステップに進みます。

## ステップ 6: 一時インスタンスを終了する

この時点で、ステップ 1 で起動した一時インスタンスは不要になります。このインスタンスに対する料金が請求されないように、インスタンスを終了することができます。

ステップ 7: 一時インスタンスを終了する

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. ステップ 1 で作成した一時インスタンスを選択し、[アクション]、[インスタンスの状態]、[終了]、[Yes, Terminate (はい、終了する)] の順に選択します。

## ステップ 7: クラスターのプレイスメントグループで EFA と NCCL 対応のインスタンスを起動する

EFA 対応の AMI、および EFA 対応のセキュリティグループを使用して、EFA および NCCL 対応のインスタンスをクラスターのプレイスメントグループに起動します。

EFA および NCCL 対応のインスタンスをクラスターのプレイスメントグループに起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. [AMI の選択] ページで、[自分の AMI] を選択し、以前に作成した AMI を探して、[選択] を選択します。
4. [Choose an Instance Type (インスタンスタイプを選択)] ページで [p3dn.24xlarge] を選択してから、[Next: Configure Instance Details (次へ：インスタンス詳細の設定)] を選択します。
5. [Configure Instance Details] ページで以下の操作を実行します。
  - a. [インスタンス数] に、起動する EFA および NCCL 対応のインスタンスの数を入力します。
  - b. [ネットワーク] および [サブネット] で、インスタンスを起動する VPC およびサブネットを選択します。

- c. [プレイスメントグループ] で、[インスタンスをプレイスメントグループに追加します] チェックボックスをオンにします。
  - d. [プレイスメントグループ名] で、[新しいプレイスメントグループに追加します] チェックボックスをオンにし、プレイスメントグループの分かりやすい名前を入力します。次に、[プレイスメントグループ戦略] で [クラスター] を選択します。
  - e. [EFA] で、[有効化] を選択します。
  - f. [ネットワークインターフェイス] セクションの [eth0] で、[新しいネットワークインターフェイス] を選択します。必要に応じて、プライマリ IPv4 アドレスと 1 つ以上のセカンダリ IPv4 アドレスを指定できます。関連付けられている IPv6 CIDR ブロックを持つサブネットにインスタンスを起動する場合は、必要に応じて、プライマリ IPv6 アドレスと 1 つ以上のセカンダリ IPv6 アドレスを指定することができます。
  - g. [次の手順: ストレージの追加] を選択します。
6. [Add Storage (ストレージの追加)] ページで、AMI で指定されたボリュームに加えてインスタンスにアタッチするボリューム (例: ルートデバイスボリューム) を指定します。次に、[Next: Add Tags (次へ: タグの追加)] を選択します。
  7. [Add Tags] ページで、ユーザーフрендリーな名前などを使ってインスタンスのタグを指定し、[Next: Configure Security Group] を選択します。
  8. [セキュリティグループの設定] ページの [セキュリティグループの割り当て] で、[既存のセキュリティグループの選択] を選択し、前に作成したセキュリティグループを選択します。
  9. [Review and Launch] を選択します。
  10. [インスタンス作成の確認] ページで設定を確認し、[起動] を選択してキーペアを選択し、インスタンスを起動します。

## Step 8: パスワードレス SSH を有効にする

アプリケーションを有効にして、クラスターのすべてのインスタンス間で実行するには、リーダーノードからメンバーノードに対してパスワードレス SSH アクセスを有効にする必要があります。リーダーノードは、アプリケーションの実行元となるインスタンスです。クラスターのその他のインスタンスは、メンバーノードになります。

クラスターのインスタンス間でパスワードレス SSH を有効にするには

1. クラスターの 1 つのインスタンスをリーダーノードとして選択し、そのインスタンスに接続します。
2. リーダーノードで `strictHostKeyChecking` を無効にし、`ForwardAgent` を有効にします。適切なテキストエディタを使用して `~/.ssh/config` を開き、以下を追加します。

```
Host *
  ForwardAgent yes
Host *
  StrictHostKeyChecking no
```

3. RSA キーペアを生成します。

```
$ ssh-keygen -t rsa -N "" -f /home/ubuntu/.ssh/id_rsa
```

`$HOME/.ssh/` ディレクトリでキーペアが作成されます。

4. リーダーノードのプライベートキーの許可を変更します。

```
$ chmod 600 ~/.ssh/id_rsa
```

5. 適切なテキストエディタを使用して `~/.ssh/id_rsa.pub` を開き、キーをコピーします。
6. クラスターの各メンバーノードで、以下を実行します。

- a. インスタンスに接続します。
  - b. 適切なテキストエディタを使用して `~/.ssh/authorized_keys` を開き、前にコピーしたパブリックキーを追加します。
7. パスワードレス SSH が予期したとおりに機能しているかテストするため、リーダーノードに接続し、次のコマンドを実行します。

```
$ ssh member_node_private_ip
```

キーまたはパスワードを求められることなく、メンバーノードに接続できるはずです。

## EFA の使用

EFA は、Amazon EC2 の他の Elastic Network Interface と同じように作成、使用、管理することができます。ただし、Elastic Network Interface とは異なり、EFAs は、実行中状態のインスタンスにアタッチしたり、実行中状態のインスタンスからデタッチしたりすることはできません。

## EFA の要件

EFA を使用するには、以下の操作を行う必要があります。

- サポートされているインスタンスタイプ (`c5n.18xlarge`, `c5n.metal`, `i3en.24xlarge`, `i3en.metal`, `inf1.24xlarge`, `m5dn.24xlarge`, `m5n.24xlarge`, `r5dn.24xlarge`, `r5n.24xlarge`, and `p3dn.24xlarge`) のいずれかを使用します。
- サポートされている AMI (Amazon Linux, Amazon Linux 2, RHEL 7.6, RHEL 7.7, CentOS 7, Ubuntu 16.04, and Ubuntu 18.04) のいずれかを使用します。
- EFA ソフトウェアコンポーネントをインストールします。詳細については、「[ステップ 3: EFA ソフトウェアをインストールする \(p. 767\)](#)」および「[ステップ 5: \(オプション\) インテル MPI をインストールする \(p. 769\)](#)」を参照してください。
- セキュリティグループ自体との間のインバウンドおよびアウトバウンドのトラフィックをすべて許可するセキュリティグループを使用します。詳細については、「[ステップ 1: EFA 対応のセキュリティグループを準備する \(p. 766\)](#)」を参照してください。

### コンテンツ

- [EFAを作成する \(p. 788\)](#)
- [EFAを停止したインスタンスにアタッチする \(p. 789\)](#)
- [インスタンス起動時にEFAをアタッチする \(p. 789\)](#)
- [EFAを起動テンプレートに追加する \(p. 790\)](#)
- [IPアドレスをEFAに割り当てる \(p. 790\)](#)
- [EFAからのIPアドレスの割り当て解除 \(p. 790\)](#)
- [セキュリティグループの変更 \(p. 790\)](#)
- [EFAのデタッチ \(p. 790\)](#)
- [EFAsの表示 \(p. 790\)](#)
- [EFAの削除 \(p. 791\)](#)

## EFAを作成する

EFA は、VPC のサブネットに作成することができます。作成後に EFA を別のサブネットに移動することはできません。また、アタッチできるのは、同じアベイラビリティーゾーンの停止したインスタンスに限ります。

## コンソールを使用して新しい EFA を作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. [Create Network Interface] を選択します。
4. [説明] に、EFA の分かりやすい名前を入力します。
5. [サブネット] で、EFA を作成するサブネットを選択します。
6. [プライベート IP] に、プライマリのプライベート IPv4 アドレスを入力します。IPv4 アドレスを指定しない場合、選択されているサブネット内で使用可能なプライベート IPv4 アドレスが選択されます。
7. (IPv6 のみ) IPv6 CIDR ブロックが関連付けられているサブネットを選択した場合は、オプションで [IPv6 IP] フィールドに IPv6 アドレスを指定できます。
8. [Security groups] で、1 つまたは複数のセキュリティグループを選択します。
9. [EFA] で、[有効化] を選択します。
10. [Yes, Create] を選択します。

## AWS CLI を使用して新しい EFA を作成するには

次の例に示されているように、`create-network-interface` コマンドを使用し、`interface-type` で `efa` を指定します。

```
$ aws ec2 create-network-interface --subnet-id subnet-01234567890 --description example_efa  
--interface-type efa
```

## EFA を停止したインスタンスにアタッチする

EFA は、サポート対象の `stopped` 状態のインスタンスにアタッチすることができます。`running` 状態のインスタンスに EFA をアタッチすることはできません。サポートされるインスタンスタイプの詳細については、[サポートされるインスタンスタイプ \(p. 765\)](#) を参照してください。

Elastic Network Interface をインスタンスにアタッチするのと同じ方法で、EFA をインスタンスにアタッチできます。詳細については、[停止したインスタンスまたは実行中のインスタンスにネットワークインターフェイスをアタッチする \(p. 730\)](#) を参照してください。

## インスタンス起動時に EFA をアタッチする

インスタンスの起動時に既存の EFA をアタッチするには (AWS CLI)

次の例に示されているように、`run-instances` コマンドを使用し、`NetworkInterfaceId` で EFA の ID を指定します。

```
$ aws ec2 run-instances --image-id ami_id --count 1 --instance-type c5n.18xlarge --key-name my_key_pair --network-interfaces DeviceIndex=0,NetworkInterfaceId=efa_id,Groups=sg_id,SubnetId=subnet_id
```

インスタンス起動時に新しい EFA をアタッチするには (AWS CLI)

次の例に示されているように、`run-instances` コマンドを使用し、`InterfaceType` で `efa` を指定します。

```
$ aws ec2 run-instances --image-id ami_id --count 1 --instance-type c5n.18xlarge --key-name my_key_pair --network-interfaces DeviceIndex=0,InterfaceType=efa,Groups=sg_id,SubnetId=subnet_id
```

## EFA を起動テンプレートに追加する

EFA 対応のインスタンスの起動に必要な設定情報を含む起動テンプレートを作成できます。EFA 対応の起動テンプレートを作成するには、新しい起動テンプレートを作成し、サポート対象のインスタンスタイプ、EFA 対応の AMI、および EFA 対応のセキュリティグループを指定します。詳細については、「[EFA および MPI の開始方法 \(p. 765\)](#)」を参照してください。

他の AWS のサービス (AWS Batch など) を使用して EFA 対応のインスタンスを起動するには、起動テンプレートを利用できます。

起動テンプレートの作成の詳細については、「[起動テンプレートの作成 \(p. 456\)](#)」を参照してください。

## IP アドレスを EFA に割り当てる

Elastic IP (IPv4) アドレスをお持ちの場合は、EFA に関連付けることができます。IPv6 CIDR ブロックに関連付けられているサブネットで EFA をプロビジョンしている場合は、1 つ以上の IPv6 アドレスを EFA に割り当てるすることができます。

IP アドレスを Elastic Network Interface に割り当てるのと同じ方法で、Elastic IP (IPv4) および IPv6 アドレスを EFA に割り当てます。詳細については、以下のトピックを参照してください。

- [Elastic IP アドレス \(IPv4\) の関連付け \(p. 733\)](#)
- [IPv6 アドレスの割り当て \(p. 734\)](#)

## EFA からの IP アドレスの割り当て解除

IP アドレスを Elastic Network Interface から割り当て解除するのと同じ方法で、Elastic IP (IPv4) および IPv6 アドレスを EFA から割り当て解除します。詳細については、以下のトピックを参照してください。

- [Elastic IP アドレス \(IPv4\) の関連付けの解除 \(p. 734\)](#)
- [IPv6 アドレスの割り当て解除 \(p. 735\)](#)

## セキュリティグループの変更

EFA に関連付けられているセキュリティグループは変更することができます。OS バイパス機能を有効にするには、EFA が、セキュリティグループ自体との間のインバウンドおよびアウトバウンドのトラフィックをすべて許可するセキュリティグループのメンバーである必要があります。

Elastic Network Interface に関連付けられているセキュリティグループを変更するのと同じ方法で、EFA に関連付けられているセキュリティグループを変更します。詳細については、「[セキュリティグループの変更 \(p. 732\)](#)」を参照してください。

## EFA のデタッチ

EFA をインスタンスからデタッチするには、まずインスタンスを停止する必要があります。実行中状態のインスタンスから EFA をデタッチすることはできません。

インスタンスから Elastic Network Interface をデタッチするのと同じ方法で、EFA をインスタンスからデタッチします。詳細については、「[ネットワークインターフェイスをインスタンスからデタッチする \(p. 731\)](#)」を参照してください。

## EFAs の表示

アカウントのすべての EFAs を表示できます。

Elastic Network Interface を表示するのと同じ方法で EFAs を表示します。詳細については、「[ネットワークインターフェイスに関する詳細の表示 \(p. 729\)](#)」を参照してください。

## EFA の削除

EFA を削除するには、まずインスタンスから削除する必要があります。インスタンスにアタッチされている場合は、EFA を削除する必要があります。

Elastic Network Interface を削除するのと同じ方法で EFAs を削除します。詳細については、「[ネットワークインターフェイスの削除 \(p. 728\)](#)」を参照してください。

## EFA のモニタリング

Elastic Fabric Adapter のパフォーマンスをモニタリングするには、次の機能を使用できます。

### Amazon VPC フローログ

Amazon VPC フローログを作成することで、EFA との間で送受信されるトラフィックに関する情報を取得できます。フローログデータは Amazon CloudWatch Logs と Amazon S3 に発行できます。フローログを作成したら、選択した送信先でそのデータを取得して表示できます。詳細については、Amazon VPC ユーザーガイドの「[VPC フローログ](#)」を参照してください。

EFA のフローログを作成する方法は、Elastic Network Interface のフローログを作成する場合と同じです。詳細については、Amazon VPC ユーザーガイドの「[フローログの作成](#)」を参照してください。

フローログエントリで、EFA エントリは、srcAddress および destAddress で識別されます。次の例に示されているように、これらはいずれも MAC アドレス形式になります。

| version  | accountId  | eniId        | srcAddress        | destAddress       | sourcePort | destPort   |
|----------|------------|--------------|-------------------|-------------------|------------|------------|
| protocol | packets    | bytes        | start             | end               | action     | log-status |
| 2        | 3794735123 | eni-10000001 | 01:23:45:67:89:ab | 05:23:45:67:89:ab | -          | -          |
| 9        | 5689       | 1521232534   | 1524512343        | ACCEPT            | OK         |            |

### Amazon CloudWatch

Amazon CloudWatch には、リアルタイムで EFAs をモニタリングできるメトリクスが用意されています。メトリクスの収集と追跡、カスタマイズしたダッシュボードの作成、および指定したメトリクスが指定したしきい値に達したときに通知またはアクションを実行するアラームの設定を行うことができます。詳細については、「[CloudWatch を使用したインスタンスのモニタリング \(p. 642\)](#)」を参照してください。

## プレイスメントグループ

新しい EC2 インスタンスを起動する場合、EC2 サービスは、相関性のエラーを最小限に抑えるために、すべてのインスタンスが基盤となるハードウェアに分散されるようにインスタンスを配置します。プレイスメントグループを使用することで、ワークロードのニーズに対応するために独立したインスタンスのグループのプレイスメントに影響を与えることができます。ワークロードのタイプに応じて、以下のいずれかのプレイスメント戦略によりプレイスメントグループを作成できます。

- クラスター – アベイラビリティゾーン内でインスタンスをまとめます。この戦略により、ワークロードは、HPC アプリケーションで典型的な緊密に組み合わされたノード間通信に必要な低レイテンシーネットワークパフォーマンスを実現できます。
- パーティション – インスタンスを複数の論理パーティションに分散させ、1 つのパーティション内のインスタンスのグループが基盤となるハードウェアを別のパーティション内のインスタンスのグループと共有しないようにします。この戦略は、Hadoop、Cassandra、Kafka などの大規模な分散および複製ワークロードで一般的に使用されます。
- 分散 – 相関性のエラーを減らすために、少数のインスタンスを厳密に基盤となるハードウェア全体に配置します。

プレイスマントグループを作成するための料金は発生しません。

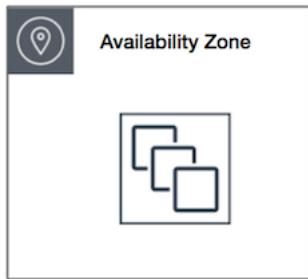
#### コンテンツ

- [クラスタープレイスマントグループ \(p. 792\)](#)
- [パーティションプレイスマントグループ \(p. 793\)](#)
- [スプレッドプレイスマントグループ \(p. 793\)](#)
- [プレイスマントグループのルールと制限 \(p. 794\)](#)
- [プレイスマントグループの作成 \(p. 795\)](#)
- [プレイスマントグループでのインスタンスの起動 \(p. 796\)](#)
- [プレイスマントグループのインスタンスを説明する \(p. 798\)](#)
- [インスタンスのプレイスマントグループの変更 \(p. 799\)](#)
- [プレイスマントグループを削除する \(p. 800\)](#)

## クラスタープレイスマントグループ

クラスタープレイスマントグループは、単一のアベイラビリティーボーン内のインスタンスを論理的にグループ化したもので、クラスタープレイスマントグループは、同じリージョン内の複数のピア VPC にまたがることができます。同じクラスタープレイスマントグループ内のインスタンスは、TCP/IP トランザクションの 10 Gbps を上限とするフローごとのスループット制限が高くなり、ネットワークの同じ高バーサイションバンド幅セグメントに配置されます。

次の図は、クラスタープレイスマントグループに配置されたインスタンスを示しています。



低いネットワークレイテンシー、高いネットワークスループット、またはその両方からメリットを受けるアプリケーションの場合は、クラスタープレイスマントグループの使用をお勧めします。また、ネットワークトランザクションの大部分がグループ内のインスタンス間で発生している場合にもお勧めします。プレイスマントグループで、最も低いレイテンシーと最も高いネットワークパフォーマンス(1秒あたりパケット数)を実現するためには、拡張ネットワーキングをサポートするインスタンスタイプを選択します。詳細については、「[拡張ネットワーキング \(p. 737\)](#)」を参照してください。

インスタンスは、次の方法で起動することをお勧めします。

- プレイスマントグループ内で必要な数のインスタンスを起動するには、1つの起動リクエストを使用します。
- プレイスマントグループ内のすべてのインスタンスに同じインスタンスタイプを使用します。

後でプレイスマントグループにさらにインスタンスを追加しようとした場合、またはプレイスマントグループ内で複数のインスタンスタイプを起動しようとした場合、容量不足エラーが発生する可能性が高くなります。

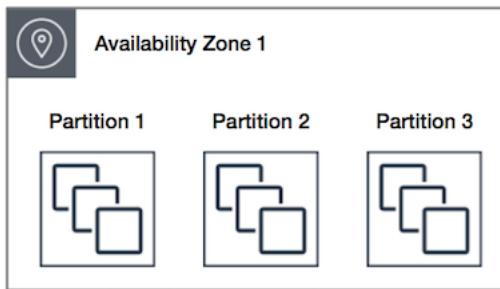
プレイスマントグループ内のインスタンスを停止して再起動しても、そのインスタンスは同じプレイスマントグループ内で実行されます。ただし、インスタンスに対して十分な容量がない場合、起動は失敗します。

既にインスタンスを実行中のプレイスメントグループ内のインスタンスを起動するときに容量エラーを受け取った場合は、プレイスメントグループ内のすべてのインスタンスを停止して開始し、もう一度起動を試みてください。インスタンスを起動すると、すべてのリクエストしたインスタンスに応じた容量があるハードウェアにインスタンスが移行される場合があります。

## パーティションプレイスメントグループ

パーティションプレイスメントグループは、アプリケーションに関連するハードウェア障害の頻度を軽減するために役立ちます。パーティションプレイスメントグループを使用する場合、Amazon EC2 は各グループをパーティションと呼ばれる論理的なセグメントに分割します。Amazon EC2 には、プレイスメントグループ内の各パーティションにそれぞれ一連のラックがあります。各ラックには独自のネットワークおよび電源があります。プレイスメントグループ内のパーティションどうしが同じラックを共有することはありません。これにより、アプリケーション内でのハードウェア障害による影響を隔離できます。

次のイメージは、単一のアベイラビリティゾーン内のパーティションプレイスメントグループのシンプルな描画を示しています。ここでは、3 つのパーティション (パーティション 1、パーティション 2、パーティション 3) があるパーティションプレイスメントグループに配置されたインスタンスを示しています。各パーティションは複数のインスタンスで構成されています。各パーティション内のインスタンスは、他のパーティション内のラックを共有しないため、単一のハードウェア障害の影響は関連付けられたパーティションのみに留まります。



パーティションプレイスメントグループは、HDFS、HBase、Cassandra などの大規模な分散および複製ワークロードを異なるラック間でデプロイするために使用できます。インスタンスをパーティションプレイスメントグループに起動すると、Amazon EC2 は、指定したパーティション数全体にインスタンスを均等に分散しようとします。インスタンスを特定のパーティションに起動して、インスタンスの配置場所をより細かく制御することもできます。

パーティションプレイスメントグループは、同じリージョン内の複数のアベイラビリティゾーンにパーティションを持つことができます。パーティションプレイスメントグループは、アベイラビリティゾーンごとに最大 7 つのパーティションを持つことができます。パーティションプレイスメントグループで起動できるインスタンス数の制限は、アカウントの制限のみです。

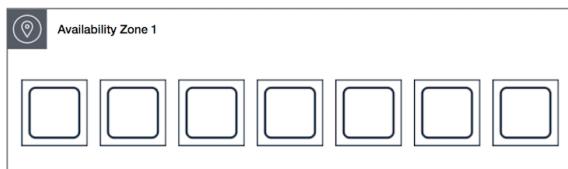
また、パーティションプレイスメントグループでは各パーティションが可視化されるため、どのインスタンスがどのパーティションにあるかを確認できます。この情報は、HDFS、HBase、Cassandra などトポロジー対応アプリケーションと共有できます。これらのアプリケーションはこの情報を利用してインテリジェントなデータレプリケーションの決定を行い、データの可用性と耐久性を向上します。

パーティションプレイスメントグループでインスタンスを開始または起動し、リクエストを実行するための固有のハードウェアが不足している場合、そのリクエストは失敗します。Amazon EC2 では、時間の経過とともににより別のハードウェアを利用できるようになりますので、後でリクエストを再試行できます。

## スプレッドプレイスメントグループ

スプレッドプレイスメントグループは、それぞれに独自のネットワークおよび電源がある異なるラックに別々に配置できるインスタンスのグループです。

次の図は、1つのアベイラビリティゾーン内の、スプレッドプレイスメントグループに配置された7つのインスタンスを示しています。7つのインスタンスは、7つの異なるラックに配置されます。



スプレッドプレイスメントグループは、少数の重要なインスタンスが互いに分離して保持される必要があるアプリケーションに推奨されます。スプレッドプレイスメントグループでインスタンスを起動すると、インスタンスが同じラックを共有するときに発生する可能性のある、同時障害のリスクが軽減されます。スプレッドプレイスメントグループは、異なるラックへのアクセスを提供するため、長時間のインスタンスタイプの混合やインスタンスの起動に適しています。

スプレッドプレイスメントグループは、同じリージョン内の複数のアベイラビリティゾーンに分散できます。グループごとのアベイラビリティゾーンごとに、最大7つの実行中のインスタンスを持つことができます。

スプレッドプレイスメントグループでインスタンスを開始または起動し、リクエストを実行するための固有のハードウェアが不足している場合、そのリクエストは失敗します。Amazon EC2 では、時間の経過とともににより別のハードウェアを利用できるようになりますので、後でリクエストを再試行できます。

## プレイスメントグループのルールと制限

### 一般的なルールと制限

プレイスメントグループを使用する前に、次のルールに注意してください。

- ・ プレイスマントグループには、リージョンの AWS アカウント内で固有の名前を付ける必要があります。
- ・ プレイスマントグループをマージすることはできません。
- ・ インスタンスは、1つのプレイスメントグループ内で一度に起動できます。複数のプレイスメントグループにまたがることはできません。
- ・ オンデマンドキャパシティー予約 (p. 432) および zonal リザーブドインスタンス (p. 282) は、特定のアベイラビリティゾーンの EC2 インスタンスに対してキャパシティーを予約します。キャパシティーの予約で、プレイスメントグループ内のインスタンスで使用できます。ただし、プレイスメントグループに対して明示的にキャパシティーを予約することはできません。
- ・ テナント `host` を持つインスタンスは、プレイスメントグループ内で起動できません。

### クラスター プレイスマントグループのルールと制限

クラスター プレイスマントグループには、以下のルールが適用されます。

- ・ クラスター プレイスマントグループ内でインスタンスを起動するときは、次のいずれかのインスタンスタイプを使用する必要があります。
  - ・ 汎用: A1、M4、M5、M5a、M5ad、M5d、M5dn、および M5n
  - ・ コンピューティングの最適化: C3、C4、C5、C5d、C5n、および cc2.8xlarge
  - ・ メモリ最適化: cr1.8xlarge、R3、R4、R5、R5a、R5ad、R5d、R5dn、R5n、X1、X1e、および z1d
  - ・ ストレージを最適化: D2、H1、hs1.8xlarge、I2、I3、および I3en
  - ・ 高速コンピューティング: F1、G2、G3、G4dn、P2、P3、および P3dn

- クラスター プレイスマント グループを、複数のアベイラビリティーゾーンで設定することはできません。
- クラスター プレイスマント グループの 2 つのインスタンス間のトラフィックの最大ネットワークスルーブット速度は、2 つのインスタンスのうち遅い方に制限されます。高スルーブットの要件があるアプリケーションの場合、要件に適合するネットワーク接続を備えたインスタンスタイプを選択します。
- 拡張ネットワーキングに対して有効になっているインスタンスには、以下のルールが適用されます。
  - クラスター プレイスマント グループ内のインスタンス間では、シングルフロートラフィックに最大 10 Gbps を使用できます。クラスター プレイスマント グループ内にないインスタンスは、シングルフロートラフィックに最大 5 Gbps を使用できます。
  - 同じリージョン内でのインスタンスと Amazon S3 バケットとの間では、パブリック IP アドレス空間または VPC エンドポイントを介したトラフィックに、使用可能なすべてのインスタンスの集計帯域幅を使用できます。
- 複数のインスタンスタイプをクラスター プレイスマント グループに起動できます。ただし、これにより起動に成功するために必要な容量が使用可能になる可能性が低くなります。クラスター プレイスマント グループ内ですべてのインスタンスで同じインスタンスタイプを使用することをお勧めします。
- インターネットへのネットワーク トラフィックとオンプレミスリソースへの AWS Direct Connect 接続は、5 Gbps に制限されます。

## パーティション プレイスマント グループのルールと制限

パーティション プレイスマント グループには、以下のルールが適用されます。

- パーティション プレイスマント グループは、アベイラビリティーゾーンごとに最大 7 つのパーティションをサポートします。パーティション プレイスマント グループで起動できるインスタンス数の制限は、アカウントの制限のみです。
- パーティション プレイスマント グループでインスタンスが起動されると、Amazon EC2 はインスタンスをすべてのパーティションに均等に分散しようとしています。Amazon EC2 では、すべてのパーティションにインスタンスが均等に分散されるとは限りません。
- ハードウェア専有インスタンスを持つパーティション プレイスマント グループは、最大 2 つのパーティションを持つことができます。
- パーティション プレイスマント グループは Dedicated Hosts ではサポートされません。

## スプレッド プレイスマント グループのルールと制限

スプレッド プレイスマント グループには、以下のルールが適用されます。

- スプレッド プレイスマント グループは、アベイラビリティーゾーンごとに最大 7 つの実行インスタンスをサポートします。たとえば、アベイラビリティーゾーンが 3 つあるリージョンでは、グループ内に合計 21 個のインスタンス (ゾーンごとに 7 個) を実行することができます。同じアベイラビリティーゾーンと同じスプレッド プレイスマント グループで 8 番目のインスタンスを開始しようとすると、インスタンスは起動しません。アベイラビリティーゾーンに 7 つ以上のインスタンスが必要な場合は、複数のスプレッド プレイスマント グループを使用することをお勧めします。複数の プレイスマント グループに分散しても、グループ間でインスタンスが分散されるとは限りませんが、グループごとに切り離されるため、特定の障害クラスからの影響は制限されます。
- ハードウェア専有インスタンス または Dedicated Hosts では、スプレッド プレイスマント グループはサポートされていません。

## プレイスメント グループの作成

プレイスメント グループは、次のいずれかの方法で作成できます。

## 新しいコンソール

コンソールを使用してプレイスメントグループを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[プレイスメントグループ]、[プレイスメントグループの作成] の順に選択します。
3. グループの名前を指定します。
4. グループのプレイスメント方法を選択します。[パーティション] を選択した場合は、グループ内のパーティション数を指定します。
5. [Create group] を選択します。

## 古いコンソール

コンソールを使用してプレイスメントグループを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Placement Group]、[Create Placement Group] の順に選択します。
3. グループの名前を指定します。
4. グループのプレイスメント方法を選択します。[パーティション] を選択する場合、グループ内のパーティションの数を指定します。
5. [作成] を選択します。

## AWS CLI

AWS CLI を使用してプレイスメントグループを作成するには

`create-placement-group` コマンドを使用します。次の例で、プレイスメントグループ名は `my-cluster`、プレイスメント方法は `cluster` です。

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

AWS CLI を使用してパーティションプレイスメントグループを作成するには

`create-placement-group` コマンドを使用します。`--strategy` パラメータに値として `partition` を指定し、`--partition-count` パラメータに必要なパーティション数を指定します。この例では、パーティションプレイスメントグループは `HDFS-Group-A` という名で、パーティションは 5 つ作成されています。

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --partition-count 5
```

## PowerShell

AWS Tools for Windows PowerShell を使用してプレイスメントグループを作成するには

`New-EC2PlacementGroup` コマンドを使用します。

# プレイスメントグループでのインスタンスの起動

[プレイスメントグループのルールと制限が満たされている場合 \(p. 794\)](#)、次のいずれかの方法を使用してプレイスメントグループ内でインスタンスを起動できます。

## Console

コンソールを使用してプレイスメントグループにインスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. [インスタンスの作成] を選択します。指示どおりにウィザードを完了し、次の操作を行うように注意します。
  - [Choose an Instance Type] ページはで、プレイスメントグループ内で起動できるインスタンスタイプを選択します。
  - [Configure Instance Details (インスタンスの詳細を設定)] ページでは、以下のフィールドがプレイスメントグループに適用できます。
    - [インスタンス数] で、このプレイスメントグループ内で必要なインスタンスの総数を入力します。これは、後でプレイスメントグループにインスタンスを追加できない場合があるためです。
    - [Placement group (プレイスメントグループ)] で、[インスタンスをプレイスメントグループに追加します] チェックボックスを選択します。このページに [プレイスメントグループ] が表示されない場合は、選択したインスタンスタイプがプレイスメントグループ内で起動できるタイプであることを確認してください。それ以外の場合、このオプションは使用できません。
    - [プレイスメントグループ名] で、既存のプレイスメントグループあるいは作成した新しいプレイスメントグループのどちらにインスタンスを追加するかを選択します。
    - [プレイスメントグループ戦略] では、適切な戦略を選択します。[partition] を選択した場合は、[Target partition] で [Auto distribution] を選択し、Amazon EC2 により、グループ内のすべてのパーティションにインスタンスができるだけ均等に分散します。または、インスタンスを起動するパーティションを指定します。

## AWS CLI

AWS CLI を使用してプレイスメントグループ内でインスタンスを起動するには

`run-instances` コマンドを使用し、`--placement "GroupName = my-cluster"` パラメータを使用してプレイスメントグループ名を指定します。次の例で、プレイスメントグループ名は `my-cluster` です。

```
aws ec2 run-instances --placement "GroupName = my-cluster"
```

AWS CLI を使用してパーティションプレイスメントグループの特定のパーティション内でインスタンスを起動するには

`run-instances` コマンドを使用して、`--placement "GroupName = HDFS-Group-A, PartitionNumber = 3"` パラメータを使用するグループプレイスメントグループ名とパーティションを指定します。この例では、パーティションプレイスメントグループは `HDFS-Group-A` という名で、パーティション数は 3 です。

```
aws ec2 run-instances --placement "GroupName = HDFS-Group-A, PartitionNumber = 3"
```

## PowerShell

AWS Tools for Windows PowerShell を使用してプレイスメントグループ内でインスタンスを起動するには

`New-EC2Instance` コマンドを使用し、`-Placement_GroupName` パラメータを使用してプレイスメントグループ名を指定します。

## プレイスメントグループのインスタンスを説明する

次のいずれかの方法を使用して、インスタンスのプレイスメント情報を表示できます。AWS CLI を使用して、パーティション番号でパーティションプレイスメントグループをフィルターすることもできます。

### Console

コンソールを使用してインスタンスのプレイスメントグループとパーティション番号を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、詳細ペインの [プレイスメントグループ] を確認します。プレイスメントグループにインスタンスがない場合、フィールドは空になります。それ以外の場合は、プレイスメントグループ名が表示されます。プレイスメントグループがパーティションプレイスメントグループの場合、このインスタンスのパーティション番号の [Partition number (パーティション番号)] を調べます。

### AWS CLI

AWS CLI を使用してパーティションプレイスメントグループのインスタンスのパーティション番号を表示するには

`describe-instances`コマンドを使用して `--instance-id` パラメータを指定します。

```
aws ec2 describe-instances --instance-id i-0123a456700123456
```

レスポンスにはプレイスメント情報が含まれています。この情報にはインスタンスのプレイスメントグループ名とパーティション番号が含まれます。

```
"Placement": {  
    "AvailabilityZone": "us-east-1c",  
    "GroupName": "HDFS-Group-A",  
    "PartitionNumber": 3,  
    "Tenancy": "default"  
}
```

AWS CLI を使用して特定のパーティションプレイスメントグループとパーティション番号のインスタンスをフィルターするには

`describe-instances` コマンドを使用して、`placement-group-name` および `placement-partition-number` フィルターを持つ `--filters` パラメータを指定します。この例では、パーティションプレイスメントグループは HDFS-Group-A という名で、パーティション数は 7 です。

```
aws ec2 describe-instances --filters "Name = placement-group-name, Values = HDFS-Group-A" "Name = placement-partition-number, Values = 7"
```

レスポンスは、指定されたプレイスメントグループ内の指定されたパーティション内にあるすべてのインスタンスをリストします。次の出力例は、返されたインスタンスのインスタンス ID、インスタンスタイプ、および配置情報のみを示しています。

```
"Instances": [  
    {  
        "InstanceId": "i-0albc23d4567e8f90",
```

```
"InstanceType": "r4.large",
},
"Placement": {
    "AvailabilityZone": "us-east-1c",
    "GroupName": "HDFS-Group-A",
    "PartitionNumber": 7,
    "Tenancy": "default"
}
{
    "InstanceId": "i-0a9b876cd5d4ef321",
    "InstanceType": "r4.large",
},
"Placement": {
    "AvailabilityZone": "us-east-1c",
    "GroupName": "HDFS-Group-A",
    "PartitionNumber": 7,
    "Tenancy": "default"
}
],
```

## インスタンスのプレイスメントグループの変更

インスタンスのプレイスメントグループは、次のいずれかの方法で変更できます。

- 既存のインスタンスをプレイスメントグループに移動する
- プレイスメントグループ間でインスタンスを移動する
- プレイスメントグループからインスタンスを削除する

インスタンスを移動または削除する前に、インスタンスを `stopped` 状態にする必要があります。インスタンスを移動または削除するには、AWS CLI または AWS SDK を使用できます。

### AWS CLI

AWS CLI を使用してプレイスメントグループにインスタンスを移動するには

- `stop-instances` コマンドを使用して、インスタンスを停止します。
- `modify-instance-placement` コマンドを使用し、インスタンスの移動先のプレイスメントグループの名前を指定します。

```
aws ec2 modify-instance-placement --instance-id i-0123a456700123456 --group-name MySpreadGroup
```

- `start-instances` コマンドを使用してインスタンスを起動します。

### PowerShell

AWS Tools for Windows PowerShell を使用してプレイスメントグループにインスタンスを移動するには

- `Stop-EC2Instance` コマンドを使用してインスタンスを停止します。
- `Edit-EC2InstancePlacement` コマンドを使用し、インスタンスの移動先のプレイスメントグループの名前を指定します。
- `Start-EC2Instance` コマンドを使用してインスタンスを起動します。

## AWS CLI

AWS CLI を使用してプレイスメントグループからインスタンスを削除するには

1. [stop-instances](#) コマンドを使用して、インスタンスを停止します。
2. [modify-instance-placement](#) コマンドを使用し、プレイスメントグループ名に空の文字列を指定します。

```
aws ec2 modify-instance-placement --instance-id i-0123a456700123456 --group-name ""
```

3. [start-instances](#) コマンドを使用してインスタンスを起動します。

## PowerShell

AWS Tools for Windows PowerShell を使用してプレイスメントグループからインスタンスを削除するには

1. [Stop-EC2Instance](#) コマンドを使用してインスタンスを停止します。
2. [Edit-EC2InstancePlacement](#) コマンドを使用し、プレイスメントグループ名に空の文字列を指定します。
3. [Start-EC2Instance](#) コマンドを使用してインスタンスを起動します。

# プレイスメントグループを削除する

プレイスメントグループを交換する必要がある場合、または不要になった場合は、そのプレイスメントグループを削除できます。プレイスメントグループを削除するには、次のいずれかの方法を使用できます。

### Important

削除するプレイスメントグループにはインスタンスが含まれていない必要があります。プレイスメントグループ内で起動したすべてのインスタンスを終了 (p. 547) し、インスタンスを別のプレイスメントグループに移動 (p. 799) するか、プレイスメントグループから削除 (p. 800) することができます。インスタンスを終了または移動する前にインスタンスがプレイスメントグループ内にあることを確認するには、[インスタンス] 画面でインスタンスを選択し、詳細ペインで [プレイスメントグループ] の値を確認します。

### 新しいコンソール

コンソールを使用してプレイスメントグループを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Placement Groups] を選択します。
3. プレイスマントグループを選択し、[削除] を選択します。
4. 確認を求められたら、「Delete」と入力し、[削除] を選択します。

### 古いコンソール

コンソールを使用してプレイスメントグループを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Placement Groups] を選択します。
3. プレイスマントグループを選択し、[Delete Placement Group] を選択します。

4. 確認を求めるメッセージが表示されたら、[削除] を選択します。

#### AWS CLI

AWS CLI を使用してプレイスメントグループを削除するには

`delete-placement-group` コマンドを使用し、削除するプレイスメントグループの名前を指定します。次の例で、プレイスメントグループ名は `my-cluster` です。

```
aws ec2 delete-placement-group --group-name my-cluster
```

#### PowerShell

AWS Tools for Windows PowerShell を使用してプレイスメントグループを削除するには

`Remove-EC2PlacementGroup` コマンドを使用してプレイスメントグループを削除します。

## EC2 インスタンスの最大ネットワーク送信単位 (MTU)

ネットワーク接続の最大送信単位 (MTU) とは、接続を介して渡すことができる最大許容パケットサイズ (バイト単位) です。接続の MTU が大きいほど、より多くのデータを単一のパケットで渡すことができます。イーサネットパケットは、フレーム (送信している実際のデータ) とそれを囲むネットワークオーバー ヘッド情報で構成されています。

イーサネットフレームの形式はさまざままで、最も一般的な形式は、標準イーサネット v2 フレーム形式です。これはインターネットのほとんどでサポートされている最大のイーサネットパケットサイズである 1500 MTU をサポートします。インスタンスでサポートされている最大 MTU は、インスタンスタイプによって異なります。すべての Amazon EC2 インスタンスタイプで 1500 MTU がサポートされており、現在の多くのインスタンスサイズで 9001 MTU またはジャンボフレームがサポートされています。

#### コンテンツ

- [ジャンボフレーム \(9001 MTU\) \(p. 801\)](#)
- [パス MTU 検出 \(p. 802\)](#)
- [2 つホスト間のパス MTU の確認 \(p. 802\)](#)
- [Linux インスタンス上の MTU の確認および設定 \(p. 803\)](#)
- [トラブルシューティング \(p. 804\)](#)

## ジャンボフレーム (9001 MTU)

ジャンボフレームでは、パケットあたりのペイロードサイズを拡張し、パケットオーバーヘッド以外のパケットの割合を高めることによって、1500 バイトを超えるデータを送信できます。同じ量の使用可能なデータを少ないパケットで送信することができます。ただし、特定の AWS リージョン (EC2-Classic)、1 つの VPC、または VPC ピア接続の外側では、最大パスが 1500 MTU になることがあります。VPN 接続およびインターネットゲートウェイを介して送信されるトラフィックは 1500 MTU に制限されます。パケットが 1500 バイト以上ある場合は、フラグメント化されます。または、`Don't Fragment` フラグが IP ヘッダーに設定されている場合は削除されます。

ジャンボフレームを、インターネットバウンドトラフィックや VPC を出るトラフィックに使用する場合は慎重に行ってください。パケットは中間システムによってフラグメント化されるため、このトラフィックの速度が低下します。VPC 外に向かうトラフィックの速度を低下させずに VPC 内のジャンボフレーム

を使用するには、ルートごとに MTU サイズを設定するか、または MTU サイズやルートの異なる複数の Elastic ネットワークインターフェイスを使用します。

クラスターープレイスマントグループ内にコロケーションされたインスタンスでは、考えられる最大のネットワークスループットの実現するうえでジャンボフレームが役立ちます。この場合は、ジャンボフレームを使用することが推奨されています。詳細については、「[プレイスメントグループ \(p. 791\)](#)」を参照してください。

AWS Direct Connect を経由した VPC とオンプレミスのネットワーク間のトラフィックにはジャンボフレームを使用できます。詳細や、ジャンボフレーム対応を確認する方法については、『AWS Direct Connect ユーザーガイド』の「[ネットワーク MTU 設定](#)」を参照してください。

すべての [現行世代のインスタンス \(p. 188\)](#) は、ジャンボフレームをサポートしています。以下の現行世代のインスタンスは、ジャンボフレームとして C3、G2、I2、M3、および R3 をサポートしています。

## パス MTU 検出

2 つのデバイス間のパス MTU を判断するために、パス MTU 検出が使用されます。パス MTU は、送信側ホストと受信側ホスト間のパスでサポートされている最大のパケットサイズです。ホストが受信側ホストの MTU よりも大きなパケット、またはデバイスの MTU よりも大きなパケットをパスに沿って送信する場合、受信側ホストまたはデバイスは ICMP メッセージ Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (タイプ 3、コード 4) を返します。これによって、パケットが送信できるようになるまで MTU を調整するように元のホストに指示されます。

デフォルトでは、セキュリティグループはインバウンド ICMP トラフィックを許可しません。インスタンスがこのメッセージを受信でき、パケットが削除されないようにするには、[Destination Unreachable] プロトコルをインスタンスのインバウンドセキュリティグループルールに設定したカスタム ICMP ルールを追加する必要があります。詳細については、「[パス MTU 検出のルール \(p. 923\)](#)」を参照してください。

### Important

インスタンスのセキュリティグループを変更してパス MTU 検出を有効にした場合、一部のルーターによってジャンボフレームが無視されることがあります。VPC のインターネットゲートウェイによって、最大 1500 バイトのパケットだけが転送されます。インターネットトラフィックには、1500 MTU パケットが推奨されています。

## 2 つホスト間のパス MTU の確認

tracepath コマンドを使用して 2 つのホスト間のパス MTU を確認できます。このコマンドは、Amazon Linux を含む、多くの Linux ディストリビューションでデフォルトで提供されている iputils パッケージの一部です。

tracepath を使用してパス MTU を確認するには

次のコマンドを使用して、EC2 インスタンスと別のホスト間のパス MTU を確認します。宛先として DNS 名または IP アドレスを使用できます。宛先が別の EC2 インスタンスの場合、セキュリティグループによりインバウンド UDP トラフィックが許可されていることを確認します。次の例では、EC2 インスタンスと amazon.com の間のパス MTU を確認します。

```
[ec2-user ~]$ tracepath amazon.com
1?: [LOCALHOST]          pmtu 9001
1:  ip-172-31-16-1.us-west-1.compute.internal (172.31.16.1)    0.187ms pmtu 1500
1:  no reply
2:  no reply
3:  no reply
4:  100.64.16.241 (100.64.16.241)                                0.574ms
5:  72.21.222.221 (72.21.222.221)                                84.447ms asymm 21
6:  205.251.229.97 (205.251.229.97)                            79.970ms asymm 19
7:  72.21.222.194 (72.21.222.194)                                96.546ms asymm 16
```

```
8: 72.21.222.239 (72.21.222.239)                                79.244ms asymmm 15
9: 205.251.225.73 (205.251.225.73)                                91.867ms asymmm 16
...
31: no reply
    Too many hops: pmtu 1500
    Resume: pmtu 1500
```

この例では、パス MTU は 1500 です。

## Linux インスタンス上の MTU の確認および設定

一部のインスタンスでは、ジャンボフレームを使用し、それ以外のドライバには標準フレームサイズを使用するように設定されています。VPC 内のネットワークトラフィックにはジャンボフレームを使用し、インターネットトラフィックには標準フレームを使用したいと思われるかもしれません。いずれにしても、予想したとおりにインスタンスが動作することを確認することをお勧めします。このセクションの手順に従って、ネットワークインターフェイスの MTU 設定を確認し、必要に応じてそれらを変更することができます。

Linux インスタンス上の MTU 設定を確認するには

以下の ip コマンドを使用して、現在の MTU 値を確認できます。出力例では、**mtu 9001** が、このインスタンスにジャンボフレームが使用されていると示していることに注意してください。

```
[ec2-user ~]$ ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode DEFAULT
    group default qlen 1000
        link/ether 02:90:c0:b7:9e:d1 brd ff:ff:ff:ff:ff:ff
```

Linux インスタンス上の MTU 値を設定するには

1. MTU 値は、ip コマンドを使用して設定できます。次のコマンドで、目的の MTU 値を 1500 に設定できますが、代わりに 9001 を使用します。

```
[ec2-user ~]$ sudo ip link set dev eth0 mtu 1500
```

2. (オプション) 再起動後もネットワーク MTU 設定を維持するには、オペレーティングシステムのタイプに基づいて、次の設定ファイルを変更します。

- Amazon Linux 2 の場合、次の行を /etc/sysconfig/network-scripts/ifcfg-eth0 ファイルに追加します。

```
MTU=1500
```

次の行を /etc/dhcp/dhclient.conf ファイルに追加します。

```
request subnet-mask, broadcast-address, time-offset, routers, domain-name, domain-search, domain-name-servers, host-name, nis-domain, nis-servers, ntp-servers;
```

- Amazon Linux の場合は、以下の行を /etc/dhcp/dhclient-eth0.conf ファイルに追加します。

```
interface "eth0" {
    supersede interface-mtu 1500;
}
```

- その他の Linux ディストリビューションの場合は、特定のドキュメントを参照してください。

3. (オプション) インスタンスを再起動し、MTU 設定が正しいことを確認します。

## トラブルシューティング

ジャンボフレームを使用したときに EC2 インスタンスと Amazon Redshift クラスター間の接続に問題が発生する場合は、『Amazon Redshift Cluster Management Guide』の「クエリがハンギングしたようになる」を参照してください。

## Virtual Private Cloud

Amazon Virtual Private Cloud ( Amazon VPC ) を使用すると、AWS クラウド内の独自の論理的に分離された領域の仮想ネットワーク ( Virtual Private Cloud ( VPC ) とも呼ばれます ) を定義できます。Amazon EC2 のリソース ( インスタンスなど ) を VPC サブネット内部で起動できます。VPC は、お客様自身のデータセンターで運用されている従来のネットワークによく似ていますが、AWS からスケーラブルなインフラストラクチャを使用できるというメリットがあります。お客様の VPC はお客様が設定できます。IP アドレスレンジの選択、サブネットの作成、ルートテーブル、ネットワークゲートウェイ、セキュリティの設定ができます。VPC のインスタンスをインターネットまたは独自のデータセンターに接続できます。

AWS アカウントを作成すると、各リージョンのデフォルト VPC が作成されます。デフォルトの VPC は、設定済みですぐに使用できる VPC です。デフォルトの VPC にすぐにインスタンスを起動できます。または、必要に応じた独自でデフォルト以外の VPC を作成および設定をすることができます。

2013 年 12 月 4 日以前に AWS アカウントを作成した場合は、リージョンによっては EC2-Classic プラットフォームのサポートがある場合もあります。2013 年 12 月 4 日以降に AWS アカウントを作成した場合は、EC2-Classic をサポートしていないため、VPC でリソースを起動する必要があります。詳細については、「[EC2-Classic \(p. 804\)](#)」を参照してください。

## Amazon VPC ドキュメント

Amazon VPC の詳細については、次のドキュメントを参照してください。

| ガイド                                               | 説明                                          |
|---------------------------------------------------|---------------------------------------------|
| <a href="#">Amazon VPC ユーザーガイド</a>                | Amazon VPC の主要な概念を説明し、その機能を使用するための手順を説明します。 |
| <a href="#">Amazon VPC Peering Guide</a>          | VPC ピアリングの主要な概念を説明し、使用するための手順を説明します。        |
| <a href="#">AWS Site-to-Site VPN ネットワーク管理者ガイド</a> | ネットワーク管理者がカスタマーゲートウェイを設定する際に役立ちます。          |

## EC2-Classic

EC2-Classic は、お客様のインスタンスは他のユーザー様と共有する単一のフラットネットワーク内で稼働します。Amazon VPC は、お客様のインスタンスはご自分の AWS アカウントから論理的に独立した仮想プライベートクラウド (VPC) 内で稼働します。

EC2-Classic プラットフォームは、Amazon EC2 のオリジナルリリースで導入されました。2013 年 12 月 4 日以降に AWS アカウントを作成した場合は、EC2-Classic をサポートしていないため、VPC で Amazon EC2 インスタンスを起動する必要があります。

アカウントが EC2-Classic をサポートしていない場合は、デフォルトの VPC が作成されます。デフォルトでは、インスタンスを起動すると、デフォルトの VPC でインスタンスが起動されます。または、デフォルト以外の VPC を作成して、インスタンスを起動するときに VPC を指定することもできます。

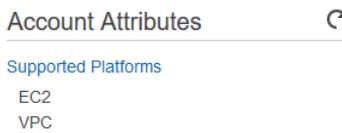
## サポートされるプラットフォームの検出

Amazon EC2 コンソールを見れば、指定したリージョンでインスタンスを起動可能なプラットフォームや、そのリージョンにデフォルト VPC があるかどうかがわかります。

使用するリージョンがナビゲーションバーで選択されていることを確認してください。Amazon EC2 コンソールダッシュボード上で、[Account Attributes (アカウント属性)] の下にある [Supported Platforms (サポートされるプラットフォーム)] を探します。

### EC2-Classic をサポートするアカウント

ダッシュボードの [Account Attributes (アカウント属性)] に次のように表示される場合は、このリージョンでアカウントが EC2-Classic プラットフォームと VPC の両方をサポートし、デフォルトの VPC がないことを示します。



`describe-account-attributes` コマンドの出力には、`supported-platforms` 属性の `EC2` 値と `VPC` 値が含まれます。

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
{
    "AccountAttributes": [
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
            ]
        }
    ]
}
```

### VPC が必要なアカウント

ダッシュボードの [Account Attributes (アカウント属性)] に次のように表示される場合は、アカウントがこのリージョンのインスタンスを起動するために VPC を必要とし、このリージョンの EC2-Classic プラットフォームをサポートしていないこと、識別子 `vpc-1a2b3c4d` を持つデフォルトの VPC があることを示します。



`describe-account-attributes` コマンドの出力には、`supported-platforms` 属性の `VPC` 値のみが含まれています。

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

```
{  
    "AccountAttributes": [  
        {  
            "AttributeValues": [  
                {  
                    "AttributeValue": "VPC"  
                }  
            ]  
            "AttributeName": "supported-platforms",  
        }  
    ]  
}
```

## EC2-Classic で利用可能なインスタンスタイプ

より新しいインスタンスタイプのほとんどには VPC が必要です。EC2-Classic でサポートされているインスタンスタイプは次のタイプのみです。

- 汎用: M1、M3、T1
- コンピューティングの最適化: C1、C3、CC2
- メモリの最適化: CR1、M2、R3
- ストレージの最適化: D2、HS1、I2
- 高速コンピューティング: G2

アカウントで EC2-Classic がサポートされる場合で、デフォルト以外の VPC を作成していない場合、次のいずれかを行って VPC を必要とするインスタンスを起動できます。

- サブネット ID またはネットワークインターフェイス ID をリクエストで指定して、デフォルト以外の VPC を作成し、VPC 専用インスタンスを起動します。デフォルトの VPC がない場合に、AWS CLI、Amazon EC2 API、または AWS SDK を使用して VPC 専用インスタンスを起動するには、デフォルト以外の VPC を作成する必要があります。
- Amazon EC2 コンソールを使用して VPC 専用インスタンスを起動します。Amazon EC2 コンソールによって、アカウントにデフォルト以外の VPC が作成され、最初のアベイラビリティーゾーン内のサブネットにインスタンスが起動します。コンソールによって、次の属性を持つ VPC が作成されます。
  - 各アベイラビリティーゾーンにパブリック IPv4 アドレス指定属性が `true` に設定されたサブネットが 1 つずつ。これにより、インスタンスがパブリック IPv4 アドレスを受け取ることができるようになります。詳細については、『Amazon VPC ユーザーガイド』の「[VPC の IP アドレス指定](#)」を参照してください。
  - インターネットゲートウェイ、およびインターネットゲートウェイに VPC のトラフィックをルーティングするメインルートテーブル。これにより、VPC で起動したインスタンスがインターネット経由で通信できるようになります。詳細については、『Amazon VPC ユーザーガイド』の「[インターネットゲートウェイ](#)」を参照してください。
  - VPC のデフォルトのセキュリティグループおよび各サブネットに関連付けられたデフォルトのネットワーク ACL。詳細については、Amazon VPC ユーザーガイドの「[VPC のセキュリティグループ](#)」を参照してください。

EC2-Classic にその他のリソースがある場合は、それらを VPC に移行する手順を実行することができます。詳細については、『[EC2-Classic の Linux インスタンスから VPC の Linux インスタンスへの移行 \(p. 825\)](#)』を参照してください。

## EC2-Classic と VPC の違い

次の表は、EC2-Classic で起動したインスタンス、デフォルト VPC で起動したインスタンス、非デフォルト VPC で起動したインスタンスの違いをまとめたものです。

| 特徴                                           | EC2-Classic                                                              | デフォルト VPC                                                                                                          | デフォルトではない VPC                                                                                                      |
|----------------------------------------------|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| パブリック IPv4 アドレス (Amazon のパブリック IP アドレスプールより) | インスタンスは、EC2-Classic パブリック IPv4 アドレスプールからパブリック IPv4 アドレスを受け取ります。          | デフォルトのサブネットで起動されたインスタンスは、起動時に特に IPv4 アドレスを指定しない場合、またはサブネットのパブリック IPv4 アドレス属性を変更しない場合、デフォルトでパブリック IPv4 アドレスを受け取ります。 | インスタンスは、起動時に特に IPv4 アドレスを指定しない場合、またはサブネットのパブリック IP アドレス属性を変更しない場合、デフォルトではパブリック IPv4 アドレスを受け取れません。                  |
| プライベート IPv4 アドレス                             | インスタンスは、起動するたびに、EC2-Classic のアドレス範囲からプライベート IPv4 アドレスを受け取ります。            | インスタンスはデフォルト VPC のアドレス範囲から静的プライベート IPv4 アドレスを受け取ります。                                                               | インスタンスは VPC のアドレス範囲から静的プライベート IPv4 アドレスを受け取れます。                                                                    |
| 複数のプライベート IPv4 アドレス                          | 1 つのインスタンスには 1 つのプライベート IP アドレスを選択します。複数の IP アドレスはサポートされません。             | 複数のプライベート IPv4 アドレスを 1 つのインスタンスに割り当てることができます。                                                                      | 複数のプライベート IPv4 アドレスを 1 つのインスタンスに割り当てることができます。                                                                      |
| Elastic IP アドレス (IPv4)                       | 停止すると、Elastic IP とインスタンスの関連付けが解除されます。                                    | 停止しても、Elastic IP とインスタンスの関連付けが維持されます。                                                                              | 停止しても、Elastic IP とインスタンスの関連付けが維持されます。                                                                              |
| Elastic IP アドレスを関連付ける                        | Elastic IP アドレスはインスタンスに関連付けます。                                           | Elastic IP アドレスはネットワークインターフェイスのプロパティの 1 つです。Elastic IP アドレスをインスタンスに割り当てるには、そのインスタンスにアタッチされているネットワークインターフェイスを更新します。 | Elastic IP アドレスはネットワークインターフェイスのプロパティの 1 つです。Elastic IP アドレスをインスタンスに割り当てるには、そのインスタンスにアタッチされているネットワークインターフェイスを更新します。 |
| Elastic IP アドレスの関連付けを解除する                    | 既に別のインスタンスに関連付けられている Elastic IP アドレスを関連付けると、アドレスは自動的に新しいインスタンスに関連付けられます。 | 既に別のインスタンスに関連付けられている Elastic IP アドレスを関連付けると、アドレスは自動的に新しいインスタンスに関連付けられます。                                           | Elastic IP アドレスがすでに別のインスタンスに関連付けられている場合は、再接続を許可した場合にのみ成功します。                                                       |
| Elastic IP アドレスのタグ付け                         | Elastic IP アドレスにタグを適用することはできません。                                         | Elastic IP アドレスにタグを適用することができます。                                                                                    | Elastic IP アドレスにタグを適用することができます。                                                                                    |
| DNS ホスト名                                     | DNS ホスト名はデフォルトで有効化されています。                                                | DNS ホスト名はデフォルトで有効化されています。                                                                                          | DNS ホスト名はデフォルトで無効化されています。                                                                                          |
| セキュリティグループ                                   | セキュリティグループは、他の AWS アカウントに属するセキュリティグループを参照できます。                           | セキュリティグループは、VPC または VPC ピアリング接続のピア VPC のセキュリティグループを参照できます。                                                         | セキュリティグループは、VPC のみのセキュリティグループを参照できます。                                                                              |

| 特徴              | EC2-Classic                                                                                                                                                                                                                                     | デフォルト VPC                                                                                                                                | デフォルトではない VPC                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| セキュリティグループの関連付け | 実行中のインスタンスのセキュリティグループは変更できません。割り当て済みのセキュリティグループのルールを変更するか、インスタンスを新しいインスタンスと置き換える必要があります(置き換えるには、まずインスタンスから AMI を作成し、この AMI から、目的のセキュリティグループを使用して新しいインスタンスを起動し、元のインスタンスから Elastic IP アドレスに関連付けられているものをすべて解除し、新しいインスタンスに関連付け、その後、元のインスタンスを終了します)。 | 最大 5 つのセキュリティグループを 1 つのインスタンスに割り当てるすることができます。<br><br>セキュリティグループをインスタンスに割り当てるのは、インスタンスの起動時と実行中です。                                         | 最大 5 つのセキュリティグループを 1 つのインスタンスに割り当てるすることができます。<br><br>セキュリティグループをインスタンスに割り当てるのは、インスタンスの起動時と実行中です。            |
| セキュリティグループのルール  | インバウンドトラフィックのみにルールを追加できます。                                                                                                                                                                                                                      | インバウンドトラフィックとアウトバウンドトラフィックのルールを追加できます。                                                                                                   | インバウンドトラフィックとアウトバウンドトラフィックのルールを追加できます。                                                                      |
| テナント            | インスタンスは共有するハードウェアで実行されます。                                                                                                                                                                                                                       | 共有ハードウェアまたはシングルテナントハードウェアでインスタンスを実行できます。                                                                                                 | 共有ハードウェアまたはシングルテナントハードウェアでインスタンスを実行できます。                                                                    |
| インターネットにアクセスする  | インスタンスはインターネットにアクセスできます。インスタンスは自動的にパブリック IP アドレスを受信し、AWS ネットワークエッジを通してインターネットに直接アクセスできます。                                                                                                                                                       | デフォルトでは、インスタンスはインターネットにアクセスできます。インスタンスはデフォルトでパブリック IP アドレスを受け取ります。インターネットゲートウェイはデフォルトの VPC にアタッチされ、デフォルトのサブネットにはインターネットゲートウェイへのルートがあります。 | デフォルトでは、インスタンスはインターネットにアクセスできません。インスタンスはデフォルトでパブリック IP アドレスを受け取れません。VPC は、作成方法によってはインターネットゲートウェイを持つ場合があります。 |
| IPv6 アドレス指定     | IPv6 アドレス指定はサポートされていません。IPv6 アドレスをインスタンスに割り当てるることはできません。                                                                                                                                                                                        | オプションで、VPC に IPv6 CIDR ブロックを関連付け、VPC のインスタンスに IPv6 アドレスを割り当てるすることができます。                                                                  | オプションで、VPC に IPv6 CIDR ブロックを関連付け、VPC のインスタンスに IPv6 アドレスを割り当てるすることができます。                                     |

## EC2-Classic 用セキュリティグループ

EC2-Classic を使用している場合は、EC2-Classic 用に作成したセキュリティグループを使用する必要があります。EC2-Classic でインスタンスを起動する場合は、インスタンスと同じリージョンのセキュリティ

グループを指定する必要があります。EC2-Classic でインスタンスを起動する場合は、VPC 用に作成したセキュリティグループは指定できません。

EC2-Classic でインスタンスを起動した後でセキュリティグループを変更することはできません。ただし、セキュリティグループルールの追加または削除は可能です。これらの変更は、セキュリティグループに関連付けられているすべてのインスタンスに短時間で自動的に適用されます。

AWS アカウントには、EC2-Classic のリージョンごとにデフォルトのセキュリティグループが自動的に設定されます。デフォルトセキュリティグループを削除しようとした場合、次のエラーが発生します: Client.InvalidGroup.Reserved: The security group 'default' is reserved.

カスタムセキュリティグループを作成できます。セキュリティグループ名はリージョンのアカウント内で一意である必要があります。EC2-Classic で使用するセキュリティグループを作成するには、VPC に [No VPC (VPC なし)]を選択します。

デフォルトおよびカスタムセキュリティグループにインバウンドルールを追加できます。EC2-Classic セキュリティグループのアウトバウンドルールは変更できません。セキュリティグループルールを作成するときは、発信元または宛先と同じリージョン内の EC2-Classic に別のセキュリティグループを使用できます。別の AWS アカウントのセキュリティグループを指定するには、111122223333/sg-edcd9784 のように、AWS アカウント ID を接頭辞として追加します。

EC2-Classic では、アカウントごとに各リージョンに最大 500 のセキュリティグループを持つことができます。ことができ、また 1 つのセキュリティグループには最大 100 のルールを追加できます。

## IP アドレスの割り当てと DNS

Amazon は、Amazon が提供する IPv4 DNS ホスト名を解決する DNS サーバーを IPv4 アドレスに提供します。EC2-Classic で、Amazon DNS サーバーは 172.16.0.23 にあります。

EC2-Classic でカスタムファイアウォール構成を作成した場合、Amazon DNS サーバーのアドレスのポート 53 (DNS) からのインバウンドトラフィック (送信先はエフェメラル範囲のポート) を許可するルールをファイアウォールに作成する必要があります。そうしない場合、インスタンスからの内部 DNS 解決に失敗します。ファイアウォールが自動的に DNS クエリレスポンスを自動的に許可しない場合は、Amazon DNS サーバーの IP アドレスからのトラフィックを許可する必要があります。Amazon DNS サーバーの IP アドレスを取得するには、インスタンス内から以下のコマンドを使用します。

```
grep nameserver /etc/resolv.conf
```

## Elastic IP アドレス

アカウントで EC2-Classic がサポートされている場合、EC2-Classic プラットフォームで使用する Elastic IP アドレスのプールと、VPC で使用する別のプールがあります。VPC で使用するために割り当てた Elastic IP アドレスを EC2-Classic のインスタンスに関連付けることや、その逆を行なうことはできません。ただし、EC2-Classic プラットフォームで使用するために割り当てた Elastic IP アドレスを VPC での使用に移行することはできます。Elastic IP アドレスを別のリージョンに移行することはできません。

コンソールを使用して、EC2-Classic で使用する Elastic IP アドレスを割り当てるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. [Allocate new address] を選択します。
4. [Classic] を選択し、次に [Allocate] を選択します。確認画面を閉じます。

## EC2-Classic からの Elastic IP アドレスの移行

アカウントで EC2-Classic がサポートされている場合、EC2-Classic プラットフォームで使用するために割り当てた Elastic IP アドレスを、同じリージョン内の VPC での使用に移行することができます。これ

は、リソースを EC2-Classic から VPC に移行するのに役立ちます。たとえば、VPC で新しいウェブサーバーを起動した後、EC2-Classic でウェブサーバーに使用していたのと同じ Elastic IP アドレスを新しい VPC ウェブサーバーに使用できます。

Elastic IP アドレスを VPC に移行した後は、EC2-Classic で使用することはできません。ただし、必要に応じて EC2-Classic に復元することができます。もともと VPC で使用するために割り当てられていた Elastic IP アドレスを EC2-Classic に移行することはできません。

Elastic IP アドレスを移行する場合、インスタンスに関連付けないでください。インスタンスからの Elastic IP アドレスの関連付け解除の詳細については、「[Elastic IP アドレスの関連付け解除 \(p. 710\)](#)」を参照してください。

アカウントに設定できる数であれば、EC2-Classic Elastic IP アドレスはいくつでも移行できます。ただし、Elastic IP アドレスを移行すると、VPC の Elastic IP アドレス制限にカウントされます。Elastic IP アドレスを移行した結果として制限を超過する場合は、移行できません。同様に、Elastic IP アドレスを EC2-Classic に復元すると、EC2-Classic の Elastic IP アドレス制限にカウントされます。詳細については、「[Elastic IP アドレスの制限 \(p. 712\)](#)」を参照してください。

24 時間以内にアカウントに割り当てられた Elastic IP アドレスを移行することはできません。

Amazon EC2 コンソールまたは Amazon VPC コンソールを使用して、EC2-Classic から Elastic IP アドレスを移動できます。このオプションは、アカウントで EC2-Classic をサポートしている場合にのみ使用できます。

Amazon EC2 コンソールを使用して Elastic IP アドレスを移動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. Elastic IP アドレスを選択し、[Actions]、[Move to VPC scope] の順に選択します。
4. 確認ダイアログボックスで、[Move Elastic IP] を選択します。

Amazon EC2 コンソールまたは Amazon VPC コンソールを使用して Elastic IP アドレスを EC2-Classic に復元できます。

Amazon EC2 コンソールを使用して Elastic IP アドレスを EC2-Classic に復元するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. Elastic IP アドレスを選択し、[Actions]、[Restore to EC2 scope] の順に選択します。
4. 確認ダイアログボックスで [Restore] を選択します。

Elastic IP アドレスを移動または復元するコマンドを実行すると、Elastic IP アドレスの移行プロセスには数分かかる場合があります。Elastic IP アドレスがまだ移動中であるか、移動が完了したかを確認するには、[describe-moving-addresses](#) コマンドを使用します。

Elastic IP アドレスを移動すると、[Elastic IPs] ページの [Allocation ID] フィールドで割り当て ID を確認できます。

Elastic IP アドレスの移動中状態が 5 分を超えた場合は、[プレミアムサポート](#) にお問い合わせください。

コマンドラインを使用して Elastic IP アドレスを移動するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [move-address-to-vpc](#) (AWS CLI)
- [Move-EC2AddressToVpc](#) (AWS Tools for Windows PowerShell)

コマンドラインを使用して Elastic IP アドレスを EC2-Classic に復元するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [restore-address-to-classic \(AWS CLI\)](#)
- [Restore-EC2AddressToClassic \(AWS Tools for Windows PowerShell\)](#)

コマンドラインを使用してアドレスの移動ステータスを表示するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-moving-addresses \(AWS CLI\)](#)
- [Get-EC2Address \(AWS Tools for Windows PowerShell\)](#)

## EC2-Classic と VPC との間でのリソースの共有とアクセス

AWS アカウントのリソースと機能は、ClassicLink などを通じて、EC2-Classic と VPC のプラットフォーム間で共有できます。詳細については、「[ClassicLink \(p. 812\)](#)」を参照してください。

アカウントが EC2-Classic をサポートしている場合、EC2-Classic 用にリソースの設定を行った場合があります。EC2-Classic から VPC に移行する場合は、VPC 内でこれらのリソースを再作成する必要があります。EC2-Classic から VPC への移行に関する詳細については、「[EC2-Classic の Linux インスタンスから VPC の Linux インスタンスへの移行 \(p. 825\)](#)」を参照してください。

以下のリソースは、EC2-Classic と VPC の間で共有したりアクセスすることができます。

| リソース                   | コメント                                                                                                                                                                                                                                     |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AMI                    |                                                                                                                                                                                                                                          |
| バンドルタスク                |                                                                                                                                                                                                                                          |
| EBS ボリューム              |                                                                                                                                                                                                                                          |
| Elastic IP アドレス (IPv4) | Elastic IP アドレスを EC2-Classic から VPC に移行できます。もともと VPC で使用するために割り当てられていた Elastic IP アドレスを EC2-Classic に移行することはできません。詳細については、「 <a href="#">EC2-Classic からの Elastic IP アドレスの移行 (p. 809)</a> 」を参照してください。                                      |
| インスタンス                 | EC2-Classic インスタンスは、パブリック IPv4 アドレスを使用して VPC 内のインスタンスと通信できます。または、ClassicLink を使用して、プライベート IPv4 アドレス経由の通信を有効にすることができます。<br><br>インスタンスを EC2-Classic から VPC に移行することはできません。しかし、アプリケーションを EC2-Classic のインスタンスから VPC のインスタンスに移行することはできます。詳細につい |

| リソース                | コメント                                                                                                                                                                                                                                                                                                 |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | ては、「EC2-Classic の Linux インスタンスから VPC の Linux インスタンスへの移行 (p. 825)」を参照してください。                                                                                                                                                                                                                          |
| キーペア                |                                                                                                                                                                                                                                                                                                      |
| ロードバランサー            | ClassicLink を使用している場合、ロードバランサーのあるリンクされた EC2-Classic インスタンスを VPC に登録することができます。こうしてこの VPC はそのインスタンスと同じアベイラビリティーゾーンにサブネットがあることになります。<br><br>ロードバランサーを EC2-Classic から VPC に移行することはできません。EC2-Classic でロードバランサーがある VPC にインスタンスを登録することはできません。                                                               |
| 配置グループ <sup>*</sup> |                                                                                                                                                                                                                                                                                                      |
| リザーブドインスタンス         | リザーブドインスタンスのネットワークプラットフォームを EC2-Classic から VPC に変更できます。詳細については、「リザーブドインスタンスの変更 (p. 306)」を参照してください。                                                                                                                                                                                                   |
| セキュリティグループ          | リンクされた EC2-Classic インスタンスは、VPC からのトラフィックを監視するためには、ClassicLink を通して VPC のセキュリティグループを使用することができます。VPC インスタンスは、EC2-Classic セキュリティグループを使用できません。<br><br>EC2-Classic から VPC にセキュリティグループを移行することはできません。EC2-Classic のセキュリティグループのルールを VPC のセキュリティグループにコピーすることはできます。詳細については、「セキュリティグループを作成する (p. 915)」を参照してください。 |
| スナップショット            |                                                                                                                                                                                                                                                                                                      |

EC2-Classic と VPC の間で以下のリソースを共有したり、移動することはできません。

- スポットインスタンス

## ClassicLink

ClassicLink を使用すると、EC2-Classic インスタンスと同じリージョンにある自アカウントの VPC にリンクできます。VPC のセキュリティグループを EC2-Classic インスタンスに関連付けると、プライベート IPv4 アドレスを使用して EC2-Classic インスタンスと VPC 内のインスタンスが通信できるようになります。ClassicLink により、パブリック IPv4 アドレスや Elastic IP アドレスを使用しなくても、これらのプラットフォーム内のインスタンス間で通信できます。

ClassicLink は、EC2-Classic プラットフォームをサポートするアカウントを持つすべてのユーザーが利用でき、任意のインスタンスタイプの EC2-Classic インスタンスで使用できます。VPC へのリソース

の移行の詳細については、「[EC2-Classic の Linux インスタンスから VPC の Linux インスタンスへの移行 \(p. 825\)](#)」を参照してください。

ClassicLink は追加料金なしで使用できます。データ転送とインスタンスの使用に対する標準料金が適用されます。

## コンテンツ

- [ClassicLink の基本 \(p. 813\)](#)
- [ClassicLink の制限事項 \(p. 815\)](#)
- [ClassicLink の操作 \(p. 816\)](#)
- [ClassicLink の IAM ポリシー例 \(p. 820\)](#)
- [例: 3 層ウェブアプリケーションの ClassicLink セキュリティグループ設定 \(p. 822\)](#)

## ClassicLink の基本

EC2-Classic を使用して ClassicLink インスタンスを VPC にリンクする場合、2 つの作業が含まれます。まず、ClassicLink 用に VPC を有効にする必要があります。デフォルトでは、アカウントのすべての VPC は、その分離を維持するために ClassicLink 用に有効になっていません。ClassicLink 用に VPC を有効になると、アカウント内の同じリージョンにある任意の実行中の EC2-Classic インスタンスをその VPC にリンクすることができます。インスタンスをリンクするには、VPC からセキュリティグループを選択して、EC2-Classic インスタンスに関連付ける作業も必要です。インスタンスをリンクすると、VPC セキュリティグループで許可されている場合は、プライベート IP アドレスを使用して VPC 内のインスタンスと通信できます。EC2-Classic インスタンスは、VPC にリンクされたときに、プライベート IP アドレスを失いません。

### Note

インスタンスを VPC にリンクすることを、インスタンスをアタッチするということもあります。

リンクされた EC2-Classic インスタンスは VPC 内のインスタンスと通信できますが、VPC の一部ではありません。たとえば、`DescribeInstances` API リクエストでインスタンスを表示して VPC でフィルタリングする場合や、Amazon EC2 コンソールの [Instances] 画面を使用して VPC でフィルタリングする場合、その結果には VPC にリンクされた EC2-Classic インスタンスは含まれません。リンクされた EC2-Classic インスタンスの表示の詳細については、「[ClassicLink が有効な VPC とリンクされたインスタンスを表示する \(p. 818\)](#)」を参照してください。

デフォルトでは、リンクされた EC2-Classic インスタンスから VPC のインスタンスに対応するためにパブリック DNS ホスト名を使用する場合、ホスト名はインスタンスのパブリック IP アドレスに解決されます。VPC のインスタンスからリンクされた EC2-Classic インスタンスに対応するためにパブリック DNS ホスト名を使用する場合も同じになります。パブリック DNS ホスト名をプライベート IP アドレスに解決するには、VPC の ClassicLink DNS サポートを有効にできます。詳細については、「[ClassicLink DNS サポートの有効化 \(p. 819\)](#)」を参照してください。

インスタンスと VPC の間の ClassicLink 接続が不要になった場合、VPC から EC2-Classic インスタンスのリンクを解除できます。これにより、EC2-Classic インスタンスから VPC セキュリティグループの関連付けが解除されます。リンクされた EC2-Classic インスタンスは、停止されたときに自動的に VPC からリンク解除されます。VPC からすべてのリンクされた EC2-Classic インスタンスのリンクを解除した後、VPC の ClassicLink を無効にすることができます。

## AWS による VPC 内の他の ClassicLink サービスの使用

リンクされた EC2-Classic インスタンスは、VPC 内の AWS、Amazon Redshift、Amazon ElastiCache、および Elastic Load Balancing の各 Amazon RDS サービスにアクセスできます。ただし、VPC 内のインスタンスは、AWS を使用して、EC2-Classic プラットフォームでプロビジョニングされる ClassicLink サービスにアクセスすることはできません。

Elastic Load Balancing を使用している場合は、リンクされた EC2-Classic インスタンスをロードバランサーに登録することができます。ClassicLink が有効な VPC にロードバランサーを作成し、インスタンスが実行されるアベイラビリティーボーンを有効にする必要があります。リンクされた EC2-Classic インスタンスを終了する場合、ロードバランサーはインスタンスの登録を解除します。

Amazon EC2 Auto Scaling を使用する場合、起動時に指定した Amazon EC2 Auto Scaling が有効な VPC に自動的にリンクされるインスタンスを含む ClassicLink グループを作成できます。詳細については、『Amazon EC2 Auto Scaling ユーザーガイド』の「[EC2-Classic インスタンスの VPC へのリンク](#)」を参照してください。

VPC で Amazon RDS インスタンスまたは Amazon Redshift クラスターを使用し、パブリックにアクセス可能（インターネットからアクセス可能）である場合、リンクされた EC2-Classic インスタンスからこれらのリソースに対応するために使用するエンドポイントは、デフォルトでパブリック IP アドレスに解決されます。これらのリソースがパブリックにアクセス可能でない場合、エンドポイントはプライベート IP アドレスに解決されます。ClassicLink を使用してプライベート IP 経由でパブリックにアクセス可能な RDS インスタンスまたは Redshift クラスターに対応するには、プライベート IP アドレスまたはプライベート DNS ホスト名を使用するか、ClassicLink VPC の DNS サポートを有効にする必要があります。

プライベート DNS ホスト名またはプライベート IP アドレスを使用して RDS インスタンスに対応する場合、リンクされた EC2-Classic インスタンスは、マルチ AZ 配置に使用できるフェイルオーバーのサポートを使用することはできません。

Amazon EC2 コンソールを使用して、Amazon Redshift、Amazon ElastiCache、または Amazon RDS リソースのプライベート IP アドレスを検索できます。

VPC 内の AWS リソースのプライベート IP アドレスを見つけるには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. [Description] 列のネットワークインターフェイスの説明を確認します。Amazon Redshift、Amazon ElastiCache、Amazon RDS で使用されるネットワークインターフェイスには、説明に含まれるサービスの名前が付けられます。たとえば、Amazon RDS インスタンスにアタッチされるネットワークインターフェイスの説明は、RDSNetworkInterface のようになります。
4. 必要なネットワークインターフェイスを選択します。
5. 詳細ペインで、[Primary private IPv4 IP] フィールドからプライベート IP アドレスを取得します。

## ClassicLink の使用の管理

デフォルトでは、IAM ユーザーには ClassicLink を使用するためのアクセス許可がありません。IAM 用の VPC の有効化と無効化、ClassicLink が有効な VPC へのインターフェイスのリンクとリンク解除、および ClassicLink が有効な VPC とリンクされた ClassicLink インスタンスの表示の権限をユーザーに付与する EC2-Classic ポリシーを作成できます。Amazon EC2 の IAM ポリシーの詳細については、「[Amazon EC2 の IAM ポリシー \(p. 842\)](#)」を参照してください。

ClassicLink を操作するためのポリシーの詳細については、次の例 ([ClassicLink の IAM ポリシー例 \(p. 820\)](#)) を参照してください。

## ClassicLink のセキュリティグループ

EC2-Classic インスタンスを VPC にリンクしても、EC2-Classic セキュリティグループには影響しません。これらのセキュリティグループは、引き続きインスタンスに入り出すすべてのトラフィックを管理します。ただし、VPC 内のインスタンスに入り出すトラフィックは例外で、EC2-Classic インスタンスに関連付けられた VPC セキュリティグループによって管理されます。同じ VPC にリンクされた複数の EC2-Classic インスタンスは、同じ VPC セキュリティグループに関連付けられているかどうかに関係なく、VPC を介して相互に通信できません。EC2-Classic インスタンス間の通信は、それらのインスタンスに関連付けられた EC2-Classic セキュリティグループによって制御されます。セキュリティグループ設定

の例については、[例: 3 層ウェブアプリケーションの ClassicLink セキュリティグループ設定 \(p. 822\)](#)を参照してください。

VPC にインスタンスをリンクすると、インスタンスに関連付けられる VPC セキュリティグループを変更することはできなくなります。インスタンスに別のセキュリティグループを関連付けるには、最初にインスタンスのリンクを解除し、次にもう一度 VPC にリンクして、必要なセキュリティグループを選択する必要があります。

## ClassicLink のルーティング

ClassicLink 用に VPC を有効にすると、VPC のすべてのルートテーブルに、送信先が 10.0.0.0/8 で、ターゲットが local である静的ルートが追加されます。これによって、VPC 内のインスタンスと、VPC にリンクされている EC2-Classic インスタンスとの間で通信が可能になります。ClassicLink が有効な VPC にカスタムルートテーブルを追加する場合、送信先が 10.0.0.0/8 で、ターゲットが local である静的ルートが自動的に追加されます。VPC の ClassicLink を無効にすると、このルートは VPC のすべてのルートテーブルから自動的に削除されます。

10.0.0.0/16 および 10.1.0.0/16 IP アドレス範囲にある VPC で ClassicLink を有効にできるのは、VPC の作成時に自動的に追加されたローカルルートを除き、10.0.0.0/8 IP アドレス範囲のルートテーブルに既存の静的ルートがない場合のみです。同様に、ClassicLink 用に VPC を有効にしている場合、10.0.0.0/8 IP アドレス範囲内のルートテーブルに、より詳細なルートを追加できない場合があります。

### Important

VPC CIDR ブロックがパブリックにルーティング可能な IP アドレス範囲である場合は、EC2-Classic インスタンスを VPC にリンクする前にセキュリティへの影響を考慮してください。たとえば、リンクされた EC2-Classic インスタンスが、VPC の IP アドレス範囲内にあるソース IP アドレスからサービス拒否 (DoS) リクエストによるフラッド攻撃を受けた場合、応答トライフィックは VPC に送信されます。[RFC 1918](#) に規定されているように、プライベート IP アドレスの範囲を使用して VPC を作成することを強くお勧めします。

ルートテーブルと VPC でのルーティングに関する詳細については、『Amazon VPC ユーザーガイド』の「[ルートテーブル](#)」を参照してください。

## ClassicLink の VPC ピア接続の有効化

2 つの VPC 間に VPC ピア接続があり、ClassicLink を介して 1 つまたは両方の VPC にリンクされた 1 つ以上の EC2-Classic インスタンスが存在する場合は、VPC ピア接続を拡大して、EC2-Classic インスタンスと VPC ピア接続の他方の側の VPC のインスタンス間の通信を有効にすることができます。これにより、EC2-Classic インスタンスと VPC のインスタンスは、プライベート IP アドレスを使用して通信することができます。これを行うには、ローカル VPC がピア VPC でリンクされた EC2-Classic インスタンスと通信できるようにするか、リンクされたローカル EC2-Classic インスタンスがピア VPC のインスタンスと通信できるようにします。

ローカル VPC が、ピア VPC 内のリンクされた EC2-Classic インスタンスと通信できるようにする場合、宛先が 10.0.0.0/8、ターゲットが local として、静的ルートが自動的にルートテーブルに追加されます。

詳細および例については、『Amazon VPC Peering Guide』の「[ClassicLink を使用した設定](#)」を参照してください。

## ClassicLink の制限事項

ClassicLink 機能を使用するには、次の制限事項に注意する必要があります。

- EC2-Classic インスタンスは、一度に 1 つの VPC にのみリンクすることができます。

- ・リンクされた EC2-Classic インスタンスを停止した場合、インスタンスは VPC から自動的にリンクが解除され、VPC セキュリティグループはインスタンスとの関連付けが失われます。インスタンスを再起動した後、インスタンスを VPC に再びリンクできます。
- ・EC2-Classic インスタンスを、別のリージョンにある VPC や別の AWS アカウントの VPC にリンクすることはできません。
- ・ClassicLink を使用して、VPC インスタンスを別の VPC または EC2-Classic リソースにリンクすることはできません。VPC 間のプライベート接続を確立するには、VPC ピア接続を使用できます。詳細については、「[Amazon VPC Peering Guide](#)」を参照してください。
- ・VPC の Elastic IP アドレスをリンクされた EC2-Classic インスタンスに関連付けることはできません。
- ・IPv6 の通信用に EC2-Classic インスタンスを有効にすることはできません。IPv6 CIDR ブロックを VPC に関連付けて、IPv6 アドレスを VPC 内のリソースに割り当てるとはできますが、ClassicLinked インスタンスと VPC 内のリソースの間で通信できるのは、IPv4 経由に限られます。
- ・EC2-Classic のプライベート IP アドレス範囲の 10/8 と競合するルートを持つ VPC は ClassicLink 用に有効にすることはできません。これには、ルートテーブルに既にローカルルートがあり、IP アドレス範囲が 10.0.0.0/16 および 10.1.0.0/16 である VPC は含まれません。詳細については、「[ClassicLink のルーティング \(p. 815\)](#)」を参照してください。
- ・専用ハードウェアテナント用に設定された VPC は、ClassicLink 用に有効にすることはできません。専用テナント VPC を ClassicLink に対して有効にするようにリクエストするには、AWS サポートまでお問い合わせください。

**Important**

EC2-Classic インスタンスは共有ハードウェアで実行されます。規制またはセキュリティ要件のために VPC のテナントを `dedicated` に設定した場合、EC2-Classic インスタンスを VPC にリンクすると、その要件に準拠しない可能性があります。この設定により、共有テナントのリソースが、プライベート IP アドレスを使用して、隔離されたリソースに直接アクセスできるためです。ClassicLink 用に専用 VPC を有効にする必要がある場合は、詳細な理由を添えて AWS サポートにリクエストしてください。

- ・172.16.0.0/16 の範囲内の VPC に EC2-Classic インスタンスをリンクした場合に、VPC 内に IP アドレス 172.16.0.23/32 で稼働中の DNS サーバーがある場合、リンクした EC2-Classic インスタンスは VPC DNS サーバーにアクセスできません。この問題を回避するためには、DNS サーバーを VPC 内の違う IP アドレスで稼働させてください。
- ・ClassicLink は VPC からの推移関係をサポートしていません。リンクされた EC2-Classic インスタンスは、VPN 接続、VPC ゲートウェイエンドポイント、NAT ゲートウェイ、または VPC に関連付けられたインターネットゲートウェイにアクセスできません。同様に、VPN 接続またはインターネットゲートウェイの他方の側のリソースは、リンクされた EC2-Classic インスタンスにアクセスできません。

## ClassicLink の操作

Amazon EC2 コンソールと Amazon VPC コンソールを使用して、ClassicLink 機能を使用できます。ClassicLink 用の VPC の有効化と無効化、および VPC と EC2-Classic インスタンスのリンクとリンク解除を行うことができます。

**Note**

ClassicLink 機能は、EC2-Classic をサポートするアカウントとリージョンのコンソールにのみ表示されます。

### タスク

- ・[ClassicLink 用の VPC の有効化 \(p. 817\)](#)
- ・[ClassicLink が有効になった VPC の作成 \(p. 817\)](#)
- ・[VPC へのインスタンスのリンク \(p. 817\)](#)
- ・[起動時にインスタンスを VPC にリンクする \(p. 818\)](#)

- ClassicLink が有効な VPC とリンクされたインスタンスを表示する (p. 818)
- ClassicLink DNS サポートの有効化 (p. 819)
- ClassicLinkDNS サポートを無効にする (p. 819)
- VPC からインスタンスのリンクを解除する (p. 819)
- VPC に対する ClassicLink の無効化 (p. 820)

## ClassicLink 用の VPC の有効化

EC2-CClassic インスタンスを VPC にリンクするには、まず ClassicLink 用の VPC を有効にする必要があります。VPC のルーティングが ClassicLink のプライベート IP アドレス範囲と競合する場合、EC2-CClassic 用の VPC を有効にすることはできません。詳細については、「ClassicLink のルーティング (p. 815)」を参照してください。

ClassicLink 用に VPC を有効にするには

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、[Actions]、[Enable ClassicLink] の順に選択します。
4. 確認ダイアログボックスで、[Yes, Enable] を選択します。
5. (オプション) パブリック DNS ホスト名をプライベート IP アドレスに解決するには、インスタンスをリンクする前に、VPC の ClassicLink DNS サポートを有効にします。詳細については、「ClassicLink DNS サポートの有効化 (p. 819)」を参照してください。

## ClassicLink が有効になった VPC の作成

ClassicLink コンソールの VPC ウィザードを使用することによって、新しい VPC を作成し、すぐに Amazon VPC 用に有効にすることができます。

ClassicLink が有効になった VPC を作成するには

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. Amazon VPC ダッシュボードで、[Start VPC Wizard] を選択します。
3. VPC 設定オプションの 1 つを選択し、[Select] を選択します。
4. ウィザードの次のページで、[Enable ClassicLink] の [Yes] を選択します。ウィザードの残りの手順を完了して、VPC を作成します。VPC ウィザードの使用の詳細については、『Amazon VPC ユーザーガイド』の「Amazon VPC のシナリオ」を参照してください。
5. (オプション) パブリック DNS ホスト名をプライベート IP アドレスに解決するには、インスタンスをリンクする前に、VPC の ClassicLink DNS サポートを有効にします。詳細については、「ClassicLink DNS サポートの有効化 (p. 819)」を参照してください。

## VPC へのインスタンスのリンク

ClassicLink 用の VPC を有効にした後、EC2-CClassic インスタンスを VPC にリンクすることができます。

### Note

実行中の EC2-CClassic インスタンスのみを VPC にリンクできます。stopped 状態にあるインスタンスをリンクすることはできません。

パブリック DNS ホスト名をプライベート IP アドレスに解決するには、インスタンスをリンクする前に、VPC の ClassicLink DNS サポートを有効にします。詳細については、「ClassicLink DNS サポートの有効化 (p. 819)」を参照してください。

### インスタンスを VPC にリンクするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. 実行中の EC2-Classical インスタンスを選択し、[Actions]、[ClassicLink]、[Link to VPC] の順に選択します。複数のインスタンスを選択して、同じ VPC にリンクすることができます。
4. 表示されたダイアログボックスで、リストから VPC を選択します。ClassicLink 用に有効になった VPC のみが表示されます。
5. インスタンスに関連付ける VPC セキュリティグループを 1 つ以上選択します。終了したら [Link to VPC] を選択します。

### 起動時にインスタンスを VPC にリンクする

Amazon EC2 コンソールの起動ウィザードを使用して、EC2-Classical インスタンスを起動し、すぐに ClassicLink が有効な VPC にリンクすることができます。

### 起動時にインスタンスを VPC にリンクするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. Amazon EC2 ダッシュボードから、[Launch Instance] を選択します。
3. AMI を選択し、インスタンスタイプを選択します。[Configure Instance Details インスタンス詳細の設定)] ページで、[ネットワーク] リストから [Launch into EC2-Classical (EC2-Classical で起動)] を選択していることを確認します。

#### Note

T2 インスタンスタイプなど、一部のインスタンスタイプは VPC でのみ起動できます。EC2-Classical で起動できるインスタンスタイプを選択していることを確認します。

4. [Link to VPC (ClassicLink)] セクションで、[Link to VPC] から VPC を選択します。ClassicLink が有効な VPC のみが表示されます。インスタンスに関連付ける VPC のセキュリティグループを選択します。ページの他の設定オプションを完了した後、ウィザードの残りの手順を完了して、インスタンスを起動します。起動ウィザードの使用の詳細については、「[AMI からのインスタンスの起動 \(p. 449\)](#)」を参照してください。

### ClassicLink が有効な VPC とリンクされたインスタンスを表示する

ClassicLink コンソールで Amazon VPC が有効な VPC をすべて表示し、EC2-Classical コンソールでリンクされた Amazon EC2 インスタンスをすべて表示できます。

### ClassicLink が有効な VPC を表示するには

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、[Summary] タブで、[ClassicLink] フィールドを探します。値 [Enabled] は、VPC が ClassicLink に対して有効であることを示します。
4. または、[ClassicLink] 列を探し、各 VPC に表示される値 ([Enabled] または [Disabled]) を確認します。この列が表示されない場合は、[Edit Table Columns] (歯車型のアイコン) をクリックし、[ClassicLink] 属性を選択してから、[Close] を選択します。

### リンクされた EC2-Classical インスタンスを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで、[インスタンス] を選択します。
3. EC2-Classical インスタンスを選択し、[Description] タブで、[ClassicLink] フィールドを探します。インスタンスが VPC にリンクされている場合、このフィールドにはインスタンスのリンク先 VPC の ID が表示されます。インスタンスが VPC にリンクされていない場合、フィールドには [Unlinked] と表示されます。
4. または、インスタンスをフィルタリングして、特定の VPC やセキュリティグループのリンクされた EC2-Classical インスタンスのみを表示できます。検索バーで「ClassicLink」と入力し、関連する ClassicLink リソース属性を選択して、セキュリティグループ ID または VPC ID を選択します。

## ClassicLink DNS サポートの有効化

VPC の ClassicLink DNS サポートを有効にして、リンクされた EC2-Classical インスタンスと VPC のインスタンス間で対応された DNS ホスト名がプライベート IP アドレスに解決され、パブリック IP アドレスに解決されないようにします。この機能を有効にするには、DNS ホスト名および DNS の解決について VPC が有効になっている必要があります。

### Note

VPC に対して ClassicLink DNS サポートを有効にした場合、リンクされた EC2-Classical インスタンスは、VPC に関連付けられた任意のプライベートホストゾーンにアクセスできます。詳細については、『Amazon Route 53 開発者ガイド』の「[プライベートホストゾーンの使用](#)」を参照してください。

### ClassicLink DNS サポートを有効にするには

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、[Actions]、[Edit ClassicLink DNS Support] の順に選択します。
4. [Yes] を選択して ClassicLink DNS サポートを有効にし、[Save] を選択します。

## ClassicLinkDNS サポートを無効にする

VPC の ClassicLink DNS サポートを無効にして、リンクされた EC2-Classical インスタンスと VPC のインスタンス間で対応された DNS ホスト名がパブリック IP アドレスに解決され、プライベート IP アドレスに解決されないようにします。

### ClassicLink DNS サポートを無効にするには

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、[Actions]、[Edit ClassicLink DNS Support] の順に選択します。
4. [No] を選択して ClassicLink DNS サポートを無効にし、[Save] を選択します。

## VPC からインスタンスのリンクを解除する

ClassicLink インスタンスと VPC の間の EC2-Classical 接続が不要になった場合、VPC からインスタンスのリンクを解除できます。インスタンスのリンクを解除すると、インスタンスから VPC セキュリティグループの関連付けが解除されます。

### Note

停止したインスタンスは、VPC から自動的にリンクが解除されます。

## VPC からインスタンスのリンクを解除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. [Actions] リストで、[ClassicLink] を選択し、[Unlink Instance] を選択します。複数のインスタンスを選択して、同じ VPC からリンクを解除することができます。
4. 確認ダイアログボックスで [Yes] を選択します。

## VPC に対する ClassicLink の無効化

EC2-Classical インスタンスと VPC の間の接続が不要になった場合は、VPC の ClassicLink を無効にすることができます。最初に、VPC にリンクされたすべての EC2-Classical インスタンスのリンクを解除します。

## VPC の ClassicLink を無効にするには

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、[Actions]、[Disable ClassicLink] の順に選択します。
4. 確認ダイアログボックスで、[Yes, Disable] を選択します。

## ClassicLink の IAM ポリシー例

ClassicLink で VPC を有効にし、EC2-Classical インスタンスと VPC にリンクできます。ClassicLink が有効な VPC と、VPC にリンクされたすべての EC2-Classical インスタンスを表示することもできます。ec2:EnableVpcClassicLink、ec2:DisableVpcClassicLink、ec2:AttachClassicLinkVpc、ec2:DetachClassicLinkVpc の各アクションのリソースレベルのアクセス許可を使用してポリシーを作成し、ユーザーがそれらのアクションを使用できるかどうかを制御できます。リソースレベルのアクセス許可は、ec2:Describe\* アクションではサポートされません。

### 例

- [ClassicLink を使用する完全なアクセス許可 \(p. 820\)](#)
- [ClassicLink で VPC を有効化および無効化する \(p. 821\)](#)
- [インスタンスをリンクする \(p. 821\)](#)
- [インスタンスのリンクの解除 \(p. 822\)](#)

## ClassicLink を使用する完全なアクセス許可

次のポリシーでは、ClassicLink が有効な VPC とリンクされた EC2-Classical インスタンスを表示するアクセス許可、ClassicLink で VPC を有効化および無効化するアクセス許可、ClassicLink が有効な VPC からインスタンスをリンクおよびリンク解除するアクセス許可をユーザーに付与します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeClassicLinkInstances", "ec2:DescribeVpcClassicLink",  
            "ec2:EnableVpcClassicLink", "ec2:DisableVpcClassicLink",  
            "ec2:AttachClassicLinkVpc", "ec2:DetachClassicLinkVpc"  
        ],  
        "Resource": "*"  
    }]  
}
```

## ClassicLink で VPC を有効化および無効化する

次のポリシーでは、特定のタグ「`purpose=classiclink`」を持つ VPC を ClassicLink で有効化および無効化することをユーザーに許可します。ユーザーは、ClassicLink で他の VPC を有効化または無効化することができます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:*VpcClassicLink",  
            "Resource": "arn:aws:ec2:region:account:vpc/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/purpose": "classiclink"  
                }  
            }  
        }  
    ]  
}
```

## インスタンスをリンクする

次のポリシーでは、インスタンスが `m3.large` インスタンスタイプの場合に限り、インスタンスを VPC にリンクするアクセス許可をユーザーに付与します。2 番目のステートメントでは、VPC にインスタンスをリンクするのに必要な、VPC およびセキュリティグループリソースを使用することをユーザーに許可します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:AttachClassicLinkVpc",  
            "Resource": "arn:aws:ec2:region:account:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:InstanceType": "m3.large"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:AttachClassicLinkVpc",  
            "Resource": [  
                "arn:aws:ec2:region:account:vpc/*",  
                "arn:aws:ec2:region:account:security-group/*"  
            ]  
        }  
    ]  
}
```

次のポリシーでは、インスタンスを特定の VPC (`vpc-1a2b3c4d`) にのみリンクするアクセス許可、VPC の特定のセキュリティグループのみインスタンス (`sg-1122aabb` と `sg-aabb2233`) に関連付けるアクセス許可をユーザーに付与します。ユーザーは、インスタンスを他の VPC にリンクすることはできず、他の VPC のセキュリティグループを指定してリクエスト内のインスタンスに関連付けることはできません。

```
{  
    "Version": "2012-10-17",
```

```
"Statement": [
    {
        "Effect": "Allow",
        "Action": "ec2:AttachClassicLinkVpc",
        "Resource": [
            "arn:aws:ec2:region:account:vpc/vpc-1a2b3c4d",
            "arn:aws:ec2:region:account:instance/*",
            "arn:aws:ec2:region:account:security-group/sg-1122aabb",
            "arn:aws:ec2:region:account:security-group/sg-aabb2233"
        ]
    }
]
```

## インスタンスのリンクの解除

次のポリシーでは、インスタンスが「`unlink=true`」タグを持つ場合にのみ、リンクされた EC2-Classic インスタンスを VPC からリンク解除するアクセス許可をユーザーに付与します。2 番目のステートメントでは、VPC からインスタンスをリンク解除するのに必要な、VPC リソースを使用するアクセス許可をユーザーに付与します。

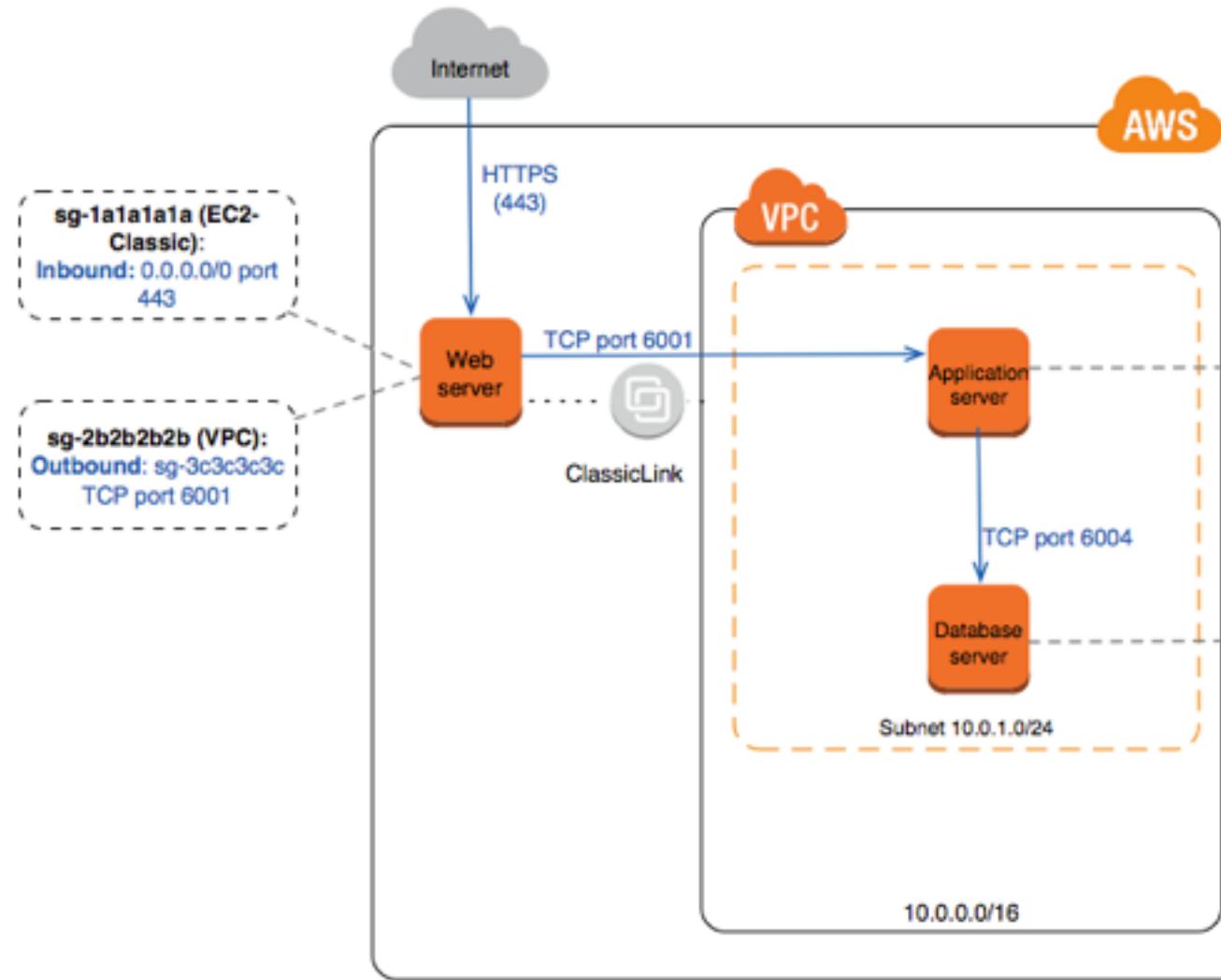
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DetachClassicLinkVpc",
            "Resource": [
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/unlink": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DetachClassicLinkVpc",
            "Resource": [
                "arn:aws:ec2:region:account:vpc/*"
            ]
        }
    ]
}
```

## 例: 3 層ウェブアプリケーションの ClassicLink セキュリティグループ設定

この例では、次の 3 つのインスタンスを持つアプリケーションを使用します。public-facing ウェブサーバー、アプリケーションサーバー、データベースサーバー。ウェブサーバーは、インターネットからの HTTPS トラフィックを受け入れ、TCP ポート 6001 を介してアプリケーションサーバーと通信します。次に、アプリケーションサーバーが TCP ポート 6004 を介してデータベースサーバーと通信します。アプリケーション全体をアカウントの VPC に移行しています。アプリケーションサーバーとデータベースサーバーは、すでに VPC に移行しました。ウェブサーバーはまだ EC2-Classic にあり、ClassicLink を介して VPC にリンクされています。

これらのインスタンス間でのみトラフィックが流れるようにセキュリティグループを設定する必要があります。次の 4 つのセキュリティグループがあります。ウェブサーバー用に 2 つ (`sg-1a1a1a1a` と `sg-2b2b2b2b`)、アプリケーションサーバー用に 1 つ (`sg-3c3c3c3c`)、およびデータベースサーバー用に 1 つ (`sg-4d4d4d4d`)。

次の図は、インスタンスのアーキテクチャーとそれらのセキュリティグループ設定を示しています。



#### ウェブサーバーのセキュリティグループ (sg-1a1a1a1a および sg-2b2b2b2b)

EC2-Classic に 1 つのセキュリティグループ、VPC にその他のセキュリティグループがあります。ClassicLink を介してインスタンスを VPC にリンクしたときに、VPC のセキュリティグループがウェブサーバーインスタンスに関連付けられています。VPC セキュリティグループを使用すると、ウェブサーバーからアプリケーションサーバーへのアウトバウンドトラフィックを制御することができます。

EC2-Classic セキュリティグループ (sg-1a1a1a1a) のセキュリティグループルールを以下に示します。

| Inbound   |       |            |                                    |
|-----------|-------|------------|------------------------------------|
| Source    | Type  | Port Range | Comments                           |
| 0.0.0.0/0 | HTTPS | 443        | インターネットトラフィックがウェブサーバーに到達できるようにします。 |

VPC セキュリティグループ (sg-2b2b2b2b) のセキュリティグループルールを以下に示します。

| Outbound    |      |            |                                                                                           |
|-------------|------|------------|-------------------------------------------------------------------------------------------|
| Destination | Type | Port Range | Comments                                                                                  |
| sg-3c3c3c3c | TCP  | 6001       | ウェブサーバーから VPC のアプリケーションサーバーへの (または sg-3c3c3c3c に関連付けられている他のインスタンスへの) アウトバウンドトラフィックを許可します。 |

#### アプリケーションサーバーのセキュリティグループ (**sg-3c3c3c3c**)

アプリケーションサーバーに関連付けられている VPC セキュリティグループのセキュリティグループルールを以下に示します。

| Inbound     |      |            |                                                                                          |
|-------------|------|------------|------------------------------------------------------------------------------------------|
| Source      | Type | Port Range | Comments                                                                                 |
| sg-2b2b2b2b | TCP  | 6001       | ウェブサーバー (または sg-2b2b2b2b に関連付けられている他のインスタンス) から、指定したタイプのトラフィックがアプリケーションサーバーに到達できるようにします。 |
| Outbound    |      |            |                                                                                          |
| Destination | Type | Port Range | Comments                                                                                 |
| sg-4d4d4d4d | TCP  | 6004       | アプリケーションサーバーからデータベースサーバー (または sg-4d4d4d4d に関連付けられている他のインスタンス) へのアウトバウンドトラフィックを許可します。     |

#### データベースサーバーのセキュリティグループ (**sg-4d4d4d4d**)

データベースサーバーに関連付けられている VPC セキュリティグループのセキュリティグループルールを以下に示します。

| Inbound     |      |            |                                                                                             |
|-------------|------|------------|---------------------------------------------------------------------------------------------|
| Source      | Type | Port Range | Comments                                                                                    |
| sg-3c3c3c3c | TCP  | 6004       | アプリケーションサーバー (または sg-3c3c3c3c に関連付けられている他のインスタンス) から、指定したタイプのトラフィックがデータベースサーバーに到達できるようにします。 |

## EC2-Classic の Linux インスタンスから VPC の Linux インスタンスへの移行

2013 年 12 月 4 日以前に AWS アカウントを作成した場合は、リージョンによっては EC2-Classic のサポートがある場合もあります。ネットワークの強化や新しいインスタンスの種類など、一部の Amazon EC2 リソースと機能には、仮想プライベートクラウド (VPC) が必要です。いくつかのリソースは EC2-Classic と VPC の間で共有できますが、ほかのリソースは共有できません。詳細については、「[EC2-Classic と VPC との間でのリソースの共有とアクセス \(p. 811\)](#)」を参照してください。

アカウントが EC2-Classic をサポートしている場合、EC2-Classic 用にリソースの設定を行った場合があります。EC2-Classic から VPC に移行する場合は、VPC 内でこれらのリソースを再作成する必要があります。

VPC への移行には 2 つの方法があります。完全移行する、または時間をかけて少しづつ移行する方法が利用できます。選択する方法は EC2-Classic アプリケーションのサイズと複雑さによって異なります。たとえば、静的なウェブサイトを 1 つまたは 2 つのインスタンスで実行しているアプリケーションの場合、短期間のダウントIMEを許容することができ、よって完全移行ができます。プロセスを中断できない多層アプリケーションの場合は、ClassicLink を使用して増分移行を実行できます。これによって、アプリケーションが完全に VPC で実行されるようになるまで、機能のコンポーネントを 1 つずつ転送できます。

Windows インスタンスを移行する必要がある場合は、『EC2-Classic』の「[Windows インスタンスの Amazon EC2 ユーザーガイド から VPC への Windows インスタンスの移行](#)」を参照してください。

### コンテンツ

- [VPC への完全移行 \(p. 825\)](#)
- [ClassicLink を使用したVPC への増分移行 \(p. 831\)](#)

## VPC への完全移行

以下のタスクを実行して、EC2-Classic から VPC へアプリケーションを完全移行します。

### タスク

- [ステップ 1: VPC を作成する \(p. 825\)](#)
- [ステップ 2: セキュリティグループを設定する \(p. 826\)](#)
- [ステップ 3: EC2-Classic インスタンスから AMI を作成する \(p. 826\)](#)
- [ステップ 4: VPC でインスタンスを起動する \(p. 827\)](#)
- [例: シンプルなウェブのアプリケーションの移行 \(p. 829\)](#)

### ステップ 1: VPC を作成する

VPC の使用を開始するには、アカウントに VPC があることを確認します。VPC は次のいずれかの方法で作成できます。

- AWS アカウントは、各リージョンで用意されているデフォルトの VPC に搭載されています。起動するインスタンスは、他に指定がない限りデフォルトでこの VPC で起動されます。デフォルト VPC の詳細については、「[デフォルト VPC とデフォルトサブネット](#)」を参照してください。独自の VPC を設定しない場合、または VPC の設定に特定の要件を必要としない場合は、このオプションを使用します。
- 既存の AWS アカウントでは、Amazon VPC コンソールを開き、VPC ウィザードを使用して、新しい VPC を作成します。詳細については、「[Amazon VPC コンソールウィザードの設定](#)」を参照してください。ウィザードで利用可能ないずれかの設定セットを使用して、既存の EC2-Classic アカウントで VPC をすぐに設定する場合は、このオプションを使用します。インスタンスを起動するたびにこの VPC を指定します。

- 既存の AWS アカウントでは、Amazon VPC コンソールを開き、要件に応じて VPC のコンポーネントを設定します。詳細については、「[VPC とサブネット](#)」を参照してください。特定のサブネットの番号など、VPC に特定の要件がある場合は、このオプションを使用します。インスタンスを起動するたびにこの VPC を指定します。

## ステップ 2: セキュリティグループを設定する

EC2-Classic と VPC で同じセキュリティグループを使用することはできません。ただし、VPC のインスタンスで EC2-Classic のインスタンスと同じセキュリティグループのルールを使用したい場合、Amazon EC2 コンソールを使用して既存の EC2-Classic のセキュリティグループのルールを新しい VPC のセキュリティグループにコピーすることができます。

### Important

同じリージョンの同じ AWS アカウントの新しいセキュリティグループにのみセキュリティグループのルールをコピーできます。新しい AWS アカウントを作成した場合、この方法を使用して新しいアカウントに既存のセキュリティグループのルールをコピーすることはできません。新しいセキュリティグループを作成し、独自ルールを追加する必要があります。新しいセキュリティグループの作成方法については、「[Linux インスタンスの Amazon EC2 セキュリティグループ \(p. 911\)](#)」を参照してください。

セキュリティグループのルールを新しいセキュリティグループにコピーするには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Security Groups] を選択します。
- EC2-Classic インスタンスに関連付けられたセキュリティグループを選択し、[Actions] を選択して、[Copy to new] を選択します。
- [Create Security Group] ダイアログボックスで、新しいセキュリティグループの名前と説明を指定します。[VPC] リストから使用している VPC を選択します。
- [Inbound] タブには、EC2-Classic セキュリティグループのルールが自動入力されます。必要に応じてルールを変更できます。[Outbound] タブには、すべてのアウトバウンドトラフィックを許可するルールが自動的に作成されます。セキュリティグループのルールの変更方法については、「[Linux インスタンスの Amazon EC2 セキュリティグループ \(p. 911\)](#)」を参照してください。

### Note

他のセキュリティグループを参照する EC2-Classic セキュリティグループのルールを定義した場合、VPC セキュリティグループでそのルールを使用することはできません。同じ VPC のセキュリティグループを参照するようにルールを変更します。

- [作成] を選択します。

## ステップ 3: EC2-Classic インスタンスから AMI を作成する

AMI はインスタンスを起動するためのテンプレートです。既存の EC2-Classic インスタンスに基づいて独自の AMI を作成し、その AMI を使用して VPC でインスタンスを起動できます。

AMI の作成に使用する方法は、インスタンスのルートデバイスタイプと、インスタンスが実行されるオペレーティングシステムのプラットフォームによって異なります。インスタンスのルートデバイスタイプを確認するには、[Instances] ページに移動し、インスタンスを選択して、[Description] タブの [Root device type] フィールドの情報を確認します。この値が ebs の場合、インスタンスは EBS-Backed です。この値が instance-store の場合、インスタンスは Instance Store-Backed です。また、[describe-instances](#) AWS CLI コマンドを使用してルートデバイスタイプを確認することもできます。

次の表は、インスタンスのルートデバイスタイプ、およびソフトウェアプラットフォームに基づいて AMI を作成するオプションを示します。

Important

インスタンスタイプには、PV と HVM 仮想化の両方をサポートするものもありますが、どちらか一方のみサポートするものもあります。AMI を使用して現在のインスタンスタイプと異なるインスタンスタイプを起動する場合、そのインスタンスタイプで AMI が提供する仮想化のタイプがサポートされていることを確認します。AMI で PV 仮想化がサポートされ、HVM 仮想化をサポートするインスタンスタイプを使用する場合、HVM ベースの AMI にソフトウェアを再インストールする必要がある場合があります。PV および HVM 仮想化に関する詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。

| インスタンスのルートデバイスタイプ | アクション                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EBS               | インスタンスから EBS-backed AMI を作成します。詳細については、「 <a href="#">Amazon EBS-Backed Linux AMI の作成 (p. 116)</a> 」を参照してください。                                                |
| インスタンスストア         | AMI ツールを使用してインスタンスから instance store-backed AMI を作成します。詳細については、「 <a href="#">Instance Store-Backed Linux AMI の作成 (p. 119)</a> 」を参照してください。                     |
| インスタンスストア         | instance store-backed インスタンスを EBS-backed インスタンスに変換します。詳細については、「 <a href="#">Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換する (p. 131)</a> 」を参照してください。 |

(オプション) Amazon EBS ボリュームにデータを保存する

Amazon EBS ボリュームを作成して、物理ハードドライブを使用するように、そのボリュームを使用してインスタンスのデータをバックアップおよび保存できます。Amazon EBS ボリュームは同アベイラビリティーボリュームのすべてのインスタンスにアタッチおよびデタッチできます。EC2-Classic のインスタンスからボリュームをデタッチして、同じアベイラビリティーボリュームの VPC で起動する新しいインスタンスにアタッチできます。

Amazon EBS ボリュームの詳細については、次のトピックを参照してください。

- [Amazon EBS ボリューム \(p. 931\)](#)
- [Amazon EBS ボリュームの作成 \(p. 949\)](#)
- [インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)

Amazon EBS ボリュームのデータをバックアップするには、定期的にボリュームのスナップショットを作成します。必要な場合は、スナップショットから Amazon EBS ボリュームを復元できます。Amazon EBS スナップショットの詳細については、次のトピックを参照してください。

- [Amazon EBS スナップショット \(p. 970\)](#)
- [Amazon EBS スナップショットの作成 \(p. 972\)](#)
- [スナップショットからの Amazon EBS ボリュームの復元 \(p. 950\)](#)

## ステップ 4: VPC でインスタンスを起動する

AMI を作成した後、VPC でインスタンスを起動できます。インスタンスは、既存の EC2-Classic インスタンスと同じデータと構成です。

既存のアカウントで作成した VPC、または新しい VPC のみの AWS アカウントで、インスタンスを起動できます。

## 既存の EC2-Classic アカウントを使用する

Amazon EC2 起動ウィザードを使用して、VPC でインスタンスを起動できます。

VPC でインスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ダッシュボードで、[Launch Instance] を選択します。
3. [Choose an Amazon Machine Image] ページで、[My AMIs] カテゴリを選択し、作成した AMI を選択します。
4. [Choose an Instance Type] ページで、インスタンスのタイプを選択し、[Next: Configure Instance Details] を選択します。
5. [Configure Instance Details] ページで、[Network] リストから VPC を選択します。[Subnet] リストから必要なサブネットを選択します。必要な他の詳細を設定し、[Configure Security Group] ページが表示されるまでウィザードの次のページに進みます。
6. [Select an existing group] を選択し、前に作成したセキュリティグループを選択します。[Review and Launch] を選択します。
7. インスタンスの詳細を確認し、[Launch] を選択して、キーペアを指定し、インスタンスを起動します。

ウィザードの各ステップで設定できるパラメータの詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。

## 新しい、VPC のみのアカウントを使用する

新しい AWS アカウントでインスタンスを起動するには、最初に新しいアカウントで作成した AMI を共有する必要があります。その後、Amazon EC2 起動ウィザードを使用して、デフォルトの VPC でインスタンスを起動できます。

新しい AWS アカウントと AMI を共有するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. AMI で作成したアカウントに切り替えます。
3. ナビゲーションペインで [AMIs] を選択します。
4. [Filter] リストで、[Owned by me] が選択されていることを確認し、AMI を選択します。
5. [Permissions] タブで、[Edit] を選択します。新しい AWS アカウントのアカウント番号を入力し、[Add Permission] を選択して、[Save] を選択します。

インスタンスをデフォルトの VPC 内に起動するには

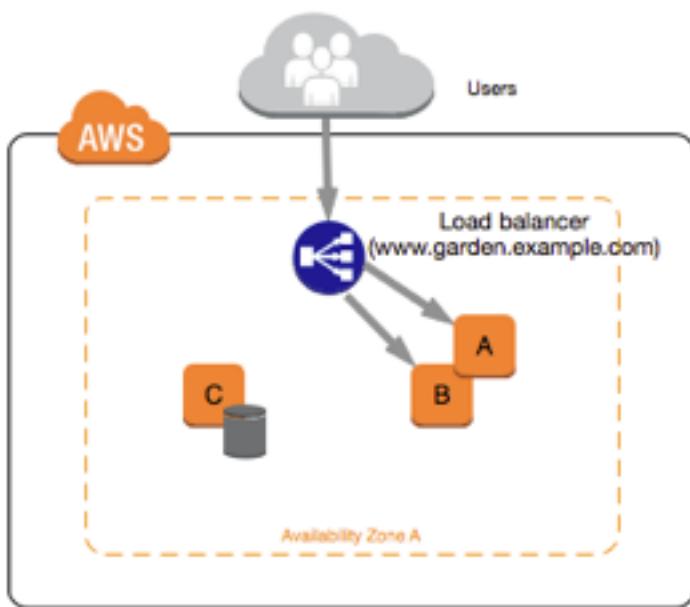
1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 新しい AWS アカウントに切り替えます。
3. ナビゲーションペインで [AMIs] を選択します。
4. [Filter] リストで [Private images] を選択します。EC2-Classic アカウントから共有した AMI を選択し、[Launch] を選択します。
5. [Choose an Instance Type] ページで、インスタンスのタイプを選択し、[Next: Configure Instance Details] を選択します。
6. [Configure Instance Details] ページの [Network] リストでデフォルトの VPC が選択されている必要があります。必要な他の詳細を設定し、[Configure Security Group] ページが表示されるまでウィザードの次のページに進みます。
7. [Select an existing group] を選択し、前に作成したセキュリティグループを選択します。[Review and Launch] を選択します。

8. インスタンスの詳細を確認し、[Launch] を選択して、キーペアを指定し、インスタンスを起動します。

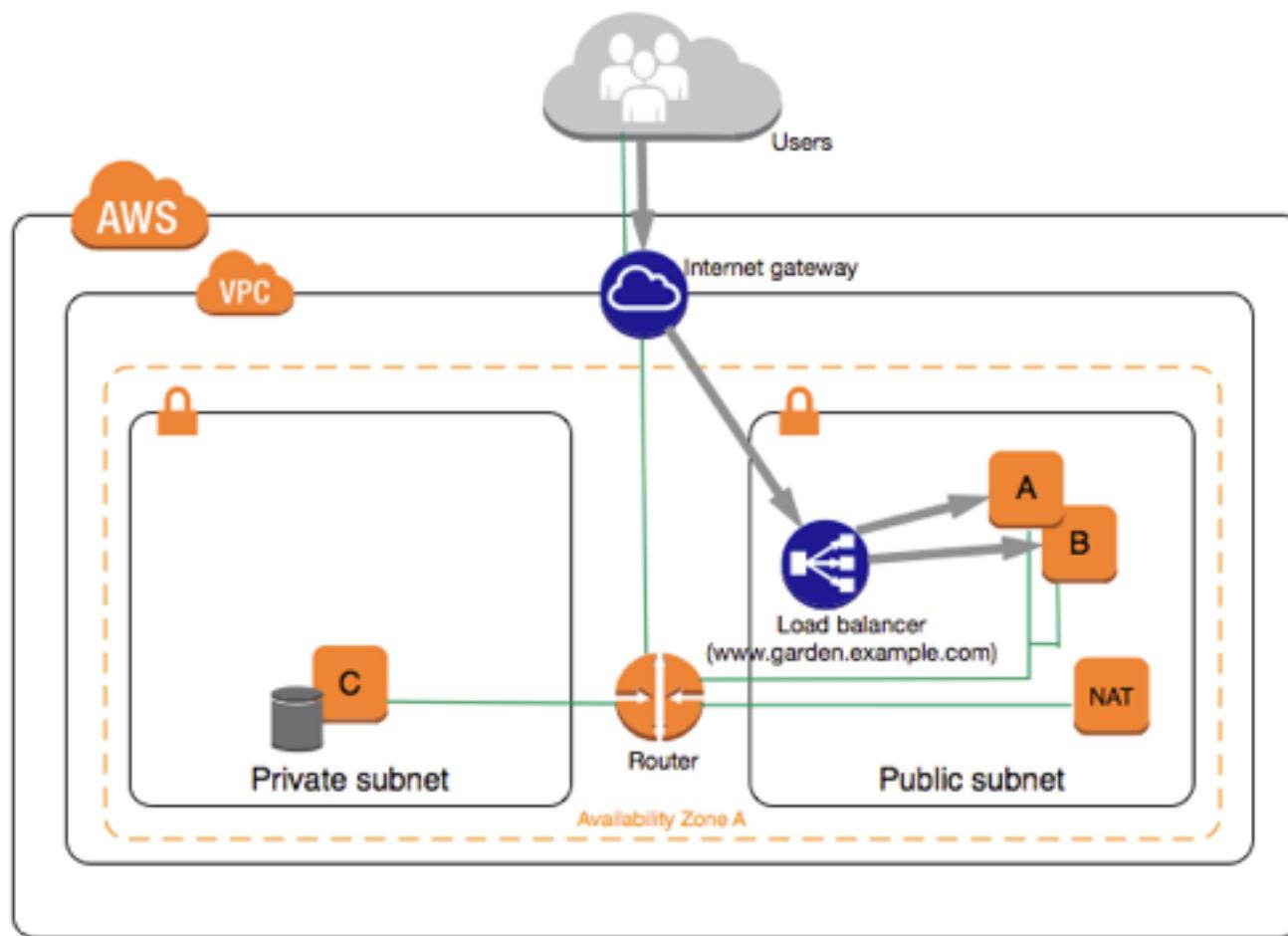
ウィザードの各ステップで設定できるパラメータの詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。

### 例: シンプルなウェブのアプリケーションの移行

この例では、AWS を使用して、ガーデニングウェブサイトをホストします。ウェブサイトを管理するには、EC2-Classic で 3 個のインスタンスを実行します。インスタンス A とインスタンス B は公開ウェブアプリケーションをホストします。これらのインスタンス間でトラフィックの負荷を分散するには Elastic Load Balancing を使用します。インスタンス A と B に Elastic IP アドレスを割り当てると、これらのインスタンスの設定と管理に使用する静的 IP アドレスが割り当てられます。インスタンス C は、ウェブサイトの MySQL データベースを保持します。ドメイン名 `www.garden.example.com` を登録し、Route 53 を使用すると、ロードバランサーの DNS 名と関連付けられたエイリアスレコードが設定されたホストゾーンが作成されます。



VPC への移行の最初の部分では、ニーズに適合する VPC アーキテクチャの種類を決定します。この場合、次のように決定しました: ウェブサーバーに 1 つのパブリックサブネット、データベースサーバーに 1 つのプライベートサブネット。ウェブサイトが成長したら、サブネットにウェブサーバーとデータベースサーバーを追加できます。デフォルトでは、プライベートサブネットのインスタンスはインターネットにアクセスできません。ただし、パブリックサブネットのネットワークアドレス変換 (NAT) デバイスを介してインターネットアクセスを有効にすることができます。インターネットから提供されるデータベースサーバーの定期的な更新やパッチをサポートするように、NAT デバイスを設定できます。Elastic IP アドレスを VPC に移行し、パブリックサブネットでロードバランサーを作成してウェブサーバー間のトラフィックの負荷を分散します。



VPC にウェブアプリケーションを移行するには、次の手順に従います。

- VPC を作成する: この場合、Amazon VPC コンソールで VPC ウィザードを使用して、VPC と サブネットを作成できます。2 番目のウィザード設定では 1 つのプライベートサブネットと 1 つのパブリックサブネットを持つ VPC を作成し、自分のパブリックサブネットの NAT デバイスを起動して、設定します。詳細については、Amazon VPC ユーザーガイドの「[パブリックサブネットとプライベートサブネットを持つ VPC \(NAT\)](#)」を参照してください。
- インスタンスから AMI を作成する: いずれかのウェブサーバーから AMI を作成し、データベースサーバーから 2 番目の AMI を作成します。詳細については、「[ステップ 3: EC2-Classic インスタンスから AMI を作成する \(p. 826\)](#)」を参照してください。
- セキュリティグループを設定する: EC2-Classic 環境では、ウェブサーバー用に 1 つのセキュリティグループ、データベースサーバー用にもう 1 つのセキュリティグループを設定します。Amazon EC2 コンソールを使用して、VPC の新しいセキュリティグループに、各セキュリティグループのルールをコピーします。詳細については、「[ステップ 2: セキュリティグループを設定する \(p. 826\)](#)」を参照してください。

Tip

最初に他のセキュリティグループから参照されるセキュリティグループを作成します。

- 新しい VPC でインスタンスを起動する: パブリックサブネットで代替ウェブサーバーを起動し、プライベートサブネットで代替データベースサーバーを起動します。詳細については、「[ステップ 4: VPC でインスタンスを起動する \(p. 827\)](#)」を参照してください。

- NAT デバイスを設定する: NAT インスタンスを使用している場合、プライベートサブネットからの HTTP および HTTPS トラフィックを許可するためのセキュリティグループを作成する必要があります。詳細については、「[NAT インスタンス](#)」を参照してください。NAT ゲートウェイを使用している場合は、プライベートサブネットからのトラフィックは自動的に許可されます。
- データベースを設定する: EC2-Classic でデータベースサーバーから AMI を作成したとき、そのインスタンスに格納されているすべての設定情報は AMI にコピーされています。新しいデータベースサーバーに接続し、設定の詳細を更新する必要がある可能性があります。たとえば、EC2-Classic のウェブサーバーに完全な読み取り、書き込み、変更のアクセス許可を付与するようにデータベースを設定した場合、代わりに新しいウェブサーバーに同じアクセス許可を付与するように設定ファイルを更新する必要があります。
- ウェブサーバーを設定する: ウェブサーバーは EC2-Classic のインスタンスと同じ設定にします。たとえば、EC2-Classic でデータベースを使用するようにウェブサーバーを構成した場合、新しいデータベースインスタンスをポイントするようにウェブサーバーの設定を更新します。

Note

起動時に別の方で指定しない限り、デフォルト以外のサブネットで起動されたインスタンスにデフォルトでパブリック IP アドレスは割り当てられません。新しいデータベースサーバーにパブリック IP アドレスが割り当てられていない可能性があります。この場合、新しいデータベースサーバーのプライベート DNS 名を使用するようにウェブサーバーの設定ファイルを更新できます。同じ VPC のインスタンスはプライベート IP アドレスを使用して互いに通信できます。

- Elastic IP アドレスを移行する: Elastic IP アドレスと EC2-Classic のウェブサーバーの関連付けを解除し、VPC に移行します。移行した後、VPC 内の新しいウェブサーバーに関連付けることができます。詳細については、「[EC2-Classic からの Elastic IP アドレスの移行 \(p. 809\)](#)」を参照してください。
- 新しいロードバランサーを作成する: インスタンスへのトラフィックの負荷を分散するために引き続き Elastic Load Balancing を使用するには、VPC のロードバランサーのさまざまな設定方法を知っている必要があります。詳細については、「[Amazon VPC の Elastic Load Balancing](#)」を参照してください。
- DNS レコードを更新する: パブリックサブネットのロードバランサーを設定した後、www.garden.example.com ドメインが新しいロードバランサーをポイントしていることを確認します。これを行うには、DNS レコードを更新して、Route 53 のエイリアスレコードを更新する必要があります。Route 53 の使用の詳細については、「[Route 53 の使用を開始する](#)」を参照してください。
- EC2-Classic のリソースをシャットダウンする: ウェブアプリケーションが VPC アーキテクチャ内で動作していることを確認した後、EC2-Classic のリソースをシャットダウンして、これらに対する課金を停止することができます。EC2-Classic インスタンスを終了し、EC2-Classic Elastic IP アドレスを解放します。

## ClassicLink を使用したVPC への増分移行

ClassicLink 機能によって、VPC への増分移行の管理が容易になります。ClassicLink を使用すると、EC2-Classic インスタンスを同じリージョンのアカウント内の VPC にリンクできます。これにより、新しい VPC リソースは、プライベート IPv4 アドレスを使用して、EC2-Classic インスタンスと通信できます。次に、機能を 1 つずつ VPC に移行できます。このトピックでは、EC2-Classic から VPC への増分移行を管理するための基本的な手順を説明し、。

ClassicLink の詳細については、「[ClassicLink \(p. 812\)](#)」を参照してください。

### トピック

- [ステップ 1: 移行シーケンスを準備する \(p. 832\)](#)
- [ステップ 2: VPC を作成する \(p. 832\)](#)
- [ステップ 3: ClassicLink 用に VPC を有効にする \(p. 832\)](#)
- [ステップ 4: EC2-Classic インスタンスから AMI を作成する \(p. 832\)](#)
- [ステップ 5: VPC でインスタンスを起動する \(p. 833\)](#)

- ステップ 6: EC2-Classic インスタンスを VPC にリンクする (p. 834)
- ステップ 7: VPC への移行を完了する (p. 834)

## ステップ 1: 移行シーケンスを準備する

ClassicLink を効果的に使用するには、最初に、VPC に移行する必要があるアプリケーションのコンポーネントを把握し、その機能を移行する順序を確認する必要があります。

たとえば、プレゼンテーションのウェブサーバー、バックエンドのデータベースサーバー、トランザクションの認証ロジックを利用するアプリケーションがあるとします。この場合、認証ロジックから移行プロセスを開始し、次にデータベースサーバー、最後にウェブサーバーの順に移行することを決定できます。

## ステップ 2: VPC を作成する

VPC の使用を開始するには、アカウントに VPC があることを確認します。VPC は次のいずれかの方法で作成できます。

- 既存の AWS アカウントでは、Amazon VPC コンソールを開き、VPC ウィザードを使用して、新しい VPC を作成します。詳細については、「[Amazon VPC コンソール ウィザードの設定](#)」を参照してください。ウィザードで利用可能ないずれかの設定セットを使用して、既存の EC2-Classic アカウントで VPC をすぐに設定する場合は、このオプションを使用します。インスタンスを起動するたびにこの VPC を指定します。
- 既存の AWS アカウントでは、Amazon VPC コンソールを開き、要件に応じて VPC のコンポーネントを設定します。詳細については、「[VPC とサブネット](#)」を参照してください。特定のサブネットの番号など、VPC に特定の要件がある場合は、このオプションを使用します。インスタンスを起動するたびにこの VPC を指定します。

## ステップ 3: ClassicLink 用に VPC を有効にする

VPC を作成した後、ClassicLink 用に VPC を有効にすることができます。ClassicLink の詳細については、「[ClassicLink \(p. 812\)](#)」を参照してください。

ClassicLink 用に VPC を有効にするには

1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、[Actions] リストから [Enable ClassicLink] を選択します。
4. 確認ダイアログボックスで、[Yes, Enable] を選択します。

## ステップ 4: EC2-Classic インスタンスから AMI を作成する

AMI はインスタンスを起動するためのテンプレートです。既存の EC2-Classic インスタンスに基づいて独自の AMI を作成し、その AMI を使用して VPC でインスタンスを起動できます。

AMI の作成に使用する方法は、インスタンスのルートデバイスタイプと、インスタンスが実行されるオペレーティングシステムのプラットフォームによって異なります。インスタンスのルートデバイスタイプを確認するには、[Instances] ページに移動し、インスタンスを選択して、[Description] タブの [Root device type] フィールドの情報を確認します。この値が ebs の場合、インスタンスは EBS-Backed です。この値が instance-store の場合、インスタンスは Instance Store-Backed です。また、[describe-instances](#) AWS CLI コマンドを使用してルートデバイスタイプを確認することもできます。

次の表は、インスタンスのルートデバイスタイプ、およびソフトウェアプラットフォームに基づいて AMI を作成するオプションを示します。

### Important

インスタンスタイプには、PV と HVM 仮想化の両方をサポートするものもありますが、どちらか一方のみサポートするものもあります。AMI を使用して現在のインスタンスタイプと異なるインスタンスタイプを起動する場合、そのインスタンスタイプで AMI が提供する仮想化のタイプがサポートされていることを確認します。AMI で PV 仮想化がサポートされ、HVM 仮想化をサポートするインスタンスタイプを使用する場合、HVM ベースの AMI にソフトウェアを再インストールする必要がある場合があります。PV および HVM 仮想化に関する詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。

| インスタンスのルートデバイスタイプ | アクション                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EBS               | インスタンスから EBS-backed AMI を作成します。詳細については、「 <a href="#">Amazon EBS-Backed Linux AMI の作成 (p. 116)</a> 」を参照してください。                                                |
| インスタンスストア         | AMI ツールを使用してインスタンスから instance store-backed AMI を作成します。詳細については、「 <a href="#">Instance Store-Backed Linux AMI の作成 (p. 119)</a> 」を参照してください。                     |
| インスタンスストア         | instance store-backed インスタンスを EBS-backed インスタンスに変換します。詳細については、「 <a href="#">Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換する (p. 131)</a> 」を参照してください。 |

### (オプション) Amazon EBS ボリュームにデータを保存する

Amazon EBS ボリュームを作成して、物理ハードドライブを使用するように、そのボリュームを使用してインスタンスのデータをバックアップおよび保存できます。Amazon EBS ボリュームは同アベイラビリティゾーンのすべてのインスタンスにアタッチおよびデタッチできます。EC2-Classic のインスタンスからボリュームをデタッチして、同じアベイラビリティゾーンの VPC で起動する新しいインスタンスにアタッチできます。

Amazon EBS ボリュームの詳細については、次のトピックを参照してください。

- [Amazon EBS ボリューム \(p. 931\)](#)
- [Amazon EBS ボリュームの作成 \(p. 949\)](#)
- [インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)

Amazon EBS ボリュームのデータをバックアップするには、定期的にボリュームのスナップショットを作成します。必要な場合は、スナップショットから Amazon EBS ボリュームを復元できます。Amazon EBS スナップショットの詳細については、次のトピックを参照してください。

- [Amazon EBS スナップショット \(p. 970\)](#)
- [Amazon EBS スナップショットの作成 \(p. 972\)](#)
- [スナップショットからの Amazon EBS ボリュームの復元 \(p. 950\)](#)

## ステップ 5: VPC でインスタンスを起動する

移行プロセスの次のステップでは、機能の転送を開始できるように VPC でインスタンスを起動します。前のステップで作成した AMI を使用して、VPC 内でインスタンスを起動できます。このインスタンスでは、既存の EC2-Classic インスタンスと同じデータおよび設定が使用されます。

カスタム AMI を使用して VPC でインスタンスを起動するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ダッシュボードで、[Launch Instance] を選択します。
3. [Choose an Amazon Machine Image] ページで、[My AMIs] カテゴリを選択し、作成した AMI を選択します。
4. [Choose an Instance Type] ページで、インスタンスのタイプを選択し、[Next: Configure Instance Details] を選択します。
5. [Configure Instance Details] ページで、[Network] リストから VPC を選択します。[Subnet] リストから必要なサブネットを選択します。必要な他の詳細を設定し、[Configure Security Group] ページが表示されるまでウィザードの次のページに進みます。
6. [Select an existing group] を選択し、前に作成したセキュリティグループを選択します。[Review and Launch] を選択します。
7. インスタンスの詳細を確認し、[Launch] を選択して、キーペアを指定し、インスタンスを起動します。

ウィザードの各ステップで設定できるパラメータの詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」を参照してください。

インスタンスを起動し、running 状態になったら、必要に応じて、インスタンスに接続してインスタンスを設定できます。

## ステップ 6: EC2-Classic インスタンスを VPC にリンクする

インスタンスを設定し、アプリケーションの機能を VPC で利用できるようにした後、ClassicLink を使用して、新しい VPC のインスタンスと EC2-Classic インスタンスの間のプライベート IP 通信を有効にすることができます。

インスタンスを VPC にリンクするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. EC2-Classic インスタンスを選択し、[Actions]、[ClassicLink]、[Link to VPC] の順に選択します。

### Note

インスタンスが running 状態であることを確認します。

4. ダイアログボックスで、ClassicLink が有効な VPC を選択します (ClassicLink が有効になっている VPC のみが表示されます)。
5. インスタンスに関連付ける VPC セキュリティグループを 1 つ以上選択します。終了したら [Link to VPC] を選択します。

## ステップ 7: VPC への移行を完了する

アプリケーションのサイズや移行する必要がある機能に応じて、アプリケーションのすべてのコンポーネントを EC2-Classic から VPC に移動するまで、ステップ 4 ~ 6 を繰り返します。

EC2-Classic インスタンスと VPC インスタンスの間で内部コミュニケーションを有効にした後、アプリケーションが EC2-Classic プラットフォーム内のサービスではなく、VPC 内の移行されたサービスを使用するように、アプリケーションを更新する必要があります。そのための詳細な手順は、アプリケーションの設計によって異なります。通常、この作業では、EC2-Classic インスタンスではなく VPC インスタンスの IP アドレスを指すように送信先 IP アドレスを更新します。EC2-Classic プラットフォームで現在使用している Elastic IP アドレスを VPC に移行できます。詳細については、「[EC2-Classic からの Elastic IP アドレスの移行 \(p. 809\)](#)」を参照してください。

このステップを完了し、アプリケーションが VPC から機能していることをテストしたら、EC2-Classic インスタンスを終了し、VPC の ClassicLink を無効にすることができます。また、不要になった EC2-Classic リソースをクリーンアップして、リソースの料金が発生することを回避できます。たとえ

ば、Elastic IP アドレスを解放し、EC2-Classic インスタンスに関連付けられていたボリュームを削除できます。

# Amazon EC2におけるセキュリティ

AWS では、クラウドのセキュリティが最優先事項です。AWS のお客様は、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャから利点を得られます。

セキュリティは、AWS とお客様の間の共有責任です。[責任共有モデル](#)では、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ – AWS は、AWS クラウド内で AWS サービスを実行するインフラストラクチャを保護する責任を担います。また、AWS は、使用するサービスを安全に提供します。[AWSコンプライアンスプログラム](#)の一環として、サードパーティの監査が定期的にセキュリティの有効性をテストおよび検証しています。Amazon EC2に適用されるコンプライアンスプログラムの詳細については、[コンプライアンスプログラムの対象範囲に含まれる AWS のサービス](#)を参照してください。
- クラウド内のセキュリティ – お客様の責任はお客様が使用する AWS のサービスによって決まります。また、お客様は、お客様のデータの機密性、企業の要件、および適用可能な法律および規制などの他の要因についても責任を担います。

このドキュメントは、Amazon EC2使用時における責任共有モデルの適用法を理解するのに役立ちます。ここでは、セキュリティやコンプライアンスに関する目標を達成できるようにAmazon EC2を設定する方法について説明します。また、Amazon EC2リソースの監視やセキュリティ確保に役立つ他のAWSサービスの用法についても学習します。

## 目次

- [Amazon EC2におけるインフラストラクチャセキュリティ \(p. 836\)](#)
- [Amazon EC2の耐障害性 \(p. 838\)](#)
- [Amazon EC2におけるデータ保護 \(p. 838\)](#)
- [Amazon EC2 の Identity and Access Management \(p. 839\)](#)
- [Amazon EC2 のキーペア \(p. 899\)](#)
- [Linux インスタンスの Amazon EC2 セキュリティグループ \(p. 911\)](#)
- [Amazon EC2での更新管理 \(p. 926\)](#)
- [Amazon EC2のコンプライアンス検証 \(p. 926\)](#)

# Amazon EC2におけるインフラストラクチャセキュリティ

マネージドサービスであるAmazon EC2は、ホワイトペーパーである[Amazon Web Services: AWS セキュリティプロセスの概要](#)に記載されているAWSグローバルネットワークセキュリティ手順で保護されています。

お客様は、AWSが公開している API コールを使用し、ネットワーク経由でAmazon EC2にアクセスすることになります。クライアントで Transport Layer Security (TLS) 1.0 以降がサポートされている必要があります。TLS 1.2 以降が推奨されています。また、Ephemeral Diffie-Hellman (DHE) や Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートもクライアントでサポートされている必要があります。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストは、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、[AWS Security Token Service](#) (AWS STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

## ネットワークの隔離

virtual private cloud (VPC) は、AWS クラウド内の論理的に隔離された領域にある仮想ネットワークです。ワークロードまたは組織エンティティ単位でインフラストラクチャを隔離するには、個別の VPC を使用します。

サブネットは、ある範囲の IP アドレスが示す VPC 内の領域です。インスタンスを起動する場合には、VPC 内のあるサブネットにおいて起動することになります。サブネットを使用すると、単一の VPC 内で多階層ウェブアプリケーションの各階層（ウェブサーバー、アプリケーションサーバーおよびデータベースサーバーなど）を隔離できます。インターネットからの直接アクセスを認めるべきでないインスタンスには、プライベートサブネットを使用します。

パブリックインターネットを介してトラフィックを送信することなく VPC から Amazon EC2 API を呼び出すには、AWS PrivateLink を使用します。

## 物理ホストでの隔離

同じ物理ホストで実行される異なる EC2 インスタンスは、個別の物理ホストで実行されるかのように隔離されます。ハイパーバイザーが CPU およびメモリを隔離し、各インスタンスには、生ディスクデバイスへのアクセスに代わる仮想ディスクへのアクセスが提供されます。

インスタンスを停止または終了すると、そのインスタンスに割り当てられていたメモリをハイパーバイザーがスクラップ（ゼロに設定）し、そのメモリが新たなインスタンスに割り当てられ、すべてのストレージブロックがリセットされます。これは、お客様のデータが誤って他のインスタンスに引き渡されないようにするための処理です。

ネットワーク MAC アドレスは、AWS ネットワークインフラストラクチャが各インスタンスに対し動的に割り当てます。IP アドレスは、AWS ネットワークインフラストラクチャが各インスタンスに対し動的に割り当てるか、要認証 API リクエストを介して EC2 管理者が割り当てます。AWS ネットワークは、インスタンスは割り当てられた MAC および IP アドレスからのみトラフィックを送信できます。それ以外のトラフィックは除外されます。

デフォルトでは、インスタンスは、そのインスタンス宛ではないトラフィックを受信することはできません。インスタンスにおいて、ネットワークアドレス変換 (NAT、network address translation)、ルーティングまたはファイアウォールといったサービスの実行が必要な場合には、ネットワークインターフェースの送信元/送信先チェックを無効化できます。

## ネットワークトラフィックの制御

EC2 インスタンスへのネットワークトラフィックを制御するには、以下のオプションを検討します。

- セキュリティグループ (p. 911) を使用してインスタンスへのアクセスを制限する。この方法を使うと、たとえば、社内ネットワークのアドレス範囲に属するアドレスからのトラフィックのみ認めるといったことができます。
- インターネットからの直接アクセスを認めるべきでないインスタンスには、プライベートサブネットを使用します。プライベートサブネット内にあるインスタンスからのインターネットアクセスに、要塞ホストまたは NAT ゲートウェイを使用する。
- AWS Virtual Private Network または AWS Direct Connectを使用して、リモートネットワークから VPC へのプライベート接続を確立する。詳細については、[ネットワークから Amazon VPC への接続オプション](#) を参照してください。
- VPC フローログを使用して、インスタンスに到達するトラフィックを監視する。
- AWS Security Hubを使用して、インスタンスからの意図しないネットワークアクセスを確認する。

- EC2 インスタンス コネクト (p. 511) を使用して、SSH キーの共有および管理が不要なセキュアシェル (SSH, Secure Shell) を使いインスタンスに接続する。
- インバウンド SSH ポートを開き、SSH キーを管理する代わりに、AWS Systems Manager セッションマネージャーを使用してインスタンスにリモートアクセスする。
- インバウンド SSH ポートを開き、SSH キーを管理する代わりに、AWS Systems Manager 実行コマンドを使用して、共通の管理タスクを自動化する。

各 Amazon EC2 インスタンスへのネットワークアクセスの制限に加えて、Amazon VPC は、オンラインゲートウェイ、プロキシサーバー、さまざまなネットワークモニタリングオプションなど、追加のネットワークセキュリティ管理の実装をサポートしています。

詳細については、ホワイトペーパー「[AWS セキュリティのベストプラクティス](#)」を参照してください。

## Amazon EC2の耐障害性

AWS のグローバルインフラストラクチャは AWS リージョンとアベイラビリティーゾーンを中心として構築されます。リージョンには、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている複数の物理的に独立および隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS のリージョンやアベイラビリティーゾーンの詳細については、[AWSグローバルインフラストラクチャ](#) を参照してください。

AWS グローバルインフラストラクチャに加え、Amazon EC2 が、データ耐障害性をサポートする以下の機能を提供します。

- リージョンで AMI をコピーする機能
- リージョン間の EBS スナップショットをコピーする機能
- Amazon Data Lifecycle Manager を使用して EBS スナップショットを自動化する機能
- Amazon EC2 Auto Scaling を使用してフリートの健全性や可用性を維持する機能
- Elastic Load Balancing を使用して、単一のまたは複数のアベイラビリティーゾーンにある複数のインスタンスの間で受信トラフィックを分散する機能

## Amazon EC2におけるデータ保護

Amazon Elastic Compute Cloud (Amazon EC2) は、データ保護に関する規制やガイドラインを含む AWS 責任共有モデル に準拠しています。AWS は、すべての AWS サービスの実行基盤となるグローバルインフラストラクチャを保護する責任を負います。AWS は、カスタマーコンテンツや個人データの取扱いに必要となるセキュリティ構成の管理を含む、このインフラストラクチャにてホストされるデータの管理を保持します。データ管理者またはデータ処理者となる AWS のお客様および APN パートナーは、AWS クラウドに保存される個人データについて責任を負います。

データ保護目的の場合、AWS アカウント認証情報を保護して IAM (AWS Identity and Access Management) で個々のユーザー アカウントをセットアップし、そのユーザーに各自の職務を果たすために必要なアクセス許可のみが付与されるようにすることをお勧めします。また、以下の方法でデータを保護することをお勧めします。

- 各アカウントで多要素認証 (MFA) を使用します。
- TLS を使用して AWS リソースと通信します。
- AWS CloudTrail で API とユーザー アクティビティ ログをセットアップします。

- AWS 暗号化ソリューションを、AWS サービス内のすべてのデフォルトのセキュリティ管理と一緒に使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これにより、Amazon S3 に保存される個人データの検出と保護が支援されます。

顧客のアカウント番号などの機密の識別情報は、関数名やタグなどのメタデータの自由形式フィールドに配置しないことを強くお勧めします。メタデータに入力したデータは、いずれも診断ログに含まれるデータとして取得される可能性があります。外部サーバーへの URL を指定するときは、そのサーバーへのリクエストを検証するための認証情報を URL に含めないでください。

データ保護の詳細については、AWS セキュリティブログのブログ投稿「[AWS の責任共有モデルと GDPR](#)」を参照してください。

## 保管時の暗号化

Amazon EBS暗号化は、EBS ボリュームおよびスナップショット向けの暗号化ソリューションです。このソリューションは、AWS Key Management Service (AWS KMS) カスタマーマスターkey (CMK, customer master key) を使用します。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

NVMe インスタンスストアボリューム内のデータは、インスタンスのハードウェアモジュールに実装されている XTS-AES-256 暗号を使用して暗号化されます。暗号化キーは、ハードウェアモジュールで作成され、NVMe インスタンスストレージデバイスごとに固有です。すべての暗号化キーは、インスタンスが停止または終了して復元できないときに破棄されます。この暗号化を無効にしたり、独自の暗号キーを指定したりすることはできません。

## 転送中の暗号化

AWS は、すべてのタイプの EC2 インスタンス間において安全でプライベートな接続を提供します。また、同じ VPC 内やピア接続された VPC 内のサポートされているインスタンス間で転送中のトラフィックを自動的に暗号化します。これには、256 ビット暗号化の AEAD アルゴリズムを使用します。この暗号化機能は、基盤となるハードウェアのオフロード機能を使用し、ネットワークパフォーマンスには影響を及ぼしません。転送中のトラフィックの暗号化をサポートしているインスタンスは、C5n、G4、I3en、M5dn、M5n、P3dn、R5dn、および R5n です。

SSH は、Linuxインスタンスへのリモートアクセスに必要なセキュア通信チャネルを提供します。AWS Systems Manager Session Manager および Run Command を使用したインスタンスへのリモートアクセスは、TLS 1.2 を使用して暗号化され、SigV4 を使用して署名される接続の確立を要求します。

クライアントとインスタンスの間で送受信される機微データは、トранスポートレイヤーセキュリティ (TLS、Transport Layer Security) といった暗号化プロトコルを使用して暗号化されます。

# Amazon EC2 の Identity and Access Management

セキュリティ認証情報により、AWS のサービスに対してお客様の身分が証明され、Amazon EC2 リソースなどの AWS リソースを無制限に使用できる許可が与えられます。Amazon EC2 および AWS Identity and Access Management (IAM) の機能を使用して、その他のユーザー、サービス、およびアプリケーションがお客様の Amazon EC2 のリソースを使用できるようにします。その際、お客様のセキュリティ認証情報は共有されません。他のユーザーが AWS アカウント内のリソースをどのように使用するかを制御するには IAM を、Amazon EC2 インスタンスへのアクセスを制御するにはセキュリティグループを使用できます。Amazon EC2 のリソースの完全使用または制限付き使用のどちらを許可するか選択できます。

### コンテンツ

- [インスタンスへのネットワークアクセス \(p. 840\)](#)

- [Amazon EC2 のアクセス許可属性 \(p. 840\)](#)
- [IAM および Amazon EC2 \(p. 840\)](#)
- [Amazon EC2 の IAM ポリシー \(p. 842\)](#)
- [Amazon EC2 の IAM ロール \(p. 888\)](#)
- [Linux インスタンス用の受信トラフィックの認可 \(p. 897\)](#)

## インスタンスへのネットワークアクセス

セキュリティグループは、1つ以上のインスタンスに到達できるトラフィックを制御するファイアウォールとして機能します。インスタンスを起動するときに、そのインスタンスに1つまたは複数のセキュリティグループを割り当てることができます。セキュリティグループのそれぞれに、そのインスタンスへのトラフィックを制御するルールを追加できます。セキュリティグループルールはいつでも変更できます。新しいルールは、そのセキュリティグループが割り当てられているインスタンスすべてに自動的に適用されます。

詳細については、「[Linux インスタンス用の受信トラフィックの認可 \(p. 897\)](#)」を参照してください。

## Amazon EC2 のアクセス許可属性

お客様の組織には複数の AWS アカウントがある場合があります。Amazon EC2 では、Amazon Machine Image (AMI) および Amazon EBS スナップショットを使用できる追加の AWS アカウントを指定できます。このアクセス許可は AWS アカウントレベルでのみ有効です。特定の AWS アカウント内の特定ユーザーのアクセス許可を制限することはできません。指定した AWS アカウントのすべてのユーザーが、AMI またはスナップショットを使用できます。

AMI ごとに `LaunchPermission` 属性があり、AMI にアクセスできる AWS アカウントを制御します。詳細については、「[AMI を一般公開する \(p. 105\)](#)」を参照してください。

Amazon EBS スナップショットごとに `createVolumePermission` 属性があり、スナップショットを使用できる AWS アカウントを制御します。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

## IAM および Amazon EC2

IAM を使って以下を行えます。

- お客様の AWS アカウントでユーザーとグループを作成する
- お客様の AWS アカウントでユーザーごとに固有のセキュリティ認証情報を割り当てる
- AWS のリソースを使用してタスクを実行するために各ユーザーのアクセス許可を制御する
- 別の AWS アカウントのユーザーがお客様の AWS のリソースを共有できるようにする
- AWS アカウントにロールを作成し、それを行えるユーザーまたはサービスを定義する
- お客様の企業用の既存のアイデンティティを使用し、AWS のリソースを使用してタスクを実行するようアクセス許可を与える

Amazon EC2 と組み合わせて IAM を使用すると、組織のユーザーが特定の Amazon EC2 API アクションを使用してタスクを実行できるかどうか、また、特定の AWS リソースを使用できるかどうかを制御できます。

このトピックには、以下の質問に対する回答があります。

- IAM でグループとユーザーを作成するには、どうすればよいですか？
- ポリシーを作成するには、どうすればよいですか？

- Amazon EC2 でタスクを実行するには、どのような IAM ポリシーが必要ですか？
- Amazon EC2 でアクションを実行するための許可を与えるには、どうすればよいですか？
- Amazon EC2 の特定のリソースでアクションを実行するための許可を与えるには、どうすればよいですか？

## IAM のグループとユーザーを作成する

IAM グループを作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで、[Groups]、[Create New Group] の順に選択します。
3. [グループ名] にグループの名前を入力し、[次のステップ] を選択します。
4. [Attach Policy (ポリシーのアタッチ)] ページで、AWS 管理ポリシーを選択し、[次のステップ] を選択します。たとえば、Amazon EC2 では以下の AWS 管理ポリシーのいずれかが、ニーズを満たす場合があります。
  - PowerUserAccess
  - ReadOnlyAccess
  - AmazonEC2FullAccess
  - AmazonEC2ReadOnlyAccess
5. [Create Group (グループの作成)] を選択します。

新しいグループは、[Group Name] の下に表示されます。

IAM ユーザーを作成するには、グループにユーザーを追加し、ユーザーのパスワードを作成します。

1. ナビゲーションペインで、[Users]、[Add user] を選択します。
2. [ユーザー名] には、ユーザー名を入力します。
3. [アクセスタイプ] で、[Programmatic access (プログラムによるアクセス) と [AWS マネジメントコンソール アクセス] の両方を選択します。
4. [Console password] で、以下のいずれかを選択します。
  - [Autogenerated password]。現在有効なパスワードポリシーがある場合、このポリシーと合致するパスワードがランダムに生成されて各ユーザーに付与されます。[Final] ページに到達すると、パスワードを表示またはダウンロードできます。
  - [Custom password]。ボックスに入力したパスワードが各ユーザーに割り当てられます。
5. [Next: Permissions (次へ: アクセス許可)] を選択します。
6. [Set permissions] ページで、[Add user to group] を選択します。先ほど作成したグループの横にあるチェックボックスを選択し、[Next: Review] を選択します。
7. [Create user] を選択します。
8. ユーザーのアクセスキー（アクセスキー ID とシークレットアクセスキー）を表示するには、各パスワードおよび表示するシークレットアクセスキーの横にある [Show] をクリックします。アクセスキーを保存するには、[Download .csv] を選択し、安全な場所にファイルを保存します。

### Important

この手順の完了後はシークレットアクセスキーを取得できません。置き場所を忘れた場合は、新しく作成しなければなりません。

9. [Close] を選択します。
10. ユーザーごとに認証情報（アクセスキーとパスワード）を与えます。これにより、IAM グループ用に指定したアクセス許可に基づいてサービスを使用できるようになります。

## 関連トピック

IAM の詳細については、以下を参照してください。

- Amazon EC2 の IAM ポリシー (p. 842)
- Amazon EC2 の IAM ロール (p. 888)
- AWS Identity and Access Management (IAM)
- IAM ユーザーガイド

## Amazon EC2 の IAM ポリシー

デフォルトでは、IAM ユーザーには Amazon EC2 リソースを作成または変更、または Amazon EC2 API を使用するタスクを実行する権限がありません。(つまり、Amazon EC2 コンソールまたは CLI を使用して実行することもできません。) IAM ユーザーがリソースを作成または変更、およびタスクを実行できるようにするには、IAM ポリシーを作成する必要があります。これによって、必要な特定のリソースおよび API アクションを使用するためのアクセス許可を IAM ユーザーに付与し、その後、ポリシーをそのアクセス許可が必要な IAM ユーザーまたはグループにアタッチします。

ポリシーをユーザーまたはユーザーのグループにアタッチする場合、ポリシーによって特定リソースの特定タスクを実行するユーザーの権限が許可または拒否されます。IAM ポリシーの一般的な情報については、IAM ユーザーガイドの「[Permissions and Policies](#)」を参照してください。カスタム IAM ポリシーの管理と作成の詳細については、「[IAM ポリシーの管理](#)」を参照してください。

### はじめに

IAM ポリシーは、1 つ以上の Amazon EC2 アクションを使用するアクセス許可を付与または拒否する必要があります。さらに、このアクションで使用できるリソース(すべてのリソースか、場合によっては特定のリソース)も指定する必要があります。このポリシーには、リソースに適用する条件も含めることができます。

Amazon EC2 では、リソースレベルのアクセス許可が部分的にサポートされます。これは、一部の EC2 API アクションでは、ユーザーがそのアクションに使用できるリソースを指定できないことを意味します。代わりに、ユーザーがそのアクションにすべてのリソースを使用することを許可する必要があります。

| タスク                                 | トピック                                                         |
|-------------------------------------|--------------------------------------------------------------|
| ポリシーの基本構造について                       | <a href="#">ポリシー構文 (p. 843)</a>                              |
| ポリシーでのアクションの定義                      | <a href="#">Amazon EC2 のアクション (p. 843)</a>                   |
| ポリシーでの特定のリソースの定義                    | <a href="#">Amazon EC2 用の Amazon リソースネーム (ARN) (p. 844)</a>  |
| リソースの使用への条件の適用                      | <a href="#">Amazon EC2 の条件キー (p. 845)</a>                    |
| Amazon EC2 での使用可能なリソースレベルのアクセス許可の使用 | <a href="#">Amazon EC2 のアクション、リソース、および条件キー (IAM ユーザーガイド)</a> |
| ポリシーのテスト                            | ユーザーが必要なアクセス許可を持っているかどうかを確認する (p. 846)                       |
| CLI または SDK のサンプルポリシー               | <a href="#">AWS CLI または AWS SDK で使用するサンプルポリシー (p. 849)</a>   |
| Amazon EC2 コンソールのサンプルポリシー           | <a href="#">Amazon EC2 コンソールで機能するサンプル ポリシー (p. 880)</a>      |

## ポリシーの構造

次のトピックでは、IAM ポリシーの簡単な構造について説明します。

### コンテンツ

- [ポリシー構文 \(p. 843\)](#)
- [Amazon EC2 のアクション \(p. 843\)](#)
- [Amazon EC2 API アクションでサポートされるリソースレベルのアクセス許可 \(p. 844\)](#)
- [Amazon EC2 用の Amazon リソースネーム \(ARN\) \(p. 844\)](#)
- [Amazon EC2 の条件キー \(p. 845\)](#)
- [ユーザーが必要なアクセス許可を持っているかどうかを確認する \(p. 846\)](#)

## ポリシー構文

IAM ポリシーは 1 つ以上のステートメントで構成される JSON ドキュメントです。各ステートメントは次のように構成されます。

```
{  
  "Statement": [  
    {  
      "Effect": "effect",  
      "Action": "action",  
      "Resource": "arn",  
      "Condition": {  
        "condition": {  
          "key": "value"  
        }  
      }  
    }  
  ]  
}
```

ステートメントはさまざまなエレメントで構成されます。

- [Effect]: effect は、Allow または Deny にすることができます。デフォルトでは、IAM ユーザーはリソースおよび API アクションを使用するアクセス許可がないため、リクエストはすべて拒否されます。明示的な許可はデフォルトに優先します。明示的な拒否はすべての許可に優先します。
- [Action]: action は、アクセス許可を付与または拒否する対象とする、特定の API アクションです。action の指定については、[Amazon EC2 のアクション \(p. 843\)](#) を参照してください。
- [Resource]: アクションによって影響を及ぼされるリソースです。Amazon EC2 API アクションの中には、アクションによって作成/変更できるリソースをポリシー内で特定できるものもあります。Amazon リソースネーム (ARN) を使用して、またはステートメントがすべてのリソースに適用されることを示すワイルドカード (\*) を使用して、リソースを指定します。詳細については、「[Amazon EC2 API アクションでサポートされるリソースレベルのアクセス許可 \(p. 844\)](#)」を参照してください。
- [Condition]: condition はオプションです。ポリシーの発効条件を指定するために使用します。Amazon EC2 の条件を指定する方法については、[Amazon EC2 の条件キー \(p. 845\)](#) を参照してください。

Amazon EC2 の IAM ポリシーステートメント例についての詳細は、「[AWS CLI または AWS SDK で使用するサンプルポリシー \(p. 849\)](#)」を参照してください。

## Amazon EC2 のアクション

IAM ポリシーステートメントで、IAM をサポートするすべてのサービスから任意の API アクションを指定できます。Amazon EC2 の場合、API アクション ec2: の名前に次のプレフィックスを使用します。例: ec2:RunInstances および ec2:CreateImage。

単一のステートメントに複数のアクションを指定するには、次のようにコンマで区切ります。

```
"Action": [ "ec2:action1", "ec2:action2" ]
```

ワイルドカードを使用して複数のアクションを指定することもできます。たとえば、以下のように「Describe」という単語で始まる名前のすべてのアクションを指定できます。

```
"Action": "ec2:Describe*"
```

Amazon EC2 API アクションをすべて指定するには、\* ワイルドカードを以下のように使用します。

```
"Action": "ec2:/*"
```

Amazon EC2 アクションのリストについては、『Amazon EC2 API Reference』の「[アクション](#)」を参照してください。

## Amazon EC2 API アクションでサポートされるリソースレベルのアクセス許可

リソースレベルのアクセス許可とは、ユーザーがアクションを実行可能なリソースを指定できることを意味します。Amazon EC2 では、リソースレベルのアクセス許可が部分的にサポートされます。これは、特定の Amazon EC2 アクションでは、満たす必要がある条件、またはユーザーが使用できる特定のリソースに基づいて、ユーザーがそれらのアクションをいつ使用できるかを制御できることを意味します。たとえば、特定の AMI のみを使用して、特定のタイプのインスタンスだけを起動するアクセス許可をユーザーに付与できます。

IAM ポリシーステートメントでリソースを指定するには、Amazon リソースネーム (ARN) を使用します。ARN 値の指定については、「[Amazon EC2 用の Amazon リソースネーム \(ARN\) \(p. 844\)](#)」を参照してください。API アクションが個々の ARN をサポートしていない場合は、ワイルドカード (\*) を使用して、アクションによってすべてのリソースが影響を受ける可能性があることを指定する必要があります。

リソースレベルのアクセス許可をサポートする Amazon EC2 API アクション、およびポリシーで使用できる ARN と条件キーがわかる表を見るには、IAM ユーザーガイドの「[Amazon EC2 のアクション、リソース、および条件キー](#)」を参照してください。Amazon EC2 の条件キーについては、後のセクションで詳しく説明します。

Amazon EC2 API アクションに対して使用する IAM ポリシーで、タグベースのリソースレベルアクセス許可を適用できます。これにより、ユーザーがどのリソースを作成、変更、または使用できるかを制御しやすくなります。詳細については、「[リソース作成時にタグ付けするアクセス許可の付与 \(p. 847\)](#)」を参照してください。

## Amazon EC2 用の Amazon リソースネーム (ARN)

各 IAM ポリシーステートメントは、ARN を使用して指定したリソースに適用されます。

ARN には以下の一般的な構文があります。

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

サービス (例: ec2)。

リージョン

リソースのリージョン (例: us-east-1)。

アカウント

ハイフンなしの AWS アカウント ID (例: 123456789012)。

resourceType

リソースの種類 (例: `instance`)。

resourcePath

リソースを識別するパス。パスにワイルドカードの `*` が使用できます。

たとえば、以下のように ARN を使用して、ステートメント内で特定のインスタンス (`i-1234567890abcdef0`) を指定することができます。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

以下のように `*` ワイルドカードを使用して、特定のアカウントに属するすべてのインスタンスを指定できます。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

また、以下のように `*` ワイルドカードを使用して、特定のアカウントに属するすべての Amazon EC2 リソースを指定することもできます。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:*
```

すべてのリソースを指定する場合、または特定の API アクションが ARN をサポートしていない場合は、以下のように、`Resource` エレメント内で `*` ワイルドカードを使用します。

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

Amazon EC2 API アクションの多くが複数のリソースと関連します。たとえば、`AttachVolume` では Amazon EBS ボリュームをインスタンスにアタッチするため、IAM ユーザーはボリュームおよびインスタンスを使用するアクセス許可が必要です。1 つのステートメントで複数のリソースを指定するには、次のように ARN をカンマで区切ります。

```
"Resource": ["arn1", "arn2"]
```

Amazon EC2 リソースの ARN のリストについては、IAM ユーザーガイドの「[Amazon EC2 で定義されるリソースタイプ](#)」を参照してください。

## Amazon EC2 の条件キー

ポリシーステートメントでは、オプションで有効になるタイミングを制御する条件を指定できます。各条件には 1 つ以上のキーと値のペアが含まれます。条件キーは大文字小文字を区別しません。私たちは AWS 範囲の条件キーに加え、追加のサービス固有の条件キーを定義しました。

Amazon EC2 のサービス固有の条件キーのリストについては、IAM ユーザーガイドの「[Amazon EC2 の条件キー](#)」を参照してください。Amazon EC2 は、AWS 全体の条件キーも実装します。詳細については、「[IAM ユーザーガイド](#)」の「[すべてのリクエストで利用可能な情報](#)」を参照してください。

IAM ポリシーで条件キーを使用するには、`Condition` ステートメントを使用します。たとえば、次のポリシーは、セキュリティグループのインバウンドルールとアウトバウンドルールを追加および削除するアクセス許可をユーザーに付与します。`ec2:Vpc` 条件キーを使用して、これらのアクションを実行できる対象は、特定の VPC 内のセキュリティグループに限ることを指定します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
"Action": [
    "ec2:AuthorizeSecurityGroupIngress",
    "ec2:AuthorizeSecurityGroupEgress",
    "ec2:RevokeSecurityGroupIngress",
    "ec2:RevokeSecurityGroupEgress"],
"Resource": "arn:aws:ec2:region:account:security-group/*",
"Condition": {
    "StringEquals": {
        "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-11223344556677889"
    }
}
]
```

複数の条件、または単一の条件に複数のキーを指定する場合、論理 AND 演算を使用してそれらを評価します。1つのキーに複数の値を使用して単一の条件を指定する場合、論理 OR 演算を使用して条件を評価します。アクセス許可が付与されるには、すべての条件を満たしている必要があります。

条件を指定する際にプレースホルダーも使用できます。たとえば、IAM ユーザーに、そのユーザーの IAM ユーザー名を指定したタグ付きのリソースを使用するアクセス許可を与えることができます。詳細については、IAM ユーザーガイドの「[ポリシー変数](#)」を参照してください。

#### Important

多くの条件キーはリソースに固有のものであり、一部の API アクションでは複数のリソースを使用します。条件キーを使用してポリシーを作成する場合は、ポリシーステートメントの Resource 要素で、条件キーが適用されるリソースを指定します。指定しない場合、そのポリシーはユーザーに対してすべてのアクションの実行を禁止します。これは、条件キーが適用されないリソースに対して条件チェックが失敗するためです。リソースを指定しない場合や、ポリシーの Action 要素に複数の API アクションを含めている場合は、...IfExists 条件タイプを使用して、条件キーが適用されないリソースに対して無視されるようにする必要があります。詳細については、『IAM ユーザーガイド』の「[...IfExists 条件](#)」を参照してください。

すべての Amazon EC2 アクションは、aws:RequestedRegion および ec2:Region 条件キーをサポートします。詳細については、「[例: 特定のリージョンへのアクセスの制限 \(p. 850\)](#)」を参照してください。

ec2:SourceInstanceARN キーは、リクエストの生成元インスタンスの ARN を指定する条件に使用できます。この条件キーは、使用可能な AWS 全体を対象しており、サービス固有ではありません。ポリシーの例については、「[EC2 インスタンスがボリュームをアタッチまたはデタッチすることを許可する](#)」と「[例: 特定のインスタンスが他の AWS サービスでリソースを表示できるようにする \(p. 876\)](#)」を参照してください。ec2:SourceInstanceARN キーは、ステートメントの Resource 要素に ARN を入力する変数として使用することはできません。

Amazon EC2 のポリシーステートメントの例については、[AWS CLI または AWS SDK で使用するサンプルポリシー \(p. 849\)](#) を参照してください。

### ユーザーが必要なアクセス許可を持っているかどうかを確認する

IAM ポリシーを作成したら、ポリシーを本稼働環境に置く前に、そのポリシーがユーザーに特定の API アクションおよび必要なリソースを使用するアクセス許可を付与しているかどうかを確認することをお勧めします。

まずテスト目的の IAM ユーザーを作成し、作成した IAM ポリシーをテストユーザーにアタッチします。次に、テストユーザーとしてリクエストを作成します。

テストしている Amazon EC2 アクションがリソースを作成または変更する場合、DryRun パラメータを使用してリクエストを作成する（または、--dry-run オプションで AWS CLI コマンドを実行する）必要があります。この場合、発信者は認証チェックを行いますが、操作は完了しません。たとえば、実際に終了させることなく、ユーザーが特定のインスタンスを終了できるかどうかを確認できます。テストユーザーに

必要なアクセス許可がある場合、リクエストで DryRunOperation が返されます。必要なアクセス許可がない場合は UnauthorizedOperation が返されます。

ポリシーが想定したアクセス許可をユーザーに付与していない場合、または過度に許可されている場合、必要に応じてポリシーを調整し、必要な結果を得るまで再テストできます。

**Important**

ポリシーの変更が反映され、有効になるには数分間かかります。したがって、ポリシーの更新をテストするには 5 分かかると見ておいてください。

認証チェックが失敗した場合、リクエストでは診断情報でエンコードされたメッセージが返されます。DecodeAuthorizationMessage アクションを使用してメッセージをデコードできます。詳細については、AWS Security Token Service API リファレンス の「[DecodeAuthorizationMessage](#)」、および「AWS CLI Command Reference」の「[decode-authorization-message](#)」を参照してください。

## リソース作成時にタグ付けするアクセス許可の付与

一部のリソース作成 Amazon EC2 API アクションでは、リソースの作成時にタグを指定できます。詳細については、「[リソースにタグを付ける \(p. 1121\)](#)」を参照してください。

ユーザーがリソースの作成時にタグを付けるには、リソースを作成するアクション (ec2:RunInstances や ec2:CreateVolume など) を使用するためのアクセス許可が必要です。タグがリソース作成アクションで指定されている場合、Amazon は ec2:CreateTags アクションで追加の承認を実行してユーザーがタグを作成するアクセス権限を持っているかどうかを確認します。そのため、ユーザーには、ec2:CreateTags アクションを使用する明示的なアクセス権限が必要です。

ec2:CreateTags アクションの IAM ポリシー定義で、Condition 要素と ec2:CreateAction 条件キーを使用して、リソースを作成するアクションにタグ付けのアクセス許可を付与します。

次のポリシー例では、インスタンスを起動し、起動時にインスタンスとボリュームにタグを適用することをユーザーに許可します。ユーザーには、既存のリソースへのタグ付けが許可されません (ec2:CreateTags アクションを直接呼び出すことはできません)。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account:/*/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction" : "RunInstances"
        }
      }
    }
  ]
}
```

同様に、次のポリシーでは、ユーザーがボリュームを作成し、ボリューム作成時にボリュームにタグを適用することができます。ユーザーには、既存のリソースへのタグ付けが許可されません (ec2:CreateTags アクションを直接呼び出すことはできません)。

```
{  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateVolume"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:*/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:CreateAction" : "CreateVolume"  
                }  
            }  
        }  
    ]  
}
```

ec2:CreateTags アクションは、タグがリソース作成アクション時に適用された場合のみ評価されます。したがって、リクエストでタグが指定されていない場合、リソースを作成するアクセス権限を持っているユーザー（タグ付け条件がないと仮定）には、ec2:CreateTags アクションを実行するアクセス権限が必要ありません。ただし、ユーザーがタグを使用してリソースを作成しようとした場合、ユーザーが ec2:CreateTags アクションを使用するアクセス権限を持っていない場合はリクエストに失敗します。

ec2:CreateTags アクションは、タグが起動テンプレートに指定されている場合にも評価されます。ポリシーの例については、「[起動テンプレートのタグ \(p. 869\)](#)」を参照してください。

## 特定のタグに対するアクセスの制御

IAM ポリシーの Condition 要素で追加の条件を使用して、リソースに適用できるタグキーとタグ値を制御できます。

次の条件キーは、前のセクションの例で使用できます。

- `aws:RequestTag`: 特定のタグキーまたはタグキーと値がリクエストに存在している必要があることを指定する場合に使用します。リクエストでは他のタグも指定できます。
- `StringEquals` 条件演算子とともに使用して、特定のタグキーと値の組み合わせを適用します。たとえば、タグ `cost-center=cc123` を適用します。

```
"StringEquals": { "aws:RequestTag/cost-center": "cc123" }
```

- `StringLike` 条件演算子とともに使用して、リクエストで特定のタグキーを適用します。たとえば、タグキー `purpose` を適用します。

```
"StringLike": { "aws:RequestTag/purpose": "*" }
```

- `aws:TagKeys`: リクエストで使用されるタグキーを適用する場合に使用します。
  - リクエストにタグが指定されている場合は、`ForAllValues` 修飾子を使用して特定のタグキーのみを適用します（リクエストにタグが指定されている場合、特定のタグキーのみが許可されます。他のタグは許可されません）。たとえば、タグキー `environment` または `cost-center` が適用されます：

```
"ForAllValues:StringEquals": { "aws:TagKeys": ["environment", "cost-center"] }
```

- ForAnyValue 修飾子とともに使用して、指定されたタグキーの少なくとも 1 つがリクエストに存在することを要求します。たとえば、タグキー environment または webserver のうち少なくとも 1 つがリクエストに存在している必要があります。

```
"ForAnyValue:StringEquals": { "aws:TagKeys": ["environment", "webserver"] }
```

これらの条件キータグ付けをサポートするリソース作成アクションと、ec2:CreateTags および ec2:DeleteTags アクションに適用できます。Amazon EC2 API アクションがタグ付けをサポートしているかどうかについては、IAM ユーザーガイドの「[Amazon EC2 のアクション、リソース、および条件キー](#)」を参照してください

リソースの作成時にタグを指定するようにユーザーに強制するには、リソース作成アクションで aws:RequestTag 修飾子とともに aws:TagKeys 条件キーまたは ForAnyValue 条件キーを使用する必要があります。ユーザーがリソース作成アクションのタグを指定しない場合、ec2:CreateTags アクションは評価されません。

条件においては、条件キーでは大文字と小文字が区別されず、条件値では大文字と小文字が区別されます。したがって、タグキーの大文字と小文字を区別するには、条件の値としてタグキーが指定される aws:TagKeys 条件キーを使用します。

IAM ポリシーの例は、「[AWS CLI または AWS SDK で使用するサンプルポリシー \(p. 849\)](#)」を参照してください。複数値の条件の詳細については、「[IAM ユーザーガイド](#)」の「[複数のキー値をテストする条件を作成する](#)」を参照してください。

## EC2 リソースタグを使用したアクセスの制御

タグに基づいてアクセスを制御するには、ポリシーの Condition 要素でタグ情報を指定することもできます。これにより、ユーザーが変更、使用、または削除できる EC2 リソースをより適切に制御できます。

たとえば、インスタンスを終了することをユーザーに許可するが、インスタンスに environment=production タグが付いている場合はアクションを拒否するポリシーを作成できます。これを行うには、ec2:ResourceTag 条件キーを使用し、リソースにアタッチされているタグに基づいてリソースへのアクセスを許可または拒否します。

```
"StringEquals": { "ec2:ResourceTag/environment": "production" }
```

Amazon EC2 API アクションが ec2:ResourceTag 条件キーを使用したアクセスの制御をサポートしているかどうかについては、IAM ユーザーガイドの「[Amazon EC2 のアクション、リソース、および条件キー](#)」を参照してください。Describe アクションはリソースレベルのアクセス許可をサポートしないため、これらのアクセス許可は、条件なしの別のステートメントで指定する必要があります。

IAM ポリシーの例は、「[AWS CLI または AWS SDK で使用するサンプルポリシー \(p. 849\)](#)」を参照してください。

### Note

タグに基づいてリソースへのユーザーのアクセスを許可または拒否する場合は、ユーザーが同じリソースに対してそれらのタグを追加または削除することを明示的に拒否することを検討する必要があります。そうしないと、ユーザーはそのリソースのタグを変更することで、制限を回避してリソースにアクセスできてしまいます。

## AWS CLI または AWS SDK で使用するサンプルポリシー

以下の例では、Amazon EC2 に対して IAM ユーザーが所有するアクセス許可を制御するために使用できるポリシーステートメントを示しています。これらのポリシーは、AWS CLI または AWS SDK で行われたリクエスト向けに設計されています。Amazon EC2 コンソールで機能するポリシーの例については、「[Amazon EC2 コンソールで機能するサンプル ポリシー \(p. 880\)](#)」を参照してください。Amazon VPC

に固有の IAM ポリシーの例については、「[Amazon VPC の Identity and Access Management](#)」を参照してください。

例

- [例: 読み取り専用アクセス \(p. 850\)](#)
- [例: 特定のリージョンへのアクセスの制限 \(p. 850\)](#)
- [インスタンスの使用 \(p. 851\)](#)
- [ボリュームの操作 \(p. 852\)](#)
- [スナップショットの操作 \(p. 855\)](#)
- [インスタンスを起動する \(RunInstances\) \(p. 862\)](#)
- [例: リザーブドインスタンス の使用 \(p. 872\)](#)
- [例: リソースのタグ付け \(p. 873\)](#)
- [例: IAM ロールの使用 \(p. 875\)](#)
- [例: ルートテーブルを操作する \(p. 876\)](#)
- [例: 特定のインスタンスが他の AWS サービスでリソースを表示できるようにする \(p. 876\)](#)
- [例: 起動テンプレートの使用 \(p. 877\)](#)
- [インスタンスマタデータの使用 \(p. 878\)](#)

### 例: 読み取り専用アクセス

次のポリシーでは、名前が `Describe` で始まるすべての Amazon EC2 API アクションを使用できるアクセス許可をユーザーに与えます。Resource エレメントにワイルドカードを使用します。これは、ユーザーが API アクションですべてのリソースを指定できることを示します。また、API アクションがリソースレベルのアクセス許可をサポートしていない場合も、\* ワイルドカードが必要です。どの Amazon EC2 API アクションでどの ARN を使用できるかの詳細については、IAM ユーザーガイドの「[Amazon EC2 のアクション、リソース、および条件キー](#)」を参照してください。

デフォルトで API アクションを使用するアクセス許可が拒否されているため、ユーザーには（別のステートメントでアクセス許可が与えられない限り）そのリソースに対してアクションを実行するアクセス許可がありません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:Describe*",  
            "Resource": "*"  
        }  
    ]  
}
```

### 例: 特定のリージョンへのアクセスの制限

次のポリシーでは、リージョンが 欧州 (フランクフルト) でない限り、すべての Amazon EC2 API アクションを使用するアクセス許可をユーザーに拒否します。これにはグローバル条件キー `aws:RequestedRegion` が使用され、このキーはすべての Amazon EC2 API アクションでサポートされています。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {  
                "aws:RequestedRegion": "eu-central-1"  
            }  
        }  
    ]  
}
```

```
"Condition": {
    "StringNotEquals": {
        "aws:RequestedRegion": "eu-central-1"
    }
}
]
```

または、条件キー `ec2:Region` を使用することもできます。これは、Amazon EC2 に固有のもので、すべての Amazon EC2 API アクションでサポートされています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "ec2:Region": "eu-central-1"
                }
            }
        }
    ]
}
```

## インスタンスの使用

### 例

- 例: すべてのインスタンスを記述、起動、停止、開始、および終了する (p. 851)
- 例: すべてのインスタンスを記述し、特定のインスタンスのみを停止、開始、および終了する (p. 852)

### 例: すべてのインスタンスを記述、起動、停止、開始、および終了する

次のポリシーでは、`Action` エレメントで指定された API アクションを使用するアクセス許可をユーザーに与えます。`Resource` エレメントでは \* ワイルドカードを使用して、ユーザーが API アクションですべてのリソースを指定できることを示します。また、API アクションがリソースレベルのアクセス許可をサポートしていない場合も、\* ワイルドカードが必要です。どの Amazon EC2 API アクションでどの ARN を使用できるかの詳細については、IAM ユーザーガイドの「[Amazon EC2 のアクション、リソース、および条件キー](#)」を参照してください。

ユーザーはデフォルトで API アクションを使用するアクセス許可を拒否されているため、ユーザーには(別のステートメントでユーザーにそのアクセス許可を与えない限り) その他の API アクションを使用するアクセス許可がありません。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances", "ec2:DescribeImages",
                "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
                "ec2:DescribeAvailabilityZones",
                "ec2:RunInstances", "ec2:TerminateInstances",
                "ec2:StopInstances", "ec2:StartInstances"
            ],
            "Resource": "*"
        }
    ]
}
```

```
}
```

#### 例: すべてのインスタンスを記述し、特定のインスタンスのみを停止、開始、および終了する

次のポリシーでは、すべてのインスタンスを表示し、i-1234567890abcdef0 と i-0598c7d356eba48d7 インスタンスのみを開始および停止し、米国東部 (バージニア北部) リージョン (us-east-1) 内でリソースタグ "purpose=test" の付いたインスタンスのみを終了する許可をユーザーに与えます。

最初のステートメントでは、Resource エレメントに \* ワイルドカードを使用して、ユーザーがそのアクションにすべてのリソースを指定できることを示しています。この場合、すべてのインスタンスをリストできます。また、API アクションがリソースレベルのアクセス許可をサポートしていない場合も、\* ワイルドカードが必要です (この場合は、ec2:DescribeInstances)。どの Amazon EC2 API アクションでどの ARN を使用できるかの詳細については、IAM ユーザーガイドの「[Amazon EC2 のアクション、リソース、および条件キー](#)」を参照してください。

2 番目のステートメントでは、StopInstances および StartInstances アクションに対してリソースレベルのアクセス許可を使用しています。Resource エレメント内で、ARN によって特定のインスタンスが指定されています。

3 番目のステートメントでは、指定された AWS アカウントに属する米国東部 (バージニア北部) リージョン (us-east-1) 内にあり、タグ "purpose=test" が付けられているすべてのインスタンスを終了する許可をユーザーに与えています。Condition エレメントは、ポリシーステートメントの発効条件を指定します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances",
                "ec2:StartInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:123456789012:instance/i-1234567890abcdef0",
                "arn:aws:ec2:us-east-1:123456789012:instance/i-0598c7d356eba48d7"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:TerminateInstances",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/purpose": "test"
                }
            }
        }
    ]
}
```

#### ボリュームの操作

例

- 例: ボリュームをアタッチおよびデタッチする (p. 853)
- 例: ボリュームの作成 (p. 853)
- 例: タグ付きのボリュームの作成 (p. 854)

#### 例: ボリュームをアタッチおよびデタッチする

API アクションが複数のリソースを指定するために発信者を必要とする場合、ユーザーがすべての必要なリソースにアクセスできるようにポリシーステートメントを作成する必要があります。1 つ以上のリソースで Condition エレメントを使用する必要がある場合、この例のとおり複数のステートメントを作成する必要があります。

以下のポリシーでは、ユーザーがタグ「volume\_user=iam-user-name」の付いたボリュームを、タグ「department=dev」の付いたインスタンスにアタッチしたり、またインスタンスからボリュームをデタッチしたりできるようにします。このポリシーを IAM グループにアタッチする場合、aws:username ポリシー変数によってグループの IAM ユーザーに、値として IAM ユーザー名を持つタグ名が volume\_user のインスタンスからボリュームをアタッチまたはデタッチするためのアクセス許可が付与されます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AttachVolume",  
                "ec2:DetachVolume"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/department": "dev"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:AttachVolume",  
                "ec2:DetachVolume"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/volume_user": "${aws:username}"  
                }  
            }  
        }  
    ]  
}
```

#### 例: ボリュームの作成

次のポリシーでは、ユーザーが [CreateVolume](#) API アクションを使用することができます。ユーザーは、ボリュームが暗号化されていて、ボリューム サイズが 20 GiB 未満の場合にのみボリュームの作成を許可されます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateVolume",  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "ec2:VolumeKmsKey": "<no value>"  
                }  
            }  
        }  
    ]  
}
```

```
"Action": [
    "ec2:CreateVolume"
],
"Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
"Condition": {
    "NumericLessThan": {
        "ec2:VolumeSize" : "20"
    },
    "Bool": {
        "ec2:Encrypted" : "true"
    }
}
]
```

#### 例: タグ付きのボリュームの作成

次のポリシーには、タグ `aws:RequestTag` および `costcenter=115` を使用して作成したすべてのボリュームへのタグ付けをユーザーに求める `stack=prod` 条件キーが含まれています。`aws:TagKeys` 条件キーは、`ForAllValues` 修飾子を使用し、キー `costcenter` および `stack` のみがリクエストで許可されることを指定します(他のタグは指定できません)。ユーザーがこれらのタグを渡さないか、タグをまったく指定しない場合、リクエストは失敗します。

タグを適用するリソース作成アクションでは、ユーザーが `CreateTags` アクションを使用するアクセス権限を持っていることも必要です。2番目のステートメントは、`ec2:CreateAction` 条件キーを使用して、ユーザーが `CreateVolume` のコンテキストでみタグを使用できるようにします。ユーザーは、既存のボリュームにも他のリソースにもタグ付けできません。詳細については、「[リソース作成時にタグ付けるアクセス許可の付与 \(p. 847\)](#)」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateTaggedVolumes",
            "Effect": "Allow",
            "Action": "ec2:CreateVolume",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/costcenter": "115",
                    "aws:RequestTag/stack": "prod"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": ["costcenter", "stack"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction" : "CreateVolume"
                }
            }
        }
    ]
}
```

次のポリシーでは、ユーザーがタグを指定しなくてもボリュームを作成することができます。CreateTags アクションは、タグが CreateVolume リクエストで指定されている場合にのみ評価されます。ユーザーがタグを指定する場合、purpose=test タグを指定する必要があります。リクエストでは他のタグは許可されません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateVolume",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:us-east-1:1234567890:volume/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/purpose": "test",  
                    "ec2:CreateAction" : "CreateVolume"  
                },  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": "purpose"  
                }  
            }  
        }  
    ]  
}
```

## スナップショットの操作

以下に、CreateSnapshot (EBS ボリュームのポイントインタイムスナップショット) と CreateSnapshots (マルチボリュームスナップショット) の両方のポリシーの例を示しています。

### 例

- 例: 手動スナップショットの作成 (p. 855)
- 例: スナップショットの作成 (p. 856)
- 例: タグ付きのスナップショットの作成 (p. 856)
- 例: タグ付きのスナップショットの作成 (p. 857)
- 例: スナップショットのアクセス許可設定を変更する (p. 862)

### 例: 手動スナップショットの作成

次のポリシーでは、お客様が CreateSnapshot API アクションを使用することができます。お客様は、ボリュームが暗号化されていて、ボリューム サイズが 20 GiB 未満の場合にのみスナップショットを作成できます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateSnapshot",  
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:DeleteSnapshot",  
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*"  
        }  
    ]  
}
```

```
"Effect": "Allow",
"Action": "ec2:CreateSnapshot",
"Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
"Condition": {
    "NumericLessThan": {
        "ec2:VolumeSize": "20"
    },
    "Bool": {
        "ec2:Encrypted": "true"
    }
}
]
```

#### 例: スナップショットの作成

次のポリシーでは、お客様が [CreateSnapshot API](#) アクションを使用することができます。インスタンス上のすべてのボリュームがタイプ GP2 の場合にのみ、お客様はスナップショットを作成できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": [
                "arn:aws:ec2:us-east-1::snapshot/*",
                "arn:aws:ec2:*:*:instance/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSchedules",
            "Resource": "arn:aws:ec2:us-east-1::*:volume/*",
            "Condition": {
                "StringLikeIfExists": {
                    "ec2:VolumeType": "gp2"
                }
            }
        }
    ]
}
```

#### 例: タグ付きのスナップショットの作成

次のポリシーには、タグ `aws:RequestTag` および `costcenter=115` をすべての新しいリクエストに適用することをお客様に求める `stack=prod` 条件キーが含まれています。`aws:TagKeys` 条件キーは、`ForAllValues` 修飾子を使用し、キー `costcenter` および `stack` のみをリクエストで指定できることを示します。リクエストは、これらの条件のいずれかに一致しない場合に失敗します。

タグを適用するリソース作成アクションでは、`CreateTags` アクションを使用するアクセス権限も持っていることが求められます。3番目のステートメントは、`ec2:CreateAction` 条件キーを使用して、お客様が `CreateSnapshot` のコンテキストでみタグを使用できるようにします。お客様は、既存のボリュームにも他のリソースにもタグ付けできません。詳細については、「[リソース作成時にタグ付けするアクセス許可の付与 \(p. 847\)](#)」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/costcenter": "115"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateAction",
            "Resource": "arn:aws:ec2:us-east-1::volume/*",
            "Condition": {
                "StringLike": {
                    "aws:TagKeys": "costcenter,stack"
                }
            }
        }
    ]
}
```

```
{  
    "Effect": "Allow",  
    "Action": "ec2:CreateSnapshot",  
    "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*"  
},  
{  
    "Sid": "AllowCreateTaggedSnapshots",  
    "Effect": "Allow",  
    "Action": "ec2:CreateSnapshot",  
    "Resource": "arn:aws:ec2:us-east-1::snapshot/*",  
    "Condition": {  
        "StringEquals": {  
            "aws:RequestTag/costcenter": "115",  
            "aws:RequestTag/stack": "prod"  
        },  
        "ForAllValues:StringEquals": {  
            "aws:TagKeys": [  
                "costcenter",  
                "stack"  
            ]  
        }  
    }  
},  
{  
    "Effect": "Allow",  
    "Action": "ec2:CreateTags",  
    "Resource": "arn:aws:ec2:us-east-1::snapshot/*",  
    "Condition": {  
        "StringEquals": {  
            "ec2:CreateAction": "CreateSnapshot"  
        }  
    }  
}  
]
```

#### 例: タグ付きのスナップショットの作成

次のポリシーには、タグ `aws:RequestTag` および `costcenter=115` をすべての新しいリクエストに適用することをお客様に求める `stack=prod` 条件キーが含まれています。`aws:TagKeys` 条件キーは、`ForAllValues` 修飾子を使用し、キー `costcenter` および `stack` のみをリクエストで指定できることを示します。リクエストは、これらの条件のいずれかに一致しない場合に失敗します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:CreateSnapshots",  
            "Resource": [  
                "arn:aws:ec2:us-east-1::snapshot/*",  
                "arn:aws:ec2:/*:*:instance/*",  
                "arn:aws:ec2:/*:*:volume/*"  
            ]  
        },  
        {  
            "Sid": "AllowCreateTaggedSnapshots",  
            "Effect": "Allow",  
            "Action": "ec2:CreateSnapshots",  
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/costcenter": "115",  
                    "aws:RequestTag/stack": "prod"  
                }  
            }  
        }  
    ]  
}
```

```
        "aws:RequestTag/stack":"prod"
    },
    "ForAllValues:StringEquals":{
        "aws:TagKeys":[
            "costcenter",
            "stack"
        ]
    }
},
{
    "Effect":"Allow",
    "Action":"ec2:CreateTags",
    "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
    "Condition":{
        "StringEquals":{
            "ec2:CreateAction":"CreateSnapshots"
        }
    }
}
]
```

次のポリシーでは、お客様がタグを指定しなくてもスナップショットを作成することができます。CreateTags アクションは、タグが CreateSnapshot または CreateSnapshots リクエストで指定されている場合にのみ評価されます。タグを指定する場合、タグは purpose=test である必要があります。リクエストでは他のタグは許可されません。

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Action":"ec2:CreateSnapshot",
            "Resource":"*"
        },
        {
            "Effect":"Allow",
            "Action":"ec2:CreateTags",
            "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
            "Condition":{
                "StringEquals":{
                    "aws:RequestTag/purpose":"test",
                    "ec2:CreateAction":"CreateSnapshot"
                },
                "ForAllValues:StringEquals":{
                    "aws:TagKeys":"purpose"
                }
            }
        }
    ]
}
```

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Action":"ec2:CreateSnapshots",
            "Resource":"*"
        },
        {
            "Effect":"Allow",
            "Action":"ec2:CreateTags",
            "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
            "Condition":{
                "StringEquals":{
                    "aws:RequestTag/purpose":"test",
                    "ec2:CreateAction":"CreateSnapshot"
                },
                "ForAllValues:StringEquals":{
                    "aws:TagKeys":"purpose"
                }
            }
        }
    ]
}
```

```
"Action":"ec2:CreateTags",
"Resource":"arn:aws:ec2:us-east-1::snapshot/*",
"Condition":{
    "StringEquals":{
        "aws:RequestTag/purpose":"test",
        "ec2:CreateAction":"CreateSnapshots"
    },
    "ForAllValues:StringEquals":{
        "aws:TagKeys":"purpose"
    }
}
]
```

次のポリシーでは、ソースボリュームにお客様の User:username がタグ付けされていて、スナップショット自体に Environment:Dev と User:username がタグ付けされている場合にのみスナップショットの作成を許可します。お客様は、スナップショットにタグを追加できます。

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect":"Allow",
            "Action":"ec2:CreateSnapshot",
            "Resource":"arn:aws:ec2:us-east-1:123456789012:volume/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/User": "${aws:username}"
                }
            }
        },
        {
            "Effect":"Allow",
            "Action":"ec2:CreateSnapshot",
            "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/Environment": "Dev",
                    "aws:RequestTag/User": "${aws:username}"
                }
            }
        },
        {
            "Effect":"Allow",
            "Action":"ec2:CreateTags",
            "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
        }
    ]
}
```

次の CreateSnapshots のポリシーでは、ソースボリュームにお客様用の User:username がタグ付けされ、スナップショット自体に Environment:Dev と User:username のタグ付けがされている場合にのみスナップショットを作成できます。

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect":"Allow",
            "Action":"ec2:CreateSnapshots",
            "Resource":"arn:aws:ec2:us-east-1::instance/*",
        }
    ]
}
```

```
{  
    "Effect": "Allow",  
    "Action": "ec2:CreateSnapshots",  
    "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
    "Condition": {  
        "StringEquals": {  
            "ec2:ResourceTag/User": "${aws:username}"  
        }  
    }  
},  
{  
    "Effect": "Allow",  
    "Action": "ec2:CreateSnapshots",  
    "Resource": "arn:aws:ec2:us-east-1::snapshot/*",  
    "Condition": {  
        "StringEquals": {  
            "aws:RequestTag/Environment": "Dev",  
            "aws:RequestTag/User": "${aws:username}"  
        }  
    }  
},  
{  
    "Effect": "Allow",  
    "Action": "ec2:CreateTags",  
    "Resource": "arn:aws:ec2:us-east-1::snapshot/*"  
}  
]  
}
```

次のポリシーでは、スナップショットにお客様の User:username がタグ付けされている場合のみスナップショットの削除を許可します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2>DeleteSnapshot",  
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/User": "${aws:username}"  
                }  
            }  
        }  
    ]  
}
```

次のポリシーでは、お客様はスナップショットを作成できますが、作成されるスナップショットにタグキー value=stack が付いている場合はアクションが拒否されます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2>CreateSnapshot",  
                "ec2>CreateTags"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": "ec2>DeleteSnapshot",  
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/Value": "stack"  
                }  
            }  
        }  
    ]  
}
```

```
"Effect":"Deny",
"Action":"ec2:CreateSnapshot",
"Resource":"arn:aws:ec2:us-east-1::snapshot/*",
"Condition": {
    "ForAnyValue:StringEquals": {
        "aws:TagKeys": "stack"
    }
}
]
```

次のポリシーでは、お客様はスナップショットを作成できますが、作成されるスナップショットにタグキー value=stack が付いている場合はアクションが拒否されます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshots",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "ec2:CreateSnapshots",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "stack"
                }
            }
        }
    ]
}
```

次のポリシーでは、複数のアクションを単一のポリシーにまとめることができます。スナップショットがリージョン us-east-1 で作成された場合にのみ、スナップショットを作成することができます (CreateSnapshots のコンテキスト内で)。スナップショットがリージョン us-east-1 に作成されている場合、およびインスタンスタイプが t2\* の場合にのみ、スナップショットを作成できます (CreateSnapshots のコンテキスト内で)。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshots",
                "ec2:CreateSnapshot",
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2::instance/*",
                "arn:aws:ec2::snapshot/*",
                "arn:aws:ec2::volume/*"
            ],
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "ec2:Region": "us-east-1"
                }
            }
        }
    ]
}
```

```
        },
        "StringLikeIfExists": {
            "ec2:InstanceType": [
                "t2.*"
            ]
        }
    }
}
```

#### 例: スナップショットのアクセス許可設定を変更する

次のポリシーでは、スナップショットに `User:username` というタグが付けられている場合にのみスナップショットを変更できます。ここで、`username` はお客様の AWS アカウントのユーザー名です。この条件が満たされない場合、リクエストは失敗します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2: ModifySnapshotAttribute",
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/user-name": "${aws:username}"
                }
            }
        }
    ]
}
```

## インスタンスを起動する (RunInstances)

`RunInstances` API アクションでは、1 つ以上のインスタンスを起動します。`RunInstances` は AMI を必要とし、インスタンスを作成します。ユーザーは、リクエスト内でキーペアとセキュリティグループを指定できます。VPC 内に起動するにはサブネットが必要であり、起動されるとネットワークインターフェイスが作成されます。Amazon EBS-Backed AMI から起動すると、ボリュームが作成されます。そのため、ユーザーにはこれらの Amazon EC2 リソースを使用するアクセス許可が必要です。ユーザーが `RunInstances` に対してオプションのパラメータを指定する必要がある、またはユーザーからパラメータの特定の値を制限するポリシーステートメントを作成できます。

インスタンスの起動に必要なリソースレベルのアクセス許可の詳細については、IAM ユーザーガイドの「[Amazon EC2 のアクション、リソース、および条件キー](#)」を参照してください。

デフォルトでは、作成したインスタンスを記述、開始、停止、または終了するアクセス許可はユーザーに付与されていません。作成したインスタンスを管理するアクセス許可をユーザーに付与する 1 つの方法としては、インスタンスごとに特定のタグを作成し、そのタグでインスタンスを管理できるようにステートメントを作成します。詳細については、「[インスタンスの使用 \(p. 851\)](#)」を参照してください。

### リソース

- [AMI \(p. 863\)](#)
- [インスタンスタイプ \(p. 864\)](#)
- [Subnets \(p. 865\)](#)
- [EBS ボリューム \(p. 866\)](#)
- [タグ \(p. 866\)](#)
- [起動テンプレートのタグ \(p. 869\)](#)

- [Elastic GPU \(p. 870\)](#)
- [起動テンプレート \(p. 870\)](#)

## AMI

次のポリシーでは、指定された AMI (`ami-9e1670f7` および `ami-45cf5c3c`) のみを使用してインスタンスを起動できます。(別のステートメントでユーザーに起動するアクセス許可が付与されない限り) ユーザーはその他の AMI を使用してインスタンスを起動することはできません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region::image/ami-9e1670f7",  
                "arn:aws:ec2:region::image/ami-45cf5c3c",  
                "arn:aws:ec2:region:account:instance/*",  
                "arn:aws:ec2:region:account:volume/*",  
                "arn:aws:ec2:region:account:key-pair/*",  
                "arn:aws:ec2:region:account:security-group/*",  
                "arn:aws:ec2:region:account:subnet/*",  
                "arn:aws:ec2:region:account:network-interface/*"  
            ]  
        }  
    ]  
}
```

一方、以下のポリシーは、Amazon が所有するすべての AMI からインスタンスを起動することをユーザーに許可します。最初のステートメントの `Condition` エレメントは、`ec2:Owner` が `amazon` であるかどうかをテストします。(別のステートメントでユーザーに起動するアクセス許可が付与されない限り) ユーザーはその他の AMI を使用してインスタンスを起動することはできません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region::image/ami-*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Owner": "amazon"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region:account:instance/*",  
                "arn:aws:ec2:region:account:subnet/*",  
                "arn:aws:ec2:region:account:volume/*",  
                "arn:aws:ec2:region:account:network-interface/*",  
                "arn:aws:ec2:region:account:key-pair/*",  
                "arn:aws:ec2:region:account:security-group/*"  
            ]  
        }  
    ]  
}
```

## インスタンスタイプ

次のポリシーにより、ユーザーは t2.micro または t2.small インスタンスタイプのみを使用してインスタンスを起動できます。これにより、コストを管理することができます。最初のステートメントの Condition エレメントは ec2:InstanceType が t2.micro または t2.small のどちらであるかをテストするため、ユーザーは大きなインスタンスを起動することはできません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region:account:instance/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:InstanceType": ["t2.micro", "t2.small"]  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region::image/ami-*",  
                "arn:aws:ec2:region:account:subnet/*",  
                "arn:aws:ec2:region:account:network-interface/*",  
                "arn:aws:ec2:region:account:volume/*",  
                "arn:aws:ec2:region:account:key-pair/*",  
                "arn:aws:ec2:region:account:security-group/*"  
            ]  
        }  
    ]  
}
```

また、ユーザーが t2.micro と t2.small のインスタンスタイプ以外のすべてのインスタンス起動へのアクセスを拒否するポリシーを作成することもできます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region:account:instance/*"  
            ],  
            "Condition": {  
                "StringNotEquals": {  
                    "ec2:InstanceType": ["t2.micro", "t2.small"]  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:region::image/ami-*",  
                "arn:aws:ec2:region:account:network-interface/*",  
                "arn:aws:ec2:region:account:instance/*",  
                "arn:aws:ec2:region:account:subnet/*",  
                "arn:aws:ec2:region:account:volume/*",  
                "arn:aws:ec2:region:account:key-pair/*",  
            ]  
        }  
    ]  
}
```

```
        "arn:aws:ec2:region:account:security-group/*"
    ]
}
}
```

## Subnets

次のポリシーにより、ユーザーは指定したサブネット subnet-12345678 のみを使用してインスタンスを起動できます。グループは、インスタンスを他のサブネットに起動することはできません (他のステートメントがそのような許可をユーザーに与えている場合はその限りではありません)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account:subnet/subnet-12345678",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:image/ami-*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
      ]
    }
  ]
}
```

また、ユーザーがその他のサブネットにインスタンスを起動するアクセス許可を拒否するポリシーを作成することもできます。ステートメントでは、サブネット subnet-12345678 が指定されている場合以外は、ネットワークインターフェイスの作成を拒否することでこれを実行します。この拒否は、他のサブネットへのインスタンスの起動を許可する他のすべてのポリシーよりも優先されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account:network-interface/*"
      ],
      "Condition": {
        "ArnNotEquals": {
          "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:image/ami-*",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/*"
      ]
    }
  ]
}
```

}

## EBS ボリューム

次のポリシーでは、インスタンスの EBS ボリュームが暗号化されている場合のみユーザーがインスタンスを起動できます。ユーザーは、ルートボリュームが暗号化されるように、暗号化されたスナップショットを使用して作成された AMI からインスタンスを起動する必要があります。ユーザーが起動時にインスタンスにアタッチする追加ボリュームも暗号化されている必要があります。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2:*::volume/*"  
            ],  
            "Condition": {  
                "Bool": {  
                    "ec2:Encrypted": "true"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2::::image/ami-*",  
                "arn:aws:ec2::::network-interface/*",  
                "arn:aws:ec2::::instance/*",  
                "arn:aws:ec2::::subnet/*",  
                "arn:aws:ec2::::key-pair/*",  
                "arn:aws:ec2::::security-group/*"  
            ]  
        }  
    ]  
}
```

## タグ

次のポリシーでは、ユーザーがインスタンスを起動し、作成時にインスタンスにタグ付けすることができます。タグを適用するリソース作成アクションには、ユーザーが CreateTags アクションを使用するアクセス権限を持っている必要があります。2 番目のステートメントは、ec2:CreateAction 条件キーを使用し、ユーザーが RunInstances のコンテキストでのみ、インスタンスに対してのみタグを作成できるようにします。ユーザーは、既存のリソースにタグ付けできることができず、RunInstances リクエストを使用してボリュームにタグ付けすることもできません。

詳細については、「[リソース作成時にタグ付けするアクセス許可の付与 \(p. 847\)](#)」を参照してください。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "  
                arn:aws:ec2:  
                instance/  
                <instance-id>  
            >/  
                tag/*"  
        }  
    ]  
}
```

```
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction" : "RunInstances"
        }
    }
}
```

次のポリシーには、`aws:RequestTag` および `RunInstances` タグを使用して `environment=production` により作成されたすべてのインスタンスおよびボリュームへのタグ付けをユーザーに求める `purpose=webserver` 条件キーが含まれています。`aws:TagKeys` 条件キーは、`ForAllValues` 修飾子を使用し、キー `environment` および `purpose` のみがリクエストで許可されることを指定します(他のタグは指定できません)。リクエストでタグが指定されていない場合、リクエストに失敗します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region::image/*",
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:network-interface/*",
                "arn:aws:ec2:region:account:security-group/*",
                "arn:aws:ec2:region:account:key-pair/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region:account:volume/*",
                "arn:aws:ec2:region:account:instance/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production" ,
                    "aws:RequestTag/purpose": "webserver"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": ["environment","purpose"]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction" : "RunInstances"
                }
            }
        }
    ]
}
```

```
        }
    }
}
```

次のポリシーは、`ForAnyValue` 条件で `aws:TagKeys` 修飾子を使用して、リクエストで少なくとも 1 つのタグが指定されている必要があり、キー `environment` または `webserver` が含まれている必要があります。タグは、インスタンスとボリュームの両方に適用される必要があります。リクエストでは、任意のタグ値を指定できます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:region::image/*",
        "arn:aws:ec2:region:account:subnet/*",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:security-group/*",
        "arn:aws:ec2:region:account:key-pair/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
      "Resource": [
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:instance/*"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": ["environment", "webserver"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2>CreateTags"
      ],
      "Resource": "arn:aws:ec2:region:account:/*/*",
      "Condition": {
        "StringEquals": {
          "ec2>CreateAction" : "RunInstances"
        }
      }
    }
  ]
}
```

次のポリシーでは、ユーザーはリクエストでタグを指定する必要はありませんが、指定する場合は `purpose=test` タグを指定する必要があります。他のタグは許可されません。ユーザーは、`RunInstances` リクエストでタグ付け可能なリソースにタグを適用できます。

```
{
  "Version": "2012-10-17",
```

```
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:RunInstances"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateTags"
        ],
        "Resource": "arn:aws:ec2:region:account:*//*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/purpose": "test",
                "ec2:CreateAction" : "RunInstances"
            },
            "ForAllValues:StringEquals": {
                "aws:TagKeys": "purpose"
            }
        }
    }
]
```

## 起動テンプレートのタグ

次の例で、ユーザーはインスタンスを起動できますが、特定の起動テンプレートを使用する場合に限りま  
す (lt-09477bcd97b0d310e)。ec2:IsLaunchTemplateResource 条件キーは、ユーザーが起動テン  
プレートで指定されたリソースを上書きしないようにします。ステートメントの 2 番目の部分では、ユー  
ザーは作成時にインスタンスにタグ付けできます — ステートメントのこの部分は、起動テンプレートでタ  
グがインスタンスに対して指定されている場合に必要になります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/
lt-09477bcd97b0d310e"
                },
                "Bool": {
                    "ec2:IsLaunchTemplateResource": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction" : "RunInstances"
                }
            }
        }
    ]
}
```

```
    ]  
}
```

## Elastic GPU

次のポリシーでは、ユーザーはインスタンスを起動させ、インスタンスにアタッチする Elastic GPU を指定できます。ユーザーは任意のリージョンでインスタンスを起動できますが、Elastic GPU をアタッチできるのはその us-east-2 リージョンでの起動中に限られます。

`ec2:ElasticGpuType` 条件キーは、`ForAnyValue` 修飾子を使用し、elastic GPU タイプ `eg1.medium` および `eg1.large` のみがリクエストで許可されることを指定します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:RunInstances"  
            ],  
            "Resource": [  
                "arn:aws:ec2:*:account:elastic-gpu/*"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Region": "us-east-2"  
                },  
                "ForAnyValue:StringLike": {  
                    "ec2:ElasticGpuType": [  
                        "eg1.medium",  
                        "eg1.large"  
                    ]  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "Resource": [  
                "arn:aws:ec2::::image/ami-*",  
                "arn:aws:ec2::*:account:network-interface/*",  
                "arn:aws:ec2::*:account:instance/*",  
                "arn:aws:ec2::*:account:subnet/*",  
                "arn:aws:ec2::*:account:volume/*",  
                "arn:aws:ec2::*:account:key-pair/*",  
                "arn:aws:ec2::*:account:security-group/*"  
            ]  
        }  
    ]  
}
```

## 起動テンプレート

次の例で、ユーザーはインスタンスを起動できますが、特定の起動テンプレートを使用する場合に限りま  
す (lt-09477bcd97b0d310e)。ユーザーは、`RunInstances` アクションでパラメータを指定すること  
で、起動テンプレートのパラメータを上書きできます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:RunInstances",  
            "TemplateReference": "lt-09477bcd97b0d310e"  
        }  
    ]  
}
```

```
"Resource": "*",
"Condition": {
    "ArnLike": {
        "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/
lt-09477bcd97b0d310e"
    }
}
]
```

この例で、ユーザーは、起動テンプレートを使用する場合に限りインスタンスを起動できます。ポリシーでは `ec2:IsLaunchTemplateResource` 条件キーを使用して、ユーザーが起動テンプレート内の既存の ARN を上書きできないようにします。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
                },
                "Bool": {
                    "ec2:IsLaunchTemplateResource": "true"
                }
            }
        }
    ]
}
```

次のサンプルポリシーによりユーザーはインスタンスを起動できますが、起動テンプレートを使用する場合に限ります。ユーザーは、リクエストでサブネットおよびネットワークインターフェイスのパラメータを上書きすることはできません。これらのパラメータは、起動テンプレートでのみ指定できます。ステートメントの最初の部分は、`NotResource` 要素を使用して、サブネットやネットワークインターフェイスを除くその他のすべてのリソースを許可します。ステートメントの 2 番目の部分は、サブネットおよびネットワークインターフェイスのリソースを許可しますが、これは起動テンプレートから取得された場合に限ります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "NotResource": [
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:network-interface/*"
            ],
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:region:account:subnet/*",
                "arn:aws:ec2:region:account:network-interface/*"
            ],
            "Condition": {

```

```
        "ArnLike": {
            "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
        },
        "Bool": {
            "ec2:IsLaunchTemplateResource": "true"
        }
    }
]
```

次の例では、起動テンプレートを使用していて、また起動テンプレートにタグがある場合に限り、ユーザーはインスタンスを起動できるようになります Purpose=Webservers。ユーザーは、RunInstances アクションで起動テンプレートパラメータを上書きすることはできません。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "NotResource": "arn:aws:ec2:region:account:launch-template/*",
            "Condition": {
                "ArnLike": {
                    "ec2:LaunchTemplate": "arn:aws:ec2:region:account:launch-template/*"
                },
                "Bool": {
                    "ec2:IsLaunchTemplateResource": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:region:account:launch-template/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Purpose": "Webservers"
                }
            }
        }
    ]
}
```

## 例: リザーブドインスタンスの使用

次のポリシーでは、アカウントで リザーブドインスタンス を表示、変更、購入するアクセス許可をユーザーに与えます。

個別の リザーブドインスタンス にリソースレベルのアクセス許可を設定することはできません。このポリシーは、ユーザーがアカウントのすべての リザーブドインスタンス にアクセスできることを意味します。

Resource 要素は \* ワイルドカードを使用して、ユーザーがそのアクションにすべてのリソースを指定できることを示しています。この場合、アカウントのすべての リザーブドインスタンス をリストして変更できます。ユーザーは、アカウント認証情報を使用して リザーブドインスタンス を購入することもできます。また、API アクションがリソースレベルのアクセス許可をサポートしていない場合も、\* ワイルドカードが必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",

```

```
    "Action": [
        "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
        "ec2:PurchaseReservedInstancesOffering", "ec2:DescribeAvailabilityZones",
        "ec2:DescribeReservedInstancesOfferings"
    ],
    "Resource": "*"
}
]
```

次のコードでは、アカウント内の リザーブドインスタンス を表示および変更できるようにユーザーに許可しています。新しい リザーブドインスタンス の購入は、許可していません。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*"
        }
    ]
}
```

### 例: リソースのタグ付け

次のポリシーでは、タグにキー `CreateTags` および値 `environment` が含まれている場合のみ、ユーザーが `production` アクションを使用してインスタンスにタグを適用できます。`ForAllValues` 修飾子は、リクエストでキー `aws:TagKeys` のみが許可される（他のタグが許可されない）ことを示すため、`environment` 条件キーとともに使用されます。ユーザーは、他のリソースタイプをタグ付けすることができます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:region:account:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/environment": "production"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "environment"
                    ]
                }
            }
        }
    ]
}
```

次のポリシーでは、ユーザーは `owner` のキーと IAM ユーザー名の値を使用したタグがすでに適用されているタグ付け可能なリソースにタグ付けできます。加えて、ユーザーはリクエストで `anycompany:environment-type` のキーと値 `test` または `prod` を持つタグを指定する必要があります。ユーザーは、リクエストで追加のタグを指定できます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:/*/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/anycompany:environment-type": ["test", "prod"],  
                    "ec2:ResourceTag/owner": "${aws:username}"  
                }  
            }  
        }  
    ]  
}
```

ユーザーがリソースの特定のタグを指定できるようにする IAM ポリシーを作成できます。たとえば、次のポリシーでは、リクエストで指定されたタグキーが `environment` または `cost-center` の場合、ユーザーがボリュームのタグを削除できます。タグにはどの値でも指定できますが、指定されたキーのいずれかにタグキーが一致する必要があります。

Note

リソースを削除すると、リソースに関連付けられているすべてのタグも削除されます。タグ付きのリソースを削除する場合、ユーザーは `ec2:DeleteTags` アクションを使用するためのアクセス許可は必要ありません。削除アクションを実行するためのアクセス許可のみが必要です。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:DeleteTags",  
            "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",  
            "Condition": {  
                "ForAllValues:StringEquals": {  
                    "aws:TagKeys": ["environment", "cost-center"]  
                }  
            }  
        }  
    ]  
}
```

このポリシーでは、リソースが `owner` のキーと IAM ユーザー名の値すでにタグ付けされている場合のみ、ユーザーが任意のリソースで `environment=prod` タグのみ削除できます。ユーザーは、リソースの他のタグを削除することはできません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DeleteTags"  
            ],  
            "Resource": "arn:aws:ec2:region:account:/*/*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:TagKeys": ["environment"]  
                }  
            }  
        }  
    ]  
}
```

```
        "aws:RequestTag/environment": "prod",
        "ec2:ResourceTag/owner": "${aws:username}"
    },
    "ForAllValues:StringEquals": {
        "aws:TagKeys": [ "environment" ]
    }
}
]
```

## 例: IAM ロールの使用

次のポリシーでは、`department=test` タグを持つインスタンスに対して IAM ロールのアタッチ、置換、デタッチを行うことをユーザーに許可します。IAM ロールの置換またはデタッチには関連 ID が必要であるため、ポリシーでは `ec2:DescribeIamInstanceProfileAssociations` アクションを使用するアクセス許可もユーザーに付与します。

IAM ユーザーは、ロールをインスタンスに渡すために `iam:PassRole` アクションを使用するためのアクセス許可が必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AssociateIamInstanceProfile",
                "ec2:ReplaceIamInstanceProfileAssociation",
                "ec2:DisassociateIamInstanceProfile"
            ],
            "Resource": "arn:aws:ec2:region:account:instance/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeIamInstanceProfileAssociations",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*"
        }
    ]
}
```

次のポリシーでは、どのインスタンスに対しても IAM ロールのアタッチまたは置換を行うことをユーザーに許可します。ユーザーは、`TestRole-` で始まる名前の IAM ロールのみアタッチまたは置換できます。IAM アクションでは、インスタンスプロファイルではなく `iam:PassRole` ロールの名前を指定します（両方の名前が異なる場合）。詳細については、「[インスタンスプロファイル \(p. 889\)](#)」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*"
        }
    ]
}
```

```
"Action": [
    "ec2:AssociateIamInstanceProfile",
    "ec2:ReplaceIamInstanceProfileAssociation"
],
"Resource": "*"
},
{
"Effect": "Allow",
"Action": "ec2:DescribeIamInstanceProfileAssociations",
"Resource": "*"
},
{
"Effect": "Allow",
"Action": "iam:PassRole",
"Resource": "arn:aws:iam::account:role/TestRole-*"
}
]
```

## 例: ルートテーブルを操作する

次のポリシーでは、VPC (vpc-ec43eb89) のみに関連付けられているルートテーブルのルートの追加、削除、置換を行うことができます。ec2:Vpc 条件キーの VPC を指定するには、VPC の完全な ARN を指定する必要があります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2>DeleteRoute",
                "ec2>CreateRoute",
                "ec2:ReplaceRoute"
            ],
            "Resource": [
                "arn:aws:ec2:region:account:route-table/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-ec43eb89"
                }
            }
        }
    ]
}
```

## 例: 特定のインスタンスが他の AWS サービスでリソースを表示できるようにする

次に示すのは、IAM ロールにアタッチできるポリシーの例です。ポリシーにより、インスタンスは AWS サービスのさまざまなリソースを表示できるようになります。ec2:SourceInstanceARN 条件キーを使用して、リクエストの実行元インスタンスが i-093452212644b0dd6 インスタンスになるように指定します。同じ IAM ロールが別のインスタンスと関連付けられている場合、他のインスタンスはこれらのどのアクションも実行できません。

ec2:SourceInstanceARN は AWS 全体を対象とする条件キーであるため、Amazon EC2 だけではなく他のサービスアクションにも使用できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
{  
    "Effect": "Allow",  
    "Action": [  
        "ec2:DescribeVolumes",  
        "s3>ListAllMyBuckets",  
        "dynamodb>ListTables",  
        "rds:DescribeDBInstances"  
    ],  
    "Resource": [  
        "*"  
    ],  
    "Condition": {  
        "ArnEquals": {  
            "ec2:SourceInstanceARN": "arn:aws:ec2:region:account:instance/  
i-093452212644b0dd6"  
        }  
    }  
}
```

## 例: 起動テンプレートの使用

次のポリシーでは、ユーザーは起動テンプレートのバージョンを作成して起動テンプレートを変更することができます。ただし、特定の起動テンプレートに限られます (lt-09477bcd97b0d3abc)。ユーザーは、他の起動テンプレートを使用することはできません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ec2>CreateLaunchTemplateVersion",  
                "ec2:ModifyLaunchTemplate"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:ec2:region:account:launch-template/lt-09477bcd97b0d3abc"  
        }  
    ]  
}
```

次のポリシーでは、ユーザーは任意の起動テンプレートと起動テンプレートのバージョンを削除できます。ただし、起動テンプレートに Purpose=Testing のタグがある場合に限ります。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": [  
                "ec2>DeleteLaunchTemplate",  
                "ec2>DeleteLaunchTemplateVersions"  
            ],  
            "Effect": "Allow",  
            "Resource": "arn:aws:ec2:region:account:launch-template/*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/Purpose": "Testing"  
                }  
            }  
        }  
    ]  
}
```

## インスタンスマタデータの使用

以下のポリシーでは、インスタンスマタデータサービスバージョン 2 (IMDSv2) を使用して、ユーザーがインスタンスマタデータ ([p. 593](#))のみを取得できるようにします。以下の 4 つのポリシーは、4 つのステートメントを使用する 1 つのポリシーに結合できます。1 つのポリシーとして結合すると、このポリシーをサービスコントロールポリシー (SCP) として使用できます。これは、既存の IAM ポリシーに適用する拒否ポリシーとして（既存のアクセス許可を削除して制限するために）使用したり、アカウント、部門単位 (OU)、組織全体にグローバルに適用する SCP として使用したりすることもできます。

### Note

以下の RunInstances メタデータオプションポリシーは、RunInstances を使用してインスタンスを起動するアクセス許可をプリンシパルに付与するポリシーと組み合わせて使用する必要があります。プリンシパルに RunInstances アクセス許可もない場合、インスタンスを起動することはできません。詳細については、「[インスタンスの使用 \(p. 851\)](#)」と「[インスタンスを起動する \(RunInstances\) \(p. 862\)](#)」のポリシーを参照してください。

### Important

Auto Scaling グループを使用し、すべての新しいインスタンスで IMDSv2 の使用を要求する必要がある場合は、Auto Scaling グループで起動テンプレートを使用する必要があります。

Auto Scaling グループが起動テンプレートを使用する場合、新しい Auto Scaling グループが作成されるときに IAM プリンシパルの `ec2:RunInstances` アクセス許可がチェックされます。また、既存の Auto Scaling グループが更新され、新しい起動テンプレートまたは新しいバージョンの起動テンプレートが使用される場合にもチェックされます。

RunInstances の IAM プリンシパルでの IMDSv1 の使用に関する制限は、起動テンプレートを使用している Auto Scaling グループが作成または更新された場合にのみチェックされます。Latest または Default 起動テンプレートを使用するように設定された Auto Scaling グループでは、起動テンプレートの新しいバージョンが作成されたときにアクセス許可はチェックされません。アクセス許可をチェックするには、特定のバージョンの起動テンプレートを使用するように Auto Scaling グループを設定する必要があります。

Auto Scaling グループによって起動されるインスタンスで IMDSv2 の使用を強制するには、以下の追加ステップが必要です。

1. 作成された新しいプリンシパルのサービスコントロールポリシー (SCP) または IAM アクセス許可の境界を使用して、組織内のすべてのアカウントの起動設定の使用を無効にします。Auto Scaling グループアクセス許可を持つ既存の IAM プリンシパルの場合、関連するポリシーをこの条件キーで更新します。起動設定の使用を無効にするには、値が `null` として指定された `"autoscaling:LaunchConfigurationName"` 条件キーを使用して、関連する SCP、アクセス許可の境界、または IAM ポリシーを作成または変更します。
2. 新しい起動テンプレートの場合は、起動テンプレートでインスタンスマタデータオプションを設定します。既存の起動テンプレートの場合は、新しいバージョンの起動テンプレートを作成し、新しいバージョンでインスタンスマタデータオプションを設定します。
3. 起動テンプレートを使用するアクセス許可を任意のプリンシパルに付与するポリシーで、`"autoscaling:LaunchTemplateVersionSpecified": "true"` を指定して `$latest` と `$default` の関連付けを制限します。使用を特定のバージョンの起動テンプレートに制限することで、インスタンスマタデータオプションが設定されているバージョンを使用して新しいインスタンスを確実に起動できます。詳細については、Amazon EC2 Auto Scaling API リファレンス (具体的には `version` パラメータ) の「[LaunchTemplateSpecification](#)」を参照してください。
4. 起動設定を使用する Auto Scaling グループの場合、起動設定を起動テンプレートに置き換えます。詳細については、Amazon EC2 Auto Scaling ユーザーガイドの「[起動設定を起動テンプレートに置き換える](#)」を参照してください。
5. 起動テンプレートを使用する Auto Scaling グループの場合、インスタンスマタデータオプションが設定された新しい起動テンプレートを使用するか、インスタンスマタデータオプションが設定された現在の起動テンプレートの新しいバージョンを使用します。詳細については、AWS CLI Command Reference の「[update-auto-scaling-group](#)」を参照してください。

例

- IMDSv2 の使用を要求する (p. 879)
- ホップ制限の最大値の指定 (p. 879)
- インスタンスマターダオプションを変更できるユーザーの制限 (p. 880)
- IMDSv2 からロール認証情報を取得することを要求する (p. 880)

## IMDSv2 の使用を要求する

次のポリシーでは、インスタンスが IMDSv2 の使用を要求するようにオプトインされていない限り ("ec2:MetadataHttpTokens": "required" で指定)、RunInstances API を呼び出せないように指定します。インスタンスが IMDSv2 を要求するように指定しないと、RunInstances API を呼び出したときに UnauthorizedOperation エラーが発生します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "RequireImdsV2",  
            "Effect": "Deny",  
            "Action": "ec2:RunInstances",  
            "Resource": "arn:aws:ec2:*:instance/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "ec2:MetadataHttpTokens": "required"  
                }  
            }  
        }  
    ]  
}
```

## ホップ制限の最大値の指定

次のポリシーでは、ホップ制限を指定しない限り、RunInstances API を呼び出せないように指定します。また、ホップ制限を 3 以下にするように指定します。これを指定しないと、RunInstances API を呼び出したときに UnauthorizedOperation エラーが発生します。

Note

次のポリシーと前のポリシーを SCP 経由でアカウントに適用した場合、EC2 コンソールは MetadataHttpTokens パラメータと MetadataHttpPutResponseHopLimit パラメータをまだサポートしていないため、EC2 コンソールを使用してインスタンスを起動することはできません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "MaxImdsHopLimit",  
            "Effect": "Deny",  
            "Action": "ec2:RunInstances",  
            "Resource": "arn:aws:ec2:*:instance/*",  
            "Condition": {  
                "NumericGreaterThan": {  
                    "ec2:MetadataHttpPutResponseHopLimit": "3"  
                }  
            }  
        }  
    ]  
}
```

## インスタンスマタデータオプションを変更できるユーザーの制限

次のポリシーでは、一般的な管理者がインスタンスマタデータオプションを変更する機能を削除し、ロール ec2-imds-admins を持つユーザーのみに変更を行うことを許可します。ec2-imds-admins ロール以外のプリンシパルが ModifyInstanceMetadataOptions API を呼び出そうとすると、UnauthorizedOperation エラーが発生します。このステートメントは、ModifyInstanceMetadataOptions API の使用を制御するために使用できます。現在、ModifyInstanceMetadataOptions API 用の詳細なアクセスコントロール（条件）はありません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AllowOnlyImdsAdminsToModifySettings",  
            "Effect": "Deny",  
            "Action": "ec2:ModifyInstanceMetadataOptions",  
            "Resource": "*",  
            "Condition": {  
                "StringNotLike": {  
                    "aws:PrincipalARN": "arn:aws:iam::*:role/ec2-imds-admins"  
                }  
            }  
        }  
    ]  
}
```

## IMDSv2 からロール認証情報を取得することを要求する

次のポリシーでは、このポリシーを適用したロールを EC2 サービスが引き受けて、結果の認証情報をリクエストの署名に使用する場合は、IMDSv2 から取得した EC2 ロールの認証情報を使用してリクエストに署名する必要があることを指定します。それ以外の場合は、すべての API コールで UnauthorizedOperation エラーが発生します。このステートメント/ポリシーは、リクエストが EC2 ロールの認証情報によって署名されていない場合は効果がないため、一般的に適用できます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "RequireAllEc2RolesToUseV2",  
            "Effect": "Deny",  
            "Action": "*",  
            "Resource": "*",  
            "Condition": {  
                "NumericLessThan": {  
                    "ec2:RoleDelivery": "2.0"  
                }  
            }  
        }  
    ]  
}
```

## Amazon EC2 コンソールで機能するサンプル ポリシー

IAM ポリシーを使用して、Amazon EC2 コンソールで特定のリソースを表示、および操作するアクセス許可をユーザーに付与することができます。上記のセクションのサンプルポリシーを使用することはできますが、これらは AWS CLI または AWS SDK で作成されたリクエスト向けに設計されています。コンソールではこの機能を実行するために追加の API アクションを使用するので、これらのポリシーは正常に動作しない可能性があります。たとえば、DescribeVolumes API アクションのみを使用するアクセス許可を持つユーザーがコンソールでボリュームを表示しようとすると、エラーが発生します。このセクションでは、コンソールの特定の部分をユーザーが操作できるようになるポリシーを説明します。

Tip

コンソールでタスクを実行するために必要な API アクションを探すには、AWS CloudTrail などのサービスを使用できます。詳細については、『[AWS CloudTrail User Guide](#)』を参照してください。ポリシーにより特定のリソースを作成または変更するアクセス許可が付与されない場合、コンソールではエンコードされた診断情報のメッセージが表示されます。AWS STS の [DecodeAuthorizationMessage](#) API アクション、または AWS CLI の [decode-authorization-message](#) コマンドを使用してメッセージをデコードできます。

例

- [例: 読み取り専用アクセス \(p. 881\)](#)
- [例: EC2 起動ウィザードを使用する \(p. 882\)](#)
- [例: ボリュームを操作する \(p. 884\)](#)
- [例: セキュリティグループを操作する \(p. 885\)](#)
- [例: Elastic IP アドレスの操作 \(p. 887\)](#)
- [例: リザーブドインスタンスを使用する \(p. 888\)](#)

Amazon EC2 向けのポリシー作成の詳細については、AWS セキュリティブログの投稿「[Granting Users Permission to Work in the Amazon EC2 Console](#)」を参照してください。

### 例: 読み取り専用アクセス

ユーザーが Amazon EC2 コンソールですべてのリソースを表示できるようにするには、次の例と同じポリシーを使用します: [例: 読み取り専用アクセス \(p. 850\)](#)。別のステートメントによりユーザーにアクセス許可が与えられない限り、ユーザーはリソースのアクションを実行したり新しいリソースを作成できません。

#### インスタンス、AMI、スナップショットを表示する

代わりに、リソースのサブセットへの読み取り専用アクセスを提供できます。これを行うには、ec2:Describe API アクションの \* (ワイルドカード) を各リソースの固有の ec2:Describe アクションに置き換えます。次のポリシーによりユーザーは Amazon EC2 コンソールですべてのインスタンス、AMI、およびスナップショットを表示できます。ec2:DescribeTags アクションにより、ユーザーはパブリック AMI を表示できます。コンソールでタグ付け情報にパブリック AMI を表示させる必要がありますが、ユーザーがプライベート AMI だけを表示できるようにするには、このアクションを削除できます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeInstances", "ec2:DescribeImages",  
            "ec2:DescribeTags", "ec2:DescribeSnapshots"  
        ],  
        "Resource": "*"  
    }]  
}
```

Note

Amazon EC2 ec2:Describe\* API アクションは、リソースレベルのアクセス許可をサポートしていません。そのため、ユーザーがコンソールで表示できる個人のリソースを制御できません。したがって、上記のステートメントの Resource エレメントには、\* (ワイルドカード) が必要です。どの Amazon EC2 API アクションでどの ARN を使用できるかの詳細については、IAM ユーザーガイドの「[Amazon EC2 のアクション、リソース、および条件キー](#)」を参照してください。

## インスタンスと CloudWatch メトリクスを表示する

以下のポリシーは、ユーザーに対して Amazon EC2 コンソールでのインスタンスの表示、[Instances] ページの [Monitoring] タブでの CloudWatch アラームおよびメトリクスの表示を許可します。Amazon EC2 コンソールでは、CloudWatch API がアラームとメトリクスの表示に使用されるため、`cloudwatch:DescribeAlarms` および `cloudwatch:GetMetricStatistics` アクションを使用するアクセス権限をユーザーに付与する必要があります。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeInstances",  
                "cloudwatch:DescribeAlarms",  
                "cloudwatch:GetMetricStatistics"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

## 例: EC2 起動ウィザードを使用する

Amazon EC2 起動ウィザードは、インスタンスを設定し、起動するためのオプションを提供する一連の画面です。ユーザーがウィザードのオプションを操作できるように、API アクションを使用するアクセス許可をポリシーに含める必要があります。ポリシーにそれらのアクションを使用するアクセス許可が含まれない場合、ウィザードの一部の項目は適切にロードされず、ユーザーは起動を完了できません。

### 起動ウィザードへの基本的なアクセスを許可する

起動を正常に完了させるには、ユーザーに `ec2:RunInstances` API アクションを使用するアクセス許可を付与し、少なくとも以下の API アクションを使用できるようにする必要があります。

- `ec2:DescribeImages`: AMI を表示して選択します。
- `ec2:DescribeVpcs`: 使用できるネットワークオプションを表示します。
- `ec2:DescribeSubnets`: 選択した VPC のすべての使用可能なサブネットを表示します。
- `ec2:DescribeSecurityGroups` または `ec2>CreateSecurityGroup`: 既存のセキュリティグループを表示および選択する、または新しいセキュリティグループを作成します。
- `ec2:DescribeKeyPairs` または `ec2>CreateKeyPair`: 既存のキーペアを選択する、または新しいキーペアを作成します。
- `ec2:AuthorizeSecurityGroupIngress`: インバウンドルールを追加します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeInstances",  
                "ec2:DescribeImages",  
                "ec2:DescribeKeyPairs",  
                "ec2:DescribeVpcs",  
                "ec2:DescribeSubnets",  
                "ec2:DescribeSecurityGroups",  
                "ec2>CreateSecurityGroup",  
                "ec2:AuthorizeSecurityGroupIngress",  
                "ec2>CreateKeyPair"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

```
        "Resource": "*"
    },
{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*"
}
]
```

ポリシーに次のような API アクションを追加して、ユーザーに追加のオプションを提供できます。

- `ec2:DescribeAvailabilityZones`: 特定のアベイラビリティゾーンを選択します。
- `ec2:DescribeNetworkInterfaces`: 選択したサブネットの既存のネットワークインターフェイスを表示および選択します。
- VPC セキュリティグループにアウトバウンドルールを追加するには、ユーザーに `ec2:AuthorizeSecurityGroupEgress` API アクションを使用するアクセス許可を付与する必要があります。既存のルールを変更または削除するには、ユーザーに関連する `ec2:RevokeSecurityGroup*` API アクションを使用するアクセス許可を付与する必要があります。
- `ec2:CreateTags`: `RunInstances` により作成されたリソースにタグ付けする場合に使用します。詳細については、「[リソース作成時にタグ付けするアクセス許可の付与 \(p. 847\)](#)」を参照してください。ユーザーにこのアクションを使用するアクセス権限がなく、起動ウィザードのタグ付けページでタグを適用しようとした場合、起動に失敗します。

#### Important

ユーザーに `ec2:CreateTags` アクションを使用するアクセス許可を付与するには注意が必要です。これにより、`ec2:ResourceTag` 条件キーを使用する能力が限定され、他のリソースの使用が制限されます。ユーザーは、リソースのタグを変更してその制限を回避できます。

現在、Amazon EC2 `Describe*` API アクションは、リソースレベルのアクセス許可をサポートしていません。そのため、ユーザーが起動ウィザードで表示できる個人のリソースを制限することはできません。ただし、`ec2:RunInstances` API アクションにリソースレベルのアクセス許可を適用して、ユーザーがインスタンスの起動に使用できるリソースを制限できます。ユーザーが使用する権限がないオプションを選択すると、起動は失敗します。

#### 特定のインスタンスタイプ、サブネット、リージョンへのアクセスを制限する

次のポリシーにより、ユーザーは Amazon が所有する AMI を使用して `t2.micro` インスタンスを特定のサブネット (`subnet-1a2b3c4d`) でのみ起動することができます。ユーザーは `sa-east-1` リージョンでのみ起動できます。ユーザーが異なるリージョンを選択するか、起動ウィザードで異なるインスタンスタイプ、AMI、サブネットを選択すると、起動は失敗します。

最初のステートメントでは、上記の例で説明したように、起動ウィザードでオプションを表示するアクセス許可または新しいオプションを作成するアクセス許可がユーザーに付与されます。2番目のステートメントでは、`ec2:RunInstances` アクションでネットワークインターフェイス、ボリューム、キーペア、セキュリティグループ、サブネットリソースを使用するアクセス許可が付与されます。これは、ユーザーが VPC でインスタンスを起動するために必要です。`ec2:RunInstances` アクションの使用方法の詳細については、「[インスタンスを起動する \(RunInstances\) \(p. 862\)](#)」を参照してください。3番目と4番目のステートメントでは、インスタンスと AMI リソースを使用するアクセス許可がそれぞれ付与されますが、インスタンスが `t2.micro` インスタンスの場合のみ、および AMI が Amazon によって所有されている場合のみ付与されます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:sa-east-1:123456789012:subnet-1a2b3c4d"
            ]
        }
    ]
}
```

```
"ec2:DescribeInstances", "ec2:DescribeImages",
"ec2:DescribeKeyPairs", "ec2>CreateKeyValuePair", "ec2:DescribeVpcs",
"ec2:DescribeSubnets", "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup",
"ec2AuthorizeSecurityGroupIngress"
],
"Resource": "*"
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1:111122223333:network-interface/*",
    "arn:aws:ec2:sa-east-1:111122223333:volume/*",
    "arn:aws:ec2:sa-east-1:111122223333:key-pair/*",
    "arn:aws:ec2:sa-east-1:111122223333:security-group/*",
    "arn:aws:ec2:sa-east-1:111122223333:subnet/subnet-1a2b3c4d"
  ]
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1:111122223333:instance/*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:InstanceType": "t2.micro"
    }
  }
},
{
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:sa-east-1::image/ami-*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:Owner": "amazon"
    }
  }
}
]
```

## 例: ボリュームを操作する

次のポリシーは、ボリュームを表示して作成し、特定のインスタンスにボリュームをアタッチ、およびデタッチするアクセス許可をユーザーに付与します。

ユーザーは、"purpose=test" というタグを含むインスタンスに対してどのボリュームもアタッチできます。同様に、それらのインスタンスからボリュームをデタッチすることもできます。Amazon EC2 コンソールを使用してボリュームをアタッチするには、ユーザーに ec2:DescribeInstances アクションを使用するアクセス許可があると、[Attach Volume] ダイアログボックスのあらかじめ用意されたリストからインスタンスを選択できるため、役立ちます。ただし、これにより、コンソールの [Instances] ページでもすべてのインスタンスが表示されるため、このアクションを省略することもできます。

最初のステートメントでは、ボリュームを作成するときにユーザーがアベイラビリティゾーンを選択できるようにするため、ec2:DescribeAvailabilityZones アクションが必要です。

ユーザーは、作成したボリュームをタグ付けできません(ボリュームの作成中も作成後も)。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeVolumes",
            "ec2:DescribeAvailabilityZones",
            "ec2>CreateVolume",
            "ec2:DescribeInstances"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:AttachVolume",
            "ec2:DetachVolume"
        ],
        "Resource": "arn:aws:ec2:region:111122223333:instance/*",
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/purpose": "test"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:AttachVolume",
            "ec2:DetachVolume"
        ],
        "Resource": "arn:aws:ec2:region:111122223333:volume/*"
    }
]
```

## 例: セキュリティグループを操作する

セキュリティグループを表示し、ルールを追加/削除する

次のポリシーは、Amazon EC2 コンソールでセキュリティグループを表示し、タグ `Department=Test` を含む既存のセキュリティグループに対してインバウンドおよびアウトバウンドのルールを追加および削除するアクセス許可をユーザーに付与します。

最初のステートメントの `ec2:DescribeTags` アクションにより、ユーザーはコンソールでタグを表示できます。これにより、ユーザーは変更できるセキュリティグループをより簡単に識別できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroups", "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupIngress", "ec2:RevokeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress", "ec2:RevokeSecurityGroupEgress"
            ],
            "Resource": [
                "arn:aws:ec2:region:111122223333:security-group/*"
            ]
        }
    ]
}
```

```
],
  "Condition": {
    "StringEquals": {
      "ec2:ResourceTag/Department": "Test"
    }
  }
}
]
```

#### [Create Security Group] ダイアログボックスを使用する

ユーザーが Amazon EC2 コンソールの [Create Security Group] ダイアログボックスを使用して作業できるようにするポリシーを作成できます。このダイアログボックスを使用するには、ユーザーに少なくとも以下の API アクションを使用するアクセス許可を付与する必要があります。

- `ec2:CreateSecurityGroup`: 新しいセキュリティグループを作成するには
- `ec2:DescribeVpcs`: [VPC] リストに既存の VPC のリストを表示します。

これらのアクセス許可で、ユーザーは新しいセキュリティグループを正常に作成できますが、ルールを追加することはできません。[Create Security Group] ダイアログボックスでルールを操作するには、ポリシーに次の API アクションを追加します。

- `ec2:AuthorizeSecurityGroupIngress`: インバウンドルールを追加します。
- `ec2:AuthorizeSecurityGroupEgress`: VPC セキュリティグループにアウトバウンドルールを追加します。
- `ec2:RevokeSecurityGroupIngress`: 既存のインバウンドルールを変更または削除します。これは、ユーザーがコンソールで [Copy to new] 機能を使用できるようにするために役に立ちます。この機能により、[Create Security Group] ダイアログボックスが開き、選択したセキュリティグループと同じルールが追加されます。
- `ec2:RevokeSecurityGroupEgress`: VPC セキュリティグループのアウトバウンドルールを変更または削除します。これは、すべてのアウトバウンドトラフィックを許可するデフォルトのアウトバウンドルールを変更または削除する場合に役に立ちます。
- `ec2>DeleteSecurityGroup`: 無効なルールを保存できないときに対応します。コンソールでは、最初にセキュリティグループを作成し、次に指定されたルールを追加します。ルールが無効である場合、アクションは失敗し、コンソールによってセキュリティグループの削除が試行されます。引き続き、[Create Security Group] ダイアログボックスが利用できるため、ユーザーは無効なルールを修正してセキュリティグループを再作成できます。この API アクションは必須ではありませんが、ユーザーにこのアクションを使用するアクセス許可が付与されておらず、無効なルールを持つセキュリティグループを作成しようとすると、ルールのないセキュリティグループが作成され、後でルールを追加することが必要になります。

現在、`ec2:CreateSecurityGroup` API アクションは、リソースのレベルのアクセス許可をサポートしていません。ただし、`ec2:AuthorizeSecurityGroupIngress` および `ec2:AuthorizeSecurityGroupEgress` アクションにリソースレベルのアクセス許可を適用してルールを作成する方法を制御できます。

次のポリシーは、[Create Security Group] ダイアログボックスを使用し、特定の VPC (`vpc-1a2b3c4d`) に関連付けられたセキュリティグループに対してインバウンドおよびアウトバウンドのルールを作成するアクセス許可をユーザーに付与します。ユーザーは EC2-Classic または別の VPC のセキュリティグループを作成できますが、ルールを追加することはできません。同様に、ユーザーは VPC `vpc-1a2b3c4d` に関連付けられていない既存のセキュリティグループにルールを追加することもできません。ユーザーには、コンソールですべてのセキュリティグループを表示するアクセス許可も付与されます。これにより、ユーザーはインバウンドルールを追加するセキュリティグループをより簡単に識別できるようになります。このポリシーは、ユーザーに VPC `vpc-1a2b3c4d` に関連付けられたセキュリティグループを削除するアクセス許可も付与します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup", "ec2:DescribeVpcs"  
        ],  
        "Resource": "*"  
    },  
    {  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",  
            "ec2:AuthorizeSecurityGroupEgress"  
        ],  
        "Resource": "arn:aws:ec2:region:111122223333:security-group/*",  
        "Condition": {  
            "ArnEquals": {  
                "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"  
            }  
        }  
    }  
]
```

## 例: Elastic IP アドレスの操作

Amazon EC2 コンソールで Elastic IP アドレスを確認することをユーザーに許可するには、`ec2:DescribeAddresses` アクションを使用するためのアクセス許可をユーザーに付与します。

Elastic IP アドレスの使用をユーザーに許可する場合は、ポリシーに次のアクションを追加できます。

- `ec2:AllocateAddress`: Elastic IP アドレスを割り当てます。
- `ec2:ReleaseAddress`: Elastic IP アドレスを解放します。
- `ec2:AssociateAddress`: Elastic IP アドレスをインスタンスまたはネットワークインターフェイスに関連付けます。
- `ec2:DescribeNetworkInterfaces` と `ec2:DescribeInstances`: [Associate address] で使用します。この画面には、Elastic IP アドレスを関連付けることができるインスタンスまたはネットワークインターフェイスが表示されます。
- `ec2:DisassociateAddress`: Elastic IP アドレスとインスタンスまたはネットワークインターフェイスの関連付けを解除します。

次のポリシーでは、Elastic IP アドレスの表示、割り当て、インスタンスとの関連付けを行うことができます。ユーザーは Elastic IP アドレスとネットワークインターフェイスの関連付け、Elastic IP アドレスの関連付けの解除、または Elastic IP アドレスの解放を行うことはできません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeAddresses",  
                "ec2:AllocateAddress",  
                "ec2:DescribeInstances",  
                "ec2:AssociateAddress"  
            ],  
            "Resource": "*"  
        }  
    ]
```

```
    ]  
}
```

## 例: リザーブドインスタンスを使用する

以下のポリシーを IAM ユーザーにアタッチすることができます。これにより、アカウントのリザーブドインスタンスの表示と変更、および AWS マネジメントコンソールでの新しいリザーブドインスタンス購入のアクセス許可がユーザーに付与されます。

このポリシーにより、ユーザーはアカウントのすべてのリザーブドインスタンスと、オンデマンドインスタンスを表示できます。個別のリザーブドインスタンスにリソースレベルのアクセス許可を設定することはできません。

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": [  
            "ec2:DescribeReservedInstances", "ec2:ModifyReservedInstances",  
            "ec2:PurchaseReservedInstancesOffering", "ec2:DescribeInstances",  
            "ec2:DescribeAvailabilityZones", "ec2:DescribeReservedInstancesOfferings"  
        ],  
        "Resource": "*"  
    }]  
}
```

`ec2:DescribeAvailabilityZones` アクションは、リザーブドインスタンスを購入できるアベイラビリティーゾーンに関する情報を Amazon EC2 コンソールで表示できるようにするために必要です。`ec2:DescribeInstances` アクションは必須ではありませんが、このアクションにより、ユーザーがアカウントのインスタンスを表示し、正しい仕様に合わせて予約を購入できるようになります。

`ec2:DescribeInstances` を削除するなど、API アクションを調整してユーザーアクセスを制限できます。`ec2:DescribeAvailabilityZones` はユーザーが読み取り専用アクセスを持っていることを意味します。

## Amazon EC2 の IAM ロール

アプリケーションは AWS 認証情報で API リクエストに署名する必要があります。したがって、アプリケーション開発者である場合、EC2 インスタンスで実行するアプリケーションの認証情報を管理する戦略が必要です。たとえば、AWS 認証情報をインスタンスに安全に配布することで、他のユーザーから認証情報を保護しながら、それらのインスタンスのアプリケーションで認証情報を使用してリクエストに署名できます。ただし、各インスタンスに認証情報を安全に配布することは難しく、特に AWS が代理で作成するスポットインスタンスや Auto Scaling グループのインスタンスなどではそれが顕著です。また、AWS 認証情報を循環させる場合、各インスタンスの認証情報を更新できる必要があります。

アプリケーションが使用するセキュリティ認証情報をお客様が管理する必要なく、アプリケーションがインスタンスから API リクエストを安全に作成できるように、IAM ロールをデザインしました。AWS 認証情報を作成および配布する代わりに、以下の方法で、IAM ロールを使用して API リクエストを作成するアクセス許可を委任できます。

1. IAM ロールを作成します。
2. ロールを行うアカウントまたは AWS サービスを定義する
3. ロールを受けた後で、アプリケーションで使用できる API アクションおよびリソースを定義します。
4. インスタンスの起動時にロールを指定するか、既存のインスタンスにロールをアタッチします。
5. アプリケーションで一時的な認証情報のセットを取得して使用します。

たとえば、IAM ロールを使用し、Amazon S3 のバケットを使用する必要のあるインスタンスで実行中のアプリケーションに、アクセス許可を与えることができます。JSON 形式のポリシーを作成することにより、IAM ロールのアクセス許可を指定できます。これらのポリシーは、IAM ユーザー用に作成するポリシーに類似しています。ロールを変更すると、その変更はすべてのインスタンスに反映されます。

IAM ロールを作成するとき、アプリケーションが必要とする特定の API コールへのアクセスを制限する最小権限の IAM ポリシーを関連付けます。

複数の IAM ロールを 1 つのインスタンスにアタッチすることはできませんが、1 つの IAM ロールを複数のインスタンスにアタッチすることはできます。IAM ロールの作成と使用の詳細については、『IAM ユーザーガイド』の「[Roles](#)」を参照してください。

リソースレベルのアクセス許可を IAM ポリシーに適用して、インスタンスの IAM ロールのアタッチ、置換、またはデタッチをユーザーに許可するかどうかを制御できます。詳細については、「[Amazon EC2 API アクションでサポートされるリソースレベルのアクセス許可 \(p. 844\)](#)」と、「[例: IAM ロールの使用 \(p. 875\)](#)」の例を参照してください。

#### トピック

- [インスタンスプロファイル \(p. 889\)](#)
- [インスタンスマタデータからセキュリティ認証情報を取得する \(p. 889\)](#)
- [IAM ロールをインスタンスに渡すためのアクセス許可を IAM ユーザーに付与する \(p. 890\)](#)
- [IAM ロールの使用 \(p. 891\)](#)

## インスタンスプロファイル

Amazon EC2 は、IAM ロールのコンテナとしてインスタンスプロファイルを使用します。IAM コンソールを使用して IAM ロールを作成すると、コンソールによりインスタンスプロファイルが自動的に作成され、対応するロールと同じ名前が付けられます。Amazon EC2 コンソールを使用して IAM ロールを持つインスタンスを起動する場合、またはインスタンスに IAM ロールをアタッチする場合は、インスタンスプロファイル名のリストに基づいてロールを選択します。

AWS CLI、API、または AWS SDK を使用してロールを作成する場合、ロールとインスタンスプロファイルを別個のアクションとして作成し、基本的に異なる名前を付けます。次に AWS CLI、API、または AWS SDK を使用して IAM ロールを持つインスタンスを起動する場合、またはインスタンスに IAM ロールをアタッチする場合は、インスタンスプロファイル名を指定します。

インスタンスプロファイルに含めることができる IAM ロールの数は 1 つのみです。この制限を増やすことはできません。

詳細については、『IAM ユーザーガイド』の「[インスタンスプロファイル](#)」を参照してください。

## インスタンスマタデータからセキュリティ認証情報を取得する

インスタンスのアプリケーションは、インスタンスマタデータアイテム `iam/security-credentials/role-name` のロールから提供されたセキュリティ認証情報を取得します。アプリケーションには、ロールに関連付けられたセキュリティ認証情報によって、ロールに対して定義したアクションおよびリソースのアクセス許可が付与されます。これらのセキュリティ認証情報は一時的なものであり、私たちが自動的に循環させます。新しい認証情報は、古い認証情報が失効する少なくとも 5 分前から有効になるようにします。

#### Warning

IAM ロールでインスタンスマタデータを使用するサービスを使用する場合は、サービスで HTTP 呼び出しが行われるときに認証情報を公開しないように注意する必要があります。認証情報を公開できるサービスの種類には、HTTP プロキシ、HTML/CSS 検証サービス、および XML インクリードをサポートする XML プロセッサーが含まれます。

以下のコマンドでは、`s3access` という名前の IAM ロールのセキュリティ認証情報を取得します。

#### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

#### IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

出力例を次に示します。

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2017-05-17T15:09:54Z"
}
```

インスタンスで実行されるアプリケーション、AWS CLI、Tools for Windows PowerShell コマンド用に、一時的なセキュリティ認証情報を明示的に取得する必要はありません。AWS SDK、AWS CLI、Tools for Windows PowerShell によって、EC2 インスタンスマタデータサービスから自動的に認証情報が取得され、使用されます。一時的なセキュリティ認証情報を使用してインスタンスの外部で呼び出しを行う (IAM ポリシーをテストするなど) には、アクセスキー、秘密キー、およびセッショントークンを提供する必要があります。詳細については、『IAM ユーザーガイド』の「[一時的なセキュリティ認証情報を使用して AWS リソースへのアクセスをリクエストする](#)」を参照してください。

インスタンスマタデータの詳細については、「[インスタンスマタデータとユーザーデータ \(p. 593\)](#)」を参照してください。

## IAM ロールをインスタンスに渡すためのアクセス許可を IAM ユーザーに付与する

IAM ユーザーに対して、IAM ロールを持つインスタンスの起動や既存インスタンスの IAM ロールの置換を許可するには、ロールをインスタンスに渡すためのアクセス許可を付与します。

次の IAM ポリシーでは、IAM ロールを持つインスタンス (`ec2:RunInstances`) を起動したり、既存のインスタンス (`ec2:AssociateIamInstanceProfile` と `ec2:ReplaceIamInstanceProfileAssociation`) の IAM ロールを置換したりするためのアクセス許可をユーザーに付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances",
        "ec2:AssociateIamInstanceProfile",
```

```
        "ec2:ReplaceIamInstanceProfileAssociation"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "*"
}
]
```

このポリシーでは、ポリシー内でリソースを「\*」と指定することで、お客様のすべてのロールに対するアクセス許可を IAM ユーザーに付与します。ただし、お客様のロール（既存のロールおよび今後作成するロールを含む）を使用してインスタンスを起動するユーザーに、必要ではない、または与えるべきではないアクセス許可が与えられる可能性があることを考慮してください。

## IAM ロールの使用

IAM ロールは、インスタンスの起動時または起動後に作成してインスタンスにアタッチできます。インスタンスの IAM ロールは、置換またはデタッチすることもできます。

### コンテンツ

- [IAM ロールを作成する \(p. 891\)](#)
- [IAM ロールを持つインスタンスを起動する \(p. 893\)](#)
- [IAM ロールをインスタンスにアタッチする \(p. 894\)](#)
- [IAM ロールを置き換える \(p. 895\)](#)
- [IAM ロールをデタッチする \(p. 896\)](#)

## IAM ロールを作成する

IAM ロールを持つインスタンスを起動したり、インスタンスにアタッチしたりするには、そのロールを事前に作成する必要があります。

IAM コンソールを使用して IAM ロールを作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [Roles]、[Create role] の順に選択します。
3. [Select role type] ページで、[EC2] および [EC2] ユースケースを選択します。[Next: Permissions (次へ: 権限)] を選択します。
4. [Attach permissions policy] ページで、必要なリソースへのアクセス権をインスタンスに付与する AWS 管理ポリシーを選択します。
5. [確認] ページで、ロールの名前を入力し、[ロールの作成] を選択します。

または、AWS CLI を使用して IAM ロールを作成することもできます。次の例では、IAM ロールを作成し、このロールに Amazon S3 バケットの使用を許可するポリシーを割り当てます。

IAM ロールおよびインスタンスプロファイルを作成するには (AWS CLI)

1. 以下の信頼ポリシーを作成し、`ec2-role-trust-policy.json` という名前のテキストファイルに保存します。

```
{
    "Version": "2012-10-17",
```

```
"Statement": [
    {
        "Effect": "Allow",
        "Principal": { "Service": "ec2.amazonaws.com" },
        "Action": "sts:AssumeRole"
    }
]
```

2. s3access ロールを作成し、[create-role](#) コマンドを使用して作成した信頼ポリシーを指定します。

```
aws iam create-role --role-name s3access --assume-role-policy-document file://ec2-role-
trust-policy.json
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
                    }
                }
            ],
            "RoleId": "AROAIIZKPBKS2LEXAMPLE",
            "CreateDate": "2013-12-12T23:46:37.247Z",
            "RoleName": "s3access",
            "Path": "/",
            "Arn": "arn:aws:iam::123456789012:role/s3access"
        }
    }
}
```

3. アクセスポリシーを作成し、ec2-role-access-policy.json という名前のテキストファイルに保存します。たとえば、このポリシーは、インスタンスで実行しているアプリケーションに対し、Amazon S3 の管理権限を与えます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3:*"],
            "Resource": ["*"]
        }
    ]
}
```

4. [put-role-policy](#) コマンドを使用して、アクセスポリシーをロールにアタッチします。

```
aws iam put-role-policy --role-name s3access --policy-name S3-Permissions --policy-
document file://ec2-role-access-policy.json
```

5. [create-instance-profile](#) コマンドを使用して、s3access-profile という名前のインスタンスプロファイルを作成します。

```
aws iam create-instance-profile --instance-profile-name s3access-profile
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJTLBPJLEGREXAMPLE",
        "Roles": []
    }
}
```

```
        "CreateDate": "2013-12-12T23:53:34.093Z",
        "InstanceProfileName": "s3access-profile",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:instance-profile/s3access-profile"
    }
}
```

6. s3access インスタンスプロファイルに s3access-profile ロールを追加します。

```
aws iam add-role-to-instance-profile --instance-profile-name s3access-profile --role-name s3access
```

または、以下の AWS Tools for Windows PowerShell コマンドを使用することもできます。

- [New-IAMRole](#)
- [Register-IAMRolePolicy](#)
- [New-IAMInstanceProfile](#)

## IAM ロールを持つインスタンスを起動する

IAM ロールを作成した後、インスタンスを起動して、起動中にそのロールをインスタンスに関連付けることができます。

### Important

IAM ロールを作成した後、適切なアクセス許可が反映されるまで数秒ほどかかります。ロールを使用した最初のインスタンスの起動が失敗した場合は、数秒待ってからもう一度試してください。詳細については、『IAM ユーザーガイド』の「[Troubleshooting Working with Roles](#)」を参照してください。

IAM ロールを使用してインスタンスを起動するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ダッシュボードで、[Launch Instance] を選択します。
3. AMI およびインスタンスタイプを選択し、[Next: Configure Instance Details] を選択します。
4. [Configure Instance Details] ページの [IAM role] で、作成した IAM ロールを選択します。

### Note

[IAM role] リストには、IAM ロールの作成時に作成したインスタンスプロファイルの名前が表示されます。コンソールを使用して IAM ロールを作成した場合、インスタンスプロファイルが自動的に作成され、ロールと同じ名前が付けられます。IAM、API、または AWS CLI SDK を使用して AWS を作成した場合、インスタンスプロファイルに異なる名前を付けた可能性があります。

5. その他の詳細を設定し、ウィザードの残りの部分の指示に従うか、[Review and Launch] を選択してデフォルト設定を受け入れ、直接 [Review Instance Launch] ページに移動します。
6. 設定を確認して [Launch] を選択し、キーペアを選択してインスタンスを起動します。
7. アプリケーションで Amazon EC2 API アクションを使用している場合、インスタンスで有効にされている AWS セキュリティ認証情報を取得し、それを使用しリクエストに署名します。これは、AWS SDK によって行われます。

### IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

または、AWS CLI を使用して起動時にロールをインスタンスに関連付けることもできます。コマンド内でインスタンスプロファイルを指定する必要があります。

IAM ロールを使用してインスタンスを起動するには (AWS CLI)

1. [run-instances](#) コマンドでインスタンスプロファイルを使用してインスタンスを起動します。以下の例は、インスタンスプロファイルを使用してインスタンスを起動する方法を示しています。

```
aws ec2 run-instances --image-id ami-11aa22bb --iam-instance-profile Name="s3access-profile" --key-name my-key-pair --security-groups my-security-group --subnet-id subnet-1a2b3c4d
```

または、[New-EC2Instance Tools for Windows PowerShell](#) コマンドを使用することもできます。

2. アプリケーションで Amazon EC2 API アクションを使用している場合、インスタンスで有効にされている AWS セキュリティ認証情報を取得し、それを使用しリクエストに署名します。これは、AWS SDK によって行われます。

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

## IAM ロールをインスタンスにアタッチする

ロールを持たないインスタンスに IAM ロールをアタッチするには、そのインスタンスを `stopped` または `running` の状態にします。

IAM ロールをインスタンスにアタッチするには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[Actions]、[Instance Settings]、[Attach/Replace IAM role] の順に選択します。
4. インスタンスにアタッチする IAM ロールを選択して、[適用] を選択します。

IAM ロールをインスタンスにアタッチするには (AWS CLI)

1. 必要に応じて、インスタンスを記述して、ロールをアタッチするインスタンスの ID を取得します。

```
aws ec2 describe-instances
```

2. [associate-iam-instance-profile](#) コマンドでインスタンスプロファイルを指定して、IAM ロールをインスタンスにアタッチします。インスタンスプロファイルの Amazon リソースネーム (ARN) またはプロファイル名を使用できます。

```
aws ec2 associate-iam-instance-profile --instance-id i-1234567890abcdef0 --iam-instance-profile Name="TestRole-1"
```

```
{  
    "IamInstanceProfileAssociation": {  
        "InstanceId": "i-1234567890abcdef0",  
        "State": "associating",  
        "AssociationId": "iip-assoc-0dbd8529a48294120",  
        "IamInstanceProfile": {  
            "Id": "AIPAJLNLDX3AMYZNWYYAY",  
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-1"  
        }  
    }  
}
```

または、以下の Tools for Windows PowerShell コマンドを使用します。

- [Get-EC2Instance](#)
- [Register-EC2IamInstanceProfile](#)

## IAM ロールを置き換える

既に IAM ロールが割り当てられているインスタンスで IAM ロールを置き換えるには、インスタンスは running 状態になっている必要があります。既存のロールをデタッチしないでインスタンスの IAM ロールを変更する場合に、これを行うことができます。たとえば、インスタンスで実行しているアプリケーションが実行する API アクションが中断されないようにするために、これを行うことができます。

インスタンスの IAM ロールを置き換えるには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[Actions]、[Instance Settings]、[Attach/Replace IAM role] の順に選択します。
4. インスタンスにアタッチする IAM ロールを選択して、[適用] を選択します。

インスタンスの IAM ロールを置き換えるには (AWS CLI)

1. 必要に応じて、IAM インスタンスプロファイルの関連付けを記述し、置き換える IAM インスタンスプロファイルの関連 ID を取得します。

```
aws ec2 describe-iam-instance-profile-associations
```

2. `replace-iam-instance-profile-association` コマンドで置換元のインスタンスプロファイルの関連 ID と置換先のインスタンスプロファイルの ARN 名またはプロファイル名を指定して、IAM インスタンスプロファイルを置き換えます。

```
aws ec2 replace-iam-instance-profile-association --association-id iip-assoc-0044d817db6c0a4ba --iam-instance-profile Name="TestRole-2"
```

```
{  
    "IamInstanceProfileAssociation": {  
        "InstanceId": "i-087711ddaf98f9489",  
        "State": "associating",  
        "AssociationId": "iip-assoc-09654be48e33b91e0",  
        "IamInstanceProfile": {  
            "Id": "AIPAJCJEDKX7QYHWYK7GS",  
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"  
        }  
    }  
}
```

```
}
```

または、以下の Tools for Windows PowerShell コマンドを使用します。

- [Get-EC2IamInstanceProfileAssociation](#)
- [Set-EC2IamInstanceProfileAssociation](#)

## IAM ロールをデタッチする

実行中または停止中のインスタンスから IAM ロールをデタッチできます。

インスタンスから IAM ロールをデタッチするには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[Actions]、[Instance Settings]、[Attach/Replace IAM role] の順に選択します。
4. [IAM role] で、[No Role] を選択します。[Apply] を選択します。
5. 確認ダイアログボックスで、[Yes, Detach] を選択します。

インスタンスから IAM ロールをデタッチするには (AWS CLI)

1. 必要に応じて、[describe-iam-instance-profile-associations](#) で IAM インスタンスプロファイルの関連付けを記述し、デタッチする IAM インスタンスプロファイルの関連 ID を取得します。

```
aws ec2 describe-iam-instance-profile-associations

{
    "IamInstanceProfileAssociations": [
        {
            "InstanceId": "i-088ce778fbfeb4361",
            "State": "associated",
            "AssociationId": "iip-assoc-0044d817db6c0a4ba",
            "IamInstanceProfile": {
                "Id": "AIPAJEDNCAA64SSD265D6",
                "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"
            }
        }
    ]
}
```

2. [disassociate-iam-instance-profile](#) コマンドで関連 ID を使用して IAM インスタンスプロファイルをデタッチします。

```
aws ec2 disassociate-iam-instance-profile --association-id iip-assoc-0044d817db6c0a4ba

{
    "IamInstanceProfileAssociation": {
        "InstanceId": "i-087711ddaf98f9489",
        "State": "disassociating",
        "AssociationId": "iip-assoc-0044d817db6c0a4ba",
        "IamInstanceProfile": {
            "Id": "AIPAJEDNCAA64SSD265D6",
            "Arn": "arn:aws:iam::123456789012:instance-profile/TestRole-2"
        }
    }
}
```

}

または、以下の Tools for Windows PowerShell コマンドを使用します。

- [Get-EC2IamInstanceProfileAssociation](#)
- [Unregister-EC2IamInstanceProfile](#)

## Linux インスタンス用の受信トラフィックの認可

セキュリティグループを使用すると、どのトラフィックがインスタンスに到達できるかなど、インスタンスへのトラフィックを制御できます。たとえば、ホームネットワークからのコンピュータのみが SSH を使用してインスタンスにアクセスできるように許可できます。インスタンスがウェブサーバーの場合、すべての IP アドレスが HTTP または HTTPS を使用してインスタンスにアクセスできるようにすることで、外部ユーザーはウェブサーバーのコンテンツを閲覧できるようになります。

デフォルトのセキュリティグループと新しく作成されたセキュリティグループには、インターネットからインスタンスにアクセスできないデフォルトのルールが含まれます。詳細については、「[デフォルトのセキュリティグループ \(p. 914\)](#)」および「[カスタムのセキュリティグループ \(p. 915\)](#)」を参照してください。インスタンスへのネットワークアクセスを有効にするには、インスタンスへのインバウンドトラフィックを許可する必要があります。受信トラフィック用のポートを開くには、起動時にインスタンスに関連付けたセキュリティグループにルールを追加します。

インスタンスに接続するには、コンピュータのパブリック IPv4 アドレスからの SSH トラフィックを承認するルールをセットアップする必要があります。追加の IP アドレス範囲からの SSH トラフィックを許可するには、承認する必要がある範囲ごとに別のルールを追加します。

IPv6 の VPC を有効にして IPv6 アドレスを使用してインスタンスを起動している場合は、パブリック IPv4 アドレスではなくインスタンスの IPv6 アドレスを使用してインスタンスに接続できます。ローカルコンピュータに IPv6 アドレスがあり、IPv6 を使用するように設定されている必要があります。

Windows インスタンスへのネットワークアクセスを利用可能にする必要がある場合は、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows インスタンス用の受信トラフィックの認可](#)」を参照してください。

## 開始する前に

インスタンスへのアクセスの要求元 (例: ローカルコンピュータのパブリック IPv4 アドレスなど、信頼する単一のホストや特定のネットワーク) を判断します。Amazon EC2 コンソールのセキュリティグループエディタは、ローカルコンピュータのパブリック IPv4 アドレスを自動的に検出できます。別の方法として、インターネットブラウザで検索文字列として「私の IP アドレスは何ですか?」を使用するか、次のサービス: [Check IP](#) を使用することもできます。ISP 経由で、またはファイアウォールの内側から静的な IP アドレスなしで接続している場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。

### Warning

0.0.0.0/0 を使用すると、すべての IPv4 アドレスから SSH 経由でインスタンスにアクセスすることができます。::/0 を使用すると、すべての IPv6 アドレスからインスタンスにアクセスできるようになります。これはテスト環境で短時間なら許容できますが、実稼働環境で行うのは安全ではありません。本番環境では、特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを限定します。

EC2 Instance Connect を使用してインスタンスへの SSH アクセスをサポートするかどうかを決定します。EC2 Instance Connect を使用しない場合は、アンインストールするか、IAM ポリシーで次のアクションを拒否することを検討してください。`ec2-instance-connect:SendSSHPublicKey`。詳細については、「[EC2 Instance Connect のアンインストール \(p. 519\)](#)」および「[EC2 Instance Connect の IAM アクセス権限を設定する \(p. 515\)](#)」を参照してください。

## Linux インスタンスに対するインバウンド SSH トラフィックのルールの追加

セキュリティグループは、関連付けられたインスタンスのファイアウォールとして動作し、インバウンド トラフィックとアウトバウンド トラフィックの両方をインスタンスレベルでコントロールします。SSH を使用して IP アドレスから Linux インスタンスに接続できるようにするためのルールをセキュリティグループに追加します。

IPv4 でインバウンド SSH トラフィック用のルールをセキュリティグループに追加するには (コンソール)

1. Amazon EC2 コンソールのナビゲーションペインで、[Instances] を選択します。インスタンスを選択し、[Description] タブを確認します。[Security groups] リストに、インスタンスに関連付けられたセキュリティグループが表示されます。[view inbound rules] を選択して、インスタンスに対して有効なルールのリストを表示します。
2. ナビゲーションペインで、[Security Groups] を選択します。インスタンスに関連付けられているセキュリティグループのいずれかを選択します。
3. 詳細ペインの [Inbound] タブで、[Edit] を選択します。ダイアログで [Add Rule] を選択し、[Type] リストから [SSH] を選択します。
4. [Source] フィールドで [My IP] を選択すると、ローカルコンピュータのパブリック IPv4 アドレスが自動的にフィールドに入力されます。別の方法として、[Custom] を選択してコンピュータまたはネットワークのパブリック IPv4 アドレスを CIDR 表記で指定することもできます。たとえば、IPv4 アドレスが 203.0.113.25 である場合、この単一の IPv4 アドレスを CIDR 表記で示すには 203.0.113.25/32 と指定します。会社が特定の範囲からアドレスを割り当てる場合、範囲全体 (203.0.113.0/24 など) を指定します。

IP アドレスを見つける方法については、[開始する前に \(p. 897\)](#) を参照してください。

5. [Save] を選択します。

IPv6 アドレスを持つインスタンスを起動して、その IPv6 アドレスを使用してインスタンスに接続する場合は、SSH でインバウンド IPv6 トラフィックを許可するルールを追加する必要があります。

IPv6 でインバウンド SSH トラフィック用のルールをセキュリティグループに追加するには (コンソール)

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] を選択します。インスタンスのセキュリティグループを選択します。
3. [Inbound]、[Edit]、[Add Rule] の順に選択します。
4. [Type] で [SSH] を選択します。
5. [Source] フィールドで、コンピュータの IPv6 アドレスを CIDR 表記で指定します。たとえば、IPv6 アドレスが 2001:db8:1234:1a00:9691:9503:25ad:1761 である場合、この単一の IP アドレスを CIDR 表記で示すには 2001:db8:1234:1a00:9691:9503:25ad:1761/128 と指定します。会社が特定の範囲からアドレスを割り当てる場合、範囲全体 (2001:db8:1234:1a00::/64 など) を指定します。
6. [Save] を選択します。

### Note

次のコマンドが、インスタンスではなく、ローカルシステムで実行されていることを確認してください。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

コマンドラインを使用してセキュリティグループにルールを追加するには

- 以下のいずれかのコマンドを使用してインスタンスに関連付けられるセキュリティグループを見つける:
  - [describe-instance-attribute \(AWS CLI\)](#)

```
aws ec2 describe-instance-attribute --instance-id instance_id --attribute groupSet
```

- [Get-EC2InstanceAttribute \(AWS Tools for Windows PowerShell\)](#)

```
PS C:\> (Get-EC2InstanceAttribute -InstanceId instance_id -Attribute groupSet).Groups
```

どちらのコマンドも、次のステップで使用できるセキュリティ グループ ID を返します。

- 以下のいずれかのコマンドを使用してセキュリティグループにルールを追加します。
  - [authorize-security-group-ingress \(AWS CLI\)](#)

```
aws ec2 authorize-security-group-ingress --group-id security_group_id --protocol tcp  
--port 22 --cidr cidr_ip_range
```

- [Grant-EC2SecurityGroupIngress \(AWS Tools for Windows PowerShell\)](#)

Grant-EC2SecurityGroupIngress コマンドには、IpPermission パラメーターが必要です。このパラメーターは、セキュリティグループのルールに使用するプロトコル、ポート範囲、IP アドレス範囲を定義します。次のコマンドでは、IpPermission パラメーターが作成されます。

```
PS C:\> $ip1 = @{ IpProtocol="tcp"; FromPort="22"; ToPort="22";  
IpRanges="cidr_ip_range" }
```

```
PS C:\> Grant-EC2SecurityGroupIngress -GroupId security_group_id -IpPermission  
@$ip1
```

## インスタンスへのセキュリティグループの割り当て

インスタンスを起動する際に、インスタンスにセキュリティグループを割り当てることができます。ルールを追加または削除すると、それらの変更は、そのセキュリティグループを割り当てたすべてのインスタンスに自動的に適用されます。

インスタンスを起動した後、そのセキュリティグループを変更することができます。詳細については、『Amazon VPC ユーザーガイド』の「[インスタンスのセキュリティグループを変更する](#)」を参照してください。

## Amazon EC2 のキーペア

Amazon EC2 はパブリックキー暗号を使用して、ログイン情報の暗号化と復号を行います。パブリックキー暗号はパブリックキーを使用してデータを暗号化し、受取人はプライベートキーを使用してデータを復号します。パブリックキーとプライベートキーは、キーペアと呼ばれます。パブリックキー暗号化では、パスワードの代わりにプライベートキーを使用して、安全にインスタンスにアクセスできます。

インスタンスを起動するとき、キーペアを指定します。既存のキーペアまたは起動時に作成する新しいキーペアを指定できます。起動時に、パブリックキーは、`~/.ssh/authorized_keys` 内の工

ントリに配置されます。インスタンスにログインするには、インスタンスに接続するときにプライベートキーを指定する必要があります。詳細については、[インスタンスの起動 \(p. 448\)](#) および [Linux インスタンスへの接続 \(p. 505\)](#)。

### キーペアを作成する

Amazon EC2 を使用してキーペアを作成できます。詳細については、「[Amazon EC2 を使用してキーペアを作成する \(p. 901\)](#)」を参照してください。

または、サードパーティー製のツールで、パブリックキーを Amazon EC2 にインポートすることもできます。詳細については、「[独自のパブリックキーを Amazon EC2 にインポートする \(p. 902\)](#)」を参照してください。

それぞれのキーペアには名前が必要です。覚えやすい名前を選択するようにしてください。Amazon EC2 は、パブリックキーをキー名として指定した名前に関連付けます。

Amazon EC2 はパブリックキーのみを保存し、お客様はプライベートキーを保存します。お客様のプライベートキーを持っていれば誰でもお客様のログイン情報を復号できるので、プライベートキーを安全な場所に保存することが重要です。

Amazon EC2 が使用するキーは、2048-bit SSH-2 RSA キーです。リージョンごとに最大 5,000 のキーペアを設定できます。

### インスタンスを起動し、接続する

インスタンスを起動するときは、インスタンスへの接続に使用するキーペアの名前を指定する必要があります。インスタンスの起動時に既存のキーペアを指定しない場合、インスタンスに接続することはできません。インスタンスに接続するときは、インスタンスの起動時に指定したキーペアに対応するプライベートキーを指定する必要があります。

#### Note

Amazon EC2 ではプライベートキーのコピーが保持されないため、プライベートキーを失った場合、復元することはできません。Instance store-Backed インスタンスのプライベートキーを失った場合は、インスタンスにアクセスできなくなります。そのため、インスタンスを終了し、新しいキーペアを使用して、別のインスタンスを起動する必要があります。EBS-Backed Linux インスタンスのプライベートキーを失った場合は、インスタンスへのアクセス権を回復することができます。詳細については、「[プライベートキーを紛失した場合の Linux インスタンスへの接続 \(p. 907\)](#)」を参照してください。

### 複数ユーザー用のキーペア

単一のインスタンスにアクセスする複数のユーザーがいる場合、インスタンスにユーザーアカウントを追加できます。詳細については、「[Linux インスタンスでのユーザーアカウントの管理 \(p. 559\)](#)」を参照してください。各ユーザー用にキーペアを作成し、インスタンスの各ユーザー用の .ssh/authorized\_keys ファイルに各キーペアからのパブリックキー情報を追加できます。その後、ユーザーに対してプライベートキーファイルを配布できます。この方法では、AWS アカウントルートユーザー用に使用しているプライベートキーファイルと同一のファイルを複数のユーザーに配布する必要はありません。

### 目次

- [Amazon EC2 を使用してキーペアを作成する \(p. 901\)](#)
- [独自のパブリックキーを Amazon EC2 にインポートする \(p. 902\)](#)
- [キーペアのパブリックキーを取得する \(Linux\) \(p. 903\)](#)
- [キーペアのパブリックキーを取得する \(Windows\) \(p. 904\)](#)
- [インスタンスからキーペアのパブリックキーを取得する \(p. 904\)](#)
- [キーペアのフィンガープリントの確認 \(p. 905\)](#)

- キーペアの削除 (p. 906)
- インスタンスのキーペアの追加または交換 (p. 907)
- プライベートキーを紛失した場合の Linux インスタンスへの接続 (p. 907)

## Amazon EC2 を使用してキーペアを作成する

キーペアを作成すると、インスタンス起動時にこのキーペアを指定できます。実行中のインスタンスにこのキーペアを追加すると、他のユーザーがインスタンスに接続できるようになります。詳細については、「[インスタンスのキーペアの追加または交換 \(p. 907\)](#)」を参照してください。

キーペアは、次のいずれかの方法で作成できます。

新しいコンソール

キーペアを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[キーペア] を選択します。
3. [キーペアの作成] を選択します。
4. [Name (名前)] に、キーペアのわかりやすい名前を入力します。
5. [File format (ファイル形式)] で、プライベートキーを保存する形式を選択します。OpenSSH で使用できる形式でプライベートキーを保存するには、[pem] を選択します。プライベートキーを PuTTY で使用できる形式で保存するには、[ppk] を選択します。
6. [キーペアの作成] を選択します。

古いコンソール

キーペアを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [NETWORK & SECURITY] で、[Key Pairs] を選択します。

Note

ナビゲーションペインは Amazon EC2 コンソールの左側にあります。ペインが表示されない場合、最小化されている可能性があります。矢印を選択してペインを展開します。

3. [Create Key Pair] を選択します。
4. [キーペア名] に新しいキーペアの名前を入力し、[作成] を選択します。
5. ブラウザによって秘密キーファイルが自動的にダウンロードされます。ベースファイル名はキーペアの名前として指定した名前となり、ファイル名の拡張子は .pem となります。プライベートキーファイルを安全な場所に保存します。

Important

これは、プライベートキーを保存する唯一のチャンスです。インスタンスと対応するプライベートキーの起動時には、毎回インスタンスに接続するたびに、キーペアの名前を入力する必要があります。

6. macOS または Linux コンピュータの SSH クライアントを使用して Linux インスタンスに接続する場合は、次のコマンドを使用してプライベートキーファイルのアクセス許可を設定すると、お客様以外のユーザーはそれを読み取ることができないようになります。

```
chmod 400 my-key-pair.pem
```

これらのアクセス権限を設定しないと、このキーペアを使用してインスタンスに接続できません。詳細については、「[エラー: Unprotected Private Key File \(保護されていないプライベートキーファイル\) \(p. 1141\)](#)」を参照してください。

#### AWS CLI

キーペアを作成するには

[create-key-pair](#) AWS CLI コマンドを使用します。

#### PowerShell

キーペアを作成するには

[New-EC2KeyPair](#) AWS Tools for Windows PowerShell コマンドを使用します。

## 独自のパブリックキーを Amazon EC2 にインポートする

Amazon EC2 を使用してキーペアを作成する代わりに、サードパーティ製のツールで RSA キーペアを作成してから、パブリックキーを Amazon EC2 にインポートすることもできます。たとえば、ssh-keygen (標準 OpenSSH インストールで提供されるツール) を使用して、キーペアを作成できます。また、Java、Ruby、Python などのさまざまなプログラミング言語では、RSA キーペアの作成に使用できる標準ライブラリが提供されています。

#### 要件

- 以下の形式がサポートされています。
  - OpenSSH パブリックキー形式 (~/.ssh/authorized\_keys の形式) EC2 Instance Connect API の使用中に SSH を使用して接続する場合は、SSH2 形式もサポートされます。
  - Base64 でエンコードされた DER 形式
  - SSH パブリックキーファイル形式 ([RFC4716](#) で指定)
  - SSH プライベートキーファイルの形式が PEM である必要があります (たとえば、ssh-keygen -m PEM を使用して、OpenSSH キーを PEM 形式に変換)。
- RSA キーを作成します。Amazon EC2 では DSA キーは使用できません。
- サポートされている長さは 1024、2048、および 4096 です。EC2 Instance Connect API の使用中に SSH を使用して接続する場合は、長さ 2048 および 4096 がサポートされます。

サードパーティツールを使用してキーペアを作成するには

- 選択したサードパーティ製のツールでキーペアを生成します。
- ローカルファイルにパブリックキーを保存します。例: ~/.ssh/my-key-pair.pub (Linux の場合)、または C:\keys\my-key-pair.pub(Windows の場合)。このファイル名の拡張子は重要ではありません。
- .pem 拡張子を持つ別のローカルファイルにプライベートキーを保存します。例: ~/.ssh/my-key-pair.pem (Linux の場合)、または C:\keys\my-key-pair.pem(Windows の場合)。プライベートキーファイルを安全な場所に保存します。インスタンスと対応するプライベートキーの起動時には、毎回インスタンスに接続するたびに、キーペアの名前を入力する必要があります。

キーペアを作成したら、次のいずれかの方法を使用してキーペアを Amazon EC2 にインポートします。

## 新しいコンソール

### パブリックキーをインポートするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[キーペア] を選択します。
3. [Import Key Pair (キーペアのインポート)] を選択します。
4. [Name (名前)] に、キーペアのわかりやすい名前を入力します。
5. [Browse (参照)] を選択してパブリックキーに移動して選択するか、パブリックキーのコンテンツを [Public key contents (パブリックキーのコンテンツ)] フィールドに貼り付けます。
6. [Import Key Pair (キーペアのインポート)] を選択します。
7. インポートしたキーペアがキーペアのリストに表示されていることを確認します。

## 古いコンソール

### パブリックキーをインポートするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [NETWORK & SECURITY] で、[Key Pairs] を選択します。
3. [Import Key Pair] を選択します。
4. [Import Key Pair] ダイアログボックスで [Browse] を選択し、前に保存したパブリックキーファイルを選択します。[Key pair name] フィールドにキーペアの名前を入力し、[Import] を選択します。
5. インポートしたキーペアがキーペアのリストに表示されていることを確認します。

## AWS CLI

### パブリックキーをインポートするには

`import-key-pair` AWS CLI コマンドを使用します。

キーペアが正常にインポートされたことを確認するには

`describe-key-pairs` AWS CLI コマンドを使用します。

## PowerShell

### パブリックキーをインポートするには

`Import-EC2KeyPair` AWS Tools for Windows PowerShell コマンドを使用します。

キーペアが正常にインポートされたことを確認するには

`Get-EC2KeyPair` AWS Tools for Windows PowerShell コマンドを使用します。

## キーペアのパブリックキーを取得する (Linux)

ローカルの Linux または macOS コンピュータで、`ssh-keygen` コマンドを使用して、キーペアのパブリックキーを取得します。プライベートキーをダウンロードしたパスを指定します (`.pem` ファイル)。

```
ssh-keygen -y -f /path_to_key_pair/my-key-pair.pem
```

コマンドは、次の例に示すように、パブリックキーを返します。

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gu8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBITntckiJ7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

コマンドが失敗した場合は、次のコマンドを実行して、自分がキーペアファイルを表示できるよう、このファイルに対するアクセス許可を変更していることを確認してください。

```
chmod 400 my-key-pair.pem
```

## キーペアのパブリックキーを取得する (Windows)

ローカルの Windows コンピュータでは、PuTTYgen を使用してキーペアのパブリックキーを取得します。

PuTTYgen を起動し、[Load] を選択します。.ppk または .pem ファイルを選択します。PuTTYgen の [Public key for pasting into OpenSSH authorized\_keys ファイル] にパブリックキーが表示されます。パブリックキーは、[パブリックキーの保存] を選択してファイルの名前を指定し、ファイルを保存後、そのファイルを開いて表示することもできます。

## インスタンスからキーペアのパブリックキーを取得する

インスタンスの起動時に指定したパブリックキーも、そのインスタンスマタデータを介して表示できます。インスタンスの起動時に指定したパブリックキーを表示するには、インスタンスから次のコマンドを使用します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

以下に出力例を示します。

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gu8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBITntckiJ7FbtJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
```

以下に出力例を示します。

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
```

```
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

インスタンスへの接続に使用するキーペアを変更しても、新しいパブリックキーを表示するようにインスタンスマタデータは更新されません。代わりに、インスタンスのメタデータは、インスタンスの起動時に指定したキーペアのパブリックキーを引き続き表示します。詳細については、「[インスタンスマタデータの取得 \(p. 600\)](#)」を参照してください。

または、Linux インスタンスでは、パブリックキーは、`~/.ssh/authorized_keys` 内のエントリに配置されます。このファイルは、エディタで開くことができます。以下に、`my-key-pair` という名前のキーペアのエントリの例を示します。パブリックキーの後にキーペアの名前が続く構成になっています。

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706V
hz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBXr
lsLnBItntckiJ7FbtxJMxLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0Fzz
qaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxxXPkX4rWi+z7wB3Rb
BQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

## キーペアのフィンガープリントの確認

Amazon EC2 コンソールの [キーペア] ページで、[フィンガープリント] 列にキーペアから生成されたフィンガープリントが表示されます。AWS では、キーペアが AWS とサードパーティツールのいずれで生成されたかによって、フィンガープリントの計算が異なります。AWS を使用してキーペアを作成した場合、フィンガープリントは、SHA-1 ハッシュ関数を使用して計算されます。サードパーティツールによってキーペアを作成し、パブリックキーを AWS にアップロードした場合、または AWS で作成した既存のプライベートキーから新しいパブリックキーを作成し、AWS にアップロードした場合、フィンガープリントは、MD5 ハッシュ関数を使用して計算されます。

[Key Pairs (キーペア)] ページに表示される SSH2 フィンガープリントを使用して、ローカルコンピュータにあるプライベートキーが、AWS に格納されているパブリックキーと一致することを確認できます。プライベートキーファイルをダウンロードしたコンピュータから、プライベートキーファイルを使用して SSH2 フィンガープリントを生成します。出力はコンソールに表示されるフィンガープリントと一致する必要があります。

AWS を使用してキーペアを作成した場合、OpenSSL ツールを使用して次の例のようにフィンガープリントを生成できます。

```
$ openssl pkcs8 -in path_to_private_key -inform PEM -outform DER -topk8 -nocrypt | openssl
sha1 -c
```

サードパーティツールを使用してキーペアを作成し、パブリックキーを AWS にアップロードした場合、OpenSSL ツールを使用して、次の例のようにフィンガープリントを生成できます。

```
$ openssl rsa -in path_to_private_key -pubout -outform DER | openssl md5 -c
```

OpenSSH 7.8 以降を使用して OpenSSH キーペアを作成し、パブリックキーを AWS にアップロードした場合、`ssh-keygen` を使用して、次の例のようにフィンガープリントを生成できます。

```
$ ssh-keygen -ef path_to_private_key -m PEM | openssl rsa -RSAPublicKey_in -outform DER |
openssl md5 -c
```

## キーペアの削除

キーペアを削除すると、Amazon EC2 のパブリックキーのコピーのみが削除されます。キーペアの削除は、コンピュータのプライベートキーにも、このキーペアを使用して既に起動している各インスタンスのパブリックキーにも影響しません。削除したキーペアを使用して新しいインスタンスを起動することはできませんが、プライベートキー (.pem) ファイルを保持している間は、削除したキーペアを使用して起動した各インスタンスに引き続き接続することができます。

### Note

Auto Scaling グループ (Elastic Beanstalk 環境など) を使用している場合、削除するキーペアが起動設定で指定されていないことを確認します。Amazon EC2 Auto Scaling は異常なインスタンスを検出した場合、代わりのインスタンスを起動します。ただし、キーペアが見つからない場合は、インスタンスの起動に失敗します。

次のいずれかの方法を使用して、キーペアを削除できます。

### 新しいコンソール

#### キーペアを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[キーペア] を選択します。
3. 削除するキーペアを選択し、[Delete (削除)] を選択します。
4. 確認フィールドで、Delete を入力し、[Delete (削除)] を選択します。

### 古いコンソール

#### キーペアを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [NETWORK & SECURITY] で、[Key Pairs] を選択します。
3. キーペアを選択し、[Delete] を選択します。
4. プロンプトが表示されたら、[Yes] を選択します。

### AWS CLI

#### キーペアを削除するには

`delete-key-pair` AWS CLI コマンドを使用します。

### PowerShell

#### キーペアを削除するには

`Remove-EC2KeyPair` AWS Tools for Windows PowerShell コマンドを使用します。

### Note

インスタンスから Linux AMI を作成し、AMI を使用して別のリージョンまたはアカウントの新しいインスタンスを起動すると、新しいインスタンスには元のインスタンスからのパブリックキーが含まれます。これにより、元のインスタンスと同じプライベートキー/ファイルを使用して新しいインスタンスに接続できます。インスタンスからこのパブリックキーを削除するには、任意のテキストエディターを使用して、`.ssh/authorized_keys` ファイルからそのエントリを削除し

ます。インスタンスでのユーザーの管理、特定のキーペアを使用したリモートアクセス権の付与については、「[Linux インスタンスでのユーザーアカウントの管理 \(p. 559\)](#)」を参照してください。

## インスタンスのキーペアの追加または交換

インスタンスのデフォルトシステムアカウントにアクセスするために使用するキーペアは変更できます。たとえば、組織のユーザーが、別のキーペアを使用してシステムユーザーアカウントにアクセスする必要がある場合は、キーペアをインスタンスに追加できます。または、.pem ファイルのコピーを持つ者が存在し、インスタンスに接続させないようにするには(たとえば、組織を去った場合)、既存のキーペアを新しいキーペアに交換することができます。

### Note

この手順では、デフォルトのユーザーアカウント(例: ec2-user)のキーペアを変更するためのものです。インスタンスにユーザーアカウントを追加する方法については、「[Linux インスタンスでのユーザーアカウントの管理 \(p. 559\)](#)」を参照してください。

### キーペアの追加または交換

1. [Amazon EC2 コンソール \(p. 901\)](#) または [サードパーティ製のツール \(p. 902\)](#) で、新しいキーペアを作成します。
2. 新しいキーペアからパブリックキーを取得します。詳細については、「[キーペアのパブリックキーを取得する \(Linux\) \(p. 903\)](#)」または「[キーペアのパブリックキーを取得する \(Windows\) \(p. 904\)](#)」を参照してください。
3. 既存のプライベートキーファイルを使用してインスタンスに接続します。
4. 任意のテキストエディタを使用して、インスタンス上にある .ssh/authorized\_keys ファイルを開きます。既存のパブリックキー情報の下の新しいキーペアからパブリックキーを貼り付けます。ファイルを保存します。
5. インスタンスから切断し、新しいプライベートキーファイルを使用してインスタンスに接続できることを確認します。
6. (オプション) 既存のキーペアを交換している場合は、インスタンスに接続し、.ssh/authorized\_keys ファイルからオリジナルのキーペアのパブリックキー情報を削除します。

### Note

Auto Scaling グループ(Elastic Beanstalk 環境など)を使用している場合、交換するキーペアが起動設定で指定されていないことを確認します。Amazon EC2 Auto Scaling は異常なインスタンスを検出した場合、代わりのインスタンスを起動します。ただし、キーペアが見つからない場合は、インスタンスの起動に失敗します。

## プライベートキーを紛失した場合の Linux インスタンスへの接続

EBS-Backed インスタンスのプライベートキーを失った場合は、インスタンスへのアクセス権を回復することができます。インスタンスを停止し、そのルートボリュームをデタッチし、データボリュームとして別のインスタンスにアタッチし、authorized\_keys ファイルを変更して、ボリュームを元のインスタンスに戻し、インスタンスを再起動する必要があります。インスタンスの起動、接続、および停止の詳細については、「[インスタンスのライフサイクル \(p. 443\)](#)」を参照してください。

この手順は、instance store-backed インスタンスではサポートされません。インスタンスのルートデバイスタイプを判断するには、Amazon EC2 コンソールを開き、[Instances] を選択してインスタンスを選択

し、詳細ペインで [Root device type] の値をチェックします。この値は ebs または instance store のどちらかです。ルートデバイスがインスタンストアボリュームである場合、インスタンスに接続するにはプライベートキーが必要です。

別のキーペアを使用して EBS-Backed インスタンスに接続するには

1. Amazon EC2 コンソールまたはサードパーティ製のツールで、新しいキーペアを作成します。新しいキーペアの名前として、紛失したプライベートキーと同じ名前を指定するには、まず既存のキーペアを削除する必要があります。
2. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
3. ナビゲーションペインで [Instances] を選択し、接続先にするインスタンスを選択します(このインスタンスを「元のインスタンス」と呼びます)。
4. [Description (説明)] タブで、この手順を完了する上で必要となる以下の情報を控えておきます。
  - 元のインスタンスのインスタンス ID、AMI ID、およびアベイラビリティーゾーンを書き留めます。
  - [Root device] フィールドで、ルートボリュームのデバイス名 (/dev/sda1 や /dev/xvda など) を記録します。リンクを選択し、[EBS ID] フィールドでボリューム ID (vol-xxxxxxxxxxxxxxx) を記録します。
5. [Actions] を選択して [Instance State] を選択し、[Stop] を選択します。[Stop] が無効になっている場合は、インスタンスが既に停止しているか、またはルートボリュームがインスタンストアボリュームです。

Warning

インスタンスを停止すると、インスタンストアボリューム上のデータは消去されます。インスタンストアボリュームのデータを保持するには、このデータを永続的ストレージに必ずバックアップしてください。

6. [Launch Instance] を選択し、起動ウィザードを使用して、以下のオプションで一時インスタンスを起動します。
    - [Choose an AMI] ページで、元のインスタンスを起動するのに使用したのと同じ AMI を選択します。その AMI を使用できない場合は、停止したインスタンスから使用可能な AMI を作成できます。詳細については、「[Amazon EBS-Backed Linux AMI の作成 \(p. 116\)](#)」を参照してください。
    - [Choose an Instance Type] ページで、ウィザードによって自動的に選択されたデフォルトのインスタンスタイプをそのままにします。
    - [Configure Instance Details] ページで、接続するインスタンスと同じアベイラビリティーゾーンを指定します。VPC のインスタンスを起動する場合、このアベイラビリティーゾーンのサブネットを選択します。
    - [Add Tags] ページで、一時インスタンスであることを示すために、インスタンスに Name=Temporary タグを追加します。
    - [Review] ページで、[Launch] を選択します。新しいキーペアを作成し、コンピューター上の安全な場所にダウンロードして、[Launch Instances] を選択します。
7. ナビゲーションペインで [Volumes] を選択し、元のインスタンスのルートデバイスボリュームを選択します(前のステップでそのボリューム ID を書き留めました)。ボリュームを選択し、[Actions (アクション)]、[Detach Volume (ボリュームのデタッチ)]、[Yes, Detach (はい、デタッチする)] の順に選択します。ボリュームの状態が available になるまで待ちます([Refresh] アイコンを選択しなければならない場合があります)。
  8. ボリュームを選択したまま [Actions] を選択し、次に [Attach Volume] を選択します。一時インスタンスのインスタンス ID を選択し、[デバイス] で指定したデバイス名(例: /dev/sdf)を書き留めて、[Yes, Attach (はい、アタッチする)] を選択します。

Note

元のインスタンスを AWS Marketplace AMI から起動して、ボリュームに AWS Marketplace のコードが含まれている場合は、ボリュームをアタッチする前に一時インスタンスを停止する必要があります。

- 
9. 一時インスタンスに接続します。
  10. 一時インスタンスから、そのファイルシステムにアクセスできるように、インスタンスにアタッチしたボリュームをマウントします。たとえば、デバイス名が /dev/sdf の場合、次のコマンドを使用してボリュームを /mnt/tempvol としてマウントします。

Note

デバイス名の表示がインスタンスでは異なる場合があります。たとえば、/dev/sdf としてマウントされているデバイスが、インスタンスでは /dev/xvdf として表示される場合があります。Red Hat の一部のバージョン（または CentOS などのバリエント）では、さらに末尾の文字が 4 文字インクリメントされる場合があります。たとえば、/dev/sd~~f~~ は /dev/xvd~~k~~ になります。

- a. lsblkコマンドを使用して、ボリュームがパーティション分割されているかどうかを判断します。

```
[ec2-user ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda   202:0    0   8G  0 disk 
##xvda1 202:1    0   8G  0 part /
xvdf   202:80   0 101G  0 disk 
##xvdf1 202:81   0 101G  0 part
xvdg   202:96   0   30G  0 disk
```

前述の例では、/dev/xvda と /dev/xvdf は、パーティション分割されたボリュームで、/dev/xvdg はパーティション分割されていません。ボリュームがパーティション分割されている場合は、次のステップで raw デバイス (/dev/xvdf) の代わりにパーティション (/dev/xvdf1) をマウントします。

- b. ボリュームをマウントするための一時ディレクトリを作成します。

```
[ec2-user ~]$ sudo mkdir /mnt/tempvol
```

- c. 以前に特定したデバイス名またはボリューム名を使用して、一時マウントポイントにボリューム（またはパーティション）をマウントします。必要なコマンドは、オペレーティングシステムのファイルシステムによって異なります。

- Amazon Linux、Ubuntu および Debian

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

- Amazon Linux 2、CentOS、SLES 12、および RHEL 7.x

```
[ec2-user ~]$ sudo mount -o nouuid /dev/xvdf1 /mnt/tempvol
```

Note

ファイルシステムが破損していることを示すエラーが表示された場合は、次のコマンドを実行して fsck ユーティリティを使用してファイルシステムをチェックし、問題を修復します。

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

11. 一時インスタンスから、次のコマンドを使用して、一時インスタンスの authorized\_keys からの新しいパブリックキーを使用し、マウントされたボリューム上で authorized\_keys を更新します。

Important

以下の例では、Amazon Linux ユーザー名 ec2-user を使用します。Ubuntu インスタンスの場合は ubuntu など、別のユーザー名への置き換えが必要になる場合があります。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
プライベートキーを紛失した場合の Linux インスタンスへの接続

```
[ec2-user ~]$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

このコピーが正常に終了すると、次のステップに進むことができます。

(オプション) または、/mnt/tempvol のファイルを編集するアクセス許可がない場合、sudo を使用してファイルを更新してから、ファイルに対するアクセス許可を確認して、元のインスタンスにログインできるかどうかを確認する必要があります。次のコマンドを使用して、ファイルに対するアクセス許可を確認します。

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh  
total 4  
-rw----- 1 222 500 398 Sep 13 22:54 authorized_keys
```

この出力例では、**222** はユーザー ID、**500** はグループ ID です。次に、sudo を使用して失敗したコピーコマンドを再実行します。

```
[ec2-user ~]$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/  
authorized_keys
```

次のコマンドを再度実行して、アクセス許可が変更されているかどうかを判断します。

```
[ec2-user ~]$ sudo ls -l /mnt/tempvol/home/ec2-user/.ssh
```

ユーザー ID とグループ ID が変更されている場合は、次のコマンドを実行して復元します。

```
[ec2-user ~]$ sudo chown 222:500 /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

12. 一時インスタンスから、元のインスタンスに再アタッチできるように、アタッチしたボリュームをアンマウントします。たとえば、/mnt/tempvol のボリュームをアンマウントするには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo umount /mnt/tempvol
```

13. Amazon EC2 コンソールから、書き留めたボリューム ID を持つボリュームを選択して、[Actions (アクション)]、[Detach Volume (ボリュームのデタッチ)] を選択し、[Yes, Detach (はい、デタッチする)] を選択します。ボリュームの状態が available になるまで待ちます ([Refresh] アイコンを選択しなければならない場合があります)。
14. ボリュームを選択したまま、[Actions]、[Attach Volume] の順に選択します。元のインスタンスのインスタンス ID を選択し、元のルートデバイスのアタッチについて先ほど記録したデバイス名 (例: /dev/sda1 または /dev/xvda) を指定してから、[Attach (アタッチする)] を選択します。

**Important**

元のアタッチと同じデバイス名を指定しない場合、元のインスタンスを起動することはできません。Amazon EC2 では、ルートデバイスボリュームが sda1 または /dev/xvda であると想定されるためです。

15. 元のインスタンスを選択し、[Actions] を選択して [InstanceState] を選択した後、[Start] を選択します。インスタンスが running 状態になったら、新しいキーペアのプライベートキーファイルを使用して、そのインスタンスに接続できます。

**Note**

新しいキーペアおよび対応するプライベートキーファイルの名前が元のキーペアの名前と異なる場合は、インスタンスに接続するときに新しいプライベートキーファイルの名前を必ず指定します。

16. (オプション) 一時インスタンスをそれ以上使用しない場合は、終了できます。一時インスタンスを選択して [Actions] を選択し、[Instance State] を選択して [Terminate] を選択します。

## Linux インスタンスの Amazon EC2 セキュリティグループ

セキュリティグループは、1つ以上のインスタンスのトラフィックを制御する仮想ファイアウォールとして機能します。インスタンスを起動するとき、1つ以上のセキュリティグループを指定できます。それ以外の場合は、デフォルトのセキュリティグループが使用されます。各セキュリティグループに対してルールを追加し、関連付けられたインスタンスに対するトラフィックを許可できます。セキュリティグループルールはいつでも変更できます。新しいルールは、セキュリティグループに関連付けられているインスタンスすべてに自動的に適用されます。インスタンスに到達できるトラフィックを許可するかどうかの判断では、インスタンスに関連付けられているすべてのセキュリティグループのすべてのルールが評価されます。

VPC でインスタンスを起動する場合は、その VPC 用に作成されたセキュリティグループを指定する必要があります。インスタンスを起動した後、そのセキュリティグループを変更することができます。セキュリティグループはネットワークインターフェイスに関連付けられます。インスタンスのセキュリティグループの変更は、プライマリネットワークインターフェイス (eth0) に関連付けられるセキュリティグループを変更することになります。詳細については、『Amazon VPC ユーザーガイド』の「[インスタンスのセキュリティグループを変更する](#)」を参照してください。あらゆるネットワークインターフェイスに関連付けられているセキュリティグループも変更できます。詳細については、「[セキュリティグループの変更](#) (p. 732)」を参照してください。

お客様の要件がセキュリティグループでは満たされない場合は、セキュリティグループを使用した上で、どのインスタンスでも、お客様独自のファイアウォールを維持できます。

Windows インスタンスへのトラフィックを許可する必要がある場合は、『Amazon EC2』の「[Windows インスタンスの Windows インスタンスの Amazon EC2 ユーザーガイド セキュリティグループ](#)」を参照してください。

### 目次

- [セキュリティグループのルール \(p. 912\)](#)
  - [接続追跡 \(p. 913\)](#)
- [デフォルトのセキュリティグループ \(p. 914\)](#)
- [カスタムのセキュリティグループ \(p. 915\)](#)
- [セキュリティグループを操作する \(p. 915\)](#)
  - [セキュリティグループを作成する \(p. 915\)](#)
  - [セキュリティグループについて説明する \(p. 916\)](#)
  - [セキュリティグループへのルールの追加 \(p. 916\)](#)
  - [セキュリティグループルールを更新する \(p. 918\)](#)
  - [セキュリティグループからのルールの削除 \(p. 918\)](#)
  - [セキュリティグループの削除 \(p. 919\)](#)
- [セキュリティグループのルールのリファレンス \(p. 919\)](#)
  - [ウェブサーバールール \(p. 920\)](#)
  - [データベースサーバールール \(p. 920\)](#)
  - [コンピューターからのインスタンスへの接続ルール \(p. 922\)](#)
  - [同じセキュリティグループを持つインスタンスからインスタンスに接続するためのルール \(p. 922\)](#)
  - [パス MTU 検出のルール \(p. 923\)](#)

- Ping/ICMP のルール ([p. 923](#))
- DNS サーバールール ([p. 924](#))
- Amazon EFS ルール ([p. 924](#))
- Elastic Load Balancing ルール ([p. 925](#))
- VPC ピア接続ルール ([p. 926](#))

## セキュリティグループのルール

セキュリティグループのルールは、セキュリティグループに関連付けられたインスタンスに到達することを許可されるインバウンドトラフィックと、外に向かうことを許可されるアウトバウンドトラフィックを制御します。

セキュリティグループのルールの特徴を次に示します。

- デフォルトで、セキュリティグループはすべてのアウトバウンドトラフィックを許可します。
- セキュリティグループのルールは常にパーミッションです。アクセスを拒否するルールを作成することはできません。
- セキュリティグループはステートフルです。インスタンスからリクエストを送信する場合、そのリクエストのレスポンストラフィックは、インバウンドセキュリティグループルールにかかわらず、流れることができます。つまり、VPC セキュリティグループの場合、アウトバウンドルールにかかわらず、許可されたインバウンドトラフィックは流れることができます。詳細については、「[接続追跡 \(p. 913\)](#)」を参照してください。
- ルールの追加と削除は隨時行うことができます。変更は、セキュリティグループに関連付けられたインスタンスに自動的に適用されます。

### Note

一部のルール変更の影響は、トラフィックの追跡方法によって異なる場合があります。詳細については、「[接続追跡 \(p. 913\)](#)」を参照してください。

- 複数のセキュリティグループをインスタンスに関連付けると、各セキュリティグループのルールが効率的に集約され、1 つのルールセットが作成されます。このルールセットを使用して、アクセスを許可するかどうかを判断します。

### Note

複数のセキュリティグループを 1 つのインスタンスに割り当てるため、インスタンスには数百単位のルールを適用できます。結果として、インスタンスにアクセスするときに問題が発生する可能性があります。そのため、ルールは可能な限り要約することをお勧めします。

ルールごとに、以下の点について指定します。

- プロトコル: 許可するプロトコル。最も一般的なプロトコルは、6 (TCP) 17 (UDP)、および 1 (ICMP) です。
- ポートの範囲: TCP、UDP、カスタムプロトコルの場合、許可するポートの範囲。1 つのポート番号 (22 など)、または一定範囲のポート番号 (7000-8000 など) を指定できます。
- ICMP タイプおよびコード: ICMP の場合、ICMP タイプおよびコードです。
- 送信元または送信先: トラフィックの送信元 (インバウンドルール) または送信先 (アウトバウンドルール)。これらのオプションの 1 つを指定します。
  - 個別の IPv4 アドレス。長さ /32 のプレフィックスを使用する必要があります (例: 203.0.113.1/32)。
  - 個別の IPv6 アドレス。長さ /128 のプレフィックスを使用する必要があります (例: 2001:db8:1234:1a00::123/128)。

- CIDR ブロック表記での IPv4 アドレスの範囲 (例: 203.0.113.0/24)。
- CIDR ブロック表記での IPv6 アドレスの範囲 (例: 2001:db8:1234:1a00::/64)。
- AWS サービスのプレフィックスリスト ID。たとえば、p1-1a2b3c4d。詳細については、『Amazon VPC ユーザーガイド』の「[ゲートウェイ VPC エンドポイント](#)」を参照してください。
- 別のセキュリティグループ。これにより、指定セキュリティグループに関連付けられたインスタンスから、このセキュリティグループに関連付けられたインスタンスへのアクセスが許可されます。その際、送信元セキュリティグループからこのセキュリティグループにルールが追加されることはありません。以下のセキュリティグループの 1 つを指定できます:
  - 現在のセキュリティグループ。
  - 同じ VPC の異なるセキュリティグループ
  - VPC ピア接続のピア VPC の別のセキュリティグループ。
- (オプション) 説明: 後に識別するためなどの目的で、このルールの説明を追加できます。説明の長さは最大 255 文字とすることができます。使用できる文字は、a ~ z、A ~ Z、0 ~ 9、スペース、\_.-:/()#@[]+=;{}!\$\* です。

ルールの送信元または送信先としてセキュリティグループを指定する場合、ルールはセキュリティグループと関連付けられるすべてのインスタンスに影響します。着信トラフィックは、ソースセキュリティグループに関連付けられたインスタンスのプライベート IP アドレスに基づいて許可されます (パブリック IP アドレスまたは Elastic IP アドレスは考慮されません)。IP アドレスについては、[Amazon EC2 インスタンスの IP アドレッシング \(p. 685\)](#) を参照してください。セキュリティグループルールでピア VPC のセキュリティグループを参照していて、参照先のセキュリティグループまたは VPC ピア接続を削除すると、ルールは古いとマークされます。詳細については、『Amazon VPC Peering Guide』の「[古いセキュリティグループルールの操作](#)」を参照してください。

特定のポートに複数のルールがある場合、最も許容度の大きいルールを適用します。たとえば、IP アドレス 203.0.113.1 からの TCP ポート 22 (SSH) に対するアクセスを許可するルールがあり、全員からの TCP ポート 22 に対するアクセスを許可する別のルールがある場合、全員が TCP ポート 22 にアクセスできます。

## 接続追跡

セキュリティグループは、接続追跡を使用してインスタンスを出入りするトラフィックに関する情報を追跡します。ルールはトラフィックの接続の状態に基づいて適用され、トラフィックを許可するか拒否するかが判断されます。これによって、セキュリティグループはステートフルーになり、セキュリティグループのアウトバウンドルールにかかわらず、インバウンドトラフィックへの応答がインスタンスから外に流れることができます。逆も同じです。たとえば、自宅のコンピュータからインスタンスへの ICMP ping コマンドを開始した場合、セキュリティグループのインバウンドルールが ICMP トラフィックを許可している場合は接続に関する情報 (ポート情報など) が追跡されます。ping コマンドに対するインスタンスからのレスポンストラフィックは、新しいリクエストではなく確立済みの接続として追跡され、セキュリティグループのアウトバウンドルールがアウトバウンド ICMP トラフィックを制限している場合でも、インスタンスから外に流れることができます。

すべてのトラフィックフローが追跡されるわけではありません。セキュリティグループのルールが、すべてのトラフィック (0.0.0.0/0) について TCP または UDP フローを許可していて、他の方向ですべてのポート (0~65535) のすべての応答トラフィック (0.0.0.0/0) を許可するルールがある場合、そのトラフィックフローは追跡されません。そのため、応答トラフィックは追跡情報に基づくのではなく、応答トラフィックを許可するインバウンドまたはアウトバウンドのルールに基づいて流れることができます。

次の例では、セキュリティグループに TCP および ICMP トラフィック別のインバウンドルールと、すべてのアウトバウンドトラフィックを許可するアウトバウンドルールがあります。

| インバウンドルール |         |        |
|-----------|---------|--------|
| プロトコルのタイプ | ポート番号   | 送信元 IP |
| TCP       | 22      | *      |
| ICMP      | 0~65535 | *      |
| All       | 0~65535 | *      |

|            |           |                |
|------------|-----------|----------------|
| TCP        | 22 (SSH)  | 203.0.113.1/32 |
| TCP        | 80 (HTTP) | 0.0.0.0/0      |
| ICMP       | すべて       | 0.0.0.0/0      |
| アウトバウンドルール |           |                |
| プロトコルのタイプ  | ポート番号     | 送信先 IP         |
| すべて        | すべて       | 0.0.0.0/0      |

インバウンドルールでは 203.0.113.1/32 からのトラフィックのみ許可されるため、インスタンスに出入りするポート 22 の TCP トラフィック (SSH) は追跡されますが、必ずしもすべての IP アドレス (0.0.0.0/0) が追跡されるとは限りません。インバウンドルールとアウトバウンドルールの両方ですべてのトラフィック (0.0.0.0/0) が許可されるため、インスタンスに出入りするポート 80 の TCP トラフィック (HTTP) は追跡されません。ICMP トラフィックは、ルールにかかわらず、常に追跡されます。セキュリティグループからアウトバウンドルールを削除すると、ポート 80 (HTTP) のトラフィックを含むインスタンスとの間のすべてのトラフィックが追跡されます。

追跡されている既存のトラフィックフローは、そのフローを有効にするセキュリティグループルールを削除しても中断されないことがあります。その代わり、ユーザーまたは他のホストによって少なくとも数分間 (または確立された TCP 接続の場合は最大 5 日間) 停止されると、フローは中断されます。UDP の場合、このためにフローのリモート側でのアクションを終了する必要があることがあります。追跡されていないトラフィックフローは、そのフローを有効にするルールが削除または変更されるとすぐに中断されます。たとえば、インスタンスへのすべてのインバウンド SSH トラフィックを許可するルールを削除すると、そのインスタンスへの既存の SSH 接続がすぐに削除されます。

TCP、UDP、または ICMP 以外のプロトコルの場合は、IP アドレスとプロトコル番号のみが追跡されます。インスタンスが別のホスト (ホスト B) にトラフィックを送信し、ホスト B が元のリクエストまたは応答の 600 秒以内に別のリクエストで同じタイプのトラフィックをインスタンスに対して開始する場合、インスタンスはインバウンドセキュリティグループルールとは無関係に、そのトラフィックを受け入れます。これは、トラフィックがレスポンストラフィックと見なされるためです。

VPC セキュリティグループの場合、セキュリティグループルールを削除するとそのトラフィックがすぐに中断されるようにするか、すべてのインバウンドトラフィックがファイアウォールのルールに従うようにするには、サブネットにネットワーク ACL を使用できます。ネットワーク ACL はステートレスであるため、自動的にレスポンストラフィックを許可しません。詳細については、Amazon VPC ユーザーガイドの「[ネットワーク ACL](#)」を参照してください。

## デフォルトのセキュリティグループ<sup>†</sup>

AWS アカウントには、各リージョンのデフォルト VPC のデフォルトセキュリティグループが自動的に設定されます。インスタンスを起動するときにセキュリティグループを指定しないと、そのインスタンスは VPC のデフォルトのセキュリティグループに自動的に関連付けられます。

デフォルトのセキュリティグループには `default` と名前が付けられ、AWS によって ID が割り当てられます。デフォルトのセキュリティグループごとのデフォルトルールを次に示します。

- デフォルトのセキュリティグループに関連付けられた他のインスタンスからのすべてのインバウンドトラフィックを許可します (セキュリティグループはインバウンドルール内の送信元セキュリティグループとしてそれ自体を指定します)。
- インスタンスからのすべてのアウトバウンドトラフィックが許可されます。

デフォルトのセキュリティグループのインバウンドおよびアウトバウンドルールは追加または削除できます。

デフォルトのセキュリティグループを削除することはできません。デフォルトセキュリティグループを削除しようとした場合、Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user のエラーが発生します。

## カスタムのセキュリティグループ

インスタンスでデフォルトのセキュリティグループを使用することを望まない場合、独自のセキュリティグループを作成して、インスタンスの起動時にそれらを指定することができます。複数のセキュリティグループを作成して、インスタンスが果たすさまざまな役割(たとえば、Web サーバーまたはデータベースサーバー)を反映させることができます。

セキュリティグループを作成する場合、名前と説明を指定する必要があります。セキュリティグループには、255 文字以下の名前と説明を指定できます。また、次の特徴の制限があります。

a-z, A-Z, 0-9、スペース、および \_:-/()#,@[]+=&{}!\$\*

セキュリティグループ名は、sg- で開始できません。セキュリティグループ名は VPC で一意である必要があります。

作成するセキュリティグループのデフォルトルールを次に示します。

- ・インバウンドトラフィックを許可しません
- ・すべてのアウトバウンドトラフィックを許可します

セキュリティグループを作成したら、関連するインスタンスに到達できる着信トラフィックのタイプを反映するように着信ルールを変更できます。アウトバウンドルールも変更できます。

セキュリティグループに追加できるルールのタイプの詳細については、「[セキュリティグループのルールのリファレンス \(p. 919\)](#)」を参照してください。

## セキュリティグループを操作する

Amazon EC2 コンソールを使用して、セキュリティグループとセキュリティグループルールを作成、表示、更新、削除できます。

### タスク

- ・[セキュリティグループを作成する \(p. 915\)](#)
- ・[セキュリティグループについて説明する \(p. 916\)](#)
- ・[セキュリティグループへのルールの追加 \(p. 916\)](#)
- ・[セキュリティグループルールを更新する \(p. 918\)](#)
- ・[セキュリティグループからのルールの削除 \(p. 918\)](#)
- ・[セキュリティグループの削除 \(p. 919\)](#)

## セキュリティグループを作成する

Amazon EC2 コンソールを使いカスタムセキュリティグループを作成することも可能です。セキュリティグループを作成する VPC を指定する必要があります。

コンソールを使用して新しいセキュリティグループを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] を選択します。
3. [Create Security Group] を選択します。
4. セキュリティグループの名前と説明を指定します。

5. [VPC] で使用する VPC の ID を選択します。
6. ルールの追加を開始するか、[Create] を選択して、セキュリティグループを作成できます（ルールは後で追加できます）。ルールの追加の詳細については、[セキュリティグループへのルールの追加 \(p. 916\)](#) を参照してください。

コマンドラインを使用してセキュリティグループを作成するには

- [create-security-group](#) (AWS CLI)
- [New-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

Amazon EC2 コンソールでは、既存のセキュリティグループから新しいセキュリティグループにルールをコピーできます。

コンソールを使用してセキュリティグループをコピーするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] を選択します。
3. コピーするセキュリティグループを選択します。[Actions] を選択し、[Copy to new] を選択します。
4. [Create Security Group] ダイアログが開き、既存のセキュリティグループのルールが自動入力されます。新しいセキュリティグループの名前と説明を指定します。[VPC] で使用する VPC の ID を選択します。終了したら、[Create] を選択します。

インスタンスを起動する際に、インスタンスにセキュリティグループを割り当てるすることができます。ルールを追加または削除すると、それらの変更は、そのセキュリティグループを割り当てたすべてのインスタンスに自動的に適用されます。

インスタンスを起動した後、そのセキュリティグループを変更することができます。詳細については、『Amazon VPC ユーザーガイド』の「[インスタンスのセキュリティグループを変更する](#)」を参照してください。

## セキュリティグループについて説明する

Amazon EC2 コンソールまたはコマンドラインを使用して、セキュリティグループに関する情報を表示できます。

コンソールを使用してセキュリティグループを説明するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] を選択します。
3. (オプション) フィルターリストから [VPC ID] を選択して、VPC の ID をリストから選択します。
4. セキュリティグループを選択します。[Description] タブには一般情報が、[Inbound] タブにはインバウンドルールが、[Outbound] タブにはアウトバウンドルール、[タグ] タブにはタグが表示されます。

コマンドラインを使用して 1 つまたは複数のセキュリティグループを記述するには

- [describe-security-groups](#) (AWS CLI)
- [Get-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

## セキュリティグループへのルールの追加

ルールをセキュリティグループに追加すると、セキュリティグループに関連付けられているすべてのインスタンスに新しいルールが短時間で自動的に適用されます。

特定のタイプのアクセスのためのセキュリティグループルールの選択の詳細については、「[セキュリティグループのルールのリファレンス \(p. 919\)](#)」を参照してください。

コンソールを使用してセキュリティグループにルールを追加するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Security Groups] を選択してセキュリティグループを選びます。
3. [Inbound] タブで、[Edit] を選択します。
4. ダイアログで、[Add Rule] を選択し、以下の作業を行います。
  - [Type] で、プロトコルを選択します。
  - カスタム TCP または UDP プロトコルを選択した場合は、[Port Range] でポートの範囲を指定します。
  - カスタム ICMP プロトコルを選択する場合は、[Protocol] から ICMP タイプ名を選択し、該当する場合は、[Port Range] からコード名を選択します。
  - [Source] で、以下のいずれかのオプションを選択します。
    - [Custom]: 表示されているフィールドで、CIDR 表記の IP アドレス、CIDR ブロック、または他のセキュリティグループを指定する必要があります。
    - [Anywhere]: 自動的に 0.0.0.0/0 IPv4 CIDR ブロックを追加します。このオプションでは、指定されたタイプのすべてのトラフィックがインスタンスに到達できます。これはテスト環境で短時間なら許容できますが、実稼働環境で行うのは安全ではありません。実稼働環境では、特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを限定します。

#### Note

セキュリティグループが、IPv6 に対して有効な VPC にある場合、[任意の場所] オプションによって 2 つのルールが作成されます。1 つは IPv4 トラフィック (0.0.0.0/0) 用、もう 1 つは IPv6 トラフィック (::/0) 用です。

- [My IP]: ローカルコンピュータのパブリック IPv4 アドレスを自動的に追加します。
- 説明では、オプションでこのルールの説明を指定できます。

追加できるルールのタイプの詳細については、「[セキュリティグループのルールのリファレンス \(p. 919\)](#)」を参照してください。

5. [Save] を選択します。
6. アウトバウンドルールも指定できます。[Outbound] タブで [Edit]、[Add Rule] を選択し、以下の操作を実行します。
  - [Type] で、プロトコルを選択します。
  - カスタム TCP または UDP プロトコルを選択した場合は、[Port Range] でポートの範囲を指定します。
  - カスタム ICMP プロトコルを選択する場合は、[Protocol] から ICMP タイプ名を選択し、該当する場合は、[Port Range] からコード名を選択します。
  - [Destination] で、以下のいずれかのオプションを選択します。
    - [Custom]: 表示されているフィールドで、CIDR 表記の IP アドレス、CIDR ブロック、または他のセキュリティグループを指定する必要があります。
    - [Anywhere]: 自動的に 0.0.0.0/0 IPv4 CIDR ブロックを追加します。このオプションでは、すべての IP アドレスへのアウトバウンドトラフィックが有効になります。

#### Note

セキュリティグループが、IPv6 に対して有効な VPC にある場合、[任意の場所] オプションによって 2 つのルールが作成されます。1 つは IPv4 トラフィック (0.0.0.0/0) 用、もう 1 つは IPv6 トラフィック (::/0) 用です。

- [My IP]: ローカルコンピュータの IP アドレスを自動的に追加します。

- 説明では、オプションでこのルールの説明を指定できます。
7. [Save] を選択します。

コマンドラインを使用してセキュリティグループに 1 つまたは複数の受信ルールを追加するには

- [authorize-security-group-ingress](#) (AWS CLI)
- [Grant-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

コマンドラインを使用してセキュリティグループに 1 つまたは複数の送信ルールを追加するには

- [authorize-security-group-egress](#) (AWS CLI)
- [Grant-EC2SecurityGroupEgress](#) (AWS Tools for Windows PowerShell)

## セキュリティグループルールを更新する

コンソールを使用して既存のセキュリティグループルールのプロトコル、ポート範囲、または送信元または送信先を変更すると、コンソールは既存のルールを削除し、新しいルールを追加します。

コンソールを使用してセキュリティグループルールを更新するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで、[Security Groups] を選択します。
- 更新するセキュリティグループを選択し、[Inbound Rules] を選択してインバウンドトラフィックのルールを更新するか、[Outbound Rules] を選択してアウトバウンドトラフィックのルールを更新します。
- [Edit] を選択します。必要に応じてルールエントリを変更して、[保存] を選択します。

Amazon EC2 API またはコマンドラインツールを使用して、既存のルールのプロトコル、ポート範囲、ソースまたは送信先を更新する場合は、ルールを変更することはできません。代わりに、既存のルールを削除して新しいルールを追加する必要があります。ルールの説明のみを更新するには、[update-security-group-rule-descriptions-ingress](#) および [update-security-group-rule-descriptions-egress](#) コマンドを使用できます。

コマンドラインを使用して受信セキュリティグループルールの説明を更新するには

- [update-security-group-rule-descriptions-ingress](#) (AWS CLI)
- [Update-EC2SecurityGroupRuleIngressDescription](#) (AWS Tools for Windows PowerShell)

コマンドラインを使用して送信セキュリティグループルールの説明を更新するには

- [update-security-group-rule-descriptions-egress](#) (AWS CLI)
- [Update-EC2SecurityGroupRuleEgressDescription](#) (AWS Tools for Windows PowerShell)

## セキュリティグループからのルールの削除

セキュリティグループからルールを削除すると、その変更内容が自動的にセキュリティグループに関連付けられているインスタンスに適用されます。

コンソールを使用してセキュリティグループルールを削除するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで、[Security Groups] を選択します。
3. セキュリティグループを選択します。
4. [Inbound] タブ (インバウンドルールの場合) または [Outbound] タブ (アウトバウンドルールの場合) で、[Edit] を選択します。各ルールの横にある [Delete] (クロスアイコン) を選択します。
5. [Save] を選択します。

コマンドラインを使用してセキュリティグループから 1 つまたは複数の受信ルールを削除するには

- [revoke-security-group-ingress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupIngress](#) (AWS Tools for Windows PowerShell)

コマンドラインを使用してセキュリティグループから 1 つまたは複数の送信ルールを削除するには

- [revoke-security-group-egress](#) (AWS CLI)
- [Revoke-EC2SecurityGroupEgress](#) (AWS Tools for Windows PowerShell)

## セキュリティグループの削除

インスタンスに関連付けられているセキュリティグループを削除することはできません。デフォルトセキュリティグループを削除することはできません。同じ VPC の他のセキュリティグループのルールによって参照されているセキュリティグループは削除できません。セキュリティグループが独自のいずれかのルールで参照されている場合は、セキュリティグループを削除する前に、まずルールを削除する必要があります。

コンソールを使用してセキュリティグループを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] を選択します。
3. セキュリティグループを選択して、[Actions]、[Delete Security Group] を選択します。
4. [Yes, Delete] を選択します。

コマンドラインを使用してセキュリティグループを削除するには

- [delete-security-group](#) (AWS CLI)
- [Remove-EC2SecurityGroup](#) (AWS Tools for Windows PowerShell)

## セキュリティグループのルールのリファレンス

セキュリティグループを作成し、そのセキュリティグループに関連付けられたインスタンスのロールを反映したルールを追加できます。たとえば、ウェブサーバーとして設定されたインスタンスは、インバウンド HTTP および HTTPS アクセスを許可するセキュリティグループルールを必要とし、データベースインスタンスは、MySQL 用のポート 3306 経由のアクセスなどのタイプのデータベースアクセスを許可するルールを必要とするなどです。

以下は、特定の種類のアクセスのセキュリティグループに追加できるルールの種類の例です。

例

- ウェブサーバールール (p. 920)
- データベースサーバールール (p. 920)
- コンピューターからのインスタンスへの接続ルール (p. 922)
- 同じセキュリティグループを持つインスタンスからインスタンスに接続するためのルール (p. 922)
- パス MTU 検出のルール (p. 923)
- Ping/ICMP のルール (p. 923)
- DNS サーバールール (p. 924)
- Amazon EFS ルール (p. 924)
- Elastic Load Balancing ルール (p. 925)
- VPC ピア接続ルール (p. 926)

## ウェブサーバールール

次のインバウンドルールでは、任意の IP アドレスからの HTTP および HTTPS アクセスを許可します。VPC が IPv6 に対して有効になっている場合、IPv6 アドレスからインバウンド HTTP および HTTPS トラフィックを制御するルールを追加できます。

| プロトコルのタイプ | プロトコル番号 | ポート         | 送信元 IP    | コメント                                    |
|-----------|---------|-------------|-----------|-----------------------------------------|
| TCP       | 6       | 80 (HTTP)   | 0.0.0.0/0 | 任意の IPv4 アドレスからのインバウンド HTTP アクセスを許可します  |
| TCP       | 6       | 443 (HTTPS) | 0.0.0.0/0 | 任意の IPv4 アドレスからのインバウンド HTTPS アクセスを許可します |
| TCP       | 6       | 80 (HTTP)   | ::/0      | 任意の IPv6 アドレスからのインバウンド HTTP アクセスを許可します  |
| TCP       | 6       | 443 (HTTPS) | ::/0      | 任意の IPv6 アドレスからのインバウンド HTTPS アクセスを許可します |

## データベースサーバールール

次のインバウンドルールは、インスタンスで実行中のデータベースのタイプに応じて、データベースアクセス用に追加するルールの例です。Amazon RDS インスタンスの詳細については、[Amazon RDS ユーザーガイド](#) を参照してください。

ソース IP には、次のいずれかを指定します。

- ローカルネットワークの特定の IP アドレスまたは IP アドレス範囲 (CIDR ブロック表記)
- データベースにアクセスするインスタンスのグループのセキュリティグループ ID

| プロトコルのタイプ | プロトコル番号 | ポート                 | コメント                                                             |
|-----------|---------|---------------------|------------------------------------------------------------------|
| TCP       | 6       | 1433 (MS SQL)       | Amazon RDS インスタンス上など、Microsoft SQL Server データベースにアクセスするデフォルトのポート |
| TCP       | 6       | 3306 (MySQL/Aurora) | Amazon RDS インスタンス上など、MySQL または Aurora データベースにアクセスするデフォルトのポート     |
| TCP       | 6       | 5439 (Redshift)     | Amazon Redshift クラスターデータベースにアクセスするデフォルトのポート。                     |
| TCP       | 6       | 5432 (PostgreSQL)   | Amazon RDS インスタンス上など、PostgreSQL データベースにアクセスするデフォルトのポート           |
| TCP       | 6       | 1521 (Oracle)       | Amazon RDS インスタンス上など、Oracle データベースにアクセスするデフォルトのポート               |

オプションで、データベースサーバーからのアウトバウンドトラフィックを制限できます。たとえば、ソフトウェアの更新のためにインターネットへのアクセスを許可し、その他すべての種類のトラフィックを制限することができます。最初に、すべてのアウトバウンドトラフィックを許可するデフォルトのアウトバウンドルールを削除する必要があります。

| プロトコルのタイプ | プロトコル番号 | ポート         | 送信先 IP    | コメント                                                      |
|-----------|---------|-------------|-----------|-----------------------------------------------------------|
| TCP       | 6       | 80 (HTTP)   | 0.0.0.0/0 | 任意の IPv4 アドレスへのアウトバウンド HTTP アクセスを許可します                    |
| TCP       | 6       | 443 (HTTPS) | 0.0.0.0/0 | 任意の IPv4 アドレスへのアウトバウンド HTTPS アクセスを許可します                   |
| TCP       | 6       | 80 (HTTP)   | ::/0      | (IPv6 が有効な VPC のみ) 任意の IPv6 アドレスへのアウトバウンド HTTP アクセスを許可します |

| プロトコルのタイプ | プロトコル番号 | ポート         | 送信先 IP | コメント                                                       |
|-----------|---------|-------------|--------|------------------------------------------------------------|
| TCP       | 6       | 443 (HTTPS) | ::/0   | (IPv6 が有効な VPC のみ)、任意の IPv6 アドレスへのアウトバウンド HTTPS アクセスを許可します |

## コンピューターからのインスタンスへの接続ルール

インスタンスに接続するには、セキュリティグループに SSH アクセス (Linux インスタンスの場合) または RDP アクセス (Windows インスタンスの場合) を許可するインバウンドルールが必要です。

| プロトコルのタイプ | プロトコル番号 | ポート        | 送信元 IP                                                                                                                             |
|-----------|---------|------------|------------------------------------------------------------------------------------------------------------------------------------|
| TCP       | 6       | 22 (SSH)   | ローカルコンピュータのパブリック IPv4 アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。VPC が IPv6 に対して有効で、インスタンスに IPv6 アドレスがある場合、IPv6 アドレスまたは範囲を入力できます。 |
| TCP       | 6       | 3389 (RDP) | ローカルコンピュータのパブリック IPv4 アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。VPC が IPv6 に対して有効で、インスタンスに IPv6 アドレスがある場合、IPv6 アドレスまたは範囲を入力できます。 |

## 同じセキュリティグループを持つインスタンスからインスタンスに接続するためのルール

同じセキュリティグループに関連付けられたインスタンスが相互に通信できるようにするには、そのためのルールを明示的に追加する必要があります。

次の表は、関連付けられたインスタンスの相互通信を可能にするセキュリティグループのインバウンドルールを示します。このルールでは、すべてのタイプのトライフィックが許可されます。

| プロトコルのタイプ | プロトコル番号 | ポート     | 送信元 IP         |
|-----------|---------|---------|----------------|
| -1(すべて)   | -1(すべて) | -1(すべて) | セキュリティグループの ID |

## パス MTU 検出のルール

パス MTU は、送信側ホストと受信側ホスト間のパスでサポートされている最大のパケットサイズです。ホストが受信側ホストの MTU よりも大きなパケット、またはデバイスの MTU よりも大きなパケットをパスに沿って送信する場合、受信側ホストは次の ICMP メッセージを返します。

```
Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
```

インスタンスがこのメッセージを受信し、パケットが削除されないようにするには、インバウンドセキュリティグループのルールに ICMP ルールを追加する必要があります。

| プロトコルのタイプ | プロトコル番号 | ICMP タイプ       | ICMP コード                            | 送信元 IP                  |
|-----------|---------|----------------|-------------------------------------|-------------------------|
| ICMP      | 1       | 3(送信先に到達できません) | 4(フラグメント化が必要で、"フラグメント化しない"が設定されました) | インスタンスと通信するホストの IP アドレス |

## Ping/ICMP のルール

ping コマンドは、ICMP トラフィックの一種です。インスタンスに ping を実行するには、次のインバウンド ICMP ルールを追加する必要があります。

| プロトコルのタイプ | プロトコル番号 | ICMP タイプ | ICMP コード | 送信元 IP                                                         |
|-----------|---------|----------|----------|----------------------------------------------------------------|
| ICMP      | 1       | 8(Echo)  | 該当なし     | コンピュータのパブリック IPv4 アドレス、またはローカルネットワークの IPv4 アドレス範囲(CIDR ブロック表記) |

ping6 コマンドを使用してインスタンスの IPv6 アドレスに ping を実行するには、次のインバウンド ICMPv6 ルールを追加する必要があります。

| プロトコルのタイプ | プロトコル番号 | ICMP タイプ  | ICMP コード | 送信元 IP                                  |
|-----------|---------|-----------|----------|-----------------------------------------|
| ICMPv6    | 58      | 128(Echo) | 0        | コンピュータの IPv6 アドレス、またはローカルネットワークの IPv6 ア |

| プロトコルのタイプ | プロトコル番号 | ICMP タイプ | ICMP コード | 送信元 IP              |
|-----------|---------|----------|----------|---------------------|
|           |         |          |          | ドレス範囲 (CIDR ブロック表記) |

## DNS サーバールール

DNS サーバーとして EC2 インスタンスをセットアップした場合、TCP および UDP のトラフィックがポート 53 経由で DNS サーバーに到達できるようにする必要があります。

ソース IP には、次のいずれかを指定します。

- ネットワークの IP アドレスまたは IP アドレス範囲 (CIDR ブロック表記)
- ネットワークで、DNS サーバーにアクセスする必要がある一連のインスタンスのセキュリティグループの ID

| プロトコルのタイプ | プロトコル番号 | ポート |
|-----------|---------|-----|
| TCP       | 6       | 53  |
| UDP       | 17      | 53  |

## Amazon EFS ルール

Amazon EC2 インスタンスで Amazon EFS ファイルシステムを使用している場合、Amazon EFS マウントターゲットに関するセキュリティグループは、NFS プロトコル経由のトラフィックを許可する必要があります。

| プロトコルのタイプ | プロトコル番号 | ポート        | 送信元 IP          | コメント                                                              |
|-----------|---------|------------|-----------------|-------------------------------------------------------------------|
| TCP       | 6       | 2049 (NFS) | セキュリティグループの ID. | このセキュリティグループに関連付けられたリソース (マウントターゲットを含む) からのインバウンド NFS アクセスを許可します。 |

Amazon EC2 インスタンスに Amazon EFS ファイルシステムをマウントするには、インスタンスに接続する必要があります。したがって、インスタンスに関連付けられているセキュリティグループには、ローカルコンピュータまたはローカルネットワークからのインバウンド SSH を許可するルールが必要です。

| プロトコルのタイプ | プロトコル番号 | ポート      | 送信元 IP                    | コメント                    |
|-----------|---------|----------|---------------------------|-------------------------|
| TCP       | 6       | 22 (SSH) | ローカルコンピュータの IP アドレス範囲、または | ローカルコンピュータからのインバウンド SSH |

| プロトコルのタイプ | プロトコル番号 | ポート | 送信元 IP                            | コメント        |
|-----------|---------|-----|-----------------------------------|-------------|
|           |         |     | はネットワークの IP アドレス範囲 (CIDR ブロック表記)。 | アクセスを許可します。 |

## Elastic Load Balancing ルール

ロードバランサーを使用している場合、ロードバランサーに関連付けられたセキュリティグループには、インスタンスやターゲットとの通信を許可するルールが必要です。

| インバウンド    |         |               |                                                                                         |                                            |
|-----------|---------|---------------|-----------------------------------------------------------------------------------------|--------------------------------------------|
| プロトコルのタイプ | プロトコル番号 | ポート           | 送信元 IP                                                                                  | 注                                          |
| TCP       | 6       | リスナーポート       | インターネット向けロードバランサーの場合: 0.0.0.0/0 (すべての IPv4 アドレス)<br>内部ロードバランサーの場合: VPC の IPv4 CIDR ブロック | ロードバランサーのリスナーポートでインバウンドトラフィックを許可します。       |
| アウトバウンド   |         |               |                                                                                         |                                            |
| プロトコルのタイプ | プロトコル番号 | ポート           | 送信先 IP                                                                                  | 注                                          |
| TCP       | 6       | インスタンスリスナーポート | インスタンスセキュリティグループの ID                                                                    | インスタンスリスナーポートでインスタンスへのアウトバウンドトラフィックを許可します。 |
| TCP       | 6       | ヘルスチェックポート    | インスタンスセキュリティグループの ID                                                                    | ヘルスチェックポートでインスタンスへのアウトバウンドトラフィックを許可します。    |

インスタンスのセキュリティグループルールは、リスナーポートとヘルスチェックポートの両方でインスタンスと通信することをロードバランサーに許可する必要があります。

| インバウンド    |         |     |        |   |
|-----------|---------|-----|--------|---|
| プロトコルのタイプ | プロトコル番号 | ポート | 送信元 IP | 注 |

|     |   |              |                         |                                       |
|-----|---|--------------|-------------------------|---------------------------------------|
| TCP | 6 | インスタンスリストポート | ロードバランサーのセキュリティグループの ID | インスタンスリストポートでロードバランサーからのトラフィックを許可します。 |
| TCP | 6 | ヘルスチェックポート   | ロードバランサーのセキュリティグループの ID | ヘルスチェックポートでロードバランサーからのトラフィックを許可します。   |

詳細については、『クラシックロードバランサー 用ユーザーガイド』の「Classic Load Balancer ターゲットグループのヘルスチェック」または『Application Load Balancer 用ユーザーガイド』の「Application Load Balancer のヘルスチェックの設定」を参照してください。

## VPC ピア接続ルール

VPC セキュリティグループのインバウンドルールまたはアウトバウンドルールを更新して、ピアリング接続 VPC のセキュリティグループを参照できます。これにより、トラフィックはピア VPC の参照されるセキュリティグループに関連付けられたインスタンスに出入りできます。VPC ピア接続のセキュリティグループを設定する方法の詳細については、「[セキュリティグループの更新によるピア VPC グループの参照](#)」を参照してください。

## Amazon EC2での更新管理

弊社は、EC2 インスタンスのオペレーティングシステムやアプリケーションに対するパッチ適用やそれらの更新およびセキュリティ確保を定期的に行うよう推奨しています。[AWS Systems Managerパッチマネージャー](#)を使うと、オペレーティングシステムとアプリケーションの双方に関するセキュリティ関連更新のインストールプロセスを自動化できます。代わりに、アプリケーションベンダーが提供している、自動更新サービスまたは推奨更新インストールプロセスを使用することもできます。

## Amazon EC2のコンプライアンス検証

サードパーティの監査者は、複数のAWSコンプライアンスプログラムの一環として、Amazon EC2のセキュリティとコンプライアンスを評価します。このプログラムには、SOC、PCI、FedRAMP、HIPAAなどがあります。

特定のコンプライアンスプログラムの範囲内にある AWS のサービスのリストについては、「[コンプライアンスプログラムによる AWS 対象範囲内のサービス](#)」を参照してください。一般的な情報については、「[AWS コンプライアンスプログラム](#)」を参照してください。

サードパーティの監査レポートをダウンロードするには、AWS Artifact を使用します。詳細については、[AWS Artifactにおけるレポートのダウンロード](#)を参照してください。

Amazon EC2を使用する際にお客様が果たすべきコンプライアンス責任は、データの機密度、所属企業のコンプライアンス目標および準拠法規制に応じて決まります。AWSは、コンプライアンスに役立つ以下のリソースを提供します。

- セキュリティおよびコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、データの機密度、所属企業のコンプライアンス目標および準拠法規制に応じて決まります。AWSは、コンプライアンスに役立つ以下のリソースを提供します。

- [HIPAA のセキュリティとコンプライアンスに関するホワイトペーパーを作成する](#) – このホワイトペーパーでは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法について説明します。
- [AWS コンプライアンスのリソース](#) – このワークブックおよびガイドのコレクションは、お客様の業界や場所に適用される場合があります。
- [AWS Config Developer Guide](#) – AWS Config の「ルールでのリソースの評価」では、リソース設定が社内のプラクティス、業界のガイドライン、規制にどの程度適合しているかを評価します。
- [AWS Security Hub](#) – この AWS サービスでは、AWS 内のセキュリティ状態を包括的に表示しており、セキュリティ業界の標準およびベストプラクティスへの準拠を確認するのに役立ちます。

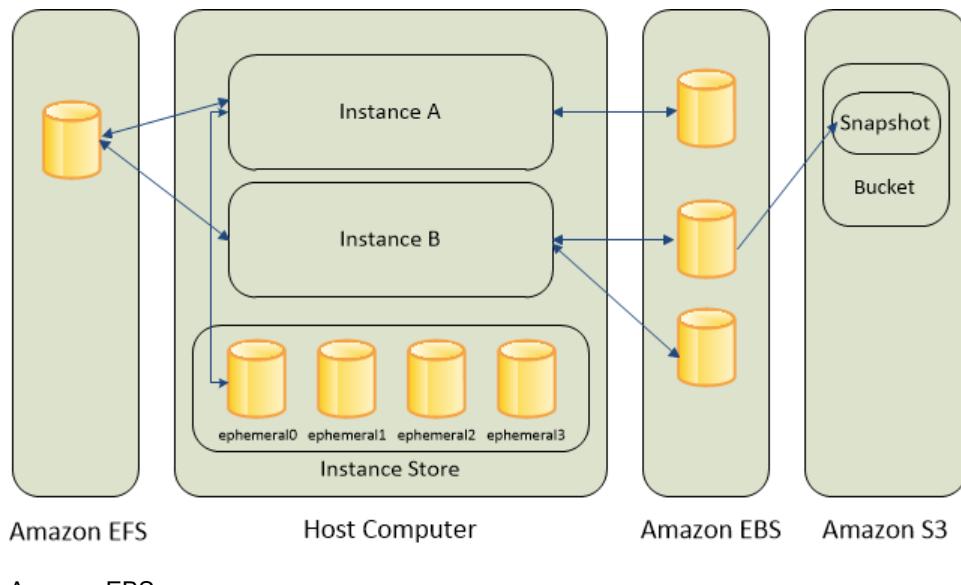
# ストレージ

Amazon EC2 にはインスタンスを格納するための、柔軟で使いやすく、コスト効率の良いデータストレージオプションが用意されています。各オプションは独自のパフォーマンスと耐久性を備えています。これらのストレージオプションは、要件に応じて個別に使用することも、組み合わせて使用することもできます。

このセクションを読むことで、Amazon EC2 がサポートするデータストレージオプションを使用して、特定の要件に対応する方法を十分に理解できるようになるはずです。たとえば、次のようなストレージオプションがあります。

- [Amazon Elastic Block Store \(p. 929\)](#)
- [Amazon EC2 インスタンスストア \(p. 1076\)](#)
- [Amazon Elastic File System \(Amazon EFS\) \(p. 1091\)](#)
- [Amazon Simple Storage Service \(Amazon S3\) \(p. 1095\)](#)

各ストレージオプションとインスタンスの関係を下の図に示します。



Amazon EBS

Amazon EBS は、実行中のインスタンスにアタッチできる、堅牢なブロックレベルのストレージボリュームを提供します。この Amazon EBS は、細かな更新を頻繁に行う必要があるデータを対象とした主要ストレージデバイスとして使用できます。たとえば、インスタンスでデータベースを実行するときに、Amazon EBS をストレージオプションとして使用することをお勧めします。

EBS ボリュームは、1つのインスタンスにアタッチできる、未加工、未フォーマットの外部ブロックデバイスのように動作します。これらのボリュームは、インスタンスの運用状況から独立した永続性を持ちます。インスタンスにアタッチした後の EBS ボリュームは、他の物理ハードドライブと同じように使用できます。前の図に示したように、複数のボリュームをインスタンスにアタッチできます。1つのインスタンスから EBS ボリュームをデタッチし、別のインスタンスにアタッチできます。インスタンスにアタッチされているボリュームの設定は動的に変更できます。EBS ボリュームは、Amazon EBS 暗号化機能を使って、暗号化されたボリュームとして作成することもできます。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

データのバックアップコピーを保持するには、EBS ボリュームのスナップショットを作成して Amazon S3 に保存します。スナップショットから EBS ボリュームを作成して、別のインスタンスにアタッチすることもできます。詳細については、「[Amazon Elastic Block Store \(p. 929\)](#)」を参照してください。

## Amazon EC2 インスタンスストア

多くのインスタンスは、ホストコンピュータに物理的にアタッチされたディスクからストレージにアクセスできます。このディスクストレージは、インスタンスストアと呼ばれます。インスタンスストアは、インスタンス用のブロックレベルの一時ストレージを提供します。インスタンスストアボリュームのデータは、関連するインスタンスの存続中にのみ保持されます。インスタンスを停止または終了すると、インスタンスストアボリュームのすべてのデータが失われます。詳細については、「[Amazon EC2 インスタンスストア \(p. 1076\)](#)」を参照してください。

## Amazon EFS ファイルシステム

Amazon EFS は、Amazon EC2 と併用できるスケーラブルなファイルストレージを提供します。EFS ファイルシステムを作成し、ファイルシステムをマウントするためにインスタンスを設定できます。複数のインスタンスで実行している作業負荷やアプリケーションの一般的なデータソースとして EFS ファイルシステムを使用できます。詳細については、「[Amazon Elastic File System \(Amazon EFS\) \(p. 1091\)](#)」を参照してください。

## Amazon S3

Amazon S3 により、低コストで信頼性に優れたデータストレージインフラストラクチャが実現します。ウェブスケールのコンピューティングをさらに簡単に実行するように設計されており、Amazon EC2 内から、またはウェブ上のどこからでも、いつでも必要な量だけデータを格納および取得できます。たとえば、Amazon S3 を使用してデータとアプリケーションのバックアップコピーを保存することができます。Amazon EC2 は、Amazon S3 を使用して EBS スナップショットとインスタンスストアバックアップ AMI を格納します。詳細については、「[Amazon Simple Storage Service \(Amazon S3\) \(p. 1095\)](#)」を参照してください。

## ストレージの追加

AMI からインスタンスを起動するたびに、そのインスタンス用のルートストレージデバイスが作成されます。ルートストレージデバイスには、インスタンスの起動に必要な情報すべてが含まれます。AMI を作成するとき、またはブロックデバイスマッピングを使用してインスタンスを起動するときに、ルートデバイスボリュームの他にストレージボリュームを指定できます。詳細については、「[ブロックデバイスマッピング \(p. 1100\)](#)」を参照してください。

実行中のインスタンスに EBS ボリュームをアタッチすることもできます。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。

# Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) は、EC2 インスタンスで使用するためのブロックレベルのストレージボリュームを提供します。EBS ボリュームの動作は、未初期化のブロックデバイスに似ています。これらのボリュームは、デバイスとしてインスタンスにマウントできます。同じインスタンスに複数のボリュームをマウントしたり、一度に複数のインスタンスにボリュームをマウントすることができます。これらのボリューム上にファイルシステムを構築できます。また、これらのボリュームをブロックデバイスを使用する場合と同じ方法で使用できます(ハードドライブとして使用するなど)。インスタンスにアタッチされているボリュームの設定は動的に変更できます。

EBS ボリュームは、同じアベイラビリティーゾーンにある実行中のどのインスタンスにもアタッチできる、可用性と信頼性に優れたストレージボリュームです。EC2 インスタンスにアタッチされた EBS ボリュームは、インスタンスの運用状況から、独立した永続性を保つストレージボリュームとして表示されます。Amazon EBS については、お客様が利用された分のみのお支払いとなります。Amazon EBS の料金の詳細については、[Amazon Elastic Block Store ページ](#)の概算費用のセクションを参照してください。

AWS アカウントによって規定される制限内であれば、同一のインスタンスに複数のボリュームをアタッチできます。アカウントには、使用できる EBS ボリュームの数と、利用可能なストレージの総量に制限があります。これらの制限と、制限を緩和するよう要求する方法の詳細については、「[Request to Increase the Amazon EBS Volume Limit](#)」を参照してください。

Amazon EBS は、データにすばやくアクセスする必要があり、長期永続性が必要な場合に推奨されます。EBS ボリュームは、ファイルシステムの主要ストレージやデータベースとしての使用に特に適しています。また、細かい更新が必要なアプリケーションや、ブロックレベルの未初期化のストレージを使用する必要があるアプリケーションにも適しています。Amazon EBS は、ランダムな読み取り/書き込みに依存するデータベーススタイルのアプリケーションと、長時間の連続読み取り/書き込みを実行するスループットが高いアプリケーションの両方に適しています。

## 目次

- [Amazon EBS の機能 \(p. 930\)](#)
- [Amazon EBS ボリューム \(p. 931\)](#)
- [Amazon EBS スナップショット \(p. 970\)](#)
- [Amazon EBS のデータサービス \(p. 1003\)](#)
- [Linux インスタンスの Amazon EBS および NVMe \(p. 1027\)](#)
- [Amazon EBS – 最適化インスタンス \(p. 1031\)](#)
- [Linux インスタンスの Amazon EBS ボリュームのパフォーマンス \(p. 1044\)](#)
- [Amazon EBS の Amazon CloudWatch メトリクス \(p. 1060\)](#)
- [Amazon EBS での Amazon CloudWatch Events \(p. 1066\)](#)

## Amazon EBS の機能

- EBS ボリュームは、特定のアベイラビリティーゾーンで作成され、そのアベイラビリティーゾーン内のインスタンスにアタッチできます。アベイラビリティーゾーンの外部でボリュームを使用できるようにするには、スナップショットを作成し、そのスナップショットをそのリージョン内の新しいボリュームに復元できます。スナップショットをその他のリージョンにコピーしてから、そのリージョン内の新しいボリュームに復元できるので、地理的な拡大やデータセンターの移行、災害復旧など、複数の AWS リージョンをより容易に活用することができます。
- Amazon EBS には、ボリュームタイプとして汎用 SSD (gp2)、プロビジョンド IOPS SSD (io1)、スループット最適化 HDD (st1)、および Cold HDD (sc1) が用意されています。ボリュームタイプ別のパフォーマンスとユースケースの概要は以下のとおりです。
  - 汎用 SSD ボリュームは、1 GiBあたり 3 IOPS のベースパフォーマンスを提供し、長時間にわたり 3,000 IOPS までバーストできます。これらのボリュームは、さまざまなユースケース（ブートボリューム、小規模および中規模のデータベース、開発環境やテスト環境など）に適しています。詳細については、「[汎用 SSD \(gp2\) ボリューム \(p. 935\)](#)」を参照してください。
  - プロビジョンド IOPS SSD ボリュームは、最大 64,000 IOPS と 1,000 MiB/秒のスループットをサポートします。これにより、EC2 インスタンスあたり何万単位の IOPS まで拡張できます。詳細については、「[プロビジョンド IOPS SSD \(io1\) ボリューム \(p. 938\)](#)」を参照してください。
  - スループット最適化 HDD ボリュームは、IOPS ではなくスループットでパフォーマンスを示す、低コストの磁気ストレージとして使用できます。これらボリュームは、Amazon EMR、ETL、データウェアハウス、ログ処理など、サイズの大きなシーケンシャルワークロードに適しています。詳細については、「[スループット最適化 HDD \(st1\) ボリューム \(p. 939\)](#)」を参照してください。
  - Cold HDD ボリュームは、IOPS ではなくスループットでパフォーマンスを示す、低コストの磁気ストレージとして使用できます。これらのボリュームは、サイズのコールドデータのシーケンシャルワークロードに適しています。データへのアクセス頻度が低くて、コストを削減したい場合は、これらのボリュームを安価なブロックストレージとして使用できます。詳細については、「[Cold HDD \(sc1\) ボリューム \(p. 941\)](#)」を参照してください。
- EBS ボリュームを暗号化されたボリュームとして作成することで、規制/監査されるデータとアプリケーションの保管時の広範な暗号化要件に対応できます。暗号化の対象となる EBS ボリュームを作成し、サポートされているインスタンスタイプに関連付けると、そのボリュームに保管されるデータ、そのボリュームとのディスク I/O、そのボリュームから作成されたスナップショットは、すべて暗号化されます。暗号化は EC2 インスタンスをホストするサーバーで行われ、EC2 インスタンスから EBS ストレージに転送されるデータが暗号化されます。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

- EBS ボリュームの特定の時点におけるスナップショットを作成し、Amazon S3 に保管できます。長期的な耐久性を実現するために、スナップショットはデータを保護します。また、スナップショットは、新しい EBS ボリュームの開始点として使用できます。1つのスナップショットからインスタンス化できるボリュームの数に制限はありません。これらのスナップショットは、複数の AWS リージョンにわたってコピーできます。詳細については、「[Amazon EBS スナップショット \(p. 970\)](#)」を参照してください。
- AWS マネジメントコンソールを介して、帯域幅、スループット、レイテンシー、平均キュー長などのパフォーマンスマトリクスを使用できます。Amazon CloudWatch のこれらのメトリクスを使用してボリュームのパフォーマンスをモニタリングすると、アプリケーションに対して十分なパフォーマンスを提供できているか、無駄なリソースにコストを費やしていないかを確認できます。詳細については、「[Linux インスタンスの Amazon EBS ボリュームのパフォーマンス \(p. 1044\)](#)」を参照してください。

## Amazon EBS ボリューム

Amazon EBS ボリュームは、1つ以上のインスタンスにアタッチできる、耐久性に優れたブロックレベルのストレージボリュームです。EBS ボリュームは、インスタンス用のシステムドライブ、データベースアプリケーションのストレージなど、頻繁に更新する必要があるデータのプライマリストレージとして使用できます。連続ディスクスキャンを実行するスループットが高いアプリケーションにも使用できます。EBS ボリュームは、EC2 インスタンスの運用状況から独立した永続性を持ちます。

複数の EBS ボリュームを1つのインスタンスにアタッチできます。ボリュームとそのアタッチ先インスタンスは同じアベイラビリティーゾーンに存在している必要があります。インスタンスにアタッチした後のボリュームは、他の物理ハードドライブと同じように使用できます。EBS ボリュームには柔軟性があります。現行世代のインスタンスタイプにアタッチされた現行世代のボリュームの場合、サイズの拡張、プロビジョンド IOPS の容量の変更、実稼働ボリュームのボリュームタイプの変更を動的に行うことができます。

Amazon EBS には、汎用 SSD (gp2)、プロビジョンド IOPS SSD (io1)、スループット最適化 HDD (st1)、Cold HDD (sc1)、およびマグネティック (standard、前世代のタイプ) というボリュームタイプが用意されています。この2つはパフォーマンス特性と料金が異なるため、アプリケーションのニーズに応じてストレージのパフォーマンスとコストを調整できます。詳細については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。

### コンテンツ

- [EBS ボリュームを使用する利点 \(p. 931\)](#)
- [Amazon EBS ボリュームの種類 \(p. 933\)](#)
- [EBS ボリュームのサイズと設定の制限 \(p. 946\)](#)
- [Amazon EBS ボリュームの作成 \(p. 949\)](#)
- [スナップショットからの Amazon EBS ボリュームの復元 \(p. 950\)](#)
- [インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)
- [Amazon EBS マルチアタッチを使用した複数のインスタンスへのボリュームのアタッチ \(p. 953\)](#)
- [Linux で Amazon EBS ボリュームを使用できるようにする \(p. 956\)](#)
- [Amazon EBS ボリュームに関する情報を表示する \(p. 959\)](#)
- [ボリュームのステータスのモニタリング \(p. 960\)](#)
- [インスタンスからの Amazon EBS ボリュームのデタッチ \(p. 967\)](#)
- [Amazon EBS ボリュームの削除 \(p. 969\)](#)

## EBS ボリュームを使用する利点

EBS ボリュームには、インスタンストアボリュームにはない利点があります。

## データの可用性

EBS ボリュームを作成すると、そのボリュームは同じアベイラビリティゾーン内で自動的にレプリケートされます。これは、1つのハードウェアコンポーネントの障害が原因でデータが失われることを防ぐためです。EBS ボリュームは、同じアベイラビリティゾーン内の任意の EC2 インスタンスにアタッチできます。アタッチしたボリュームは、ハードドライブや他の物理デバイスと同じようなネイティブブロックとして表示されます。その時点で、インスタンスはローカルドライブと同じようにボリュームとやり取りできます。このインスタンスに接続し、ext3 などのファイルシステムを使用して EBS ボリュームをフォーマットして、アプリケーションをインストールできます。

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

io1 EBS ボリュームは、最大 16 個の Nitro ベースのインスタンスにアタッチできます。詳細については、「[Amazon EBS マルチアタッチを使用した複数のインスタンスへのボリュームのアタッチ \(p. 953\)](#)」を参照してください。それ以外の場合は、EBS ボリュームを 1 つのインスタンスにアタッチできます。

EBS ボリュームのモニタリングデータは無料で取得できます (EBS-backed インスタンスのルートデバイスボリュームのデータも含まれます)。メトリクスのモニタリングの詳細については、「[Amazon EBS の Amazon CloudWatch メトリクス \(p. 1060\)](#)」を参照してください。ボリュームのステータスの追跡の詳細については、「[Amazon EBS での Amazon CloudWatch Events \(p. 1066\)](#)」を参照してください。

## データの永続性

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

EC2 コンソール上で使用する EBS ボリュームを設定するときに [Delete on Termination (終了時に削除)] チェックボックスをオフにした場合、実行中のインスタンスにアタッチされている EBS ボリュームを、インスタンスの終了時にデータがそのままの状態でインスタンスから自動的にデタッチすることができます。デタッチされたボリュームは新しいインスタンスに再アタッチできるので、迅速な復旧が可能です。[Delete on Termination (終了時に削除)] のチェックボックスがオンの場合、ボリュームは EC2 インスタンスの終了後に削除されます。EBS-backed インスタンスを使用している場合は、アタッチしたボリュームに格納されているデータに影響を与えることなく、インスタンスを停止および再起動できます。ボリュームは停止/起動のサイクルを通じてアタッチされたままです。これにより、必要なときに処理リソースとストレージリソースを使用するだけで、ボリュームでのデータの処理と格納を永続的に実行できるようになります。データは、ボリュームを明示的に削除するまでボリュームに保持されます。削除した EBS ボリュームが使用していた物理的なブロックストレージは、別のアカウントに割り当てられる前に、ゼロで上書きされます。機密データを扱っている場合は、手動によるデータの暗号化や、Amazon EBS 暗号化で保護されているボリュームへのデータの格納を検討してください。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

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

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

## データの暗号化

簡素化されたデータの暗号化を使用するには、Amazon EBS 暗号化機能を使用して、暗号化の対象となる EBS ボリュームを作成できます。暗号化は、すべての EBS ボリュームタイプでサポートされています。暗号化された EBS ボリュームを使用して、規制/監査されたデータとアプリケーションに関連した、保管されるデータの幅広い暗号化要件に対応することができます。Amazon EBS 暗号化では、256 ビット

の Advanced Encryption Standard (AES-256) アルゴリズムと、Amazon に管理されたキーインフラストラクチャが使用されます。暗号化は EC2 インスタンスをホストするサーバーで行われ、EC2 インスタンスから Amazon EBS ストレージに転送されるデータが暗号化されます。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

Amazon EBS 暗号化は、暗号化されたボリュームと、暗号化されたボリュームから作成されるスナップショットを作成するときに AWS Key Management Service (AWS KMS) マスターキーを使用します。暗号化された EBS ボリュームをリージョン内に初めて作成するときは、デフォルトのマスターキーが自動的に作成されます。AWS KMS を使用して別途作成したカスタマーマスターキー (CMK) を選択しない限り、このキーが Amazon EBS 暗号化で使用されます。独自の CMK を作成すると、アクセスコントロールを作成、使い回し、無効化、定義できるほか、データの保護に使用される暗号化キーを監査できるなど、より高い柔軟性が得られます。詳細については、「[AWS Key Management Service Developer Guide](#)」を参照してください。

## スナップショット

Amazon EBS は、Amazon S3 ボリュームのスナップショット（バックアップ）を作成し、ボリューム内のデータのコピーを EBS に書き込む機能を備えています。そこで、データは複数のアベイラビリティゾーンに冗長的に保存されます。スナップショットを作成するために、対象のボリュームが実行中のインスタンスにアタッチされている必要があります。ボリュームにデータを書き込み続けながら、そのボリュームのスナップショットを定期的に作成して、新しいボリュームのベースラインとして使用できます。このスナップショットは、新しい EBS ボリュームを複数作成したり、アベイラビリティゾーン間でボリュームを移動したりするときに使用できます。暗号化された EBS ボリュームのスナップショットは自動的に暗号化されます。

スナップショットから新規ボリュームを作成する場合、このボリュームはスナップショット作成時における元のボリュームの正確なコピーになります。暗号化されたスナップショットから復元された EBS ボリュームは、自動的に暗号化されます。別のアベイラビリティゾーンを指定し、この機能を使用してそのゾーンにボリュームを複製することもできます。スナップショットは特定の AWS アカウントと共有するか、一般公開することができます。スナップショットを作成すると、Amazon S3 でボリュームの合計サイズに基づいて、料金がかかります。ボリュームのその後のスナップショットについては、ボリュームの元のサイズを超える追加データ分にのみ料金がかかります。

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

ボリュームとスナップショットを分類および管理しやすくするため、任意のメタデータでタグ付けすることができます。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。

ボリュームを自動的にバックアップするには、[Amazon Data Lifecycle Manager \(p. 992\)](#) または [AWS Backup](#) を使用できます。

## 柔軟性

EBS ボリュームは、実稼働環境での設定変更をサポートします。サービスを中断せずに、ボリュームタイプ、ボリュームサイズ、IOPS 容量を変更できます。詳細については、「[Amazon EBS Elastic Volumes \(p. 1003\)](#)」を参照してください。

## Amazon EBS ボリュームの種類

Amazon EBS では以下のボリュームタイプを提供しており、これらはパフォーマンス特性と料金が異なるため、アプリケーションのニーズに応じてストレージのパフォーマンスとコストを調整できます。これらのボリュームタイプは、次の 2 つのカテゴリに分類されます。

- I/O サイズの小さい頻繁な読み取り/書き込み操作を含むトランザクションワークロード用に最適化された SSD-Backed ボリューム。主要なパフォーマンス属性は IOPS です。
- 大きなストリーミングワークロード用に最適化された HDD-Backed ボリューム。パフォーマンスの測定単位としては、IOPS よりスループット (MiB/秒単位で計測) の方が適しています。

インスタンスの構成、I/O 特性、ワークロードのデマンドなど、EBS ボリュームのパフォーマンスに影響を与える可能性がある要因は複数存在します。EBS ボリュームを最大限活用するための詳細については、「[Linux インスタンスの Amazon EBS ボリュームのパフォーマンス \(p. 1044\)](#)」を参照してください。

料金の詳細については、「[Amazon EBS 料金表](#)」を参照してください。

## ボリューム特性

次の表は、各ボリュームタイプのユースケースとパフォーマンス特性をまとめたものです。デフォルトのボリュームタイプは 汎用 SSD (gp2) です。

|                   | ソリッドステートドライブ (SSD)                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                     | ハードディスクドライブ (HDD)                                                                                                                                                           |                                                                                                                                               |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| ボリュームタイプ          | 汎用 SSD (gp2)                                                                                                                                                               | プロビジョンド IOPS SSD (io1)                                                                                                                                                                                                                                                                                                                              | スループット最適化 HDD (st1)                                                                                                                                                         | Cold HDD (sc1)                                                                                                                                |
| 説明                | さまざまなワークロードに適した、価格とパフォーマンスのバランスが取れている汎用 SSD ボリューム                                                                                                                          | ミッションクリティカルな低レイテンシーまたは高スループットワークロードに適した、最高パフォーマンスの SSD ボリューム                                                                                                                                                                                                                                                                                        | 高いスループットを必要とするアクセス頻度の高いワークロード向けの低コストの HDD ボリューム                                                                                                                             | アクセス頻度の低いワークロード用に設計された低コストの HDD ボリューム                                                                                                         |
| ユースケース            | <ul style="list-style-type: none"> <li>• ほとんどのワークロードに推奨される</li> <li>• システムブートボリューム</li> <li>• 仮想デスクトップ</li> <li>• 低レイテンシーのインタラクティブなアプリケーション</li> <li>• 開発・テスト環境</li> </ul> | <ul style="list-style-type: none"> <li>• 持続的な IOPS パフォーマンス、またはボリュームあたり 16,000 IOPS または 250 MiB/秒以上のスループットを必要とする重要なビジネスアプリケーション</li> <li>• などの大規模なデータベースワークロード <ul style="list-style-type: none"> <li>• MongoDB</li> <li>• Cassandra</li> <li>• Microsoft SQL Server</li> <li>• MySQL</li> <li>• PostgreSQL</li> <li>• Oracle</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• 低コストで安定した高速スループットを必要とするストリーミングワークロード</li> <li>• ビッグデータ</li> <li>• データウェアハウス</li> <li>• ログ処理</li> <li>• ブートボリュームには使用できない</li> </ul> | <ul style="list-style-type: none"> <li>• アクセス頻度の低い大量データ用のスループット指向ストレージ</li> <li>• 低いストレージコストが重視されるシナリオ</li> <li>• ブートボリュームには使用できない</li> </ul> |
| API 名             | gp2                                                                                                                                                                        | io1                                                                                                                                                                                                                                                                                                                                                 | st1                                                                                                                                                                         | sc1                                                                                                                                           |
| ボリュームサイズ          | 1GiB - 16TiB                                                                                                                                                               | 4 GiB ~ 16 TiB                                                                                                                                                                                                                                                                                                                                      | 500 GiB ~ 16 TiB                                                                                                                                                            | 500 GiB ~ 16 TiB                                                                                                                              |
| ボリュームあたりの最大 IOPS  | 16,000 (16 KiB I/O) <sup>*</sup>                                                                                                                                           | 64,000 (16 KiB I/O) †                                                                                                                                                                                                                                                                                                                               | 500 (1 MiB I/O)                                                                                                                                                             | 250 (1 MiB I/O)                                                                                                                               |
| ボリュームあたりの最大スループット | MiB/秒 *                                                                                                                                                                    | 1,000 MiB/秒 †                                                                                                                                                                                                                                                                                                                                       | 500 MiB/秒                                                                                                                                                                   | 250 MiB/秒                                                                                                                                     |

|                       | ソリッドステートドライブ (SSD) |            | ハードディスクドライブ (HDD) |            |
|-----------------------|--------------------|------------|-------------------|------------|
| インスタンスあたりの最大 IOPS ††  | 80,000             | 80,000     | 80,000            | 80,000     |
| インスタンスあたりの最大スループット †† | 2,375 MB/秒         | 2,375 MB/秒 | 2,375 MB/秒        | 2,375 MB/秒 |
| 主要なパフォーマンス属性          | IOPS               | IOPS       | MiB/秒             | MiB/秒      |

\* スループットの制限は、ボリュームサイズに応じて 128 MiB/秒～250 MiB/秒です。170 GiB より小さいボリュームは、最大スループット 128 MiB/秒を提供します。170 GiB より大きく 334 GiB より小さいボリュームは、バーストクレジットを利用する場合、最大スループット 250 MiB/秒を提供します。334 GiB 以上のボリュームは、バーストクレジットに関係なく、250 MiB/秒を提供します。ボリュームを変更しない限り、古い gp2 ボリュームはパフォーマンスが完全にはならないことがあります。詳細については、「[Amazon EBS Elastic Volumes \(p. 1003\)](#)」を参照してください。

† 最大 IOPS とスループットは、32,000 IOPS を超える[Nitro ベースのインスタンス \(p. 187\)](#)プロビジョニングされた場合にのみ保証されます。他のインスタンスは、最大 32,000 IOPS および 500 MiB/秒を保証します。ボリュームを変更しない限り、古い io1 ボリュームはパフォーマンスが完全にはならないことがあります。詳細については、「[Amazon EBS Elastic Volumes \(p. 1003\)](#)」を参照してください。

†† このスループットを達成するには、[EBS 最適化 \(p. 1031\)](#)をサポートするインスタンスが必要です。

### 旧世代のボリュームタイプ

次の表は、旧世代の EBS ボリュームタイプを示しています。旧世代のボリュームより高いパフォーマンスまたはパフォーマンスの安定性が必要であれば、汎用 SSD (gp2) など現行のボリュームタイプの使用を検討するようお勧めします。詳細については、「[Amazon EBS の旧世代ボリューム](#)」を参照してください。

| ハードディスクドライブ (HDD)  |                      |
|--------------------|----------------------|
| ボリュームタイプ           | マグネットィック             |
| ユースケース             | データへのアクセス頻度が低いワークロード |
| API 名              | standard             |
| ボリュームサイズ           | 1 GiB ~ 1 TiB        |
| ボリュームあたりの最大 IOPS   | 40 ~ 200             |
| ボリュームあたりの最大スループット  | 40 ~ 90 MiB/秒        |
| インスタンスあたりの最大 IOPS  | 80,000               |
| インスタンスあたりの最大スループット | 1,750 MiB/秒          |
| 主要なパフォーマンス属性       | IOPS                 |

### 汎用 SSD (gp2) ボリューム

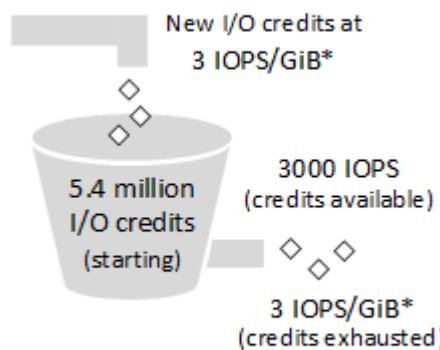
汎用 SSD (gp2) ボリュームは、さまざまなワークロードに対応できるコスト効率の高いストレージとして使用できます。これらのボリュームでは、レイテンシーは 1 枠台のミリ秒であり、長時間 3,000 IOPS に

バーストできます。最小 100 IOPS (33.33 GiB 以下) から最大 16,000 IOPS (5,334 GiB 以上) まで、ベースラインパフォーマンスは 3 IOPS/GiB (ボリュームサイズ) の割合で線形に拡大します。AWS では、プロビジョニングされたパフォーマンスをほぼ間違なく実現する gp2 ボリュームを設計しています。gp2 ボリュームのサイズ範囲は、1 GiB ~ 16 TiB です。

### I/O クレジットおよびバーストパフォーマンス

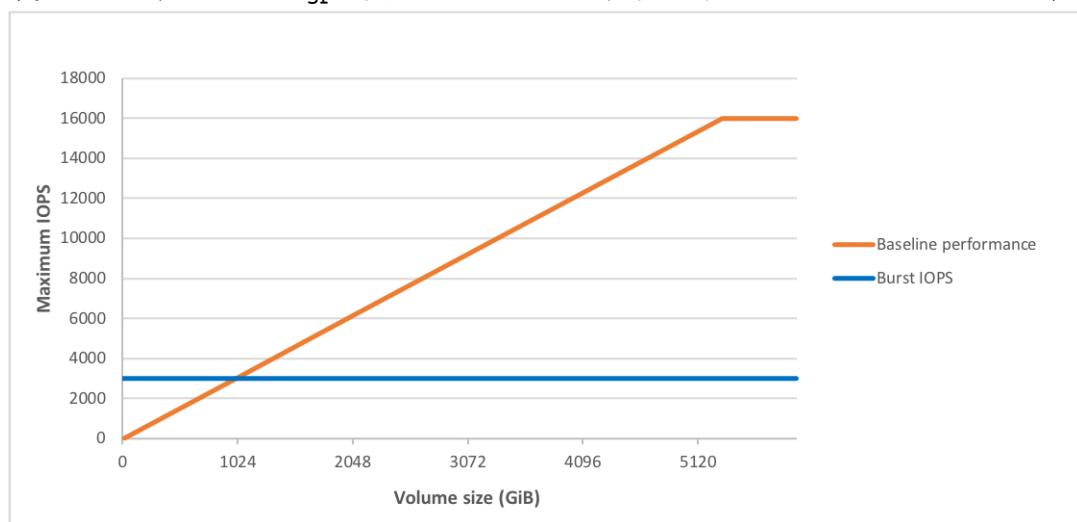
gp2 ボリュームのパフォーマンスにはボリュームサイズが反映されます。ボリュームサイズによって、ボリュームのベースラインパフォーマンスレベルや I/O クレジットを取得する速さが決まります。ボリュームサイズが大きいほどベースラインパフォーマンスレベルが高くなり、I/O クレジットの取得速度も速くなります。I/O クレジットとは、ベースラインパフォーマンスでは不十分な場合、大量の I/O をバーストする際に gp2 ボリュームで使用できる帯域幅を表します。ボリュームが I/O に対して保持しているクレジットが多いほど、長い時間ベースラインパフォーマンスレベルを超えたバーストが可能で、より高いパフォーマンスが必要な場合パフォーマンスも向上します。次の図は、gp2 のバーストバケット動作を示しています。

#### GP2 burst bucket



\* Scaling linearly between minimum 100 IOPS and maximum 16,000 IOPS

各ボリュームは、初期 I/O クレジットバランス (540 万 I/O クレジット) を受け取ります。これは、30 分間で 3,000 IOPS という最大バーストパフォーマンスを持続するには十分な数のクレジットです。この初期クレジットバランスは、ブートボリュームでの高速な初期起動サイクル、および他のアプリケーションでの優れたブートストラップエクスペリエンスを実現するために設計されました。ボリュームは、ボリュームサイズの 1 GiBあたり 3 IOPS というベースラインパフォーマンスレートで、I/O クレジットを取得します。たとえば、100 GiB の gp2 ボリュームではベースラインパフォーマンスは 300 IOPS になります。



ベースラインパフォーマンスの I/O レベルよりも高いレベルが必要となる場合は、ボリュームはクレジットバランスの I/O クレジットを利用して、必要なパフォーマンスレベル (最大 3,000 IOPS) までバーストします。ボリュームで使用される I/O クレジットが毎秒取得される I/O クレジットよりも少ない場合、未使用的 I/O クレジットは I/O クレジットバランスに追加されます。ボリュームの最大 I/O クレジットバランスは、初期クレジットバランス (540 万 I/O クレジット) と同じです。

ボリュームのベースラインパフォーマンスが最大バーストパフォーマンスより高い場合、I/O クレジットは消費されません。ボリュームが [Nitro ベースのインスタンス \(p. 187\)](#) にアタッチされている場合、バーストバランスは報告されません。Nitro ベースのインスタンス以外の場合、バーストバランスは 100% で報告されます。

ボリュームのバースト期間は、ボリュームのサイズ、必要なバースト IOPS、およびバーストが開始された時点のクレジットバランスによって異なります。これを次の式で示します。

$$\text{Burst duration} = \frac{\text{(Credit balance)}}{(\text{Burst IOPS}) - 3(\text{Volume size in GiB})}$$

次の表に、いくつかのボリュームサイズとボリュームに関連するベースラインパフォーマンス (I/O クレジットを取得するレート) を示します。また、最大のパフォーマンスレベルである 3,000 IOPS (完全なクレジットバランスで開始された時点のレベル) でのバースト期間、および空のクレジットバランスをボリュームが再補充する際にかかる秒数も示します。

| ボリュームサイズ (GiB)             | ベースラインパフォーマンス (IOPS) | 持続的な 3,000 IOPS のバースト期間 (秒数) | IO がない場合に空のクレジットバランスを満たすまでの秒数 |
|----------------------------|----------------------|------------------------------|-------------------------------|
| 1                          | 100                  | 1802                         | 54,000                        |
| 100                        | 300                  | 2,000                        | 18,000                        |
| 250                        | 750                  | 2,400                        | 7,200                         |
| 334 (最大スループットの最小サイズ)       | 1002                 | 2703                         | 5389                          |
| 500                        | 1,500                | 3,600                        | 3,600                         |
| 750                        | 2,250                | 7,200                        | 2,400                         |
| 1,000                      | 3,000                | 該当なし*                        | 該当なし*                         |
| 5,334 (最大 IOPS の最小サイズ)     | 16,000               | 該当なし*                        | 該当なし*                         |
| 16,384 (16 TiB、最大ボリュームサイズ) | 16,000               | 該当なし*                        | 該当なし*                         |

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

I/O クレジットバランスが空になつたらどうなりますか。

gp2 ボリュームが I/O クレジットバランスをすべて使用している場合、ボリュームの最大 IOPS パフォーマンスはベースライン IOPS パフォーマンスレベルにとどまり (ボリュームがクレジットを取得するレート)、ボリュームの最大スループットはベースライン IOPS と最大 I/O サイズをかけ合わせた数に減少します。スループットは 250 MiB/秒を超えることはできません。I/O 需要がベースラインレベル未満になり、未使用的クレジットが I/O クレジットバランスに追加されると、ボリュームの最大 IOPS パフォーマンスはベースラインを再度上回ります。たとえば、空のクレジットバランスがある 100 GiB gp2 ボリューム

は、ベースラインパフォーマンスが 300 IOPS で、スループット制限は 75 MiB/秒です (1 秒あたり 300 I/O オペレーション \* I/O オペレーションあたり 256 KiB = 75 MiB/秒)。ボリュームが大きくなると、ベースラインパフォーマンスが高くなり、クレジットバランスがより速く補充されるようになります。IOPS の測定方法の詳細については、「[I/O の特性とモニタリング \(p. 1047\)](#)」を参照してください。

ボリュームのパフォーマンスがベースラインレベルに頻繁に制限されること (空の I/O クレジットバランスが原因) が確認される場合は、より大きな (ベースラインパフォーマンスレベルが高い) gp2 ボリュームの使用を考慮するか、16,000 IOPS を超える持続的な IOPS パフォーマンスが必要となるワークロードに適した io1 ボリュームに切り替えることを考慮してください。

CloudWatch メトリクスとアラームを使用してバーストバケットバランスをモニタリングする方法については、「[gp2、st1、および sc1 ボリュームのバーストバケットバランスをモニタリングする \(p. 946\)](#)」を参照してください。

## スループットパフォーマンス

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

```
Throughput in MiB/s = ((Volume size in GiB) × (IOPS per GiB) × (I/O size in KiB))
```

V = ボリュームサイズ、I = I/O サイズ、R = I/O 料金、T = スループット、とします。これにより以下が簡単になります。

```
T = VIR
```

最大スループットを実現する最小ボリュームサイズは次のように求めることができます。

$$\begin{aligned} V &= \frac{T}{I \cdot R} \\ &= \frac{250 \text{ MiB/s}}{(256 \text{ KiB})(3 \text{ IOPS/GiB})} \\ &= \frac{[(250)(2^{20})(\text{Bytes})]/\text{s}}{(256)(2^{10})(\text{Bytes})([3 \text{ IOP/s}]/[(2^{30})(\text{Bytes})])} \\ &= \frac{(250)(2^{20})(2^{30})(\text{Bytes})}{(256)(2^{10})(3)} \\ &= 357,913,941,333 \text{ Bytes} \\ &= 333\# \text{ GiB (334 GiB in practice because volumes are provisioned in whole gibibytes)} \end{aligned}$$

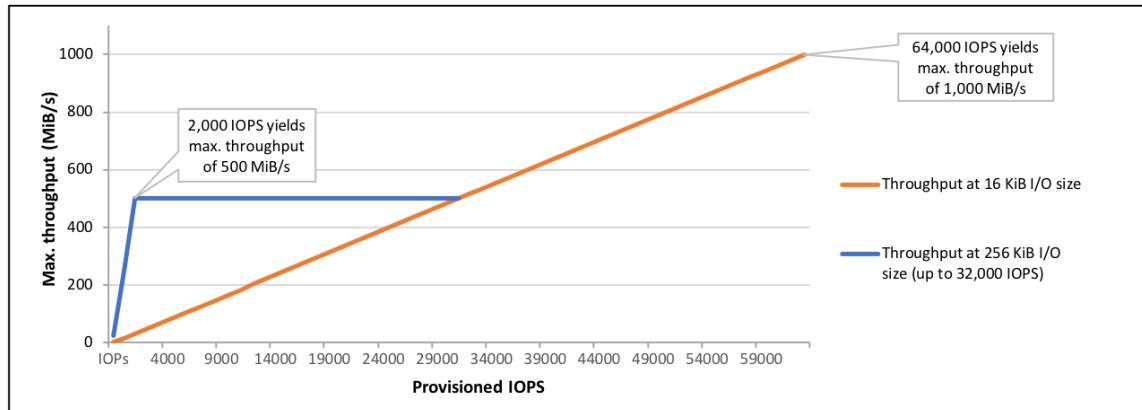
## プロビジョンド IOPS SSD (io1) ボリューム

プロビジョンド IOPS SSD (io1) ボリュームは、ランダムアクセス I/O スループットにおけるストレージパフォーマンスと整合性が重要な、I/O 集約型ワークロード (特にデータベースワークロード) のニーズを満たすように設計されています。バケットとクレジットのモデルを使用してパフォーマンスを計算する gp2 とは異なり、io1 ボリュームでは、ボリュームの作成時に一定の IOPS レートを指定できます。Amazon EBS は、プロビジョンド IOPS のパフォーマンスを 99.9% 提供します。

io1 ボリュームのサイズは、4 GiB ~ 16 TiB になります。[Nitro ベースのインスタンス \(p. 187\)](#) のインスタンスでは 1 つのボリュームにつき 100 IOPS から最大 64,000 IOPS まで、他のインスタンスでは最

最大 32,000 までプロビジョニングできます。リクエストされたボリュームサイズに対するプロビジョンド IOPS の最大割合 (GiB 単位) は 50:1 です。たとえば、100 GiB のボリュームは最大 5,000 IOPS でプロビジョニングできます。サポートされるインスタンスタイプで、サイズが 1,280 GiB 以上のボリュームでは、最大 64,000 IOPS ( $50 \times 1,280 \text{ GiB} = 64,000$ ) までプロビジョニングできます。

最大 32,000 IOPS でプロビジョニングされた io1 ボリュームは、最大 256 KiB の I/O サイズをサポートし、最大 500 MiB/s のスループットを生み出します。最大の I/O サイズでは、ピークのスループットが 2,000 IOPS に達します。32,000 を超える IOPS (最高で上限の 64,000 IOPS) でプロビジョニングされたボリュームは、最大 16 KiB の I/O サイズをサポートし、最大 1,000 MiB/s のスループットを生み出します。次のグラフは、これらのパフォーマンスの特長を示しています。



発生する I/O あたりのレイテンシーは、プロビジョニングされる IOPS とワークロードのパターンによって異なります。最適な I/O レイテンシーを実現するためには、IOPS と GiB の比率を 2:1 より大きくしてプロビジョニングすることをお勧めします。たとえば、2,000 IOPS のボリュームは 1,000 GiB よりも小さくします。

#### Note

2012 年以前に作成された一部の AWS アカウントでは、us-west-1 または ap-northeast-1 でプロビジョンド IOPS SSD (io1) ボリュームをサポートしていないアベイラビリティーゾーンにアクセスできる可能性があります。これらのリージョンの 1 つに io1 ボリュームを作成できない場合 (またはブロックデバイスマッピングに io1 ボリュームのあるインスタンスを起動できない場合) は、リージョンの別のアベイラビリティーゾーンを試します。アベイラビリティーゾーンが io1 ボリュームをサポートするかどうかは、4 GiB の io1 ボリュームをそのゾーンに作成することで確認できます。

## スループット最適化 HDD (st1) ボリューム

スループット最適化 HDD (st1) ボリュームは、IOPS ではなくスループットでパフォーマンスを示す、低コストの磁気ストレージに使用できます。このボリュームタイプは、Amazon EMR、ETL、データウェアハウス、ログ処理など、サイズの大きなシーケンシャルワークロードに適しています。ブート可能な st1 ボリュームはサポートされていません。

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

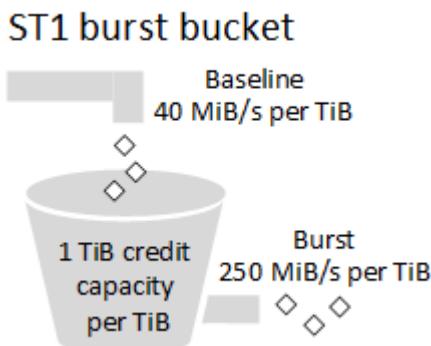
このボリュームタイプは、サイズの大きなシーケンシャル I/O が含まれるワークロードに適しており、サイズの小さなランダム I/O を実行するワークロードのお客様には、gp2 の使用をお勧めします。詳細については、「[HDD に対する読み取り/書き込みサイズが小さい場合の非効率性 \(p. 945\)](#)」を参照してください。

## スループットクレジットとバーストパフォーマンス

gp2 と同様、st1 でもパフォーマンスのためにバーストバケットモデルが使用されます。ボリュームのベースラインスループット (ボリュームのスループットクレジットが蓄積されるレート) は、ボリュームサ

イズによって決まります。ボリュームのバーストスループット（クレジットがある場合に可能な消費レート）もボリュームサイズによって決まります。ボリュームが大きいほど、ベースラインとバーストスループットの値も大きくなります。また、ボリュームのクレジットが多いほど、バーストレベルでドライブ I/O に使用できる時間が長くなります。

次の図は、st1 のバーストバケット動作を示しています。



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

$$\text{(Volume size)} \times (\text{Credit accumulation rate per TiB}) = \text{Throughput}$$

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

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

ボリュームサイズが 0.5 ~ 16 TiB の場合、ベースラインスループットの範囲は 20 ~ 500 MiB/秒（上限）です。次に示すように、この上限には 12.5 TiB で到達します。

$$12.5 \text{ TiB} \times \frac{40 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

バーストスループットの範囲は、125 MiB/秒 ~ 500 MiB/秒（上限）です。次に示すように、この上限には 2 TiB で到達します。

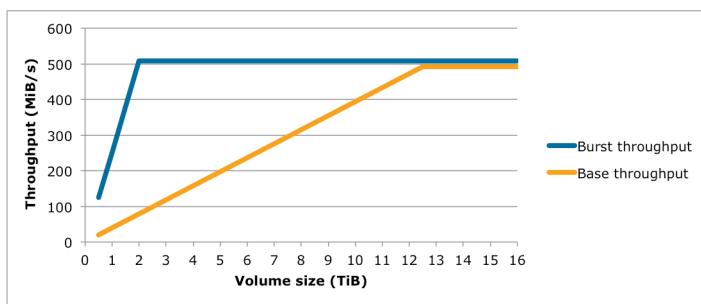
$$2 \text{ TiB} \times \frac{250 \text{ MiB/s}}{1 \text{ TiB}} = 500 \text{ MiB/s}$$

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

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

| ボリュームサイズ (TiB) | ST1 ベーススループット (MiB/秒) | ST1 バーストスループット (MiB/秒) |
|----------------|-----------------------|------------------------|
| 3              | 120                   | 500                    |
| 4              | 160                   | 500                    |
| 5              | 200                   | 500                    |
| 6              | 240                   | 500                    |
| 7              | 280                   | 500                    |
| 8              | 320                   | 500                    |
| 9              | 360                   | 500                    |
| 10             | 400                   | 500                    |
| 11             | 440                   | 500                    |
| 12             | 480                   | 500                    |
| 12.5           | 500                   | 500                    |
| 13             | 500                   | 500                    |
| 14             | 500                   | 500                    |
| 15             | 500                   | 500                    |
| 16             | 500                   | 500                    |

次の図は、テーブルの値をグラフで示したものです。



#### Note

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

CloudWatch メトリクスとアラームを使用してバーストバケットバランスをモニタリングする方法については、「[gp2、st1、および sc1 ボリュームのバーストバケットバランスをモニタリングする \(p. 946\)](#)」を参照してください。

## Cold HDD (sc1) ボリューム

Cold HDD (sc1) ボリュームは、IOPS ではなくスループットでパフォーマンスを示す、低コストの磁気ストレージに使用できます。sc1 は、st1 よりスループット制限が低く、サイズの大きなコールドデータのシーケンシャルワークロードに適しています。データへのアクセス頻度が低く、コストの削減が必要です。

る場合は、低コストなブロックストレージとして sc1 を使用できます。ポート可能な sc1 ボリュームはサポートされていません。

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

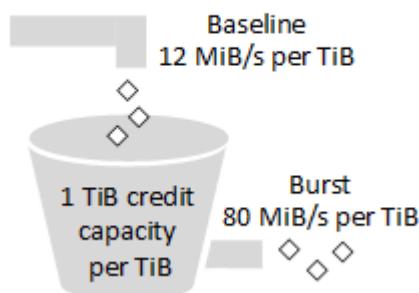
#### Note

このボリュームタイプは、サイズの大きなシーケンシャル I/O が含まれるワークロードに適しており、サイズの小さなランダム I/O を実行するワークロードのお客様には、gp2 の使用をお勧めします。詳細については、「[HDD に対する読み取り/書き込みサイズが小さい場合の非効率性 \(p. 945\)](#)」を参照してください。

### スループットクレジットとバーストパフォーマンス

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

#### SC1 burst bucket



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

$$(\text{Volume size}) \times (\text{Credit accumulation rate per TiB}) = \text{Throughput}$$

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

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

ボリュームサイズが 0.5 ~ 16 TiB の場合、ベースラインスループットの範囲は 6 MiB/秒 ~ 192 MiB/秒 (最大値) です。次に示すように、この最大値には 16 TiB で到達します。

$$16 \text{ TiB} \times \frac{12 \text{ MiB/s}}{1 \text{ TiB}} = 192 \text{ MiB/s}$$

バーストループットの範囲は、40 MiB/秒 ~ 250 MiB/秒 (上限) です。次に示すように、この上限には 3.125 TiB で到達します。

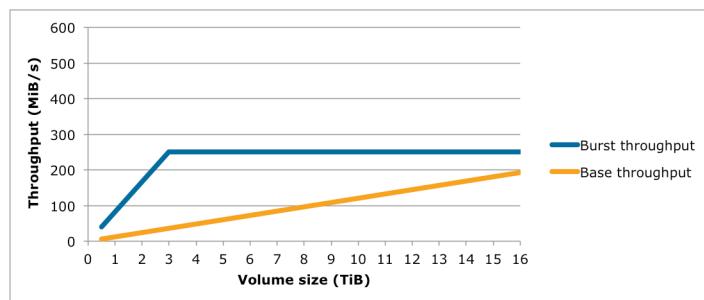
$$80 \text{ MiB/s}$$

3.125 TiB x ----- = 250 MiB/s  
1 TiB

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

| ボリュームサイズ (TiB) | SC1 ベーススループット (MiB/秒) | SC1 バーストスループット (MiB/秒) |
|----------------|-----------------------|------------------------|
| 0.5            | 6                     | 40                     |
| 1              | 12                    | 80                     |
| 2              | 24                    | 160                    |
| 3              | 36                    | 240                    |
| 3.125          | 37.5                  | 250                    |
| 4              | 48                    | 250                    |
| 5              | 60                    | 250                    |
| 6              | 72                    | 250                    |
| 7              | 84                    | 250                    |
| 8              | 96                    | 250                    |
| 9              | 108                   | 250                    |
| 10             | 120                   | 250                    |
| 11             | 132                   | 250                    |
| 12             | 144                   | 250                    |
| 13             | 156                   | 250                    |
| 14             | 168                   | 250                    |
| 15             | 180                   | 250                    |
| 16             | 192                   | 250                    |

次の図は、テーブルの値をグラフで示したものです。



#### Note

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

CloudWatch メトリクスとアラームを使用してバーストバケットバランスをモニタリングする方法については、「[gp2、st1、および sc1 ボリュームのバーストバケットバランスをモニタリングする \(p. 946\)](#)」を参照してください。

## マグネティック (standard)

マグネティック ボリュームは磁気ドライブを利用しています。データにシーケンシャルアクセスするワークロードや、小さなボリュームサイズで低コストのストレージが必要となるシナリオに最適です。これらのボリュームは、平均約 100 IOPS を実現し、バースト能力は最大約数百 IOPS です。ボリュームのサイズは 1 GiB~1 TiB です。

### Note

マグネティック は、旧世代のボリュームタイプです。新しいアプリケーションには、いずれかの新しいボリュームタイプの使用をお勧めします。詳細については、「[旧世代ボリューム](#)」を参照してください。

CloudWatch メトリクスとアラームを使用してバーストバケットバランスをモニタリングする方法については、「[gp2、st1、および sc1 ボリュームのバーストバケットバランスをモニタリングする \(p. 946\)](#)」を参照してください。

## HDD ボリュームを使用するときのパフォーマンスに関する考慮事項

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

### スループット最適化 HDD と Cold HDD

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

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

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

スキャン時間は、一般的にこの式で示します。

|             |            |             |
|-------------|------------|-------------|
| Volume size | -----      | = Scan time |
|             | Throughput |             |

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

|                                                                            |                  |                                                 |
|----------------------------------------------------------------------------|------------------|-------------------------------------------------|
| 5 TiB                                                                      | 5 TiB            | ----- = ----- = 10,486 s = 2.91 hours (optimal) |
| 500 MiB/s                                                                  | 0.00047684 TiB/s |                                                 |
| 2.91 hours + ----- = 3.27 hours (minimum expected)                         |                  |                                                 |
| (0.90)(0.99) <-- From expected performance of 90% of burst 99% of the time |                  |                                                 |

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

```
5 TiB
----- = 20972 s = 5.83 hours (optimal)
0.000238418 TiB/s

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

| ボリュームサイズ (TiB) | ST1 のスキャン時間、バーストを含む (時間)* | SC1 のスキャン時間、バーストを含む (時間)* |
|----------------|---------------------------|---------------------------|
| 1              | 1.17                      | 3.64                      |
| 2              | 1.17                      | 3.64                      |
| 3              | 1.75                      | 3.64                      |
| 4              | 2.33                      | 4.66                      |
| 5              | 2.91                      | 5.83                      |
| 6              | 3.50                      | 6.99                      |
| 7              | 4.08                      | 8.16                      |
| 8              | 4.66                      | 9.32                      |
| 9              | 5.24                      | 10.49                     |
| 10             | 5.83                      | 11.65                     |
| 11             | 6.41                      | 12.82                     |
| 12             | 6.99                      | 13.98                     |
| 13             | 7.57                      | 15.15                     |
| 14             | 8.16                      | 16.31                     |
| 15             | 8.74                      | 17.48                     |
| 16             | 9.32                      | 18.64                     |

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

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

#### HDD に対する読み取り/書き込みサイズが小さい場合の非効率性

st1 ボリュームおよび sc1 ボリュームのパフォーマンスマネージャーは、シーケンシャル I/O 用に最適化され、高スループットのワークロードに適しています。多様な IOPS およびスループットのワークロードに対し

て許容範囲のパフォーマンスを提供しますが、サイズの小さなランダム I/O のワークロードには向いていません。

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

### インスタンスごとのスループット制限

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

- ボリュームのスループット制限
- インスタンスのスループット制限

ネットワークボトルネックを回避するには、すべての Amazon EBS ボリュームで、EBS 最適化 EC2 インスタンスを選択することをお勧めします。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

### gp2、st1、および sc1 ボリュームのバーストバケットバランスをモニタリングする

gp2、st1、および sc1 ボリュームのバーストバケットレベルをモニタリングするには、Amazon CloudWatch の EBS BurstBalance メトリクスを使用します。このメトリクスは、バーストバケットに残っている I/O クレジット (gp2 用) またはスループットクレジット (st1 および sc1 用) の割合を示しています。BurstBalance メトリクスおよび I/O に関連するその他のメトリクスの詳細については、「[I/O の特性とモニタリング \(p. 1047\)](#)」を参照してください。CloudWatch では、BurstBalance 値が特定のレベルに達した場合に通知するアラームを設定できます。詳細については、「[Amazon CloudWatch アラームの作成](#)」を参照してください。

## EBS ボリュームのサイズと設定の制限

Amazon EBS ボリュームのサイズ変更は、ブロックデータストレージの物理とアリスマティック、オペレーティングシステム (OS) とファイルシステムの設計者の実装に関する意思決定によって制限されます。AWS では、サービスの信頼性を保護するためにボリュームサイズの制約を追加しています。

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

### コンテンツ

- ストレージキャパシティー (p. 946)
- サービスの制約事項 (p. 947)
- パーティションスキーム (p. 947)
- データブロックサイズ (p. 948)

## ストレージキャパシティー

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

| パーティションスキーム | アドレス可能な最大ブロック | 理論的な最大サイズ (ブロック × ブロックサイズ) | Ext4 に実装される最大サイズ* | XFS に実装される最大サイズ** | NTFS に実装される最大サイズ | EBS による最大サポート数 |
|-------------|---------------|----------------------------|-------------------|-------------------|------------------|----------------|
| MBR         | $2^{32}$      | 2 TiB                      | 2 TiB             | 2 TiB             | 2 TiB            | 2 TiB          |

| パーティションスキーム | アドレス可能な最大ブロック | 理論的な最大サイズ(ブロック×ブロックサイズ) | Ext4 に実装される最大サイズ*                               | XFS に実装される最大サイズ**      | NTFS に実装される最大サイズ | EBS による最大サポート数 |
|-------------|---------------|-------------------------|-------------------------------------------------|------------------------|------------------|----------------|
| GPT         | $2^{64}$      | 64 ZiB                  | 1 EiB = $1024^2$ TiB<br>(RHEL7 で認証されている 50 TiB) | 500 TiB<br>(RHEL7 で認証) | 256 TiB          | 16 TiB         |

\* [https://ext4.wiki.kernel.org/index.php/Ext4\\_Howto](https://ext4.wiki.kernel.org/index.php/Ext4_Howto) および <https://access.redhat.com/solutions/1532>

\*\* <https://access.redhat.com/solutions/1532>

## サービスの制約事項

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

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

- EBS では現在、最大 16 TiB のボリュームサイズがサポートされています。つまり、最大 16 TiB の EBS ボリュームを作成することはできますが、OS でそのキャパシティーが認識されるかどうかは、その独自設計の特性と、ボリュームのパーティションスキームによって異なります。
- Linux ブートボリュームは、MBR または GPT パーティションスキームを使用する場合があります。MBR は最大 2047 GiB(2 TiB ~ 1 GiB)までのブートボリュームをサポートします。GRUB 2 を備えた GPT は、2 TiB 以上のブートボリュームをサポートします。Linux AMI が MBR を使用する場合、ブートボリュームは 2047 GiB に制限されますが、非ブートボリュームにはこの制限はありません。詳細については、「[Linux で Amazon EBS ボリュームを使用できるようにする\(p. 956\)](#)」を参照してください。

## パーティションスキーム

他にも影響がある中で、このパーティションスキームは、単一ボリュームで一意にアドレス解決できる論理データブロックの数を決定します。詳細については、「[データブロックサイズ\(p. 948\)](#)」を参照してください。使用中の一般的なパーティションスキームは、マスターブートレコード(MBR)と GUID パーティションテーブル(GPT)です。これらのパーティションスキームの重要な違いは次のようにまとめることができます。

### MBR

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

$$(2^{32} - 1) \times \text{Block size} = \text{Number of addressable blocks}$$

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

$$(2^{32} - 1) \times 512 \text{ bytes} = 2 \text{ TiB} - 512 \text{ bytes}$$

この MBR ボリュームの 2 TiB の制限を増やすための回避策は、一般的に広く普及していません。したがって、AWS で大きな値が示されていたとしても、Linux や Windows で、2 TiB 以上の MBR が検知されることはありません。

## GPT

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

$$(2^{64} - 1) \times \text{Block size} = \text{Number of addressable blocks}$$

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

$$\begin{aligned} & (2^{64} - 1) \times 4,096 \text{ bytes} \\ &= 2^{64} \times 4,096 \text{ bytes} - 1 \times 4,096 \text{ bytes} \\ &= 2^{64} \times 2^{12} \text{ bytes} - 4,096 \text{ bytes} \\ &= 2^{70} \times 2^6 \text{ bytes} - 4,096 \text{ bytes} \\ &= 64 \text{ Zib} - 4,096 \text{ bytes} \end{aligned}$$

実際のコンピュータシステムでは、この理論上の最大値のような大きな値はサポートされていません。実装されたファイルシステムのサイズは現在、ext4 では 50 TiB、NTFS では 256 TiB に制限されており、いずれも、AWS 指定の 16 TiB 制限を超過しています。

## データブロックサイズ

現代のハードドライブ上のデータストレージは、論理ブロックアドレスや、オペレーティングシステムで基礎となるハードウェアをほとんど把握することなく論理ブロック内のデータを読み書きできる抽象化レイヤーによって管理されています。OS は、ストレージデバイスを使用して、このブロックを物理セクタにマッピングしています。EBS は 512 バイトのセクタをオペレーティングシステムに割り当てます。これにより、セクタサイズの倍数であるデータブロックを使用してディスクへのデータの読み書きを行うことができます。

論理データブロックの一般的なデフォルトサイズは、現在 4,096 バイト (4 KiB) です。ワーカーロードによっては、ブロックサイズが小さいまたは大きい方がメリットを得られるため、ファイルシステムはデフォルト以外のブロックサイズをサポートしています。このサイズはフォーマット時に指定できます。デフォルト以外のブロックサイズのシナリオは、このトピックの対象外ですが、指定したブロックサイズによっては、ボリュームのストレージキャパシティーに影響を及ぼす場合があります。次の表に、ブロックサイズの機能としてストレージキャパシティーを示します。

| ブロックサイズ       | 最大ボリュームサイズ |
|---------------|------------|
| 4 KiB (デフォルト) | 16 TiB     |
| 8 KiB         | 32 TiB     |
| 16 KiB        | 64 TiB     |
| 32 KiB        | 128 TiB    |
| 64 KiB (最大)   | 256 TiB    |

EBS で指定されているボリュームサイズ (16 TiB) の制限は、現在 4 KiB のデータブロックで使用できる最大サイズと同等です。

## Amazon EBS ボリュームの作成

新しい Amazon EBS ボリュームを作成して、同じアベイラビリティーボリューム内に任意の EC2 インスタンスにアタッチできます。暗号化された EBS ボリュームを作成することを選択できますが、暗号化されたボリュームがアタッチできるのは、サポートされているインスタンスタイプのみです。詳細については、「[サポートされるインスタンスタイプ \(p. 1016\)](#)」を参照してください。

高性能のストレージシナリオ用にボリュームを作成する場合、プロビジョンド IOPS SSD (io1) ボリュームを使用して、アプリケーションをサポートするために十分な帯域幅を持つインスタンス (EBS 最適化インスタンス、10 ギガビットネットワーク接続を備えたインスタンスなど) にアタッチしてください。スループット最適化 HDD (st1) ボリュームと Cold HDD (sc1) ボリュームの場合も同じです。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

新しい EBS ボリュームは、利用可能になるとすぐに最大のパフォーマンスを発揮し、初期化 (以前は事前ウォーミングと呼ばれました) を必要としません。ただし、スナップショットから復元されたボリュームのストレージブロックは、アクセスするためには事前に初期化する必要があります (Amazon S3 からフルダウンしてボリュームに書き込みます)。この準備処理には時間がかかるため、初めて各ブロックにアクセスした場合に、I/O 操作のレイテンシーの著しい増加が発生する可能性があります。ほとんどのアプリケーションにとって、ボリュームの存続期間全体でこのコストを割り当てるることは、許容範囲内です。一度データにアクセスされると、パフォーマンスは元に戻ります。詳細については、「[Amazon EBS ボリュームの初期化 \(p. 1049\)](#)」を参照してください。

### ボリュームを作成する方法

- EBS ボリュームを作成して、実行中のインスタンスにアタッチできます。詳細については、以下の手順を参照してください。
- ブロックデバイスマッピングを指定すると、インスタンスの起動時に、EBS ボリュームを作成してアタッチすることができます。詳細については、「[インスタンス起動ウィザードを使用してインスタンスを起動する \(p. 449\)](#)」および「[ブロックデバイスマッピング \(p. 1100\)](#)」を参照してください。
- 以前作成したスナップショットからボリュームを復元できます。詳細については、「[スナップショットからの Amazon EBS ボリュームの復元 \(p. 950\)](#)」を参照してください。

### コンソールを使用して、新しい(空の) EBS ボリュームを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、ボリュームを作成するリージョンを選択します。一部の Amazon EC2 リソースはリージョン間で共有できるため、この選択は重要です。詳細については、「[リソースの場所 \(p. 1110\)](#)」を参照してください。
3. ナビゲーションペインで、[Elastic Block Store (Elastic Block Store (EBS))]、[Volumes (ボリューム)] を選択します。
4. [Create Volume (ボリュームの作成)] を選択します。
5. [Volume Type (ボリュームタイプ)] で、ボリュームタイプを選択します。詳細については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。
6. [Size (GiB) (サイズ (GiB))] に、ボリュームのサイズを入力します。詳細については、「[EBS ボリュームのサイズと設定の制限 \(p. 946\)](#)」を参照してください。
7. プロビジョンド IOPS SSD ボリュームの場合、[IOPS] に、ボリュームがサポートする IOPS (1 秒あたりの入力/出力オペレーションの数) の最大数を入力します。
8. [Availability Zone] で、ボリュームを作成するアベイラビリティーボリュームを選択します。EBS ボリュームをアタッチできる EC2 インスタンスは、同じアベイラビリティーボリュームにあるものに限られます。
9. (オプション) インスタンスタイプが EBS 暗号化をサポートしており、ボリュームを暗号化する場合は、[Encrypt this volume (このボリュームを暗号化する)] を選択して、CMK を選択します。このリージョンでデフォルトで暗号化が有効になっている場合、EBS 暗号化が有効になり、EBS 暗号化

のデフォルト CMK が選択されます。[Master Key (マスターキー)] から別の CMK を選択するか、アクセス可能なキーの完全な ARN を貼り付けることができます。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

10. (オプション) [Create additional tags] を選択してボリュームにタグを追加します。タグごとに、タグキーとタグの値を指定します。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。
11. [Create Volume (ボリュームの作成)] を選択します。ボリュームステータスが [Available (利用可能)] になったら、ボリュームをインスタンスにアタッチできます。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。

コマンドラインを使用して、新しい(空の)EBS ボリュームを作成するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- `create-volume`AWS CLI
- `New-EC2Volume`AWS Tools for Windows PowerShell

## スナップショットからの Amazon EBS ボリュームの復元

Amazon S3 に格納されたスナップショットから、データが存在する Amazon EBS ボリュームを復元できます。スナップショットの ID を知っていること、およびスナップショットへのアクセス許可を持っていることが必要です。スナップショットの詳細については、「[Amazon EBS スナップショット \(p. 970\)](#)」を参照してください。

EBS スナップショットは、速度、利便性、コストに優れるため、Amazon EC2 で推奨されるバックアップツールです。スナップショットからボリュームを復元すると、すべてのデータをそのままの状態で、過去の特定時点の状態が再作成されます。復元されたボリュームをインスタンスにアタッチすることで、リージョン間でのデータの複製、テスト環境の作成、損傷または破損した本稼働ボリュームの完全な置換、特定のファイルとディレクトリの取得とアタッチされた別のボリュームへの転送を行うことができます。詳細については、「[Amazon EBS スナップショット \(p. 970\)](#)」を参照してください。

既存の EBS スナップショットを基に作成された新しいボリュームは、バックグラウンドで時間をかけて読み込まれます。つまり、スナップショットを基にボリュームを作成した後は、Amazon S3 から EBS ボリュームにすべてのデータが転送されるのを待たなくても、アタッチしたインスタンスからボリュームとそのすべてのデータへのアクセスを開始できます。まだ読み込まれていないデータに対してインスタンスからのアクセスがあった場合、ボリュームは要求されたデータを Amazon S3 から即座にダウンロードし、引き続き残りのボリュームデータをバックグラウンドで読み込みます。

## EBS パフォーマンス

新しい EBS ボリュームは、利用可能になるとすぐに最大のパフォーマンスを発揮し、初期化(以前は事前ウォーミングと呼ばれました)を必要としません。

スナップショットから復元されたボリュームへのアクセスは、ストレージブロックが Amazon S3 からブルダウンされてボリュームに書き込まれると可能になります。この事前処理には一定の時間がかかるため、各ブロックへの初回アクセス時には、I/O 操作のレイテンシーが著しく増加する可能性があります。ボリュームのパフォーマンスは、すべてのブロックがダウンロードされてボリュームに書き込まれると正常値に達します。

ほとんどのアプリケーションにとって、ボリュームの存続期間全体で初期化コストを割り当てることは、許容範囲内です。本番環境におけるこの初期パフォーマンスヒットは、以下のいずれかの方法で回避できます。

- ボリューム全体の即時初期化を強制する。詳細については、「[Amazon EBS ボリュームの初期化 \(p. 1049\)](#)」を参照してください。

- スナップショットの高速スナップショット復元を有効化して、スナップショットから作成される EBS ボリュームが作成時に完全に初期化され、各ボリュームのあらゆるプロビジョンドパフォーマンスが即座に発揮されるようにします。詳細については、「[Amazon EBS 高速スナップショット復元 \(p. 1024\)](#)」を参照してください。

## EBS 暗号化

暗号化されたスナップショットから復元された新しい EBS ボリュームは、自動的に暗号化されます。暗号化されていないスナップショットからボリュームを復元しながら、その場でボリュームを暗号化することもできます。暗号化されたボリュームは、EBS 暗号化をサポートするインスタンスタイプにのみアタッチできます。詳細については、「[サポートされるインスタンスタイプ \(p. 1016\)](#)」を参照してください。

## スナップショットからボリュームを作成する

スナップショットからボリュームを作成するには、次の手順を使用します。

コンソールを使用してスナップショットから EBS ボリュームを作成するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションバーから、スナップショットのリージョンを選択します。

スナップショットを別のリージョンのボリュームに復元するには、スナップショットを新しいリージョンにコピーし、そのリージョンのボリュームに復元することができます。詳細については、「[Amazon EBS スナップショットのコピー \(p. 977\)](#)」を参照してください。

- ナビゲーションペインで、[Elastic Block Store (Elastic Block Store (EBS))]、[Volumes (ボリューム)] を選択します。
- [Create Volume (ボリュームの作成)] を選択します。
- [Volume Type (ボリュームタイプ)] で、ボリュームタイプを選択します。詳細については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。
- [Snapshot (スナップショット)] に、ボリュームの復元元となるスナップショットの ID または説明を入力し、表示されたオプションリストから選択します。
- (オプション) [Encrypted this volume (このボリュームを暗号化)] を選択して、ボリュームの暗号化状態を変更します。デフォルトで暗号化 (p. 1017) が有効になっている場合、これはオプションです。[Master Key (マスターキー)] から CMK を選択して、EBS 暗号化のデフォルトの CMK 以外の CMK を指定します。
- [Size (GiB) (サイズ (GiB))] に、ボリュームのサイズを入力するか、スナップショットのデフォルトサイズが適切であるか確認します。

ボリュームサイズとスナップショットの両方を指定した場合は、スナップショットサイズ以上のサイズにする必要があります。ボリュームの種類とスナップショットを選択すると、ボリュームの最小サイズと最大サイズが [Size] の横に表示されます。詳細については、「[EBS ボリュームのサイズと設定の制限 \(p. 946\)](#)」を参照してください。

- プロビジョンド IOPS SSD ボリュームの場合、[IOPS] に、ボリュームがサポートする IOPS (1 秒あたりの入力/出力オペレーションの数) の最大数を入力します。
- [Availability Zone] で、ボリュームを作成するアベイラビリティゾーンを選択します。EBS ボリュームをアタッチできる EC2 インスタンスは、同じアベイラビリティゾーンに存在するものに限られます。
- (オプション) [Create additional tags] を選択してボリュームにタグを追加します。タグごとに、タグキーとタグの値を指定します。
- [Create Volume (ボリュームの作成)] を選択します。
- スナップショットからボリュームを復元すると、インスタンスに添付して使用を開始できます。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。

- 
14. スナップショットを、そのスナップショットのデフォルトよりも大きなボリュームに復元する場合、追加容量の利点を活用できるように、ボリュームのファイルシステムを拡張する必要があります。詳細については、「[Amazon EBS Elastic Volumes \(p. 1003\)](#)」を参照してください。

コマンドラインを使用してスナップショットから EBS ボリュームを作成するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [create-volume](#)AWS CLI
- [New-EC2Volume](#)Windows PowerShell

## インスタンスへの Amazon EBS ボリュームのアタッチ

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

### 前提条件

- インスタンスにアタッチできるボリューム数を決定します。詳細については、「[インスタンスボリューム数の制限 \(p. 1097\)](#)」を参照してください。
- ボリュームが暗号化されている場合、Amazon EBS 暗号化をサポートするインスタンスだけにアタッチできます。詳細については、「[サポートされるインスタンスタイプ \(p. 1016\)](#)」を参照してください。
- ボリュームに AWS Marketplace 製品コードがある場合は、次のようにになります。
  - ボリュームは停止されたインスタンスにのみアタッチできます。
  - ボリューム上に存在する AWS Marketplace コードをサブスクライブしている必要があります。
  - インスタンスの構成(インスタンスタイプ、オペレーティングシステム)は、その特定の AWS Marketplace コードをサポートするものでなければなりません。たとえば、Windows インスタンスからのボリュームを Linux インスタンスにアタッチすることはできません。
- AWS Marketplace 製品コードがボリュームからインスタンスにコピーされます。

コンソールを使用して、EBS ボリュームをインスタンスにアタッチするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Elastic Block Store (Elastic Block Store (EBS))]、[Volumes (ボリューム)] の順に選択します。
3. 利用可能なボリュームを選択し、[アクション]、[ボリュームのアタッチ] の順に選択します。
4. [インスタンス] に、インスタンスの名前または ID を入力します。オプションのリストからインスタンスを選択します(ボリュームと同じアベイラビリティーゾーンにあるインスタンスのみ表示されます)。
5. [デバイス] で、提示されたデバイス名のままにするか、サポートされる別のデバイス名を入力します。詳細については、「[Linux インスタンスでのデバイスの名前付け \(p. 1098\)](#)」を参照してください。
6. [アタッチ] を選択します。
7. インスタンスに接続し、ボリュームをマウントします。詳細については、「[Linux で Amazon EBS ボリュームを使用できるようにする \(p. 956\)](#)」を参照してください。

コマンドラインを使用して、EBS ボリュームをインスタンスにアタッチするには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [attach-volume \(AWS CLI\)](#)
- [Add-EC2Volume \(AWS Tools for Windows PowerShell\)](#)

## Amazon EBS マルチアタッチを使用した複数のインスタンスへのボリュームのアタッチ

Amazon EBS マルチアタッチを使用すると、単一のプロビジョンド IOPS SSD (io1) ボリュームを、同じアベイラビリティーボーンにある最大 16 の Nitro ベースのインスタンスにアタッチできます。複数のマルチアタッチが有効なボリュームを 1 つのインスタンスまたはインスタンスセットにアタッチできます。ボリュームがアタッチされている各インスタンスには、共有ボリュームに対する完全な読み取りおよび書き込みアクセス許可があります。マルチアタッチを使用すると、同時書き込みオペレーションを管理するクラスター化された Linux アプリケーションで、アプリケーションの可用性を高めることができます。

マルチアタッチが有効なボリュームは、通常の Amazon EBS ボリュームでサポートされる以下のような多くの機能をサポートしています。

- [タグ付け \(p. 1120\)](#)
- [Amazon EBS スナップショット \(p. 970\)](#)
- [Amazon EBS スナップショットライフサイクルの自動化 \(p. 992\)](#)
- [Amazon EBS Encryption \(p. 1014\)](#)
- [Amazon EBS の Amazon CloudWatch メトリクス \(p. 1060\)](#)
- [Amazon EBS での Amazon CloudWatch Events \(p. 1066\)](#)

### トピック

- [考慮事項と制約事項 \(p. 273\)](#)
- [パフォーマンス \(p. 954\)](#)
- [マルチアタッチの操作 \(p. 954\)](#)
- [モニタリング \(p. 956\)](#)
- [料金と請求 \(p. 956\)](#)

## 考慮事項と制約事項

- マルチアタッチが有効なボリュームは I/O フェンスをサポートしていません。I/O フェンスプロトコルは、データの一貫性を維持するために、共有ストレージ環境での書き込みアクセスを制御します。アプリケーションは、データの整合性を維持するために、アタッチされたインスタンスの書き込み順序を提供する必要があります。
- マルチアタッチが有効なボリュームは、同じアベイラビリティーボーンにある最大 16 の Nitro ベースのインスタンス (p. 187) にアタッチできます。
- マルチアタッチは、[プロビジョンド IOPS SSD \(io1\) ボリューム \(p. 938\)](#) でのみサポートされます。
- マルチアタッチは、us-east-1、us-east-2、us-west-2、eu-west-1、および ap-northeast-2 リージョンで使用できます。
- マルチアタッチが有効なボリュームは、ブートボリュームとして作成できません。
- マルチアタッチ対応のボリュームは、インスタンスあたり 1 つのブロックデバイスマッピングにアタッチできます。
- ボリュームの作成後に、マルチアタッチを有効または無効にすることはできません。
- マルチアタッチが有効なボリュームのボリュームタイプ、サイズ、プロビジョンド IOPS を変更することはできません。
- マルチアタッチは、Amazon EC2 コンソールまたは RunInstances API を使用してインスタンスの起動時に有効にすることはできません。

- Amazon EBS インフラストラクチャレイヤーに問題があるマルチアタッチが有効なボリュームは、アタッチされているすべてのインスタンスで使用できません。Amazon EC2 またはネットワークレイヤーでの問題は、一部のアタッチされたインスタンスにのみ影響する可能性があります。

## パフォーマンス

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

たとえば、50,000 プロビジョンド IOPS で io1 マルチアタッチ対応のボリュームを作成し、それを m5.8xlarge インスタンスと c5.12xlarge インスタンスにアタッチするとします。m5.8xlarge および c5.12xlarge インスタンスは、それぞれ最大 30,000 および 40,000 IOPS をサポートします。各インスタンスは、ボリュームのプロビジョンド IOPS 50,000 を下回るため、最大 IOPS を駆動できます。ただし、両方のインスタンスがボリュームへの I/O を同時に駆動する場合、それらの合計 IOPS は、ボリュームのプロビジョニングされたのパフォーマンス 50,000 IOPS を超えることはできません。ボリュームは 50,000 IOPS を超えません。

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

## マルチアタッチの操作

マルチアタッチが有効なボリュームは、他の Amazon EBS ボリュームを管理する場合とほぼ同じ方法で管理できます。ただし、マルチアタッチ機能を使用するには、ボリュームに対してマルチアタッチ機能を有効にする必要があります。新しいボリュームを作成する場合、マルチアタッチはデフォルトで無効になっています。

### 目次

- マルチアタッチの有効化 (p. 954)
- インスタンスへのボリュームのアタッチ (p. 955)
- 終了時に削除 (p. 955)

### マルチアタッチの有効化

Amazon EBS ボリュームに対してマルチアタッチを有効にできるのは、作成時のみです。

Amazon EBS ボリュームの作成時にマルチアタッチを有効にするには、次のいずれかの方法を使用します。

#### Console

##### ボリューム作成中にマルチアタッチを有効にするには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインの [Volumes (ボリューム)] を選択します。
- [Create Volume (ボリュームの作成)] を選択します。
- [Volume Type (ボリュームタイプ)] で、[Provisioned IOPS SSD (io1) (プロビジョンド IOPS SSD (io1))] を選択します。
- [Size (サイズ)] と [IOPS] で、必要なボリュームサイズとプロビジョニングする IOPS 数を選択します。
- [Availability Zone (アベイラビリティーゾーン)] で、インスタンスと同じアベイラビリティーゾーンを選択します。

7. [Multi-Attach (マルチアタッチ)] で、[Enable (有効)] を選択します。
8. [Create Volume (ボリュームの作成)] を選択します。

#### Command line

ボリューム作成中にマルチアタッチを有効にするには

`create-volume` コマンドを使用して、`--multi-attach-enabled` パラメータを指定します。

```
$ aws ec2 create-volume --volume-type io1 --multi-attach-enabled --size 100 --iops 2000  
--region us-west-2 --availability-zone us-west-2b
```

#### インスタンスへのボリュームのアタッチ

マルチアタッチが有効なボリュームは、通常のボリュームをアタッチするのと同じ方法でインスタンスにアタッチします。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。

#### 終了時に削除

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

終了時の削除を予測できるようにするには、ボリュームがアタッチされているすべてのインスタンスについて、終了時の削除を有効または無効にします。

デフォルトでは、ボリュームがインスタンスにアタッチされると、ロックデバイスマッピングの「終了時に削除」の設定は `false` に設定されます。マルチアタッチが有効なボリュームの終了時に削除をオンにするには、ロックデバイスマッピングを修正します。

アタッチされたインスタンスの終了時にボリュームを削除する場合は、アタッチされたすべてのインスタンスのロックデバイスマッピングで「終了時に削除」を有効にします。アタッチされたインスタンスの終了後にボリュームを保持する場合は、アタッチされたすべてのインスタンスのロックデバイスマッピングで、終了時に削除を無効にします。詳細については、「[インスタンスの削除で Amazon EBS ボリュームを保持する \(p. 549\)](#)」を参照してください。

インスタンスの終了時の削除設定は、起動時または起動後に変更できます。インスタンスの起動時に終了時に削除を有効または無効にした場合、設定は起動時にアタッチされたボリュームにのみ適用されます。起動後にインスタンスにボリュームをアタッチする場合は、そのボリュームの終了時の削除動作を明示的に設定する必要があります。

終了時のインスタンスの削除設定は、コマンドラインツールでのみ変更できます。

既存のインスタンスの [終了時に削除] 設定を変更するには

`modify-instance-attribute` コマンドを使用して、`--block-device-mappings` option で `DeleteOnTermination` 属性を指定します。

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

`mapping.json` で、以下を指定します。

```
[
```

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

## モニタリング

Amazon EBS ボリュームの CloudWatch メトリクスを使用して、マルチアタッチが有効なボリュームをモニタリングできます。詳細については、「[Amazon EBS の Amazon CloudWatch メトリクス \(p. 1060\)](#)」を参照してください。

データは、アタッチされたすべてのインスタンスにわたって集約されます。アタッチされた個々のインスタンスのメトリクスをモニタリングすることはできません。

## 料金と請求

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

## Linux で Amazon EBS ボリュームを使用できるようにする

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

EBS ボリュームのスナップショットは、バックアップ目的で作成したり、別のボリュームを作成する際のベースラインとして使用したりできます。詳細については、「[Amazon EBS スナップショット \(p. 970\)](#)」を参照してください。

Windows インスタンスのボリュームに関する手順は、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows でボリュームを使用できるようにする](#)」を参照してください。

## アタッチ済みボリュームのフォーマットとマウント

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

Linux で EBS ボリュームをフォーマットしてマウントするには

1. SSH を使用してインスタンスに接続します。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。
2. ブロックデバイスマッピングで指定したものとは異なるデバイス名を使用して、デバイスをインスタンスにアタッチすることができます。詳細については、「[Linux インスタンスでのデバイスの名前付け \(p. 1098\)](#)」を参照してください。lsblk コマンドを使用して、使用可能なディスクデバイスとマウントポイント (該当する場合) を表示し、使用する正しいデバイス名を決定します。lsblk の出力は、フルデバイスパスから /dev/ プレフィックスを削除します。

EBS ボリュームを NVMe ブロックデバイスとして公開する [Nitro ベースのインスタンス \(p. 187\)](#) の出力例を次に示します。ルートデバイスは /dev/nvme0n1 です。接続されているボリュームは /dev/nvme1n1 で、まだマウントされていません。

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

```
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme1n1   259:0    0   10G  0 disk
nvme0n1   259:1    0   8G  0 disk
-nvme0n1p1 259:2    0   8G  0 part /
-nvme0n1p128 259:3   0   1M  0 part
```

以下は T2 インスタンスの出力例です。ルートデバイスは /dev/xvda です。接続されているボリュームは /dev/xvdf で、まだマウントされていません。

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda     202:0    0   8G  0 disk
-xvda1   202:1    0   8G  0 part /
xvdf     202:80   0  10G  0 disk
```

- ボリュームにファイルシステムがあるかどうかを確認します。新しいボリュームは未加工のブロックデバイスであるため、マウントして使用する前に、ボリュームにファイルシステムを作成する必要があります。スナップショットから復元されたボリュームは、ファイルシステムを備えている可能性が高くなります。既存のファイルシステムの上に新しいファイルシステムを作成すると、データが上書きされます。

デバイスに関する情報(ファイルシステムの種類など)を一覧表示するには、file -s コマンドを使用します。次の出力例のように、出力に data だけが表示されている場合は、デバイスにはファイルシステムが存在していないため、ファイルシステムを作成する必要があります。

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

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

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

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

#### Warning

すでにデータが入っているボリューム(たとえば、スナップショットから復元されたボリューム)をマウントしている場合は、このコマンドを使用しないでください。ステップ 1 を実行した場合、ボリュームがフォーマットされ、既存のデータが削除されます。

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

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

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

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

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

6. 次のコマンドを使用して、前のステップで作成したディレクトリにボリュームをマウントします。

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

7. 新しいボリュームマウントのファイルのアクセス許可をプレビューして、ユーザーとアプリケーションがボリュームに書き込みできることを確認します。ファイルのアクセス許可の詳細については、Linux Documentation Project の「[ファイルセキュリティ](#)」を参照してください。
8. インスタンスを再起動した後にマウントポイントが自動的に保存されることはありません。再起動後にこの EBS ボリュームを自動的にマウントするには、「[再起動後に接続ボリュームを自動的にマウントする \(p. 958\)](#)」を参照してください。

## 再起動後に接続ボリュームを自動的にマウントする

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

/etc/fstab でシステムの現在のデバイス名 (/dev/xvdf など) は使用できますが、代わりにデバイスの 128 ビット汎用一意識別子 (UUID) を使用することをお勧めします。デバイス名は変更される可能性がありますが、UUID はパーティションの存続期間を通じて持続します。UUID を使用することで、ハードウェアの再構成後にシステムが起動できなくなる可能性を減らすことができます。詳細については、「[EBS デバイスの特定 \(p. 1029\)](#)」を参照してください。

再起動後に接続ボリュームを自動的にマウントするには

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

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

2. blkid コマンドを使用してデバイスの UUID を見つけます。

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID="ca774df7-756d-4261-a3f1-76038323e572" TYPE="xfs"
PARTLABEL="Linux" PARTUUID="02dc3d67-e87c-4f2e-9a72-a3cf8f299c10"
/dev/xvdf: UUID="aebf131c-6957-451e-8d34-ec978d9581ae" TYPE="xfs"
```

Ubuntu 18.04 では、lsblk コマンドを使用します。

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

3. 任意のテキストエディタ (例: nano または vim など) を使って /etc/fstab ファイルを開きます。

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

4. 指定されたマウントポイントにデバイスをマウントするために、/etc/fstab に次のエントリを追加します。フィールドは、blkid (Ubuntu 18.04 の場合は lsblk) から返される UUID 値、マウントポイント、ファイルシステム、および推奨されるファイルシステムマウントオプションです。詳細については、「[fstab \(man fstab の実行\)](#)」のマニュアルページを参照してください。

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

### Note

このボリュームをアタッチしないでインスタンスを起動することを目的としている場合 (たとえば、ボリュームを別のインスタンスに移動した後)、nofail マウントオプションを追加し、ボリュームのマウントでエラーが発生してもインスタンスが起動できるようにし

てください。また、Debian から派生した OS (16.04 より前の Ubuntu バージョンなど) では、nobootwait マウントオプションを追加する必要があります。

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

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

エラーメッセージが表示されたら、ファイル内のエラーに対処してください。

Warning

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

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

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

## Amazon EBS ボリュームに関する情報を表示する

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

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

### 詳細情報の表示

コンソールを使用して、EBS ボリュームについての情報を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Volumes (ボリューム)] を選択します。
3. ボリュームの詳細情報を表示するには、そのボリュームを選択します。詳細ペインで、ボリュームに関する情報を確認できます。
4. 詳細ペインで、ボリュームに関する情報を確認できます。

インスタンスにアタッチされている EBS ボリュームを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスの詳細情報を表示するには、そのインスタンスを選択します。
4. 詳細ペインで、ルートデバイスとブロックドバイスに関する情報を確認できます。

コマンドラインを使用して、EBS ボリュームについての情報を表示するには

次のいずれかのコマンドを使用してボリュームの属性を表示できます。詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [describe-volumes](#) (AWS CLI)
- [Get-EC2Volume](#) (AWS Tools for Windows PowerShell)

## 空きディスク容量の表示

インスタンスの Linux オペレーティングシステムから、どのくらいのディスク容量が使用可能かなどの EBS ボリュームの詳細情報を取得できます。たとえば、以下のコマンドを使用します。

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

## ボリュームのステータスのモニタリング

Amazon ウェブ サービス (AWS) では、Amazon Elastic Block Store (Amazon EBS) ボリュームのモニタリングに使用できるデータを自動的に提供します。

### 目次

- [EBS ボリュームステータスチェック \(p. 960\)](#)
- [EBS ボリュームイベント \(p. 962\)](#)
- [障害のあるボリュームの操作 \(p. 964\)](#)
- [AutoEnable IO ボリューム属性の操作 \(p. 966\)](#)

モニタリングの詳細については、「[Amazon EBS の Amazon CloudWatch メトリクス \(p. 1060\)](#)」と「[Amazon EBS での Amazon CloudWatch Events \(p. 1066\)](#)」を参照してください。

## EBS ボリュームステータスチェック

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

ボリュームステータスチェックは 5 分ごとに自動的に試行され、成功または失敗のステータスを返します。すべてのチェックが成功した場合、ボリュームのステータスは `ok` です。チェックが失敗した場合、ボリュームのステータスは `impaired` です。ステータスが `insufficient-data` の場合、ボリュームのチェックがまだ実行中である可能性があります。ボリュームステータスチェックの結果を表示して、障害のあるボリュームを特定し、必要なアクションを行うことができます。

ボリュームのデータが潜在的に不整合であると Amazon EBS が判断した場合、デフォルトでは、アタッチされたすべての EC2 インスタンスからそのボリュームへの I/O が無効になります。これにより、データの破損を防ぐことができます。I/O が無効になると、次のボリュームステータスチェックが失敗し、ボリュームステータスは `impaired` になります。さらに、I/O が無効になったこと、およびボリュームへの I/O を有効にすることによってボリュームの障害ステータスを解決できることを伝えるイベントが表示されます。ユーザーが I/O を有効にするまでシステムは待機するため、ユーザーはインスタンスによるボリュームの使用を継続するか、その前に `fsck` などのコマンドを使用して整合性チェックを実行するかを判断することができます。

### Note

ボリュームステータスはボリュームステータスチェックに基づいており、ボリュームの状態を反映していません。従って、ボリュームステータスではボリュームが `error` 状態 (たとえば、I/O を受け付けできない) であることは判りません。

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

I/O パフォーマンスのステータスチェックでは、実際のボリュームパフォーマンスと期待されるボリュームパフォーマンスが比較され、ボリュームのパフォーマンスが期待を下回っている場合は警告が生成されます。このステータスチェックは、インスタンスにアタッチされた io1 ボリュームにのみ使用でき、汎用 SSD (gp2) ボリューム、スループット最適化 HDD (st1) ボリューム、Cold HDD (sc1) ボリューム、およびマグネティック (standard) ボリュームには有効ではありません。I/O パフォーマンスのステータスチェックは 1 分に 1 回実行され、CloudWatch はこのデータを 5 分おきに収集するため、io1 ボリュームをインスタンスにアタッチしてから、このチェックにより I/O パフォーマンスのステータスが報告されるまで最大 5 分かかる可能性があります。

#### Important

スナップショットから復元された io1 ボリュームを初期化している間は、ボリュームのパフォーマンスが想定レベルの 50% を下回る場合があります。このため、ボリュームの [I/O Performance] ステータスチェックでは warning 状態が表示されます。これは想定の動作です。初期化中の io1 ボリュームの warning 状態は無視してかまいません。詳細については、「[Amazon EBS ボリュームの初期化 \(p. 1049\)](#)」を参照してください。

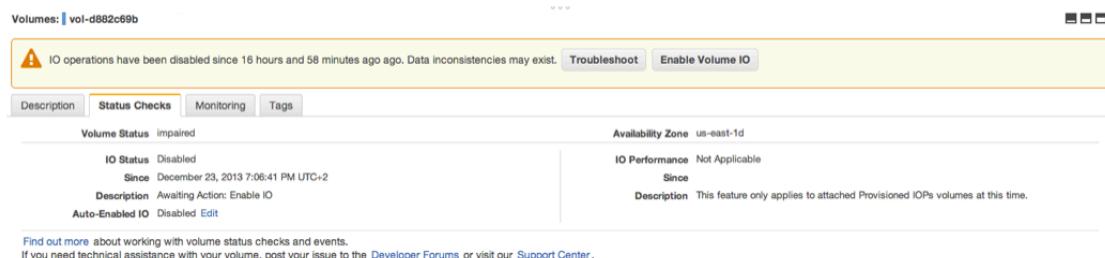
次の表に、Amazon EBS ボリュームのステータスを示します。

| ボリュームのステータス       | I/O 有効ステータス                                                                                           | I/O パフォーマンスステータス (プロビジョンド IOPS ボリュームでのみ使用可能)                                            |
|-------------------|-------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| ok                | Enabled (I/O Enabled または I/O Auto-Enabled)                                                            | Normal (ボリュームパフォーマンスは想定どおり)                                                             |
| warning           | Enabled (I/O Enabled または I/O Auto-Enabled)                                                            | Degraded (ボリュームのパフォーマンスが想定を下回っている)<br>Severely Degraded (ボリュームのパフォーマンスが想定をかなり下回っている)    |
| impaired          | Enabled (I/O Enabled または I/O Auto-Enabled)<br><br>Disabled (ボリュームがオフラインで復旧の保留中、またはユーザーによる I/O の有効化待ち) | Stalled (ボリュームのパフォーマンスは致命的な影響を受けている)<br><br>Not Available (I/O が無効なため、I/O パフォーマンスの判定不能) |
| insufficient-data | Enabled (I/O Enabled または I/O Auto-Enabled)<br><br>Insufficient Data                                   | Insufficient Data                                                                       |

ステータスチェックを表示または操作するには、Amazon EC2 コンソール、API、またはコマンドラインインターフェイスを使用します。

## コンソールでステータスチェックを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Volumes (ボリューム)] を選択します。[Volume Status (ボリュームのステータス)] 列に、各ボリュームの動作状況が表示されます。
3. ボリュームのステータスの詳細を表示するには、ボリュームを選択して、[Status Checks (ステータスチェック)] を選択します。



4. ステータスチェックが失敗したボリュームがある場合 (ステータスが impaired (障害) として示されている) は、[障害のあるボリュームの操作 \(p. 964\)](#) を参照してください。

ナビゲータで [Events (イベント)] を選択して、インスタンスとボリュームのすべてのイベントを表示することもできます。詳細については、「[EBS ボリュームイベント \(p. 962\)](#)」を参照してください。

## コマンドラインを使用してボリュームステータスに関する情報を表示するには

Amazon EBS ボリュームのステータスを表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-volume-status](#) (AWS CLI)
- [Get-EC2VolumeStatus](#) (AWS Tools for Windows PowerShell)

## EBS ボリュームイベント

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

データが潜在的に不整合であるボリュームで I/O を自動的に有効にするには、Auto-Enabled IO ボリューム属性 (API の `autoEnableIO`) の設定を変更します。この属性の変更の詳細については、「[障害のあるボリュームの操作 \(p. 964\)](#)」を参照してください。

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

ボリュームステータスイベントには、次の説明のいずれかが含まれています。

Awaiting Action: Enable IO

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

IO Enabled

このボリュームに対する I/O 操作が明示的に有効にされました。

## IO Auto-Enabled

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

### Normal

io1 ボリュームのみ。ボリュームのパフォーマンスは想定どおりです。

### Degraded

io1 ボリュームのみ。ボリュームのパフォーマンスは想定を下回っています。

### Severely Degraded

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

### Stalled

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

Amazon EC2 コンソール、API、またはコマンドラインインターフェイスを使用して、ボリュームのイベントを表示できます。

コンソールでボリュームのイベントを表示するには

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

The screenshot shows the Amazon EC2 console's 'Events' page. A table lists three events for a specific volume. The second event, 'vol-3682c675', is selected. A callout box highlights this event with the message: 'IO operations have been disabled since 30 days, 15 hours and 22 minutes ago. Data inconsistencies may exist.' Below the table, detailed event information is shown:

| Availability Zone | us-east-1d                         |
|-------------------|------------------------------------|
| Event Type        | potential-data-inconsistency       |
| Event Status      | Awaiting Action: Enable IO         |
| IO status         | IO Disabled                        |
| Attached to       | i-93aae4ea                         |
| Start Time        | December 23, 2013 7:09:20 PM UTC+2 |
| End time          |                                    |

*Find out more about [monitoring volume events](#).*

I/O が無効になっているボリュームがある場合は、「[障害のあるボリュームの操作 \(p. 964\)](#)」を参照してください。I/O パフォーマンスが通常の状態を下回っているボリュームがある場合、実行したアクションを原因とする一時的な状態である可能性があります(ピーク使用時にボリュームのスナップショットを作成した、必要な I/O 帯域幅をサポートできないインスタンスでボリュームを実行した、ボリュームのデータに初めてアクセスした、など)。

コマンドラインを使用してボリュームのイベントを表示するには

Amazon EBS ボリュームのイベント情報を表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-volume-status \(AWS CLI\)](#)
- [Get-EC2VolumeStatus \(AWS Tools for Windows PowerShell\)](#)

## 障害のあるボリュームの操作

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

### オプション

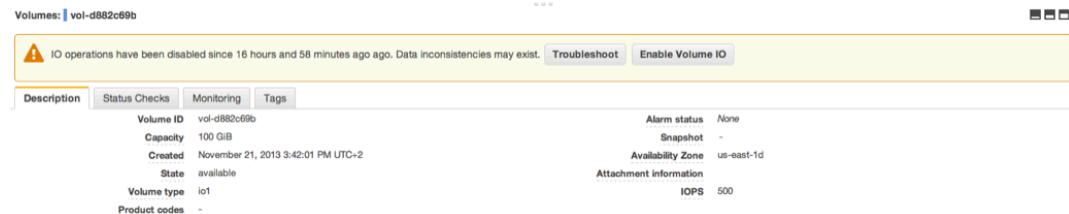
- [オプション 1: インスタンスにアタッチされたボリュームで整合性チェックを実行する \(p. 964\)](#)
- [オプション 2: 別のインスタンスを使用してボリュームで整合性チェックを実行する \(p. 965\)](#)
- [オプション 3: 不要なボリュームを削除する \(p. 966\)](#)

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

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

アタッチされたボリュームで整合性チェックを実行するには

1. アプリケーションによるボリュームの使用を停止します。
2. ボリュームの I/O を有効にします。
  - a. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
  - b. ナビゲーションペインの [Volumes (ボリューム)] を選択します。
  - c. I/O 操作を有効にするボリュームを選択します。
  - d. 詳細ペインで、[Enable Volume I/O (ボリューム I/O を有効化する)] を選択し、次に [Yes, Enable (はい、有効化します)] を選択します。



3. ボリュームのデータを確認します。

- a. fsck コマンドを実行します。
- b. (オプション) 関連するエラーメッセージがないか、使用可能なアプリケーションログまたはシステムログを確認します。
- c. ボリュームの障害が 20 分以上続く場合は、AWS サポートセンターに連絡してください。[Troubleshoot (トラブルシューティング)] をクリックしてから、[Troubleshoot Status Checks (ステータスチェックのトラブルシューティング)] ダイアログボックスの [Contact Support (サポートに問い合わせる)] を選択してサポートケースを送信します。

コマンドラインを使用してボリュームの I/O を有効にするには

Amazon EBS ボリュームのイベント情報を表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [enable-volume-io \(AWS CLI\)](#)

- [Enable-EC2VolumeIO \(AWS Tools for Windows PowerShell\)](#)

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

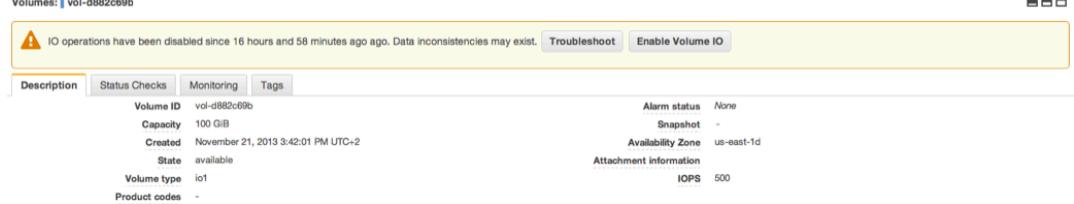
実動環境外部のボリュームをチェックするには、次の手順に従います。

### Important

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

分離されたボリュームで整合性チェックを実行するには

1. アプリケーションによるボリュームの使用を停止します。
2. ボリュームをインスタンスからデタッチします。
  - a. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
  - b. ナビゲーションペインの [Volumes (ボリューム)] を選択します。
  - c. デタッチするボリュームを選択します。
  - d. [Actions]、[Force Detach Volume] を選択します。確認のためのメッセージが表示されます。
3. ボリュームの I/O を有効にします。
  - a. ナビゲーションペインの [Volumes (ボリューム)] を選択します。
  - b. 前の手順でデタッチしたボリュームを選択します。
  - c. 詳細ペインで、[Enable Volume I/O (ボリューム I/O を有効化する)] を選択し、次に [Yes, Enable (はい、有効化します)] を選択します。



4. ボリュームを別のインスタンスにアタッチします。詳細については、「[インスタンスの起動 \(p. 448\)](#)」および「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。
5. ボリュームのデータを確認します。
  - a. fsck コマンドを実行します。
  - b. (オプション) 関連するエラーメッセージがないか、使用可能なアプリケーションログまたはシステムログを確認します。
  - c. ボリュームの障害が 20 分以上続く場合は、AWS サポートセンターに連絡してください。[Troubleshoot (トラブルシューティング)] を選択し、トラブルシューティングのダイアログボックスで [Contact Support (サポートに問い合わせる)] を選択して、サポートケースを送信します。

コマンドラインを使用してボリュームの I/O を有効にするには

Amazon EBS ボリュームのイベント情報を表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [enable-volume-io \(AWS CLI\)](#)
- [Enable-EC2VolumeIO \(AWS Tools for Windows PowerShell\)](#)

### オプション 3: 不要なボリュームを削除する

環境からボリュームを削除するには、単にそれを削除します。ボリュームの削除の詳細については、「[Amazon EBS ボリュームの削除 \(p. 969\)](#)」を参照してください。

ボリュームのデータをバックアップするスナップショットを最近作成した場合、そのスナップショットから新しいボリュームを作成できます。スナップショットからのボリュームの作成の詳細については、「[スナップショットからの Amazon EBS ボリュームの復元 \(p. 950\)](#)」を参照してください。

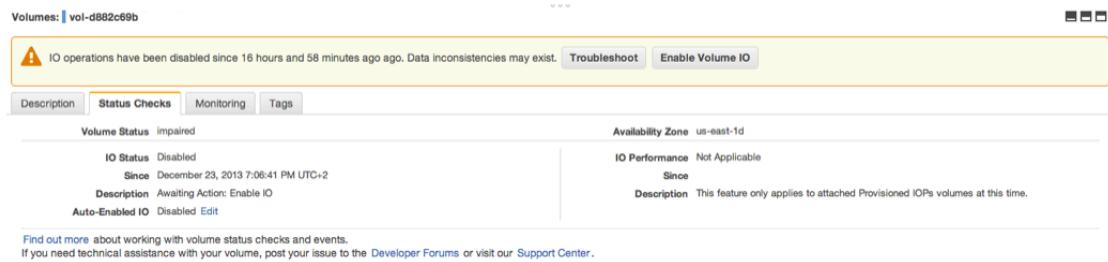
### AutoEnable IO ボリューム属性の操作

ボリュームのデータが潜在的に不整合であると Amazon EBS によって判断された場合、デフォルトでは、アタッチされているすべての EC2 インスタンスからそのボリュームへの I/O が無効になります。これにより、ボリュームステータスチェックが失敗し、障害の原因を示すボリュームステータスイベントが生成されます。あるボリュームの整合性について心配しているわけではなく、そのボリュームに障害が発生した際にそのボリュームをすぐに利用できるようにしたい場合は、デフォルトの動作を上書きして I/O を自動的に有効にするようにボリュームを設定することができます。Auto-Enabled IO ボリューム属性 (API の `autoEnableIO`) を有効にすると、ボリュームとインスタンス間の I/O が自動的に再び有効になり、ボリュームの状態チェックが成功します。また、ボリュームが潜在的に不整合な状態であること、ただしそのボリュームの I/O が自動的に有効になったことを伝えるイベントも表示されます。このイベントが発生した場合は、ボリュームの整合性をチェックし、必要に応じて置き換えます。詳細については、「[EBS ボリュームイベント \(p. 962\)](#)」を参照してください。

この手順では、ボリュームの Auto-Enabled IO 属性を表示して変更する方法について説明します。

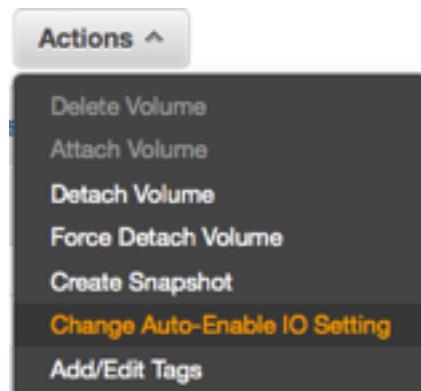
#### コンソールでボリュームの Auto-Enabled IO 属性を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Volumes (ボリューム)] を選択します。
3. ボリュームを選択して、[Status Checks (ステータスチェック)] を選択します。[Auto-Enabled IO] には、ボリュームの現在の設定 ([Enabled (有効)] または [Disabled (無効)] ) が表示されます。

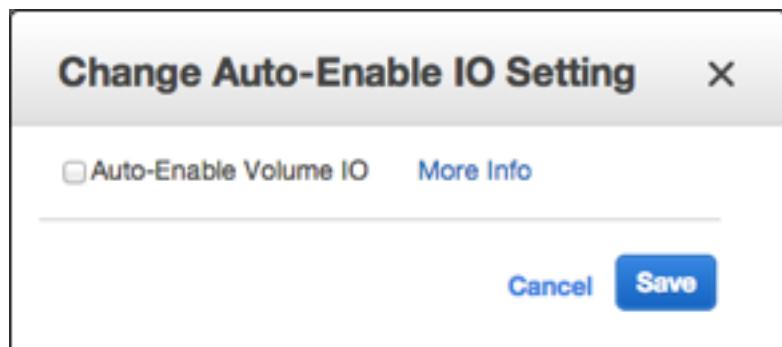


#### コンソールでボリュームの Auto-Enabled IO 属性を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Volumes (ボリューム)] を選択します。
3. ボリュームを選択して、[Actions (アクション)]、[Change Auto-Enable IO Setting (Auto-Enable IO を変更する)] の順に選択します。または、[Status Checks (ステータスチェック)] タブを選択し、[Auto-Enabled IO] で、[Edit (編集)] を選択します。



- 障害のあるボリュームの I/O を自動的に有効にするには、[Auto-Enable Volume IO (Auto-Enable ボリューム I/O)] チェックボックスをオンにします。この機能を無効にするには、チェックボックスをオフにします。



- [Save (保存)] を選択します。

コマンドラインを使ってボリュームの AutoEnableIO 属性を表示または変更するには

Amazon EBS ボリュームの autoEnableIO 属性を表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [describe-volume-attribute \(AWS CLI\)](#)
- [Get-EC2VolumeAttribute \(AWS Tools for Windows PowerShell\)](#)

ボリュームの autoEnableIO 属性を変更するには、次のコマンドのいずれかを使用できます。

- [modify-volume-attribute \(AWS CLI\)](#)
- [Edit-EC2VolumeAttribute \(AWS Tools for Windows PowerShell\)](#)

## インスタンスからの Amazon EBS ボリュームのデタッチ

インスタンスから Amazon EBS ボリュームをデタッチするには、明示的にデタッチするか、インスタンスを終了します。ただし、インスタンスが実行中の場合、最初にインスタンスからボリュームをアンマウントする必要があります。

EBS ボリュームがインスタンスのルートデバイスである場合、ボリュームをデタッチする前に、インスタンスを停止する必要があります。

AWS Marketplace 製品コードのボリュームがインスタンスからデタッチされている場合、製品コードはインスタンスには関連付けられません。

#### Important

ボリュームをデタッチした後でも、ストレージ量が AWS 無料利用枠の上限を超えており、料金が発生します。不要な料金の発生を防ぐために、ボリュームを削除する必要があります。詳細については、「[Amazon EBS ボリュームの削除 \(p. 969\)](#)」を参照してください。

この例では、ボリュームをアンマウントし、インスタンスから明示的にデタッチします。この操作は、インスタンスを終了するときや、ボリュームを別のインスタンスにアタッチするときに便利です。ボリュームがインスタンスにアタッチされていないことを確認する方法については、「[Amazon EBS ボリュームに関する情報を表示する \(p. 959\)](#)」を参照してください。

(アンマウントせずに) 切り離したボリュームを再接続することはできますが、同じマウントポイントが得られない可能性があります。進行中のボリュームがデタッチされたときにそのボリュームへの書き込みがあった場合、ボリューム上のデータは同期していない可能性があります。

コンソールを使用して、EBS ボリュームをデタッチするには

1. 次のコマンドを使用して /dev/sdh デバイスをアンマウントします。

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

2. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
3. ナビゲーションペインの [Volumes (ボリューム)] を選択します。
4. ボリュームを選択し、[Actions]、[Detach Volume] の順に選択します。
5. 確認ダイアログボックスで、[Yes, Detach] を選択します。

コマンドラインを使用してインスタンスから EBS ボリュームをデタッチするには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [detach-volume](#) (AWS CLI)
- [Dismount-EC2Volume](#) (AWS Tools for Windows PowerShell)

## トラブルシューティング

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

#### Note

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

- Amazon EC2 コンソールからボリュームを切り離しているときに問題が発生した場合は、[describe-volumes](#) CLI コマンドを使用して問題を診断すると便利です。詳細については、「[describe-volumes](#)」を参照してください。
- ボリュームの状態が `detaching` 状態のまま変わらない場合は、[Force Detach] を選択して、強制的にアタッチ解除することもできます。障害が発生したインスタンスからボリュームをアタッチ解除するた

めの最後の手段として、またはボリュームを削除するためにデタッチする場合のみ、このオプションを使用してください。インスタンスは、ファイルシステムキャッシュやファイルシステムメタデータをフラッシュする機会を失います。このオプションを使用する場合は、ファイルシステムのチェックと修復の手順を手動で実行する必要があります。

- ボリュームを数分間何度も強制的に切断しようとしたが、`detaching` 状態のままになっている場合は、[Amazon EC2 forum](#)へのヘルプのリクエストを送信できます。迅速に解決できるようにするために、ボリューム ID と、これまでに実行した手順を記述してください。
- まだマウントされたボリュームをデタッチしようとすると、ボリュームはデタッチを実行しようとして `busy` 状態でスタックする可能性があります。`describe-volumes` からの次の出力は、この状態の例を示しています。

```
aws ec2 describe-volumes --region us-west-2 --volume-ids vol-1234abcd
{
  "Volumes": [
    {
      "AvailabilityZone": "us-west-2b",
      "Attachments": [
        {
          "AttachTime": "2016-07-21T23:44:52.000Z",
          "InstanceId": "i-fedc9876",
          "VolumeId": "vol-1234abcd",
          "State": "busy",
          "DeleteOnTermination": false,
          "Device": "/dev/sdf"
        }
      ],
      ...
    }
  ]
}
```

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

## Amazon EBS ボリュームの削除

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

ボリュームを削除するには、`available` 状態（インスタンスにアタッチされない）である必要があります。詳細については、「[インスタンスからの Amazon EBS ボリュームのデタッチ \(p. 967\)](#)」を参照してください。

コンソールを使用して EBS ボリュームを削除するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインの [Volumes (ボリューム)] を選択します。
- ボリュームを選択し、[Actions]、[Delete Volume] の順に選択します。
- 確認ダイアログボックスで、[Yes, Delete] を選択します。

コマンドラインを使用して、EBS ボリュームを削除するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `delete-volume` (AWS CLI)
- `Remove-EC2Volume` (AWS Tools for Windows PowerShell)

## Amazon EBS スナップショット

ポイントインタイムスナップショットを作成することで、Amazon EBSボリュームのデータを Amazon S3 にバックアップできます。スナップショットは増分バックアップです。つまり、最後にスナップショットを作成した時点から、ボリューム上で変更のあるブロックだけが保存されます。これにより、スナップショットを作成するのに要する時間が最小限に抑えられ、データを複製しないことで、ストレージコストが節約されます。スナップショットを削除すると、そのスナップショットに固有のデータだけが削除されます。各スナップショットには、(スナップショットを作成した瞬間から) データを新しい EBS ボリュームに復元するために必要な情報がすべて含まれます。

スナップショットに基づいて EBS ボリュームを作成すると、新しいボリュームは、スナップショットの作成に使用された元のボリュームの完全なレプリカとなります。すぐに使用を開始できるよう、レプリケートされたボリュームはバックグラウンドでデータを読み込みます。まだ読み込まれていないデータにアクセスした場合、ボリュームは要求されたデータを Amazon S3 から即座にダウンロードし、引き続きボリュームの残りのデータをバックグラウンドで読み込みます。詳細については、「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」を参照してください。

### マルチボリュームスナップショット

スナップショットを使用すると、大規模データベースや複数の EBS ボリュームにわたるファイルシステムなど、重要なワークロードのバックアップを作成できます。マルチボリュームスナップショットを使用すると、EC2 インスタンスにアタッチされている複数の EBS ボリュームにわたって、正確なポイントインタイムで、データ調整済みのクラッシュ整合性スナップショットを取得できます。スナップショットは複数の EBS ボリュームにわたって自動的に作成されるため、クラッシュの一貫性を確保するためにインスタンスを停止したり、ボリューム間で調整したりする必要がなくなります。詳細については、「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」のマルチボリューム EBS スナップショットを作成する手順を参照してください。

EBS スナップショットの状態は、CloudWatch イベントを通じて追跡できます。詳細については、「[Amazon EBS での Amazon CloudWatch Events \(p. 1066\)](#)」を参照してください。

### 目次

- [増分スナップショットの仕組み \(p. 970\)](#)
- [スナップショットのコピーおよび共有 \(p. 972\)](#)
- [スナップショットの暗号化サポート \(p. 972\)](#)
- [Amazon EBS スナップショットの作成 \(p. 972\)](#)
- [Amazon EBS スナップショットの削除 \(p. 975\)](#)
- [Amazon EBS スナップショットのコピー \(p. 977\)](#)
- [Amazon EBS スナップショットに関する情報を表示する \(p. 982\)](#)
- [Amazon EBS スナップショットの共有 \(p. 982\)](#)
- [EBS スナップショットのコンテンツへのアクセス \(p. 985\)](#)
- [Amazon EBS スナップショットライフサイクルの自動化 \(p. 992\)](#)

## 増分スナップショットの仕組み

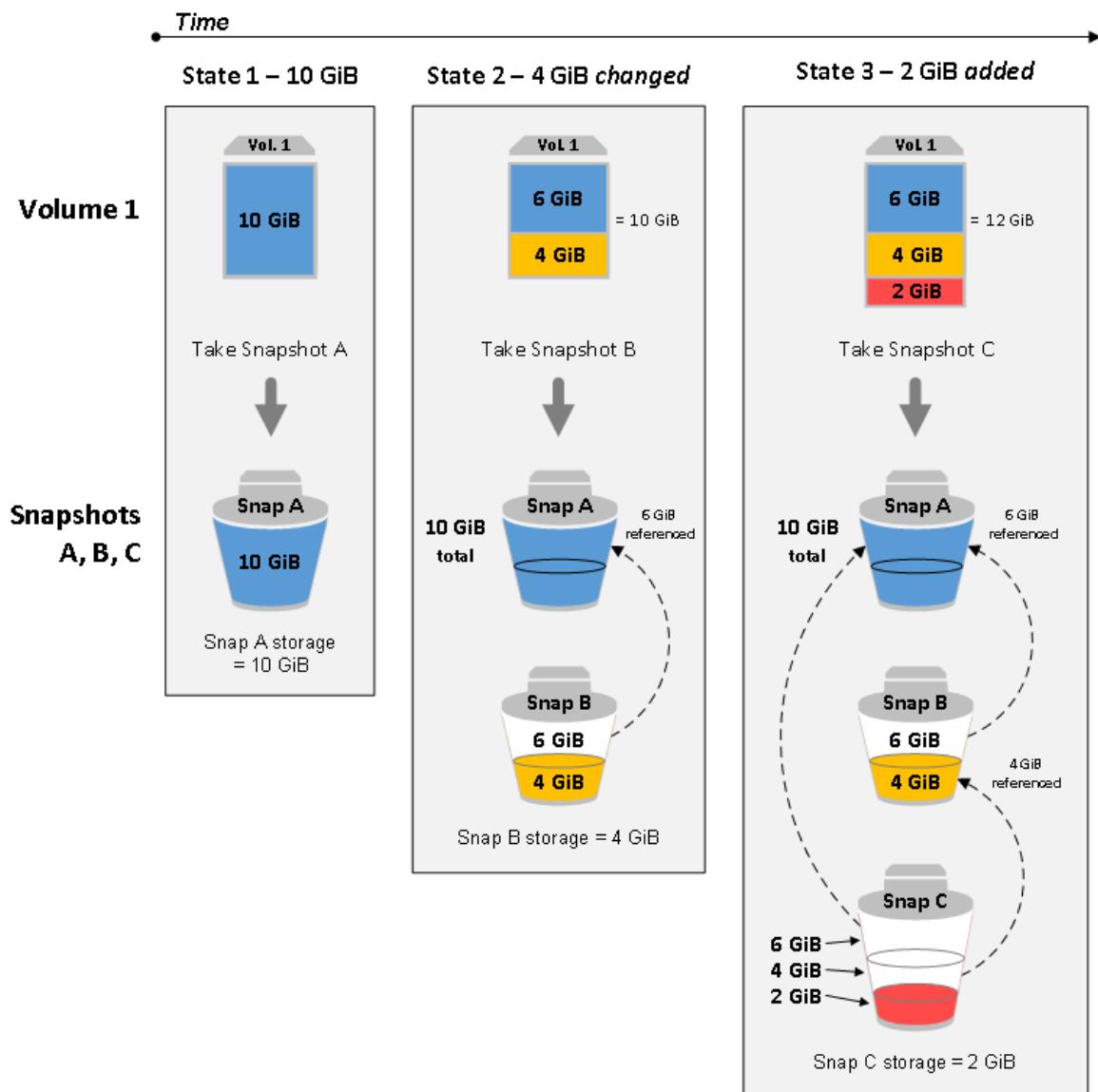
このセクションは、EBS スナップショットがある時点でのボリュームの状態をキャプチャする方法、および変化するボリュームの連続するスナップショットが変更の履歴を作成する方法に関する図を示します。

次の図では、ボリューム 1 は 3 つの時点に示されています。これら 3 つのボリューム状態それぞれのスナップショットが作成されます。

- 状態 1 では、ボリュームに 10 GiB のデータがあります。スナップ A がボリュームで作成された最初のスナップショットであるため、10 GiB のデータ全体をコピーする必要があります。

- 状態 2 では、ボリュームにはまだ 10 GiB のデータが含まれていますが、4 GiB 分が変更されました。スナップショット B は、スナップショット A が作成されたあとに変更された 4 GiB 分のみをコピーして格納する必要があります。すでにスナップショット A にコピーおよび格納されている、変更がなかった残り 6 GiB 分のデータは、スナップショット B によって、(再度) コピーされるのではなく、参照されます。これは点線の矢印によって示されます。
- 状態 3 では、2 GiB 分のデータがボリュームに追加され、合計 12 GiB となっています。スナップショット C は、スナップショット B が作成されたあとに追加された 2 GiB 分をコピーする必要があります。点線の矢印で示されているように、スナップショット C はスナップショット B に格納された 4 GiB 分のデータと、スナップショット A に格納された 6 GiB 分のデータの両方を参照します。
- 3 つのスナップショットに必要な合計ストレージ量は 16 GiB です。

ボリュームの複数のスナップショット間の関係



#### Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な(増分なし)コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

データが管理されている方法とスナップショットを削除するタイミングについては、[Amazon EBS スナップショットの削除 \(p. 975\)](#)を参照してください。

## スナップショットのコピーおよび共有

アクセス権限を変更することで、スナップショットを AWS アカウント間で共有することができます。自分のスナップショットのコピーと、他のユーザーから共有されたスナップショットのコピーを作成できます。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

スナップショットは、作成された AWS リージョンに制限されます。EBS ボリュームのスナップショットを作成した後、そのスナップショットを使って、同じリージョンで新規ボリュームを作成できます。詳細については、「[スナップショットからの Amazon EBS ボリュームの復元 \(p. 950\)](#)」を参照してください。スナップショットはリージョン間でコピーすることもできるため、地理的な拡大、データセンターの移行、災害対策など、複数のリージョンを使用することが可能になります。`completed` 状態であるアクセス可能なすべてのスナップショットをコピーできます。詳細については、「[Amazon EBS スナップショットのコピー \(p. 977\)](#)」を参照してください。

## スナップショットの暗号化サポート

EBS スナップショットは、EBS 暗号化を完全にサポートします。

- 暗号化されたボリュームのスナップショットは自動的に暗号化されます。
- 暗号化されたスナップショットから作成されたボリュームは、自動的に暗号化されます。
- 自分が所有している、またはアクセス権がある暗号化されていないスナップショットから作成したボリュームは、オンザフライで暗号化できます。
- 自分が所有している暗号化されていないスナップショットをコピーした場合、コピー処理中に暗号化できます。
- 自分が所有している、またはアクセス権がある暗号化されたスナップショットをコピーした場合、コピー処理中に異なるキーを使用して再暗号化できます。
- 暗号化されていないスナップショットから作成した暗号化されたボリュームの最初のスナップショットは常に完全なスナップショットです。
- ソーススナップショットと比べて CMK が異なる再暗号化されたボリュームの最初のスナップショットは常に完全なスナップショットです。

### Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な(増分なし)コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

スナップショット暗号化の可能性のあるシナリオの完全なドキュメントは [Amazon EBS スナップショットの作成 \(p. 972\)](#) と [Amazon EBS スナップショットのコピー \(p. 977\)](#) にあります。

詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

## Amazon EBS スナップショットの作成

EBS ボリュームのポイントインタイムスナップショットを作成して、新規ボリュームやデータバックアップ用のベースラインとして使用することができます。ボリュームのスナップショットを定期的に作成する場合、スナップショットは増分です。新しいスナップショットは、最後のスナップショット以降に変更されたブロックのみを保存します。

スナップショットは非同期に行われます。ポイントインタイムのスナップショットはすばやく作成されますが、スナップショットが完了する(変更されたすべてのブロックが Amazon S3 に転送される)まで、スナップショットのステータスは `pending` です。大きな初期スナップショットや、多数のブロックが変更されている後続のスナップショットの場合は、数時間かかることがあります。この処理の完了中に、進行中のスナップショットはボリュームに対する継続的な読み取りと書き込みの影響を受けません。

使用中のアタッチ済みボリュームのスナップショットを取ることができます。ただし、スナップショットでは、スナップショットコマンドを実行した時点での Amazon EBS ボリュームに書き込まれているデータのみがキャプチャされます。そのため、アプリケーションやオペレーティングシステムによってキャッシュされたデータは除外される可能性があります。スナップショットを取る間、ボリュームへのすべてのファイルの書き込みを停止できれば、完全なスナップショットを取ることができます。ただし、ボリュームへのすべてのファイルの書き込みを停止できない場合は、一貫した完全なスナップショットを取ることができるように、インスタンス内からボリュームをアンマウントし、スナップショットコマンドを実行して、ボリュームを再マウントします。スナップショットのステータスが `pending` の間は、ボリュームを再マウントして使用できます。

スナップショット管理を容易にするために、作成中にスナップショットにタグを付けたり、後でタグを追加したりすることができます。たとえば、スナップショットの作成元のボリュームや、元のボリュームをインスタンスに添付するために使用するデバイス名を説明するタグを追加することができます。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。

## スナップショットの暗号化

暗号化されたボリュームから作成されたスナップショットは、自動的に暗号化されます。暗号化されたスナップショットから作成されたボリュームも、自動的に暗号化されます。暗号化されたボリュームのデータと関連付けられたスナップショットは、保管時も送信時も保護されます。詳細については、「[Amazon EBS Encryption \(p. 1014\)](#)」を参照してください。

デフォルトでは、自分が所有するスナップショットからのみボリュームを作成できます。ただし、暗号化されていないスナップショットを、特定の AWS アカウントと共有したり、一般公開することで AWS コミュニティ全体で共有したりすることができます。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

暗号化されたスナップショットは、特定の AWS アカウントとのみ共有できます。他のユーザーが共有された暗号化スナップショットを使用できるようにするには、暗号化に使用された CMK キーも共有する必要があります。暗号化されたスナップショットへのアクセス権を持つユーザーが、そのスナップショットの独自のコピーを作成し、そのコピーを使用してボリュームを復元する必要があります。共有された暗号化スナップショットのコピーは、別のキーを使用して再暗号化することもできます。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

### Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な(増分なし)コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

## マルチボリュームスナップショット

マルチボリュームスナップショットを作成できます。これは、EC2 インスタンスにアタッチされているすべての EBS ボリュームのポイントインタイムスナップショットです。ライフサイクルポリシーを作成して、マルチボリュームスナップショットの作成と保持を自動化することもできます。詳細については、「[Amazon EBS スナップショットライフサイクルの自動化 \(p. 992\)](#)」を参照してください。

スナップショットが作成されると、各スナップショットは個別のスナップショットとして扱われます。单一ボリュームのスナップショットと同じように、復元、削除、クロスリージョン/アカウントコピーなど、すべてのスナップショットオペレーションを実行できます。单一ボリュームのスナップショットと同じように、マルチボリュームスナップショットにタグを付けることもできます。復元、コピー、または保存中にマルチボリュームスナップショットをまとめて管理するためにタグを付けることをお勧めします。

マルチボリュームのクラッシュコンシステントスナップショットは通常、セットとして復元されます。インスタンスにインスタンス ID、名前、またはその他の関連する詳細をタグ付けして、クラッシュコンシステントセット内にあるスナップショットを識別することは、役に立ちます。ソースボリュームから対応するスナップショットにタグを自動的にコピーすることもできます。これにより、アクセスポリシー、添付情報、コスト割り当てなどのスナップショットメタデータをソースボリュームと一致するように設定できます。

作成されたら、マルチボリュームスナップショットはそれ以外のスナップショットのように動作します。リージョンやアカウント間で復元や削除、コピーなど、すべてのオペレーションを実行できます。スナップ

プロジェクトにタグ付けすることもできます。復元、コピー、または保存中にマルチボリュームスナップショットをまとめて管理するためにタグを付けることをお勧めします。

スナップショットを作成したら、それらは正確な時点で作成された EC2 コンソールに表示されます。スナップショットはまとめて管理されるため、ボリュームセットのいずれかのスナップショットが失敗する、他のすべてのスナップショットにはエラーステータスが表示されます。

## 考慮事項

スナップショットの作成には、次の考慮事項が適用されます。

- ルートデバイスとして機能する EBS ボリュームのスナップショットを作成する場合は、スナップショットを取る前にインスタンスを停止します。
- 休止が有効にされているインスタンスからスナップショットを作成することはできません。
- 休止したインスタンスからスナップショットを作成することはできません。
- ボリュームの前のスナップショットが pending 状態の間でもボリュームのスナップショットを作成できますが、1つのボリュームで複数の pending スナップショットを作成すると、スナップショットが完了するまでボリュームのパフォーマンスが低下する場合があります。
- スナップショットの数は、gp2、io1、またはマグネティック のボリュームごとに 5 つの pending、st1 または sc1 のボリュームごとに 1 つの pending に制限されています。同一のボリュームで複数のスナップショットを同時に作成する際に ConcurrentSnapshotLimitExceeded エラーが発生した場合は、pending スナップショットが 1 つ以上完了するまで待ってから、そのボリュームの次のスナップショットを作成します。
- スナップショットが AWS Marketplace 製品コードのボリュームから作成された場合、製品コードはスナップショットに反映されます。

## スナップショットの作成

次の手順に従って、指定されたボリュームからスナップショットを作成します。

コンソールを使用してスナップショットを作成するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで [Elastic Block Store] の [Snapshots (スナップショット)] を選択します。
- [スナップショットの作成] を選択します。
- [リソースタイプの選択] で、[ボリューム] を選択します。
- [ボリューム] で、ボリュームを選択します。
- (オプション) スナップショットの説明を入力します。
- (オプション) [Add Tag (タグの追加)] を選択して、タグをスナップショットに追加します。タグごとに、タグキーとタグの値を指定します。
- [スナップショットの作成] を選択します。

コマンドラインを使用してスナップショットを作成するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [create-snapshot \(AWS CLI\)](#)
- [New-EC2Snapshot \(AWS Tools for Windows PowerShell\)](#)

## マルチボリュームスナップショットを作成する

次の手順に従って、インスタンスのボリュームからスナップショットを作成します。

## コンソールを使用してマルチボリュームスナップショットを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic Block Store] の [Snapshots (スナップショット)] を選択します。
3. [スナップショットの作成] を選択します。
4. [リソースタイプの選択] で、[インスタンス] を選択します。
5. アタッチされているすべての EBS ボリュームの同時バックアップを作成するインスタンス ID を選択します。マルチボリュームスナップショットは、インスタンスあたり最大 40 の EBS ボリュームをサポートします。
6. (オプション) [Exclude root volume (ルートボリュームの除外)] を設定します。
7. (オプション) ソースボリュームから対応するスナップショットにタグを自動的にコピーするには、[Copy tags from volume (ボリュームからタグをコピーする)] フラグを設定します。これにより、ソースボリュームに合わせてスナップショットメタデータ (アクセスポリシー、添付情報、コスト割り当てなど) が設定されます。
8. (オプション) [Add Tag (タグの追加)] を選択して、タグをスナップショットに追加します。タグごとに、タグキーとタグの値を指定します。
9. [スナップショットの作成] を選択します。

スナップショットの作成中、スナップショットはまとめて管理されます。ボリュームセット内のスナップショットの 1 つが失敗すると、他のスナップショットはそのボリュームセットのエラーステータスに移動します。スナップショットの進行状況は、[CloudWatch イベント](#) を使用してモニタリングできます。スナップショット作成プロセスが完了すると、CloudWatch は影響を受けるインスタンスのステータスと関連するすべてのスナップショットの詳細を含むイベントを生成します。

## コマンドラインを使用してマルチボリュームスナップショットを作成するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `create-snapshots` (AWS CLI)
- `New-EC2SnapshotBatch` (AWS Tools for Windows PowerShell)

## Amazon EBS スナップショットの削除

スナップショットを削除すると、そのスナップショットのみが参照するデータのみが削除されます。一意のデータは、そのデータを参照するすべてのスナップショットを削除しない限り削除されません。ボリュームの過去のスナップショットを削除しても、それ以降のスナップショットからボリュームを復元する機能に影響することはありません。

ボリュームのスナップショットを削除しても、ボリュームには影響しません。ボリュームを削除しても、そのボリュームが作成したスナップショットには影響しません。

ボリュームのスナップショットを定期的に作成する場合、スナップショットは差分になります。最後にスナップショットを作成した時点から、デバイス上で変更があったブロックだけが、新しいスナップショットに保存されます。スナップショットの保存は増分ベースで行われるもの、最新のスナップショットさえあればボリュームを復元できるようにスナップショット削除プロセスは設計されています。以前のスナップショットに保持されていて、後でボリュームから削除される、そのボリュームに存在するデータも以前のスナップショットの一意のデータとみなされます。この一意のデータは、一意のデータを参照するすべてのスナップショットが削除されない限り、一連のスナップショットから削除されません。

スナップショットを削除しても、組織のデータストレージコストが減少しない場合があります。他のスナップショットはそのスナップショットのデータを参照する場合があります。参照されたデータは常に保持されます。後から作成したスナップショットが使用しているデータを含むスナップショットを削除す

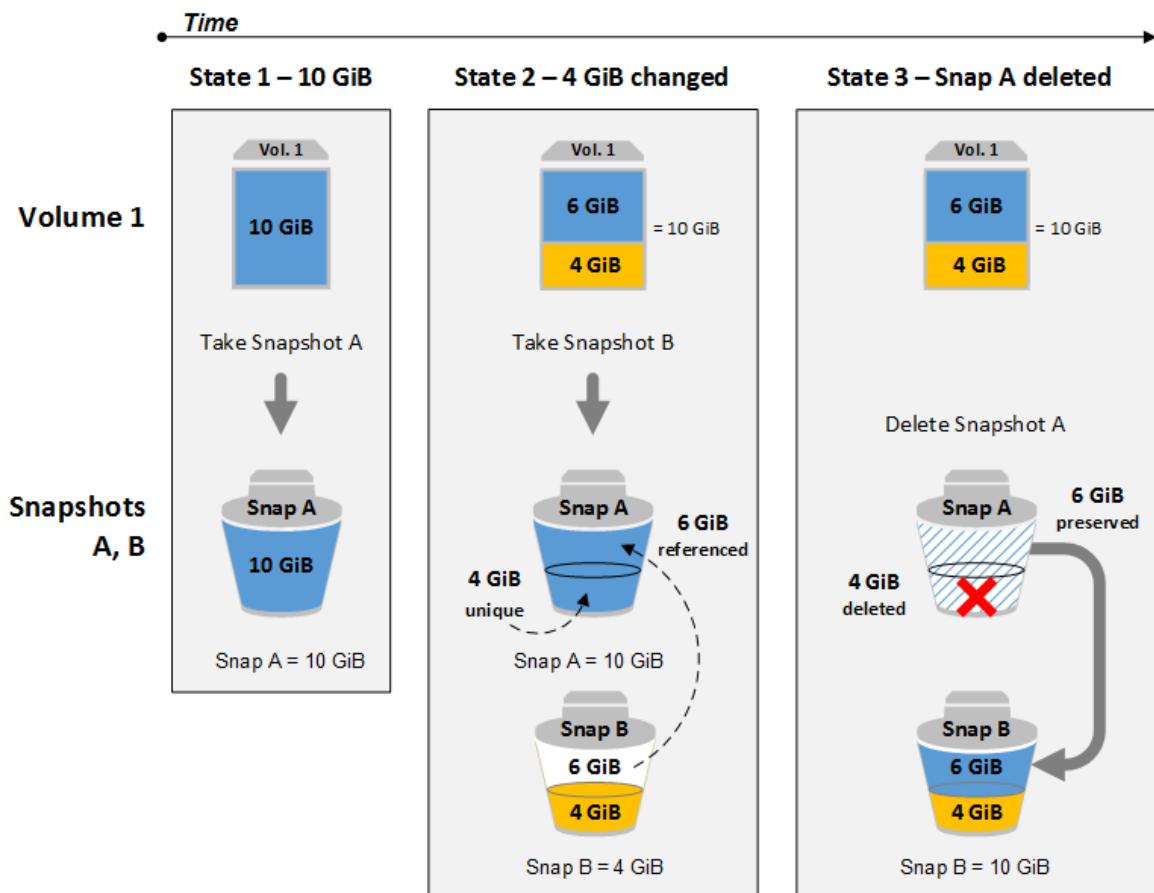
ると、参照されるデータに関連付けられたコストは後から作成されたスナップショットに割り当てられます。スナップショットがデータを格納する方法については、[増分スナップショットの仕組み \(p. 970\)](#) および以下の例を参照してください。

マルチボリュームスナップショットを削除するには、スナップショットを作成したときにグループに適用したタグを使用して、マルチボリュームグループのすべてのスナップショットを取得します。その後、スナップショットを個別に削除します。マルチボリュームスナップショットグループ内の個々のスナップショットの削除を妨げられることはできません。

次の図では、ボリューム 1 は 3 つの時点に示されています。スナップショットが最初の 2 つの状態をキャプチャし、3 つめでは、スナップショットが削除されています。

- 状態 1 では、ボリュームに 10 GiB のデータがあります。スナップ A がボリュームで作成された最初のスナップショットであるため、10 GiB のデータ全体をコピーする必要があります。
- 状態 2 では、ボリュームにはまだ 10 GiB のデータが含まれていますが、4 GiB 分が変更されました。スナップ B は、スナップ A が作成されたあとに変更された 4 GiB 分のみをコピーして格納する必要があります。すでにスナップ A にコピーおよび格納されている、変更がなかった残り 6 GiB 分のデータは、スナップ B によって、(再度) コピーされるのではなく、参照されます。これは点線の矢印によって示されます。
- 状態 3 では、ボリュームは状態 2 から変更されていませんが、スナップショット A は削除されています。スナップショット A に格納されている、スナップショット B によって参照された 6 GiB のデータは、大きい矢印に示されているように、スナップショット B に移動しました。その結果、未だに 10 GiB のデータを格納しなければなりません。スナップ A から保持されている、変更がなかった 6 GiB のデータと、スナップ B から変更があった 4 GiB のデータです。

例 1: 別のスナップショットによって一部のデータが参照されているスナップショットの削除



登録された AMI によって使用される EBS ボリュームのルートデバイスのスナップショットを削除することはできません。スナップショットを削除するには、まず AMI の登録を解除する必要があります。詳細については、「[Linux AMI の登録解除 \(p. 163\)](#)」を参照してください。

コンソールを使用してスナップショットを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Snapshots (スナップショット)] を選択します。
3. スナップショットを選択し、[Actions] リストから [Delete] を選択します。
4. [Yes, Delete] を選択します。

コマンドラインを使用してスナップショットを削除するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [delete-snapshot](#) (AWS CLI)
- [Remove-EC2Snapshot](#) (AWS Tools for Windows PowerShell)

#### Note

進行中のスナップショットを削除することはできますが、スナップショットが完了しなければ削除は実行されません。これには時間がかかる場合があります。同時スナップショット制限(5つのスナップショットが進行中)に達していて、追加のスナップショットを作成しようとした場合、`ConcurrentSnapshotLimitExceeded` エラーを受け取る場合があります。

## Amazon EBS スナップショットのコピー

Amazon EBS を使用すると、Amazon S3 に格納するボリュームのポイントインタイムスナップショットを作成できます。スナップショットを作成し、Amazon S3 へのコピーを完了すると(スナップショットの状態は [completed])、1 つの AWS リージョンから別のリージョン、または同じリージョン内にスナップショットをコピーできます。Amazon S3 サーバー側暗号化(256 ビット AES)により、スナップショットの送信中データがコピー操作中に保護されます。スナップショットコピーは、元のスナップショットの ID とは異なる ID を受け取ります。

マルチボリュームスナップショットを別の AWS リージョンにコピーするには、作成時にマルチボリュームスナップショットグループに適用したタグを使用してスナップショットを取得します。ツトを取得します。次に、スナップショットを個別に別のリージョンにコピーします。

Amazon RDS スナップショットのコピーについては、『Amazon RDS ユーザーガイド』の「[DB スナップショットのコピー](#)」を参照してください。

他のアカウントがスナップショットをコピーできるようにするには、スナップショットのアクセス許可を変更してそのアカウントへのアクセスを許可するか、スナップショットを公開してすべての AWS アカウントがスナップショットをコピーできるようにする必要があります。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

AWS リージョンおよびアカウント間におけるスナップショットのコピーの料金については、「[Amazon EBS 料金](#)」を参照してください。単一リージョンの単一アカウント内のスナップショットコピー操作では、スナップショットコピーの暗号化ステータスが変更しない限り、実際のデータはコピーされないため料金は発生しないことに注意してください。

#### Note

スナップショットを新しいリージョンにコピーする場合、完全な(増分なし)コピーが常に作成されるため、遅延とストレージコストがさらに生じる原因になります。

#### Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な（増分なし）コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

#### ユースケース

- 地理的拡張: アプリケーションを新しい AWS リージョンで起動します。
- 移行: アプリケーションを新しいリージョンに移動して、可用性を向上させ、コストを最小化します。
- 災害対策: 異なる地理的場所にまたがって定期的にデータをバックアップし、ログを記録します。災害が発生した場合、二次的なリージョンに格納された特定の時点でのバックアップを使用してアプリケーションを復元できます。これにより、データ損失と復旧時間を最小限に抑えることができます。
- 暗号化: 以前に暗号化されたスナップショットを暗号化し、スナップショットの暗号化に使用されるキーを変更します。または、他のユーザーから共有された暗号化スナップショットの場合、自分が所有するコピーを作成し、そのコピーからボリュームを復元できるようにします。
- データ保持および監査要件: 暗号化された EBS スナップショットを AWS アカウント間でコピーし、監査およびデータ保持のためにデータログや他のファイルを保持します。別のアカウントを使用すると、スナップショットの誤った削除を防止でき、メイン AWS アカウントが侵害された場合に保護できます。

#### 前提条件

- 共有スナップショットや作成したスナップショットを含む "completed" ステータスのアクセス可能な任意のスナップショットをコピーできます。
- また、AWS Marketplace、VM Import/Export、および AWS Storage Gateway のスナップショットをコピーできます。ただし、そのスナップショットがコピー先のリージョンでサポートされていることを確認する必要があります。

#### 制限

- 各アカウントは、1つのコピー先のリージョンに対して最大で 5 つの現行スナップショットコピーリクエストを行うことができます。
- ユーザー定義タグは元のスナップショットから新しいスナップショットにコピーされません。コピー操作中または操作後に、ユーザー定義タグを追加することができます。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。
- CopySnapshot アクションによって作成されたスナップショットの任意のボリューム ID はいずれの目的にも使用しないでください。

#### リージョンをまたがる増分コピー

スナップショットコピーが増分かどうかは、最近完了したスナップショットコピーによって決定されます。リージョンにわたってスナップショットをコピーする場合、次の条件に合致すればコピーは増分となります。

- スナップショットがコピー先のリージョンにコピーされたことがある。
- 最近のスナップショットコピーがコピー先のリージョンにまだ存在する。
- コピー先リージョンのスナップショットのコピーがすべて、暗号化されていないか、あるいは同じ CMK を使って暗号化されていた。

最近のスナップショットコピーが削除され、次のコピーがフルコピーである場合、増分コピーではありません。別のコピーを開始した時点で最初のコピーがまだ保留中の場合、2番目のコピーもフルコピーです。最初のコピーが完了したが、別のコピーを開始した時点で2番目のコピーがまだ保留中の場合、3番目のコピーは最初のコピーに対して増分です。

コピー先リージョンで最近のボリュームのスナップショットコピーをトラッキングできるよう、スナップショットにボリューム ID と作成時刻をタグ付けすること推奨します。

スナップショットコピーが増分かどうか確認するには、[copySnapshot \(p. 1071\)](#)CloudWatchイベントをチェックします。

## 暗号化とスナップショットのコピー

スナップショットをコピーする場合、コピーを暗号化するか、元のスナップショットとは異なる CMK を指定してコピーされたスナップショットで新しい CMK が使用されるようにもできます。ただし、コピー操作中にスナップショットの暗号化状態を変更すると、より大規模なデータ転送およびストレージ料金が発生する可能性がある完全コピー（増分ではない）が返されます。

他の AWS アカウントから暗号化されたスナップショットをコピーするには、スナップショットの暗号化に使用されたスナップショットとカスタマーマスターキー (CMK) の使用権限が必要です。共有された暗号化されたスナップショットを使用する場合は、自分が所有する CMK を使用してスナップショットを再暗号化することをお勧めします。これにより、元の CMK が侵害された場合、または所有者が取り消した場合に保護され、スナップショットを使用して作成した暗号化されたボリュームへのアクセスが失われる可能性があります。詳細については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

Encrypted パラメータを true に設定して、EBS スナップショットに暗号化を適用します。（デフォルトで暗号化 (p. 1017) が有効になっている場合、Encrypted パラメータはオプションです）。

オプションで、KmsKeyId を使用してスナップショットのコピーの暗号化に使用するカスタムキーを指定できます。（デフォルトで暗号化が有効になっている場合でも、Encrypted パラメータも true に設定する必要があります）。KmsKeyId が指定されていない場合、暗号化に使用されるキーはソーススナップショットの暗号化状態とその所有権によって異なります。

次の表では、考えられる設定の組み合わせごとの暗号化の結果について説明しています。

### 暗号化の結果: スナップショットのコピー

| Encrypted パラメータは設定されていますか？ | 暗号化はデフォルトで設定されていますか？ | ソーススナップショット                 | デフォルト (KmsKeyId は指定されていません) | カスタム (KmsKeyId が指定されています) |
|----------------------------|----------------------|-----------------------------|-----------------------------|---------------------------|
| いいえ                        | いいえ                  | 所有する暗号化されていないスナップショット       | 暗号化されていない                   | 該当なし                      |
| いいえ                        | いいえ                  | 所有する暗号化されたスナップショット          | 同じキーで暗号化されている               |                           |
| いいえ                        | いいえ                  | 自分と共有されている暗号化されていないスナップショット | 暗号化されていない                   |                           |
| いいえ                        | いいえ                  | 自分と共有されている暗号化されたスナップショット    | デフォルト CMK* で暗号化されている        |                           |
| はい                         | いいえ                  | 所有する暗号化されていないスナップショット       | デフォルト CMK で暗号化されている         | 指定された CMK** で暗号化されている     |
| はい                         | いいえ                  | 所有する暗号化されたスナップショット          | 同じキーで暗号化されている               |                           |

| <b>Encrypted</b><br>パラメータは<br>設定されていますか？ | 暗号化はデ<br>フォルトで設<br>定されていますか？ | ソーススナップ<br>ショット                     | デフォルト (KmsKeyId<br>は指定されていません) | カスタム (KmsKeyId<br>が指定されています) |
|------------------------------------------|------------------------------|-------------------------------------|--------------------------------|------------------------------|
| はい                                       | いいえ                          | 自分と共有されてい<br>る暗号化されていな<br>いスナップショット | デフォルト CMK で暗<br>号化されている        |                              |
| はい                                       | いいえ                          | 自分と共有されて<br>いる暗号化されたス<br>ナップショット    | デフォルト CMK で暗<br>号化されている        |                              |
| いいえ                                      | はい                           | 所有する暗号化され<br>ていないスナップ<br>ショット       | デフォルト CMK で暗<br>号化されている        | 該当なし                         |
| いいえ                                      | はい                           | 所有する暗号化され<br>たスナップショット              | 同じキーで暗号化され<br>ている              |                              |
| いいえ                                      | はい                           | 自分と共有されてい<br>る暗号化されていな<br>いスナップショット | デフォルト CMK で暗<br>号化されている        |                              |
| いいえ                                      | はい                           | 自分と共有されて<br>いる暗号化されたス<br>ナップショット    | デフォルト CMK で暗<br>号化されている        |                              |
| はい                                       | はい                           | 所有する暗号化され<br>ていないスナップ<br>ショット       | デフォルト CMK で暗<br>号化されている        | 指定された CMK で<br>暗号化されている      |
| はい                                       | はい                           | 所有する暗号化され<br>たスナップショット              | 同じキーで暗号化され<br>ている              |                              |
| はい                                       | はい                           | 自分と共有されてい<br>る暗号化されていな<br>いスナップショット | デフォルト CMK で暗<br>号化されている        |                              |
| はい                                       | はい                           | 自分と共有されて<br>いる暗号化されたス<br>ナップショット    | デフォルト CMK で暗<br>号化されている        |                              |

\* これは、AWS アカウントおよびリージョンでの EBS 暗号化に使用されるデフォルトの CMK です。これはデフォルトでは、EBS 用の一意の AWS 管理の CMK です。または、カスタマー管理の CMK を指定できます。詳細については、「[EBS 暗号化のデフォルトキー \(p. 1017\)](#)」を参照してください。

\*\* これは、コピーアクションに対して指定されたカスタマー管理の CMK です。この CMK は、AWS アカウントおよびリージョンのデフォルトの CMK の代わりに使用されます。

## スナップショットのコピー

Amazon EC2 コンソールを使用してスナップショットをコピーするには、次の手順を使用します。

コンソールを使用してスナップショットをコピーするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Snapshots] を選択します。

3. コピーするスナップショットを選択し、[Actions] リストから [Copy] を選択します。
4. [Copy Snapshot] ダイアログボックスで、必要に応じて次のものを更新します。
  - [Destination region (送信先リージョン)]: スナップショットのコピーを書き込むリージョンを選択します。
  - [Description]: デフォルトでは、スナップショットとコピーを見分けられるよう、元のスナップショットに関する情報が説明に含まれています。この説明は、必要に応じて変更できます。
  - 暗号化: 元のスナップショットが暗号化されていない場合は、コピーを暗号化することもできます。デフォルトで暗号化 (p. 1017) を有効にしている場合は、[Encryption (暗号化)] オプションが設定され、この設定をスナップショットコンソールから解除することはできません。[Encryption (暗号化)] オプションが設定されている場合は、以下で説明するように、フィールドで選択してカスタム CMK に暗号化することを選択できます。

暗号化されたスナップショットから暗号化を取り除くことはできません。

Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な (増分なし) コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

- [マスターキー]: このスナップショットの暗号化に使用されるカスタマーマスターキー (CMK)。アカウントのデフォルトキーが最初に表示されますが、必要に応じてアカウントのマスターキーから選択するか、別のアカウントのキーの ARN を入力するか貼り付けることができます。IAM コンソール <https://console.aws.amazon.com/iam/> に新規のマスター暗号化キーを作成することができます。
5. [Copy (コピー)] を選択します。
  6. [Copy Snapshot (スナップショットのコピー)] 確認ダイアログボックスで、[Snapshots (スナップショット)] を選択して指定したリージョンの [Snapshots (スナップショット)] ページに移動するか、[Close (閉じる)] を選択します。

コピー処理の進行状況を表示するには、コピー先のリージョンに切り替えて、[Snapshots (スナップショット)] ページを更新します。進行中のコピーがページの上部に一覧表示されます。

#### エラーをチェックするには

暗号化キーを使用する権限なしで、暗号化されたスナップショットをコピーしようとすると、メッセージが表示されずに操作に失敗します。ページを更新するまでエラー状態はコンソールに表示されません。次の例のように、コマンドラインからスナップショットの状態を確認することもできます。

```
aws ec2 describe-snapshots --snapshot-id snap-0123abcd
```

キー権限が足りないためにコピーに失敗した場合は、以下のメッセージが表示されます。"StateMessage": "「指定されたキー ID にアクセスできません」

暗号化されたスナップショットをコピーするときに、デフォルト CMK の DescribeKey 権限が必要です。明示的にこれらのアクセス許可を拒否するとコピーの障害が発生します。CMK キーの管理については、「[カスタマーマスターキーへのアクセスを制御する](#)」を参照してください。

#### コマンドラインを使用してスナップショットをコピーするには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [copy-snapshot \(AWS CLI\)](#)
- [Copy-EC2Snapshot \(AWS Tools for Windows PowerShell\)](#)

## Amazon EBS スナップショットに関する情報を表示する

スナップショットに関する詳細情報を表示できます。

コンソールを使用してスナップショットに関する詳細情報を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Snapshots (スナップショット)] を選択します。
3. リストを縮小するには、[Filter (フィルタ)] リストからオプションを選択します。たとえば、ご自分のスナップショットだけを表示するには、[Owned by Me (自分が所有)] を選択します。高度な検索オプションを使用して、スナップショットをさらにフィルタできます。使用可能なフィルタを表示するには、検索バーを選択します。
4. スナップショットの詳細情報を表示するには、そのスナップショットを選択します。

コマンドラインを使用してスナップショットに関する情報を表示するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `describe-snapshots` (AWS CLI)
- `Get-EC2Snapshot` (AWS Tools for Windows PowerShell)

## Amazon EBS スナップショットの共有

スナップショットの権限を変更することで、指定した AWS アカウントとスナップショットを共有できます。許可を受けたユーザーは、共有するスナップショットを元の EBS ボリュームを作成するための基礎として使用できますが、元のスナップショットは影響を受けません。

必要に応じて、暗号化されていないスナップショットをすべての AWS ユーザーに一般公開することもできます。暗号化されたスナップショットを公開することはできません。

暗号化されたスナップショットを共有する場合は、スナップショットの暗号化に使用するカスタマー管理の CMK も共有する必要があります。カスタマー管理の CMK を作成したときまたは後で CMK にクロスアカウント権限を適用することができます。

### Important

スナップショットを共有すると、スナップショットのすべてのデータに他人がアクセスできるようになります。スナップショットの共有は、自分のスナップショットデータすべてを共有したい人とだけ行ってください。

### 考慮事項

スナップショットの共有には、次の考慮事項が適用されます。

- スナップショットは、スナップショットが作成されたリージョンに制限されます。別のリージョンとスナップショットを共有するには、そのリージョンにスナップショットをコピーします。詳細については、「[Amazon EBS スナップショットのコピー \(p. 977\)](#)」を参照してください。
- スナップショットで長いリソース ID 形式を使用している場合は、長い ID をサポートする別のアカウントとのみ、これを共有できます。詳細については、「[リソース ID \(p. 1111\)](#)」を参照してください。
- AWS では、デフォルト CMK を使用して暗号化されたスナップショットを共有することができません。共有する予定のスナップショットは、代わりにカスタマー管理の CMK を使用して暗号化する必要があります。詳細については、『AWS Key Management Service Developer Guide』の「[キーの作成](#)」を参照してください。
- 暗号化されたスナップショットにアクセスしている共有 CMK のユーザーには、そのキーに対して `kms:DescribeKey`、`kms>CreateGrant`、`GenerateDataKey`、および `kms:ReEncrypt` の操作

を実行するためのアクセス許可が与えられている必要があります。詳細については、『AWS Key Management Service Developer Guide』の「[カスタマーマスターキーへのアクセスを制御する](#)」を参照してください。

## コンソールを使用して暗号化されていないスナップショットを共有する

コンソールを使用してスナップショットを共有するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Snapshots (スナップショット)] を選択します。
3. スナップショットを選択し、[アクション] リストから [Modify Permissions (権限の変更)] を選択します。
4. スナップショットを公開するか、次のように特定の AWS アカウントと共有します。
  - スナップショットを公開するには、[Public] を選択します。

このオプションは、暗号化されたスナップショットや AWS Marketplace 製品コードのスナップショットでは有効になりません。
  - 1つ以上の AWS アカウントでスナップショットを共有するには、[Private (プライベート)] を選択し、AWS アカウントの ID をハイフンなしで [AWS アカウント番号] フィールドに入力して、[Add Permission (アクセス許可の追加)] を選択します。追加の AWS アカウントでこの手順を繰り返します。
5. [Save (保存)] を選択します。

自分とプライベートに共有されている非暗号化されたスナップショットを使用するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Snapshots (スナップショット)] を選択します。
3. [プライベートスナップショット] フィルタを選択します。
4. ID または説明でスナップショットを見つけます。このスナップショットは、他の方法と同様に使用できます。たとえば、スナップショットからボリュームを作成したり、スナップショットを別のリージョンにコピーしたりすることができます。

## コンソールを使用して暗号化されているスナップショットを共有する

コンソールを使用して暗号化されたスナップショットを共有するには

1. AWS KMS コンソール (<https://console.aws.amazon.com/kms>) を開きます。
2. AWS リージョンを変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマー管理型のキー] を選択します。
4. [エイリアス] 列で、スナップショットの暗号化に使用したカスタマー管理キーのエイリアス (テキストリンク) を選択します。キーの詳細が新しいページで開きます。
5. [キー・ポリシー] セクションに、ポリシービューまたはデフォルトビューのいずれかが表示されます。ポリシービューは、キーポリシードキュメントを表示します。デフォルトビューは、[キー管理者]、[キーの削除]、[キーの使用]、[その他の AWS アカウント] の各セクションを表示します。デフォルトビューは、コンソールでポリシーを作成し、それをカスタマイズしていない場合に表示されます。デフォルトビューが使用できない場合は、ポリシービューでポリシーを手動で編集する必要があります。詳細については、AWS Key Management Service Developer Guide の「[キー・ポリシーの表示 \(コンソール\)](#)」を参照してください。

アクセスできるビューに応じて、ポリシービューまたはデフォルトビューのいずれかを使用し、次に示すように 1 つ以上の AWS アカウント ID をポリシーに追加します。

- (ポリシー) [編集] を選択します。1つ以上の AWS アカウント ID を "Allow use of the key" ステートメントと "Allow attachment of persistent resources" ステートメントに追加します。[Save changes] を選択します。次の例では、AWS アカウント ID 444455556666 をポリシーに追加します。

```
{  
    "Sid": "Allow use of the key",  
    "Effect": "Allow",  
    "Principal": {"AWS": [  
        "arn:aws:iam::111122223333:user/CMKUser",  
        "arn:aws:iam::444455556666:root"  
    ]},  
    "Action": [  
        "kms:Encrypt",  
        "kms:Decrypt",  
        "kms:ReEncrypt*",  
        "kms:GenerateDataKey*",  
        "kms:DescribeKey"  
    ],  
    "Resource": "*"  
},  
{  
    "Sid": "Allow attachment of persistent resources",  
    "Effect": "Allow",  
    "Principal": {"AWS": [  
        "arn:aws:iam::111122223333:user/CMKUser",  
        "arn:aws:iam::444455556666:root"  
    ]},  
    "Action": [  
        "kms>CreateGrant",  
        "kms>ListGrants",  
        "kms:RevokeGrant"  
    ],  
    "Resource": "*",  
    "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}  
}
```

- (デフォルトビュー) [その他の AWS アカウント] まで下にスクロールします。[別の AWS アカウントを追加] を選択し、プロンプトに従って AWS アカウント ID を入力します。別のアカウントを追加するには、[別の AWS アカウントを追加する] を選択し、AWS アカウント ID を入力します。すべての AWS アカウントを追加したら、[変更の保存] を選択します。

6. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
7. ナビゲーションペインで、[Snapshots (スナップショット)] を選択します。
8. スナップショットを選択し、[アクション] リストから [Modify Permissions (権限の変更)] を選択します。
9. AWS アカウントごとに、[AWS アカウント番号] に AWS アカウント ID を入力し、[Add Permission (アクセス許可の追加)] を選択します。すべての AWS アカウントを追加する場合は、[保存] を選択します。

#### 自分と共有されている暗号化されたスナップショットを使用するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Snapshots (スナップショット)] を選択します。
3. [プライベートスナップショット] フィルタを選択します。オプションで、[暗号化済み] フィルタを追加します。
4. ID または説明でスナップショットを見つけます。
5. スナップショットを選択して、[アクション]、[コピー] の順に選択します。
6. (オプション) 送信先リージョンを選択します。

7. スナップショットのコピーは、[Master Key (マスターキー)] フィールドに表示されているキーに暗号化されます。デフォルトでは、選択したキーはアカウントのデフォルトの CMK です。カスタマー管理の CMK を選択するには、入力ボックス内をクリックして利用可能なキーのリストを表示します。
8. [Copy (コピー)] を選択します。

## コマンドラインを使用してスナップショットを共有する

スナップショットのアクセス許可は、スナップショットの `createVolumePermission` 属性を使用して指定します。スナップショットを公開するには、グループを `all` に設定します。スナップショットを特定の AWS アカウントと共有するには、そのユーザーを AWS アカウントの ID に設定します。

コマンドラインを使用して、スナップショットのアクセス許可を変更するには

以下のいずれかのコマンドを使用します。

- [modify-snapshot-attribute](#) (AWS CLI)
- [Edit-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

コマンドラインを使用してスナップショットに関するアクセス許可を表示するには

以下のいずれかのコマンドを使用します。

- [describe-snapshot-attribute](#) (AWS CLI)
- [Get-EC2SnapshotAttribute](#) (AWS Tools for Windows PowerShell)

これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

## EBS スナップショットのコンテンツへのアクセス

Amazon Elastic Block Store (EBS) ダイレクト API を使用して EBS スナップショットのデータを直接読み取り、2 つのスナップショットの違いを特定できます。EBS スナップショット内のブロックの詳細を表示し、2 つのスナップショットのブロックの違いを比較して、スナップショット内のデータに直接アクセスできます。EBS のバックアップサービスを提供する独立系ソフトウェアベンダー (ISV) の場合は、EBS direct APIs を使用すると、EBS スナップショットを介した EBS ボリュームの増分変更をより容易にコスト効率よく追跡できます。これを行うために、EBS スナップショットから新しいボリュームを作成したり、EC2 インスタンスを使用して違いを比較したりする必要はありません。

このユーザーガイドでは、EBS direct APIs を構成する要素の概要と、これらの要素を効果的に使用する方法の例を示します。API のアクション、データ型、パラメータ、およびエラーの詳細については、[EBS direct APIs リファレンス](#) を参照してください。EBS direct APIs でサポートされる AWS リージョン、エンドポイント、およびサービスクオータの詳細については、AWS 全般のリファレンスの「[Amazon Elastic Block Store エンドポイントとクォータ](#)」を参照してください。

### 目次

- [EBS direct APIs の概要 \(p. 985\)](#)
- [IAM ユーザーのアクセス権限 \(p. 986\)](#)
- [コマンドラインによる EBS direct APIs の使用 \(p. 989\)](#)
- [API または AWS SDK による EBS direct APIs の使用 \(p. 991\)](#)
- [EBS direct APIs に関するよくある質問 \(FAQ\) \(p. 991\)](#)

## EBS direct APIs の概要

EBS direct APIs の使用を開始する前に、以下の主な要素を理解しておく必要があります。

## スナップショット

スナップショットは、EBS ボリュームからデータをバックアップするための主な手段です。ストレージコストを節約するために、連続するスナップショットは増分で、以前のスナップショット以降に変更されたボリュームデータのみが含まれています。詳細については、「[Amazon EBS スナップショット \(p. 970\)](#)」を参照してください。

### Note

パブリックスナップショットは、EBS direct APIs ではサポートされていません。

## ブロック

ブロックは、スナップショット内のデータのフラグメントです。各スナップショットには、何千ものブロックを含めることができます。スナップショット内のすべてのブロックは固定サイズです。

## ブロックインデックス

ブロックインデックスは、スナップショット内のブロックのオフセット位置であり、ブロックを識別するために使用されます。論理ブロック内のデータの論理オフセットを識別するには、`BlockIndex` 値に `BlockSize` 値を掛けます ( $\text{BlockIndex} * \text{BlockSize}$ )。

## ブロックトークン

ブロックトークンは、スナップショット内のブロックの識別ハッシュであり、ブロックデータの検索に使用されます。

### Note

EBS direct APIs から返されるブロックトークンは一時的なものです。ブロックトークンは、同じスナップショットの別の `ListSnapshotBlocks` リクエストや `ListChangedBlocks` リクエストを実行すると、変更されます。

## スナップショットブロックの一覧表示

`ListSnapshotBlocks` API オペレーションは、指定されたスナップショット内のブロックのブロックインデックスとブロックトークンを返します。詳細については、EBS direct APIs リファレンスの「[ListSnapshotBlocks](#)」を参照してください。

## 変更されたブロックの一覧表示

`ListChangedBlocks` API オペレーションは、同じボリューム/スナップショット系列の 2 つの指定されたスナップショット間で異なるブロックのブロックインデックスとブロックトークンを返します。詳細については、EBS direct APIs リファレンスの「[ListChangedBlocks](#)」を参照してください。

## スナップショットブロックの取得

`GetSnapshotBlock` API オペレーションは、指定されたスナップショット ID、ブロックインデックス、およびブロックトークンに関するブロック内のデータを返します。詳細については、EBS direct APIs リファレンスの「[GetSnapshotBlock](#)」を参照してください。

## API の使用

`ListSnapshotBlocks` または `ListChangedBlocks` API オペレーションを使用して、データを取得するブロックのブロックインデックスとブロックトークンを識別します。次に、`GetSnapshotBlock` API オペレーションを使用して、スナップショット内のブロックからデータを取得します。AWS CLI を使用してこれらの操作を実行する方法の例については、このガイドの後半で示します。

## IAM ユーザーのアクセス権限

IAM ユーザーが EBS direct APIs を使用するには、以下のポリシーが必要です。

### Important

以下のポリシーを IAM ユーザーに割り当てる際には注意が必要です。これらのポリシーを割り当てることで、EC2 API (CopySnapshot オペレーションや CreateVolume オペレーションなど) を介して、同じリソースへのアクセスが拒否されているユーザーにアクセスが許可される場合があります。

次のポリシーでは、EBS direct APIs へのフルアクセスを許可します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ebs:*"  
            ],  
            "Resource": "arn:aws:ec2:*:snapshot/*"  
        }  
    ]  
}
```

次のポリシーでは、特定の AWS リージョンにおいて特定のスナップショットへのアクセスを許可します。このポリシーで、<SnapshotID> はスナップショットの ID に置き換え、<Region> はスナップショットのリージョンに置き換えます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ebs>ListSnapshotBlocks"  
            ],  
            "Resource": "arn:aws:ec2:<Region>::snapshot/<SnapshotID>"  
        }  
    ]  
}
```

次のポリシーでは、特定のキー/値のタグを持つスナップショットへのアクセスを EBS direct APIs に許可します。このポリシーで、<Key> はタグのキー値に置き換え、<Value> はタグの値に置き換えます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ebs>ListChangedBlocks",  
                "ebs>ListSnapshotBlocks",  
                "ebs:GetSnapshotBlock"  
            ],  
            "Resource": "arn:aws:ec2:*:snapshot/*",  
            "Condition": {  
                "StringEqualsIgnoreCase": {  
                    "aws:ResourceTag/<Key>": "<Value>"  
                }  
            }  
        }  
    ]  
}
```

次のアクセス許可では、EBS direct APIs に対して、特定のスナップショットへのアクセスを拒否します。このポリシーで、**<SnapshotID>** はスナップショットの ID に置き換えます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ebs>ListSnapshotBlocks",  
                "ebs>ListChangedBlocks",  
                "ebs>GetSnapshotBlock"  
            ],  
            "Resource": "arn:aws:ec2:*:snapshot/*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": [  
                "ebs>ListSnapshotBlocks",  
                "ebs>ListChangedBlocks",  
                "ebs>GetSnapshotBlock"  
            ],  
            "Resource": "arn:aws:ec2:us-east-2:<SnapshotID>"  
        }  
    ]  
}
```

次のポリシーでは、特定の時間範囲内のすべてのスナップショットへのアクセスを許可します。このポリシーで、表示されている日時範囲は、必ずポリシーの日時範囲に置き換えます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ebs>ListChangedBlocks",  
                "ebs>ListSnapshotBlocks",  
                "ebs>GetSnapshotBlock"  
            ],  
            "Resource": "arn:aws:ec2:*:snapshot/*",  
            "Condition": {  
                "DateGreaterThan": {  
                    "aws:CurrentTime": "2018-05-29T00:00:00Z"  
                },  
                "DateLessThan": {  
                    "aws:CurrentTime": "2020-05-29T23:59:59Z"  
                }  
            }  
        }  
    ]  
}
```

次のポリシーでは、AWS Key Management Service (AWS KMS) の特定のキー ID を使用して、暗号化されたスナップショットを復号するためのアクセスを許可します。このポリシーで、**<AccountId>** は AWS KMS キーの AWS アカウントの ID に置き換え、**<KeyId>** は EBS direct APIs でアクセスするスナップショットの暗号化に使用されたキーの ID に置き換えます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "kmsDecrypt",  
                "kmsDescribeKey",  
                "kmsGetPublicKey",  
                "kmsListAliases",  
                "kmsListKeys",  
                "kmsListResourceTags",  
                "kmsPutResourcePolicy",  
                "kmsTagResource",  
                "kmsUntagResource",  
                "kmsUpdateKeyDescription",  
                "kmsUpdateKeyPolicy",  
                "kmsUpdateKeyState"  
            ],  
            "Resource": "arn:aws:kms:<Region>:<AccountId>:key/<KeyId>"  
        }  
    ]  
}
```

```
        "Sid": "VisualEditor0",
        "Effect": "Allow",
        "Action": [
            "kms:Decrypt",
            "kms:DescribeKey"
        ],
        "Resource": "arn:aws:kms:us-west-2:<AccountID>:key/<KeyId>"
    }
}
```

詳細については、IAM ユーザーガイドの「[IAM ユーザーのアクセス許可の変更](#)」を参照してください。

## コマンドラインによる EBS direct APIs の使用

以下の例では、AWS Command Line Interface (AWS CLI) で EBS direct APIs を使用する方法を示します。AWS CLI のインストールおよび設定の詳細については、「[AWS CLI バージョン 1 のインストール](#)」および「[AWS CLI のかんたん設定](#)」を参照してください。

Example 例: スナップショット内にあるブロックのブロックインデックスとブロックトークンを得する

次の list-snapshot-blocks コマンド例は、AWS リージョン **us-east-1** のスナップショット **snap-0987654321** 内にあるブロックのブロックインデックスとブロックトークンを返します。--starting-block-index パラメータおよび --max-results パラメータは、ブロックインデックスが **1000** より大きい最初の **100** ブロックに結果を制限します。

```
aws ebs list-snapshot-blocks --region us-east-1 --snapshot-id snap-0987654321 --starting-block-index 1000 --max-results 100
```

以下に、応答の例を示します。ブロック内のデータを取得するには、get-snapshot-block コマンドを使用して、ブロックのブロックインデックスとブロックトークンを指定します。ブロックトークンは、表示されている有効期限まで有効です。

```
{
    "Blocks": [
        {
            "BlockIndex": 1001,
            "BlockToken": "AAABAV3/PNhXOynVdMYHUpPsetaSvjLB1dtIGfbJv5OJ0sX855EzGTWos4a4"
        },
        {
            "BlockIndex": 1002,
            "BlockToken": "AAABATGQIgwr0WwIuqIMjCA/Sy7e/YoQFZsHejzGNvjKauzNgzeI13YHBfQB"
        },
        {
            "BlockIndex": 1007,
            "BlockToken": "AAABAZ9CTuQtUvp/dXqRWw4d07e0gTZ3jvn6hiW30W9duM8MiMw6yQayzF2c"
        },
        {
            "BlockIndex": 1012,
            "BlockToken": "AAABAQdzxhw0rVV6PNmsfo/YRIxo9JPR85XxPf1BLjg0Hec6pygYr6laE1p0"
        },
        {
            "BlockIndex": 1030,
            "BlockToken": "AAABAAyvPax6mv+iGWLdTUjQtFWouQ7Dqz6nSD9L+CbXnvpkswA6iDID523d"
        },
        {
            "BlockIndex": 1031,
            "BlockToken": "AAABATgWZC0XcFwUKvTJbUXMiSPg59KVxJGL+BWBClkw6spzCxJVqDVaTskJ"
        },
        ...
    ]
}
```

```
],
"ExpiryTime": 1576287332.806,
"VolumeSize": 32212254720,
"BlockSize": 524288
}
```

Example 例: 同じボリューム/スナップショット系列の 2 つのスナップショット間で異なるブロックのブロックインデックスとブロックトークンを取得する

次の `list-changed-blocks` コマンド例は、AWS リージョン `us-east-1` で、スナップショット `snap-1234567890` とスナップショット `snap-0987654321` 間で異なるブロックのブロックインデックスとブロックトークンを返します。`--starting-block-index` パラメータと `--max-results` パラメータは、ブロックインデックスが 0 より大きい最初の 500 ブロックに結果を制限します。

```
aws ebs list-changed-blocks --region us-east-1 --first-snapshot-id snap-1234567890 --second-snapshot-id snap-0987654321 --starting-block-index 0 --max-results 500
```

以下に、応答の例を示します。2 つのスナップショット間でブロックインデックス 0、6000、6001、6002、および 6003 が異なることを示しています。さらに、ブロックインデックス 6001、6002、および 6003 は、指定された最初のスナップショット ID にのみ存在し、2 番目のスナップショット ID には存在しません。これは、応答に 2 番目のブロックトークンが表示されないためです。

ブロック内のデータを取得するには、`get-snapshot-block` コマンドを使用して、ブロックのブロックインデックスとブロックトークンを指定します。ブロックトークンは、表示されている有効期限まで有効です。

```
{
    "ChangedBlocks": [
        {
            "BlockIndex": 0,
            "FirstBlockToken": "AAABAVahm9SO60Dyi0ORySzn2ZjGjW/KN3uygGls0QOYWesbzBbDnX2dGpmC",
            "SecondBlockToken":
                "AAABAf8o0o6UFi1rDbSZGIRaCEdDyBu9TlvtCQxxoKV8qrUPQP7vcM6iWGsr"
        },
        {
            "BlockIndex": 6000,
            "FirstBlockToken": "AAABAbYSiZvJ0/R9tz8suI8dSzecLjN4kkazK8inFXVintPkdaVFLfCMQsKe",
            "SecondBlockToken":
                "AAABAZnqTdzFmKRpsaMAsDxviVqeI/3jJzI2crq2eFDCgHmyNf777elD9oVR"
        },
        {
            "BlockIndex": 6001,
            "FirstBlockToken": "AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jb5Q6FRXFqAIAqE04hJoR"
        },
        {
            "BlockIndex": 6002,
            "FirstBlockToken": "AAABASqX4/NWjvNceoyMULjcRd0DnwbSwNnes1UkoP62CrQXvn47BY5435aw"
        },
        {
            "BlockIndex": 6003,
            "FirstBlockToken":
                "AAABASmJ005JxAOce25rF4P1sdRtyIDsX12tFEDunnePYUKOf4PBROuiCb2A"
        },
        ...
    ],
    "ExpiryTime": 1576308931.973,
    "VolumeSize": 32212254720,
```

```
"BlockSize": 524288,  
"NextToken": "AAADARqElNng/sV98CYk/bJDCXeLJmLJHnNSkHvLzVaO0zsPH/QM3Bi3zF//O6Mdi/  
BbJarBnp8h"  
}
```

### Example 例: ブロック内のデータを取得する

次の get-snapshot-block コマンド例は、AWS リージョン **us-east-1** で、スナップショット **snap-1234567890** 内のブロックインデックスが **6001**、ブロックインデックスが **AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR** のブロックのデータを返します。バイナリデータは、Windows コンピュータの **C:\Temp** ディレクトリ内の **output.txt** ファイルに出力されます。Linux または UNIX コンピュータでコマンドを実行する場合は、出力バスを **/tmp**、**output.txt** に置き換えて、データを **/tmp** ディレクトリ内の **output.txt** ファイルに出力します。

```
aws ebs get-snapshot-block --region us-east-1 --snapshot-id snap-1234567890 --block-index 6001 --block-token AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR C:/  
Temp/output.txt
```

以下に、応答の例を示します。返されるデータのサイズ、データを検証するためのチェックサム、チェックサムの生成に使用されたチェックサムアルゴリズムが表示されます。バイナリデータは、リクエストコマンドで指定したディレクトリとファイルに自動的に保存されます。

```
{  
    "DataLength": "524288",  
    "Checksum": "cf0Y6/Fn0oFa4VyjQPOa/iD0zhTf1PTKzxGv2OKowXc=",  
    "ChecksumAlgorithm": "SHA256"  
}
```

## API または AWS SDK による EBS direct APIs の使用

サービスの各アクションとデータ型については、[EBS direct APIs リファレンス](#)に説明と構文があります。AWS SDK の 1 つを使用して、使用しているプログラミング言語またはプラットフォームにカスタマイズした API にアクセスすることもできます。詳細については、[AWS SDK](#) を参照してください。

EBS direct APIs には、AWS 署名バージョン 4 の署名が必要です。これらの署名の作成の詳細については、AWS 全般のリファレンスで「[署名バージョン 4 署名プロセス](#)」を参照してください。

HTTP リクエストの署名方法を知る必要があるのは、手動で HTTP リクエストを作成する場合のみです。AWS コマンドラインインターフェイス (AWS CLI) またはいずれかの AWS SDK を使用して AWS へのリクエストを作成する場合、これらのツールにより、ツールの設定時に指定したアクセキーを使用して自動的にリクエストが署名されます。これらのツールを使う場合は、自分でリクエストに署名する方法を学ぶ必要はありません。

## EBS direct APIs に関するよくある質問 (FAQ)

ステータスが保留中になっているスナップショットに、EBS direct APIs からアクセスできますか？

いいえ。スナップショットは、ステータスが「完了」の場合のみアクセスできます。

ブロックインデックスは、EBS direct APIs から数値順に返されますか？

はい。返されるブロックインデックスは一意で、数値順になっています。

**MaxResults** パラメータ値が 100 未満のリクエストを送信できますか？

いいえ。使用できる **MaxResult** パラメータの最小値は 100 です。**MaxResult** パラメータ値が 100 未満のリクエストを送信し、スナップショット内に 100 を超えるブロックがあった場合、API は最低 100 の結果を返します。

複数の API リクエストを同時に実行できますか？

API リクエストは複数を同時に実行できます。ボトルネックを避けるために、アカウントで実行されている可能性のある他のワークロードを考慮してください。また、EBS direct APIs ワークフロー内に再試行メカニズムを組み込んで、スロットル、タイムアウト、およびサービスの使用不能を処理する必要があります。

**ListChangedBlocks** オペレーションを実行したときに、スナップショット内にロックがあっても空の応答が返されることがありますか？

はい。変更されたロックがスナップショット内でまばらである場合、応答は空になる場合があります。ただし、API は次ページのトークン値を返します。次ページのトークン値を使用して、結果の次ページに進みます。API から返された次ページのトークン値が null である場合は、結果の最終ページに達したことを確認できます。

**NextToken** パラメータと **StartingBlockIndex** パラメータと一緒に指定した場合、どちらのパラメータが使用されますか？

**NextToken** が使用され、**StartingBlockIndex** は無視されます。

ロックトークンとネクストトークンの有効期間はどれくらいですか？

ロックトークンの有効期間は 7 日で、ネクストトークンの有効期間は 60 分です。

暗号化されたスナップショットはサポートされますか？

はい。暗号化されたスナップショットには、EBS ダイレクト API を使用してアクセスできます。

暗号化されたスナップショットにアクセスするには、スナップショットの暗号化に使用されたキーおよび AWS KMS 復号オペレーションが必要です。ユーザーに割り当てる AWS KMS ポリシーについては、このガイドの前半にある「

[ListSnapshotBlocks](#) または [ListChangedBlocks](#) API オペレーションを使用して、データを取得するロックのロックインデックスとロックトークンを識別します。次に、[GetSnapshotBlock](#) API オペレーションを使用して、スナップショット内のロックからデータを取得します。AWS CLI を使用してこれらの操作を実行する方法の例については、このガイドの後半で示します。

(p. 1)」セクションを参照してください。

パブリックスナップショットはサポートされていますか？

いいえ。パブリックスナップショットはサポートされていません。

スナップショットロックのリストは、スナップショット内のすべてのロックインデックスとロックトークンを返すのですか、それともデータが書き込まれたものだけを返すのですか？

データが書き込まれたロックインデックスとロックトークンのみを返します。

## Amazon EBS スナップショットライフサイクルの自動化

Amazon Data Lifecycle Manager を使用して、Amazon EBS ボリュームをバックアップするスナップショットの作成、保持、削除を自動化できます。スナップショット管理を自動化すると、次のことが可能になります。

- 定期的なバックアップスケジュールを実施して貴重なデータを保護する。
- 監査担当者または社内のコンプライアンスが必要とするバックアップを保持する。
- 古いバックアップを削除してストレージコストを削減する。

Amazon CloudWatch Events と AWS CloudTrail のモニタリング機能と組み合わせることで、Amazon Data Lifecycle Manager は EBS ボリューム用の完全バックアップソリューションを追加コストなしで提供します。

## コンテンツ

- [Amazon Data Lifecycle Managerの仕組み \(p. 993\)](#)
- [Amazon Data Lifecycle Managerに関する考慮事項 \(p. 994\)](#)
- [前提条件 \(p. 995\)](#)
- [コンソールを使用したバックアップの管理 \(p. 997\)](#)
- [AWS CLI を使用したバックアップの管理 \(p. 999\)](#)
- [API を使用したバックアップの管理 \(p. 1002\)](#)
- [スナップショットライフサイクルの監視 \(p. 1002\)](#)

## Amazon Data Lifecycle Managerの仕組み

以下はAmazon Data Lifecycle Managerの主要な要素です。

### 要素

- [スナップショット \(p. 993\)](#)
- [ターゲットのリソースタグ \(p. 993\)](#)
- [スナップショットタグ \(p. 993\)](#)
- [ライフサイクルポリシー \(p. 994\)](#)

## スナップショット

スナップショットは、EBS ボリュームからデータをバックアップするための主な手段です。ストレージコストを節約するために、連続するスナップショットは増分で、以前のスナップショット以降に変更されたボリュームデータのみが含まれています。ボリュームの一連のスナップショットでスナップショットを1つ削除すると、そのスナップショットに固有のデータだけが削除されます。キャプチャされたボリュームの残りの部分は保存されます。

詳細については、「[Amazon EBS スナップショット \(p. 970\)](#)」を参照してください。

### ターゲットのリソースタグ

Amazon Data Lifecycle Manager はリソースタグを使用して、バックアップする EBS ボリュームを識別します。タグは、AWS リソース (EBS ボリュームとスナップショットを含む) に割り当てる事のできるカスタマイズ可能なメタデータです。Amazon Data Lifecycle Manager ポリシー (以下で説明) は、1 つのタグを使用するバックアップのボリュームをターゲットとします。ボリュームで複数のポリシーを実行する場合は、複数のタグをボリュームに割り当てることができます。

タグキーに「\」または「=」文字を使用することはできません。

詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)」を参照してください。

### スナップショットタグ

Amazon Data Lifecycle Manager は、ポリシーによって作成されたすべてのスナップショットに次のタグを適用し、他の方法で作成されたスナップショットと区別します。

- `aws:dlm:lifecycle-policy-id`
- `aws:dlm:lifecycle-schedule-name`

作成時に、スナップショットに適用するカスタムタグを指定することもできます。

タグキーに「\」または「=」文字を使用することはできません。

Amazon Data Lifecycle Manager がボリュームをポリシーに関連付けるために使用するターゲットタグは、オプションで、ポリシーによって作成されたスナップショットに適用できます。

## ライフサイクルポリシー

ライフサイクルポリシーは、以下のコア設定で構成されています。

- リソースタイプ—ポリシーに基づき管理される AWS リソースのタイプ。VOLUME を使用して個々のボリュームのスナップショットを作成するか、INSTANCE を使用してインスタンスのボリュームからマルチボリュームスナップショットを作成します。詳細については、「[マルチボリュームスナップショット \(p. 973\)](#)」を参照してください。
- ターゲットタグ—ポリシーに基づき管理されるようにするには EBS ボリュームまたは EC2 インスタンスに関連付ける必要のあるタグ。
- スケジュール—スナップショット作成の開始時刻や間隔。
- 保持—スナップショットの合計数または各スナップショットの保存期間に基づいて、スナップショットを保持できます。

たとえば、account=Finance タグですべての EBS ボリュームを管理し、09:00 に 24 時間ごとのスナップショットを作成し、最新の 5 つのスナップショットを保持するポリシーを作成できます。スナップショットの作成は、遅くとも 09:59 から開始できます。

## Amazon Data Lifecycle Managerに関する考慮事項

お客様の AWS アカウントには、Amazon Data Lifecycle Manager に関する以下のようなクォータがあります。

- リージョンごとに最大 100 のライフサイクルポリシーを作成できます。
- リソースごとに最大 45 個のタグを追加できます。
- ライフサイクルポリシーごとに 1 つのスケジュールを作成できます。

ライフサイクルポリシーには、次の考慮事項が適用されます。

- アクティベーションステータスを有効に設定するまで、ポリシーはスナップショットの作成を開始しません。作成時にポリシーを有効にするように設定できます。
- 最初のスナップショットは、対応するポリシーに基づき、指定開始時刻から 1 時間以内に作成されます。
- ターゲットタグを削除または変更してポリシーを変更すると、そのタグが付いている EBS ボリュームはポリシーの影響を受けなくなります。
- ポリシーのスケジュール名を変更すると、古いスケジュール名で作成されたスナップショットはポリシーの影響を受けなくなります。
- 時刻ベースの保持スケジュールを、新たな時間間隔を使用するスケジュールに修正すると、新たな時間間隔は、その後新たに作成されるスナップショットにのみ適用されます。新たなスケジュールが、このポリシーに基づき作成された既存スナップショットの保持スケジュールに影響を及ぼすことはありません。
- ポリシーの保持スケジュールを、スナップショットの数から各スナップショットの保存期間に変更することはできません。この変更を行うには、新たなポリシーを作成する必要があります。
- 各スナップショットの保存期間に基づく保持スケジュールを持つポリシーを無効にすると、ポリシーが無効になっている間に保持期間が終了したスナップショットは無期限に保持されます。これらのスナップショットは手動で削除する必要があります。ポリシーを再度有効にすると、Amazon Data Lifecycle Manager は保持期間の終了時にスナップショットの削除を再開します。
- ポリシーの適用先のリソースを削除すると、以前に作成したスナップショットはポリシーで管理されなくなります。不要になったスナップショットは手動で削除する必要があります。
- EBS ボリュームまたは EC2 インスタンスをバックアップするために複数のポリシーを作成できます。たとえば、EBS ボリュームに 2 つのタグがあり、タグ A が 12 時間ごとにスナップショットを作成する

ポリシー A のターゲットで、タグ B が 24 時間ごとにスナップショットを作成するポリシー B のターゲットである場合、Amazon Data Lifecycle Manager は両方のポリシーのスケジュールに従ってスナップショットを作成します。

以下は、ライフサイクルポリシーや[高速スナップショット復元 \(p. 1024\)](#)に関する考慮事項です。

- 高速スナップショット復元が有効化されているスナップショットについては、対応するライフサイクルポリシーを削除もしくは無効化した場合、対応するライフサイクルポリシーの高速スナップショット復元を無効化した場合、または対応するアベイラビリティーゾーンの高速スナップショット復元を無効化した場合であっても、当該復元は有効に保たれます。このようなスナップショットについては、手動で高速スナップショット復元を無効化できます。
- 高速スナップショット復元の有効化中に、有効化できるスナップショットの最大数を超えると、Amazon Data Lifecycle Manager はスケジュールに沿ったスナップショット作成は行うものの、作成したスナップショットの高速スナップショット復元は有効化しません。高速スナップショット復元が有効化されているスナップショットが削除されると、その次に Amazon Data Lifecycle Manager が作成するスナップショットの高速スナップショット復元が有効化されます。
- あるスナップショットの高速スナップショット復元を有効化すると、当該スナップショットが最適化されるまでに、TiBあたり 60 分の時間がかかります。弊社は、Amazon Data Lifecycle Manager が次のスナップショットを作成する前に各スナップショットが完全に最適化されるようなスケジュールの作成を推奨しています。

ライフサイクルポリシーおよび[マルチアタッチ \(p. 953\)](#) が有効なボリュームには、次の考慮事項が適用されます。

- マルチボリュームスナップショットのインスタンスタグに基づいてライフサイクルポリシーを作成する場合、Amazon Data Lifecycle Manager はアタッチされたインスタンスごとにボリュームのスナップショットを開始します。timestamp タグを使用して、アタッチされたインスタンスから作成された時間整合性のあるスナップショットのセットを識別します。

## 前提条件

以下は、Amazon Data Lifecycle Manager 使用の前提条件です。

### 前提条件

- [Amazon Data Lifecycle Manager アクセス許可 \(p. 995\)](#)
- [IAM ユーザーのアクセス権限 \(p. 996\)](#)

### Amazon Data Lifecycle Manager アクセス許可

Amazon Data Lifecycle Manager は IAM ロールを使用して、ユーザーの代わりにスナップショットを管理するために必要なアクセス許可を取得します。Amazon Data Lifecycle Manager は、最初に AWS マネジメントコンソールを使用してライフサイクルポリシーを作成するときに、[AWSDataLifecycleManagerDefaultRole] ロールを作成します。次のように、`create-default-role` コマンドを使用してこのロールを作成することもできます。

```
aws dlm create-default-role
```

また、必要なアクセス権限を持つカスタム IAM ロールを作成し、ライフサイクルポリシーを作成するときにはそのロールを選択することもできます。

カスタム IAM ロールを作成するには

- 次のアクセス許可でロールを作成します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateSnapshot",  
                "ec2:CreateSnapshots",  
                "ec2>DeleteSnapshot",  
                "ec2:DescribeVolumes",  
                "ec2:DescribeInstances",  
                "ec2:DescribeSnapshots"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateTags"  
            ],  
            "Resource": "arn:aws:ec2:*::snapshot/*"  
        }  
    ]  
}
```

詳細については、『IAM ユーザーガイド』の「[ロールの作成](#)」を参照してください。

2. ロールに信頼関係を追加します。
  - a. IAM コンソールで、[ロール] を選択します。
  - b. 作成したロールを選択し、[信頼関係] を選択します。
  - c. [信頼関係の編集] を選択して、次のポリシーを追加し、[信頼ポリシーの更新] を選択します。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "dlm.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

## IAM ユーザーのアクセス権限

IAM ユーザーが Amazon Data Lifecycle Manager を使用するには、次のアクセス許可が必要です。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": ["iam:PassRole", "iam>ListRoles"],  
            "Resource": "arn:aws:iam::123456789012:role/AWSDataLifecycleManagerDefaultRole"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iam:ListGroups",  
            "Resource": "arn:aws:iam::123456789012:group/AWSDataLifecycleManagerDefaultGroup"  
        }  
    ]  
}
```

```
        "Action": "dlm:*",
        "Resource": "*"
    }
}
```

詳細については、『IAM ユーザーガイド』の「[IAM ユーザーのアクセス許可の変更](#)」を参照してください。

## コンソールを使用したバックアップの管理

次の例は、Amazon Data Lifecycle Manager を使用して、EBS ボリュームのバックアップを AWS マネジメントコンソールで管理する方法を示しています。

### タスク

- [ライフサイクルポリシーの作成 \(p. 997\)](#)
- [ライフサイクルポリシーの表示 \(p. 998\)](#)
- [ライフサイクルポリシーの変更 \(p. 998\)](#)
- [ライフサイクルポリシーの削除 \(p. 998\)](#)

### ライフサイクルポリシーの作成

次の手順に従ってライフサイクルポリシーを作成します。

#### ライフサイクルポリシーを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Elastic Block Store]、[Lifecycle Manager]、[スナップショットライフサイクルポリシーの作成] の順に選択します。
3. 必要に応じて、ポリシーに次の情報を入力します。
  - [説明] - ポリシーの説明。
  - [リソースタイプ] - バックアップするリソースのタイプ。VOLUME を使用して個々のボリュームのスナップショットを作成するか、INSTANCE を使用してインスタンスのボリュームからマルチボリュームスナップショットを作成します。
  - [タグ付きターゲット] - バックアップするボリュームまたはインスタンスの識別情報となるリソースタグ。
  - [ライフサイクルポリシータグ] - ライフサイクルポリシーのタグ。
  - [スケジュール名] - スケジュールの名称。
  - [n 時間ごとにポリシーを実行] - ポリシー実行の時間間隔。サポートされている値は 1、2、3、4、6、8、12、24 です。
  - [XX 時XX 分 (UTC) に開始] - ポリシー実行の開始予定時刻。初回のポリシー実行は、予定時刻から 1 時間以内に開始されます。
  - 保持 - スナップショットの合計数または各スナップショットの保存期間に基づいて、スナップショットを保持できます。数に基づく保持の場合、数の範囲は 1 ~ 1000 です。この最大数に達すると、新しいスナップショットの作成時に最も古いスナップショットが削除されます。保存期間に基づく保持の場合、保存時間の範囲は 1 ~ 100 年です。各スナップショットの保持期間が終了すると、スナップショットは削除されます。保持期間は、作成間隔以上でなければなりません。
  - [リージョン間コピー] - 最大で 3 つの別リージョンに各スナップショットをコピーできます。リージョンごとに、異なる保持ポリシーを選択できます。また、すべてのタグをコピーするか、いずれのタグもコピーしないかを選択できます。ソーススナップショットが暗号化されている場合、または暗号化がデフォルトで有効化されている場合には、コピーースナップショットも暗号化されます。ソーススナップショットが暗号化されていない場合には、暗号化できます。CMK を指定しない場

合、スナップショットは、各コピー先リージョンにおける EBS 暗号化用のデフォルトキーを使用して暗号化されます。リージョンごとのスナップショットの同時コピー数を超えないようにする必要があります。

- [タグ付け情報] –ソースボリュームに付いているすべてのユーザー定義タグを、このポリシーに基づき作成されるスナップショットにコピーするかどうかを選択できます。Amazon Data Lifecycle Manager が適用するタグに加え、該当スナップショット用の追加タグを指定することもできます。リソースタイプが INSTANCE の場合には、`instance-id` および `timestamp` の変数タグを使用してスナップショットに自動的にタグを付けることを選択できます。変数タグの値は、タグの追加時に決定されます。
- [高速スナップショット復元] –高速スナップショット復元を有効化するかどうかや、どのアベイラビリティーゾーンにおいて有効化するかを選択できます。高速スナップショット復元を有効化できるスナップショットの最大数も指定できます。
- [IAM ロール] –スナップショットを作成、削除および記述できる権限と、ボリュームを記述できる権限を持つ IAM ロール。AWS からデフォルトロール `AWSDataLifecycleManagerDefaultRole` が提供されます。または、カスタム IAM ロールを作成できます。
- [作成後のポリシー状態] –[ポリシーの有効化] を選択して、次の予定時刻にポリシー実行が開始されるようにするか、[ポリシーの無効化] を選択してポリシーが実行されないようにすることができます。

4. [Create Policy (ポリシーの作成)] を選択します。

## ライフサイクルポリシーの表示

次の手順に従ってライフサイクルポリシーを表示します。

### ライフサイクルポリシーを表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic Block Store]、[ライフサイクルマネージャー] の順に選択します。
3. リストからライフサイクルポリシーを選択します。[詳細] タブには、ポリシーに関する情報が表示されます。

## ライフサイクルポリシーの変更

次の手順に従ってライフサイクルポリシーを変更します。

### ライフサイクルポリシーを変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic Block Store]、[ライフサイクルマネージャー] の順に選択します。
3. リストからライフサイクルポリシーを選択します。
4. [アクション]、[スナップショットライフサイクルポリシーの修正] の順に選択します。
5. 必要に応じてポリシー設定を修正します。具体例を挙げると、スケジュールを修正する、タグを追加もしくは削除する、またはポリシーを有効化もしくは無効化することができます。
6. [ポリシーの更新] を選択します。

## ライフサイクルポリシーの削除

次の手順に従ってライフサイクルポリシーを削除します。

### ライフサイクルポリシーを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで [Elastic Block Store]、[ライフサイクルマネージャー] の順に選択します。
3. リストからライフサイクルポリシーを選択します。
4. [アクション]、[スナップショットライフサイクルポリシーの削除] の順に選択します。
5. 確認を求められたら、[スナップショットライフサイクルポリシーの削除] を選択します。

## AWS CLI を使用したバックアップの管理

次の例は、Amazon Data Lifecycle Manager を使用して EBS ボリュームのバックアップを AWS CLI で管理する方法を示しています。

### 例

- ライフサイクルポリシーの作成 (p. 999)
- ライフサイクルポリシーの表示 (p. 1000)
- ライフサイクルポリシーの変更 (p. 1001)
- ライフサイクルポリシーの削除 (p. 1001)

### ライフサイクルポリシーの作成

ライフサイクルポリシーを作成するには、`create-lifecycle-policy` コマンドを使用します。構文を簡略化するために、この例では、ポリシーの詳細を含む JSON ファイル、`policyDetails.json` を使用しています。

この例では、VOLUME のリソースタイプを使用して、指定されたターゲットタグを持つすべてのボリュームのスナップショットを作成します。指定されたターゲットタグを持つすべてのインスタンスのすべてのボリュームのスナップショットを作成するには、代わりに INSTANCE のリソースタイプを使用します。

```
aws dlm create-lifecycle-policy --description "My volume policy" --state ENABLED --  
execution-role-arn arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole --  
policy-details file://policyDetails.json
```

次は、`policyDetails.json` ファイルの例です。

```
{  
    "ResourceTypes": [  
        "VOLUME"  
    ],  
    "TargetTags": [  
        {  
            "Key": "costcenter",  
            "Value": "115"  
        }  
    ],  
    "Schedules": [  
        {  
            "Name": "DailySnapshots",  
            "TagsToAdd": [  
                {  
                    "Key": "type",  
                    "Value": "myDailySnapshot"  
                }  
            ],  
            "CreateRule": {  
                "Interval": 24,  
                "IntervalUnit": "HOURS",  
                "Times": [  
                    "03:00"  
                ]  
            }  
        }  
    ]  
}
```

```
        ],
        "RetainRule": {
            "Count": 5
        },
        "CopyTags": false
    }
}
```

成功すると、このコマンドは新しく作成されたポリシーの ID を返します。出力例を次に示します。

```
{
    "PolicyId": "policy-0123456789abcdef0"
}
```

## ライフサイクルポリシーの表示

ライフサイクルポリシーに関する情報を表示するには、[get-lifecycle-policy](#) コマンドを使用します。

```
aws dlm get-lifecycle-policy --policy-id policy-0123456789abcdef0
```

出力例を次に示します。これには、指定した情報と AWS によって挿入されたメタデータが含まれます。

```
{
    "Policy": {
        "Description": "My first policy",
        "DateCreated": "2018-05-15T00:16:21+0000",
        "State": "ENABLED",
        "ExecutionRoleArn": "arn:aws:iam::210774411744:role/AWSDataLifecycleManagerDefaultRole",
        "PolicyId": "policy-0123456789abcdef0",
        "DateModified": "2018-05-15T00:16:22+0000",
        "PolicyDetails": {
            "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
            "ResourceTypes": [
                "VOLUME"
            ],
            "TargetTags": [
                {
                    "Value": "115",
                    "Key": "costcenter"
                }
            ],
            "Schedules": [
                {
                    "TagsToAdd": [
                        {
                            "Value": "myDailySnapshot",
                            "Key": "type"
                        }
                    ],
                    "RetainRule": {
                        "Count": 5
                    },
                    "CopyTags": false,
                    "CreateRule": {
                        "Interval": 24,
                        "IntervalUnit": "HOURS",
                        "Times": [
                            "03:00"
                        ]
                    }
                }
            ]
        }
    }
}
```

```
        },
        "Name": "DailySnapshots"
    }
}
}
```

## ライフサイクルポリシーの変更

ライフサイクルポリシーに関する情報を変更するには、[update-lifecycle-policy](#) コマンドを使用します。構文を簡略化するために、この例では、ポリシーの詳細を含む JSON ファイル、policyDetailsUpdated.json を参照しています。

```
aws dlm update-lifecycle-policy --state DISABLED --execution-role-arn
arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole" --policy-details
file://policyDetailsUpdated.json
```

次は、policyDetailsUpdated.json ファイルの例です。

```
{
    "ResourceTypes": [
        "VOLUME"
    ],
    "TargetTags": [
        {
            "Key": "costcenter",
            "Value": "120"
        }
    ],
    "Schedules": [
        {
            "Name": "DailySnapshots",
            "TagsToAdd": [
                {
                    "Key": "type",
                    "Value": "myDailySnapshot"
                }
            ],
            "CreateRule": {
                "Interval": 12,
                "IntervalUnit": "HOURS",
                "Times": [
                    "15:00"
                ]
            },
            "RetainRule": {
                "Count": 5
            },
            "CopyTags": false
        }
    ]
}
```

更新されたポリシーを表示するには、[get-lifecycle-policy](#) コマンドを使用します。状態、タグの値、スナップショットの間隔、およびスナップショットの開始時刻が変更されたことがわかります。

## ライフサイクルポリシーの削除

ライフサイクルポリシーを削除し、ポリシーで指定されたターゲットタグを解放して再利用できるようにするには、[delete-lifecycle-policy](#) コマンドを使用します。

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

## API を使用したバックアップの管理

Amazon Data Lifecycle Manager API リファレンスには、Amazon Data Lifecycle Manager クエリ API の各アクションとデータ型の説明と構文があります。

代わりに、いずれかの AWS SDK を使用して、使用中のプログラミング言語またはプラットフォームに対応する API にアクセスすることもできます。詳細については、[AWS SDK](#) を参照してください。

## スナップショットライフサイクルの監視

次の機能を使用して、スナップショットのライフサイクルをモニタリングできます。

### 機能

- [コンソールと AWS CLI \(p. 1002\)](#)
- [CloudWatch イベント \(p. 1002\)](#)
- [AWS CloudTrail \(p. 1003\)](#)

### コンソールと AWS CLI

ライフサイクルポリシーは、Amazon EC2 コンソールまたは AWS CLI を使用して表示できます。ポリシーによって作成された各スナップショットには、タイムスタンプとポリシー関連のタグがあります。タグを使用してスナップショットをフィルタリングして、意図したとおりにバックアップが作成されていることを確認できます。コンソールを使用したライフサイクルポリシーの表示の詳細については、「[ライフサイクルポリシーの表示 \(p. 998\)](#)」を参照してください。CLI を使用したライフサイクルポリシーに関する情報の表示の詳細については、「[ライフサイクルポリシーの表示 \(p. 1000\)](#)」を参照してください。

### CloudWatch イベント

Amazon EBS と Amazon Data Lifecycle Manager は、ライフサイクルポリシーアクションに関するイベントを発行します。AWS Lambda と Amazon CloudWatch Events を使用すると、プログラムによるイベント通知を処理できます。詳細については、「[Amazon CloudWatch Events ユーザーガイド](#)」を参照してください。

利用できるイベントは次のとおりです。

- `createSnapshot` – `CreateSnapshot` アクションが成功または失敗したときに生成される Amazon EBS イベント。詳細については、「[Amazon EBS での Amazon CloudWatch Events \(p. 1066\)](#)」を参照してください。
- `DLM Policy State Change` – ライフサイクルポリシーがエラー状態になったときに生成される Amazon Data Lifecycle Manager イベント。このイベントには、エラーを引き起こした原因の説明が含まれています。次に、IAM ロールによって付与されたアクセス権限が不十分な場合のイベントの例を示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "DLM Policy State Change",
  "source": "aws.dlm",
  "account": "123456789012",
  "time": "2018-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
  ],
}
```

```
"detail": {  
    "state": "ERROR",  
    "cause": "Role provided does not have sufficient permissions",  
    "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"  
}  
}
```

制限を超えた場合のイベントの例を次に示します。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "DLM Policy State Change",  
    "source": "aws.dlm",  
    "account": "123456789012",  
    "time": "2018-05-25T13:12:22Z",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"  
    ],  
    "detail":{  
        "state": "ERROR",  
        "cause": "Maximum allowed active snapshot limit exceeded",  
        "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"  
    }  
}
```

## AWS CloudTrail

AWS CloudTrail を使用すると、ユーザーのアクティビティや API の使用状況を追跡して、社内ポリシーや規制基準への準拠を実証することができます。詳細については、『[AWS CloudTrail User Guide](#)』を参照してください。

# Amazon EBS のデータサービス

Amazon EBS は以下のデータサービスを提供します。

## データサービス

- [Amazon EBS Elastic Volumes \(p. 1003\)](#)
- [Amazon EBS Encryption \(p. 1014\)](#)
- [Amazon EBS 高速スナップショット復元 \(p. 1024\)](#)

## Amazon EBS Elastic Volumes

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

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

## 目次

- [ボリューム変更時の要件 \(p. 1004\)](#)
- [EBS ボリュームの変更をリクエストする \(p. 1005\)](#)

- ボリューム変更の進行状況のモニタリング (p. 1008)
- ボリュームサイズ変更後の Linux ファイルシステムの拡張 (p. 1011)

## ボリューム変更時の要件

Amazon EBS ボリュームを変更すると、以下の要件と制約事項が適用されます。EBS ボリュームの一般的な要件についての詳細は、「[EBS ボリュームのサイズと設定の制限 \(p. 946\)](#)」を参照してください。

### Amazon EC2 インスタンスのサポート

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

- 現行世代のインスタンス (p. 184)
- 旧世代のインスタンスアミリー C1、C3、CC2、CR1、G2、I2、M1、M3、R3

インスタンスタイプが Elastic Volumes をサポートしていない場合は、「[Elastic Volumes がサポートされない場合は EBS ボリュームを変更する \(p. 1008\)](#)」を参照してください。

### Linux ボリュームの要件

Linux AMI では、2 TiB (2048 GiB) 以上のブートボリュームについて GUID パーティションテーブル (GPT) と GRUB 2 が必要です。現在の多くの Linux AMI は依然として MBR パーティションスキームを使用しており、2 TiB までのブートボリュームのみをサポートしています。インスタンスが 2 TiB を超えるブートボリュームで起動しない場合、使用中の AMI は、2 TiB のブートボリュームサイズに制限されている可能性があります。ブートボリューム以外のボリュームには、Linux インスタンスでこの制限はありません。Windows ボリュームに影響する要件については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows ボリュームの要件](#)」を参照してください。

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

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

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

```
GPT fdisk (gdisk) version 0.8.10

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

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

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

```
GPT fdisk (gdisk) version 0.8.8

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

## 制約事項

- Elastic Volume オペレーションは、マルチアタッチが有効な Amazon EBS ボリュームではサポートされません。
- 新しいボリュームサイズが、サポートされているボリュームのキャパシティーを超えることはできません。詳細については、「[EBS ボリュームのサイズと設定の制限 \(p. 946\)](#)」を参照してください。
- ボリュームが 2016 年 11 月 3 日 23:40 (UTC) 以前にアタッチされていた場合は、Elastic Volumes サポートを初期化する必要があります。詳細については、「[Elastic Volumes サポートの初期化 \(p. 1007\)](#)」を参照してください。
- サポートされていない前世代のインスタンスタイプを使用している場合や、ボリュームの変更を試みているときにエラーが発生した場合は、「[Elastic Volumes がサポートされていない場合は EBS ボリュームを変更する \(p. 1008\)](#)」を参照してください。
- ルートボリュームとしてインスタンスに接続されている gp2 ボリュームを st1 または sc1 ボリュームに変更することはできません。切り離して st1 または sc1 に変更した場合、インスタンスにルートボリュームとしてアタッチすることはできません。
- 要求されたボリュームサイズが st1 および sc1 ボリュームの最小サイズを下回る場合、gp2 ボリュームを st1 または sc1 ボリュームに変更することはできません。
- 変更を行うために、ボリュームのデタッチやインスタンスの停止が必要になる場合もあります。EBS ボリュームを変更する際にエラーメッセージが表示された場合や前世代のインスタンスタイプにアタッチされた EBS ボリュームを変更する場合は、以下のいずれかのステップを行ってください。
  - ルート以外のボリュームの場合は、ボリュームをインスタンスからデタッチして、変更を適用した後で、ボリュームを再アタッチします。
  - ルート(ブート)ボリュームの場合は、インスタンスを停止し、変更を適用した後で、インスタンスを再起動します。
- 既存の io1 ボリュームに 32,000 IOPS 以上のプロビジョニングを行ったら、パフォーマンスを最大限に引き出せるように、次のいずれかを実行する必要があります。
  - ボリュームをデタッチしてアタッチします。
  - インスタンスを再起動します。
- EBS ボリュームのサイズを小さくすることはできません。ただし、より小さなボリュームを作成し、そのボリュームに対して rsync などのアプリケーションレベルのツールを使用してデータを移行することができます。
- 完全に初期化されていないボリュームを変更する場合、変更時間は長くなります。詳細については、「[Amazon EBS ボリュームの初期化 \(p. 1049\)](#)」を参照してください。
- ボリュームを変更したら、6 時間以上待機してから、同じボリュームにさらに変更を加える前にそのボリュームの状態が in-use または available であることを確認してください。
- m3.medium インスタンスはボリュームの変更を完全にサポートしていますが、m3.large、m3.xlarge、および m3.2xlarge インスタンスでは、すべてのボリューム変更機能をサポートしているわけではありません。

## EBS ボリュームの変更をリクエストする

Elastic Volumes では、Amazon EBS ボリュームのサイズやパフォーマンス、ボリュームタイプをデタッチすることなく動的に変更することができます。

ボリュームを変更する場合は、次のプロセスで行います。

1. (オプション) 重要なデータを含むボリュームを変更する前に、変更をロールバックする必要がある場合に備えて、ボリュームのスナップショットを作成するのがベストプラクティスです。詳細については、「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」を参照してください。
2. ボリュームの変更をリクエストします。
3. ボリューム変更の進行状況をモニタリングします。詳細については、「[ボリューム変更の進行状況のモニタリング \(p. 1008\)](#)」を参照してください。

4. ボリュームのサイズが変更された場合、増加されたストレージ容量を利用するには、ボリュームのファイルシステムを拡張します。詳細については、[ボリュームサイズ変更後の Linux ファイルシステムの拡張 \(p. 1011\)](#)を参照してください。

## 目次

- [Elastic Volumes を使用して EBS ボリュームを変更する \(コンソール\) \(p. 1006\)](#)
- [Elastic Volumes を使用して EBS ボリュームを変更する \(AWS CLI\) \(p. 1006\)](#)
- [Elastic Volumes サポートの初期化 \(必要な場合\) \(p. 1007\)](#)
- [Elastic Volumes がサポートされていない場合は EBS ボリュームを変更する \(p. 1008\)](#)

### Elastic Volumes を使用して EBS ボリュームを変更する (コンソール)

以下の手順に従って、EBS ボリュームを変更します。

コンソールを使用して、EBS ボリュームを変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [Volumes] を選択し、変更するボリュームを選択したら、[Actions]、[Modify Volume] の順に選択します。
3. [Modify Volume] ウィンドウに、ボリューム ID とボリュームの現在の設定 (タイプ、サイズ、IOPS など) が表示されます。これらの設定のいずれかまたはすべてを 1 回のアクションで変更できます。新しい設定値を以下のように設定します。
  - タイプを変更するには、[Volume Type] の値を選択します。
  - サイズを変更するには、許可された整数値を [Size] に入力します。
  - ボリュームタイプとして [プロビジョンド IOPS SSD (io1)] を選択した場合は、許可された整数値を [IOPS] に入力します。
4. ボリューム設定を変更したら、[変更] を選択します。確認を求めるメッセージが表示されたら、[Yes] を選択します。
5. ボリュームサイズを変更しても、ボリュームのファイルシステムを拡張して新しいストレージ容量を利用するまでは、実際の効果はありません。詳細については、[ボリュームサイズ変更後の Linux ファイルシステムの拡張 \(p. 1011\)](#)を参照してください。

### Elastic Volumes を使用して EBS ボリュームを変更する (AWS CLI)

ボリュームの設定を 1 つ以上変更するには、`modify-volume` コマンドを使用します。たとえば、サイズが 100 GiB で、タイプが gp2 のボリュームがある場合は、次のコマンドでこの設定を 10,000 IOPS、サイズが 200 GiB のタイプ io1 のボリュームに変更します。

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

出力例を次に示します。

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

```
        "OriginalIops": 300,  
        "OriginalSize": 100  
    }  
}
```

ボリュームサイズを変更しても、ボリュームのファイルシステムを拡張して新しいストレージ容量を利用するまでは、実際の効果はありません。詳細については、[ボリュームサイズ変更後の Linux ファイルシステムの拡張 \(p. 1011\)](#)を参照してください。

### Elastic Volumes サポートの初期化 (必要な場合)

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

- ボリュームをデタッチしてアタッチする
- インスタンスの停止と起動

インスタンスでボリュームを変更する準備が完了していることを確認するには、次のいずれかの手順を使用します。

コンソールを使用してインスタンスの準備が完了していることを確認するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. [Show/Hide Columns] アイコン (歯車) を選択します。[Launch Time] および [Block Devices] 属性を選択し、[Close] を選択します。
4. [Launch Time] 列でインスタンスの一覧をソートします。カットオフ日以前に起動したインスタンスについては、デバイスがいつアタッチされたか確認します。次の例では、最初のインスタンスのボリューム変更を初期化する必要があります。これはカットオフ日よりも前に開始され、カットオフ日より前にそのルートボリュームがアタッチされていたためです。他のインスタンスはカットオフ日以降に開始されたため、準備は完了しています。

| Instance ID         | Launch Time                           | Block Devices                                                                                                                              |
|---------------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| i-e905622e          | February 25, 2016 at 1:49:35 PM UTC-8 | /dev/xvda=vol-e6b46410:attached:2016-02-25T21:49:35.000Z:true                                                                              |
| i-719f99a8          | December 8, 2016 at 2:21:51 PM UTC-8  | /dev/xvda=vol-bad60e7a:attached:2016-01-15T18:36:12.000Z:true                                                                              |
| i-006b02c1b78381e57 | May 17, 2017 at 1:52:52 PM UTC-7      | /dev/sda1=vol-0de9250441c73024c:attached:2017-05-17T20:52:53.000Z:true, xvdb=vol-0863a86c393496d3d:attached:2017-05-17T20:52:53.000Z:false |
| i-e3d172ed          | May 17, 2017 at 2:48:54 PM UTC-7      | /dev/sda1=vol-04c34d0b:attached:2015-01-21T21:19:46.000Z:true                                                                              |

CLI を使用してインスタンスの準備が完了していることを確認するには

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

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

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

|            |       |
|------------|-------|
| i-e905622e | True  |
| True       |       |
| i-719f99a8 | False |

```
True
i-006b02c1b78381e57      False
False
False
i-e3d172ed                False
True
```

### Elastic Volumes がサポートされていない場合は EBS ボリュームを変更する

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

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

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

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

新しく拡張したボリュームがサイズに反映されていない場合は、デバイスのファイルシステムを拡張して、インスタンスで新しいスペースを使えるようにします。詳細については、[ボリュームサイズ変更後の Linux ファイルシステムの拡張 \(p. 1011\)](#)を参照してください。

### ボリューム変更の進行状況のモニタリング

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

#### Note

まれに、一時的な AWS エラーのために failed 状態になる場合があります。これは、ボリュームのヘルスステータスを示すものではなく、ボリュームの変更に失敗したことを単に示しています。この場合は、再度ボリュームの変更を行います。

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

ボリュームの変更による影響は次のとおりです。

- 通常、ボリュームが Optimizing 状態になってから、サイズの変更が完了して反映されるまでには数秒かかります。
- パフォーマンス (IOPS) の変更は、設定の変更内容に応じて、完了するまでに数分から数時間かかる場合があります。
- 新しい設定が有効になるには最大 24 時間かかり、場合によっては (ボリュームが完全に初期化されない場合など) それ以上かかることがあります。通常、完全に使用された 1 TiB ボリュームが新しいパフォーマンス設定に移行するまでには約 6 時間かかります。

ボリュームの変更の進行状況をモニタリングするには、次のいずれかのメソッドを使用します。

#### 目次

- ボリューム変更の進行状況のモニタリング (コンソール) (p. 1009)
- ボリューム変更の進行状況のモニタリング (AWS CLI) (p. 1009)
- ボリューム変更の進行状況のモニタリング (CloudWatch イベント) (p. 1010)

## ボリューム変更の進行状況のモニタリング (コンソール)

1つ以上のボリュームの変更の進行状況を表示するには、次の手順を使用します。

コンソールを使用して変更の進行状況をモニタリングするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Volumes] を選択します。
3. ボリュームを選択します。ボリュームの変更状態は、詳細ペインの [状態] 列と [状態] フィールドに表示されます。次のスクリーンショットでは、選択したボリュームの変更状態が [optimizing] になっています。リスト内の次のボリュームは [modifying] 状態です。

### Note

[状態] 列と [状態] フィールドには、ボリュームの可用性ステータスも表示されます。このステータスは、[creating]、[available]、[in-use]、[deleting]、[deleted]、[error] のいずれかです。

4. 最新の変更アクションに関する前後の情報を表示するには、このスクリーンショットに示すように、[状態] フィールドのテキストを選択します。

The screenshot shows the AWS EC2 Volumes page. A specific volume, 'vol-02940f6ee433f...', is selected and highlighted with a blue border. In the 'Description' tab of the detailed view, under 'Volume modification details', it shows the current state as 'in-use - optimizing (1%)'. Other tabs include 'Status Checks', 'Monitoring', and 'Tags'.

| Name                 | Volume ID            | Size    | Volume Type | IOPS            | Snapshot               | Created    | Availability Zone          | State |
|----------------------|----------------------|---------|-------------|-----------------|------------------------|------------|----------------------------|-------|
| vol-0ddaa54cd90f5... | 8 GiB                | gp2     | 100         | snap-09aa45c... | January 9, 2020 at ... | eu-west-1b | in-use                     |       |
| vol-02940f6ee433f... | 16 GiB               | gp2     | 100         | snap-076d641... | January 9, 2020 at ... | eu-west-1c | in-use - optimizing (1%)   |       |
| Windows-ins...       | vol-0b01f92e8e62...  | 8 GiB   | gp2         | 100             | October 11, 2019 at... | eu-west-1a | available - modifying (0%) |       |
| attach-vol-te...     | vol-0f39fa9b39454... | 100 GiB | gp2         | 300             | January 30, 2019 at... | eu-west-1b | available                  |       |

## ボリューム変更の進行状況のモニタリング (AWS CLI)

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

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

次の出力例では、これらのボリュームの変更の状態は、引き続き `modifying` になっています。

```
{  
  "VolumesModifications": [  
    {
```

```
"TargetSize": 200,  
"TargetVolumeType": "io1",  
"ModificationState": "modifying",  
"VolumeId": "vol-1111111111111111",  
"TargetIops": 10000,  
"StartTime": "2017-01-19T22:21:02.959Z",  
"Progress": 0,  
"OriginalVolumeType": "gp2",  
"OriginalIops": 300,  
"OriginalSize": 100  
},  
{  
    "TargetSize": 2000,  
    "TargetVolumeType": "sc1",  
    "ModificationState": "modifying",  
    "VolumeId": "vol-2222222222222222",  
    "StartTime": "2017-01-19T22:23:22.158Z",  
    "Progress": 0,  
    "OriginalVolumeType": "gp2",  
    "OriginalIops": 300,  
    "OriginalSize": 1000  
}  
]  
}
```

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

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

2 つのボリュームに関する情報を含む出力例を以下に示します。

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

### ボリューム変更の進行状況のモニタリング (CloudWatch イベント)

CloudWatch イベントでは、ボリューム変更イベントの通知ルールを作成できます。ルールを使用して [Amazon SNS](#) で通知メッセージを生成するか、一致したイベントに応答して [Lambda 関数](#) を呼び出します。

CloudWatch イベントを使用して変更の進行状況をモニタリングするには

1. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
2. [Events]、[Create rule] の順に選択します。
3. [Build event pattern to match events by service] で、[Custom event pattern] を選択します。
4. [カスタムイベント/パターンの構築] で、コンテンツを次のように置き換え、[保存] を選択します。

```
{
```

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

以下にイベントデータの例を示します。

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-012345678901",
    "detail-type": "EBS Volume Notification",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "2017-01-12T21:09:07Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"
    ],
    "detail": {
        "result": "optimizing",
        "cause": "",
        "event": "modifyVolume",
        "request-id": "01234567-0123-0123-0123-0123456789ab"
    }
}
```

## ボリュームサイズ変更後の Linux ファイルシステムの拡張

EBS ボリュームのサイズを増やしたら、ファイルシステム固有のコマンドを使用して、ファイルシステムを大きなサイズに拡張します。ボリュームの状態が `optimizing` になった時点で、ファイルシステムのサイズを変更できます。

### Important

重要なデータを含むファイルシステムを拡張する前に、変更をロールバックする必要がある場合に備えて、ベストプラクティスとしてボリュームのスナップショットを作成することをお勧めします。詳細については、「[Amazon EBS スナップショットの作成 \(p. 972\)](#)」を参照してください。Linux AMI で MBR partitioning scheme を使用する場合、起動ボリュームサイズが最大 2 TiB に制限されます。詳細については、「[Linux ボリュームの要件 \(p. 1004\)](#)」および「[EBS ボリュームのサイズと設定の制限 \(p. 946\)](#)」を参照してください。

Windows ファイルシステムの拡張については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[ボリュームサイズ変更後の Windows ファイルシステムの拡張](#)」を参照してください。

次のタスクで、インスタンスのブートボリュームのサイズを 8 GB から 16 GB に、追加のボリュームのサイズを 8 GB から 30 GB に変更したとします。

### タスク

- [ボリュームのファイルシステムの識別 \(p. 1012\)](#)
- [パーティションの拡張 \(必要な場合\) \(p. 1012\)](#)
- [ファイルシステムの拡張 \(p. 1013\)](#)

## ボリュームのファイルシステムの識別

インスタンスの各ボリュームで使用するファイルシステムを識別するには、[インスタンスに接続](#)(p. 505)し、file -s コマンドを実行します。

例: Nitro ベースのインスタンスのファイルシステム

以下の例では、XFS ファイルシステムを備えたブートボリュームと、XFS ファイルシステムを備えた追加のボリュームを含む [Nitro ベースのインスタンス](#)(p. 187)を示しています。

```
[ec2-user ~]$ sudo file -s /dev/nvme?n*
/dev/nvme0n1:      x86 boot sector ...
/dev/nvme0n1p1:    SGI XFS filesystem data ...
/dev/nvme0n1p128:  data
/dev/nvme1n1:      SGI XFS filesystem data ...
```

例: T2 ベースのインスタンスのファイルシステム

以下の例では、ext4 ファイルシステムを備えたブートボリュームと、XFS ファイルシステムを備えた追加のボリュームを含む T2 インスタンスを示しています。

```
[ec2-user ~]$ sudo file -s /dev/xvd*
/dev/xvda:  DOS/MBR boot sector ..
/dev/xvda1: Linux rev 1.0 ext4 filesystem data ...
/dev/xvdf:  SGI XFS filesystem data ...
```

## パーティションの拡張(必要な場合)

EBS ボリュームには、ファイルシステムとデータを含むパーティションが存在する場合があります。ボリュームのサイズを拡張しても、パーティションのサイズは拡張されません。サイズ変更されたボリュームのファイルシステムを拡張する前に、新しいサイズのボリュームに拡張する必要があるパーティションがそのボリュームにあるかどうかを確認します。

インスタンスにアタッチされたブロックデバイスに関する情報を表示するには、lsblk コマンドを使用します。サイズ変更されたボリュームにパーティションが存在し、新しいサイズのボリュームがそのパーティションに反映されていない場合は、growpart コマンドを使用してパーティションを拡張します。LVM パーティションの拡張については、「[論理ボリュームの拡張](#)」を参照してください。

例: Nitro ベースのインスタンスのパーティション

以下の例では、Nitro ベースのインスタンスのボリュームを示します。

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

- ルートボリュームである /dev/nvme0n1 には、パーティション /dev/nvme0n1p1 があります。ルートボリュームのサイズに新しいサイズ(16 GB)が反映されている場合、パーティションのサイズには元のサイズ(8 GB)が反映されるため、ファイルシステムを拡張する前に拡張する必要があります。
- ボリューム /dev/nvme1n1 にはパーティションはありません。ボリュームのサイズには、新しいサイズ(30 GB)が反映されます。

ルートボリュームのパーティションを拡張するには、次の growpart コマンドを使用します。デバイス名とパーティション番号の間にスペースがある点に注意してください。

```
[ec2-user ~]$ sudo growpart /dev/nvmeOn1 1
```

拡張されたボリュームサイズがパーティションに反映されていることを確認するには、再度 lsblk コマンドを使用します。

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

#### 例: T2 インスタンスのパーティション

以下の例では、T2 インスタンスのボリュームを示します。

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda     202:0    0  16G  0 disk
##xvda1  202:1    0   8G  0 part /
xvdf     202:80   0  30G  0 disk
##xvdf1  202:81   0   8G  0 part /data
```

- ルートボリュームである /dev/xvda には、パーティション /dev/xvda1 があります。ボリュームのサイズが 16 GB の場合、パーティションのサイズは 8 GB のままになるため、拡張する必要があります。
- ボリューム /dev/xvdf には、パーティション /dev/xvdf1 があります。ボリュームのサイズが 30 GB の場合、パーティションのサイズは 8 GB のままになるため、拡張する必要があります。

各ボリュームのパーティションを拡張するには、次の growpart コマンドを使用します。デバイス名とパーティション番号の間にスペースがある点に注意してください。

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

拡張されたボリュームサイズがパーティションに反映されていることを確認するには、再度 lsblk コマンドを使用します。

```
[ec2-user ~]$ lsblk
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda     202:0    0  16G  0 disk
##xvda1  202:1    0  16G  0 part /
xvdf     202:80   0  30G  0 disk
##xvdf1  202:81   0  30G  0 part /data
```

#### ファイルシステムの拡張

ファイルシステム固有のコマンドを使用して、各ファイルシステムのサイズを新しいボリューム容量に変更します。次に示す例以外のファイルシステムの場合の手順については、該当するファイルシステムのドキュメントを参照してください。

#### 例: ext2、ext3、ext4 ファイルシステムの拡張

各ボリュームのファイルシステムのサイズを確認するには、df -h コマンドを使用します。この例では、/dev/xvda1 と /dev/xvdf のいずれにも、ボリュームの元のサイズ (8 GB) が反映されています。

```
[ec2-user ~]$ df -h
```

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      8.0G  1.9G  6.2G  24% /
/dev/xvdf1      8.0G   45M  8.0G   1% /data
...
```

各ボリュームのファイルシステムを拡張するには、`resize2fs` コマンドを使用します。

```
[ec2-user ~]$ sudo resize2fs /dev/xvda1
[ec2-user ~]$ sudo resize2fs /dev/xvdf1
```

拡張されたボリュームサイズが各ファイルシステムに反映されていることを確認するには、再度 `df -h` コマンドを使用します。

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      16G  1.9G  14G  12% /
/dev/xvdf1      30G   45M  30G   1% /data
...
```

#### 例: XFS ファイルシステムの拡張

各ボリュームのファイルシステムのサイズを確認するには、`df -h` コマンドを使用します。この例では、元のボリュームサイズ (8 GB) が各ファイルシステムに反映されています。

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p1   8.0G  1.6G  6.5G  20% /
/dev/nvme1n1    30G   33M  8.0G   1% /data
...
```

XFS ファイルシステムを拡張するには、次のように XFS ツールをインストールします (インストールされていない場合)。

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

各ボリュームのファイルシステムを拡張するには、`xfs_growfs` コマンドを使用します。この例では、`/` と `/data` は、`df -h` の出力で示されているボリュームのマウントポイントです。

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

拡張されたボリュームサイズが各ファイルシステムに反映されていることを確認するには、再度 `df -h` コマンドを使用します。

```
[ec2-user ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme0n1p1   16G  1.6G  15G  10% /
/dev/nvme1n1    30G   33M  30G   1% /data
...
```

## Amazon EBS Encryption

Amazon EBS 暗号化 は、EBS リソースのために、独自のキー管理インフラストラクチャの構築、保守、および保護を必要としない、簡単な暗号化ソリューションを提供します。暗号化されたボリュームとスナップショットを作成する際に、AWS Key Management Service (AWS KMS) カスタマーマスターキー (CMK) が使用されます。

暗号化オペレーションは EC2 インスタンスをホストするサーバー上で実行され、インスタンスとそれに接続された EBS ストレージ間でのデータの保存と転送中のデータの両方のセキュリティを保証します。

#### コンテンツ

- [EBS 暗号化の仕組み \(p. 1015\)](#)
- [要件 \(p. 1015\)](#)
- [EBS 暗号化のデフォルトキー \(p. 1017\)](#)
- [デフォルトでの暗号化 \(p. 1017\)](#)
- [EBS リソースの暗号化 \(p. 1018\)](#)
- [暗号化シナリオ \(p. 1019\)](#)
- [API と CLI を使用した暗号化のデフォルトの設定 \(p. 1024\)](#)

## EBS 暗号化の仕組み

EC2 インスタンスのブートボリュームとデータボリュームの両方を暗号化できます。暗号化された EBS ボリュームを作成し、サポートされるインスタンスタイプにアタッチする場合、以下のタイプのデータが暗号化されます。

- ボリューム内の保存データ
- ボリュームとインスタンスの間で移動されるすべてのデータ
- ボリュームから作成されたすべてのスナップショット
- それらのスナップショットから作成されたすべてのボリューム

EBS は、業界標準の AES-256 アルゴリズムを使用してデータキーでボリュームを暗号化します。データキーは、EBS が CMK で暗号化する前ではなく、暗号化されたデータとともにディスク上に保存されます。データキーはプレーンテキストでディスクに表示されません。同じデータキーは、ボリュームとそのスナップショットから作成された後続のボリュームのスナップショットによって共有されます。詳細については、『AWS Key Management Service Developer Guide』の「[データキー](#)」を参照してください。

Amazon EBS は AWS KMS と連携して、次のように EBS ボリュームを暗号化および復号します。

1. Amazon EBS は [CreateGrant](#) リクエストを AWS KMS に送信し、データキーを復号できるようにします。
2. Amazon EBS は [GenerateDataKeyWithoutPlaintext](#) リクエストを AWS KMS に送信し、ボリュームの暗号化に使用する CMK を指定します。
3. AWS KMS は新しいデータキーを生成し、指定された CMK で暗号化して、その暗号化されたデータキーを Amazon EBS に送信してボリュームのメタデータとともに保存します。
4. 暗号化されたボリュームをインスタンスにアタッチすると、Amazon EC2 は暗号化されたデータキーを指定して [Decrypt](#) リクエストを AWS KMS に送信します。
5. AWS KMS は、暗号化されたデータキーを復号して、Amazon EC2 に復号されたデータキーを送信します。
6. Amazon EC2 は、ハイパーテザーメモリ内のプレーンテキストデータキーを使用して、ディスク I/O をボリュームに暗号化します。プレーンテキストデータキーは、ボリュームがインスタンスにアタッチされる限り、メモリ内で維持されます。

詳細については、AWS Key Management Service Developer Guide の「[Amazon Elastic Block Store \(Amazon EBS\) で AWS KMS を使用する方法](#)」および「[AWS KMSログファイルエントリ](#)」を参照してください。

## 要件

開始する前に、以下の要件が満たされていることを確認します。

## サポートされるボリュームタイプ

暗号化は、すべての EBS ボリュームタイプでサポートされます。暗号化されたボリュームでは、暗号化されていないボリュームと同じ IOPS パフォーマンスが期待できます。遅延に対する影響は最小限に抑えられます。暗号化されていないボリュームにアクセスするのと同じ方法で、暗号化されたボリュームにアクセスできます。暗号化と復号は透過的に処理され、ユーザーやアプリケーションから追加の操作を必要としません。

## サポートされるインスタンスタイプ

Amazon EBS 暗号化は、以下に示すインスタンスタイプで使用できます。これらのインスタンスタイプには、暗号化されたボリュームと暗号化されていないボリュームを同時にアタッチすることができます。

- 汎用: A1、M3、M4、M5、M5a、M5ad、M5d、M5dn、M5n、T2、T3、および T3a
- コンピューティングの最適化: C3、C4、C5、C5d、および C5n
- メモリの最適化: cr1.8xlarge、R3、R4、R5、R5a、R5ad、R5d、R5dn、R5n、u-6tb1.metal、u-9tb1.metal、u-12tb1.metal および z1d
- ストレージの最適化: D2、h1.2xlarge、h1.4xlarge、I2、I3、および I3en
- 高速コンピューティング: F1、G2、G3、G4、P2、および P3

## IAM ユーザーのアクセス権限

EBS 暗号化のデフォルトキーとして CMK を設定すると、デフォルトのキー policy により、必要な KMS アクションにアクセスできるすべての IAM ユーザーがこのキーを使用して EBS リソースを暗号化または復号できるようになります。EBS 暗号化を使用するには、次のアクションを呼び出すアクセス権限を IAM ユーザーに付与する必要があります。

- kms:CreateGrant
- kms:Decrypt
- kms:DescribeKey
- kms:GenerateDataKeyWithoutPlainText
- kms:ReEncrypt

最小権限のプリンシパルに従うには、kms:CreateGrant へのフルアクセスを許可しないでください。代わりに、次の例に示すように、AWS のサービスによってユーザーに代わって許可が作成された場合のみ、CMK に許可を作成できるようにします。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "kms>CreateGrant",  
            "Resource": [  
                "arn:aws:kms:us-east-2:123456789012:key/abcd1234-a123-456d-a12b-a123b4cd56ef"  
            ],  
            "Condition": {  
                "Bool": {  
                    "kms:GrantIsForAWSResource": true  
                }  
            }  
        }  
    ]  
}
```

詳細については、AWS Key Management Service Developer Guide の「[デフォルトキー](#)」を参照してください。

## EBS 暗号化のデフォルトキー

Amazon EBS は、AWS リソースを保存する各リージョンに一意の AWS 管理の CMK を自動的に作成します。このキーにはエイリアス alias/aws/ebs があります。デフォルトでは、Amazon EBS は暗号化にこのキーを使用します。または、作成したカスタマー管理の対称 CMK を EBS 暗号化のデフォルトキーとして指定することもできます。独自の CMK を使用することにより、キーの作成、更新、無効化ができるなど、より高い柔軟性が得られます。

### Important

Amazon EBS は非対称 CMK をサポートしていません。詳細については、AWS Key Management Service 開発者ガイドの「[対称キーと非対称キーの使用](#)」を参照してください。

リージョンの EBS 暗号化用にデフォルトキーを設定するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、使用するリージョンを選択します。
3. [Account Attributes (アカウント属性)]、[Settings (設定)] の順に選択します。
4. [Change the default key (デフォルトキーの変更)] を選択してから、利用可能なキーを選択します。
5. [Update (更新)] を選択します。

## デフォルトでの暗号化

作成した新しい EBS ボリュームとスナップショットコピーの暗号化を強制するように AWS アカウントを設定できます。たとえば、Amazon EBS は、インスタンスの起動時に作成された EBS ボリュームと、暗号化されていないスナップショットからコピーしたスナップショットを暗号化します。暗号化されていない EBS リソースから暗号化された EBS リソースへの移行の例については、「[暗号化されていないリソースの暗号化 \(p. 1018\)](#)」を参照してください。

デフォルトでは、暗号化は既存の EBS ボリュームまたはスナップショットには影響しません。

### 考慮事項

- デフォルトでの暗号化はリージョン固有の設定です。リージョンに対して有効にした場合、そのリージョン内の個々のボリュームまたはスナップショットに対して無効にすることはできません。
- デフォルトで暗号化を有効にすると、インスタンスタイプが EBS 暗号化をサポートしている場合にのみ、インスタンスを起動できます。詳細については、「[サポートされるインスタンスタイプ \(p. 1016\)](#)」を参照してください。
- AWS Server Migration Service (SMS) を使用してサーバーを移行する場合は、デフォルトでの暗号化を有効にしないでください。デフォルトでの暗号化がすでに有効になっていて、デルタレプリケーションエラーが発生している場合は、デフォルトでの暗号化を無効にしてください。代わりに、レプリケーションジョブの作成時に AMI 暗号化を有効にします。

リージョンの暗号化をデフォルトで有効にするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、使用するリージョンを選択します。
3. ナビゲーションペインの [EC2 Dashboard (EC2 ダッシュボード)] を選択します。
4. ページの右上で、[Account Attributes (アカウントの属性)]、[Settings (設定)] の順に選択します。
5. [EBS Storage (EBS ストレージ)] で、[Always encrypt new EBS volumes (常に新しい EBS ボリュームを暗号化する)] を選択します。

## 6. [Update (更新)] を選択します。

既存のスナップショットまたは暗号化されたボリュームに関連付けられている CMK を変更することはできません。ただし、スナップショットコピー操作中に別の CMK を関連付けて、コピーしたスナップショットを新しい CMK で暗号化できます。

## EBS リソースの暗号化

EBS ボリュームを暗号化するには、[デフォルトでの暗号化 \(p. 1017\)](#)を使用するか、暗号化するボリュームを作成するときに暗号化を有効にします。

ボリュームを暗号化する場合、ボリュームの暗号化に使用する対称 CMK を指定できます。CMK が指定されていない場合、暗号化に使用されるキーはソーススナップショットの暗号化状態とその所有権によって異なります。詳細については、「[暗号化結果の表 \(p. 1022\)](#)」を参照してください。

既存のスナップショットまたはボリュームに関連付けられている CMK を変更することはできません。ただし、スナップショットコピー操作中に別の CMK を関連付けて、コピーしたスナップショットを新しい CMK で暗号化できます。

## 暗号化を使用して新しい空のボリュームを作成する

新しい空の EBS ボリュームを作成するときは、特定のボリューム作成オペレーションで暗号化を有効にすることで暗号化できます。デフォルトで EBS 暗号化を有効にした場合、ボリュームは自動的に暗号化されます。デフォルトでは、ボリュームは EBS 暗号化のデフォルトキーに暗号化されます。または、ボリュームの作成オペレーションごとに異なる対称 CMK を指定することもできます。ボリュームは最初に使用可能になった時点で暗号化されているため、データは常に保護されています。詳細な手順については、「[Amazon EBS ボリュームの作成 \(p. 949\)](#)」を参照してください。

デフォルトでは、ボリュームの作成時に選択した CMK が、ボリュームから作成したスナップショットとそれらの暗号化されたスナップショットから復元したボリュームを暗号化します。暗号化されたボリュームまたはスナップショットから暗号化を削除することはできません。つまり、暗号化されたスナップショット、または暗号化されたスナップショットのコピーから復元されたボリュームは、常に暗号化されます。

暗号化されたボリュームのパブリックスナップショットはサポートされていませんが、暗号化されたスナップショットを特定のアカウントと共有できます。詳細な手順については、「[Amazon EBS スナップショットの共有 \(p. 982\)](#)」を参照してください。

## 暗号化されていないリソースの暗号化

暗号化されていない既存のボリュームまたはスナップショットを直接暗号化する方法はありませんが、ボリュームまたはスナップショットを作成することで暗号化できます。暗号化をデフォルトで有効にした場合、Amazon EBS は EBS 暗号化のデフォルトキーを使用して、作成された新しいボリュームまたはスナップショットを暗号化します。デフォルトで暗号化を有効にしていない場合でも、個々のボリュームまたはスナップショットを作成するときに暗号化を有効にすることができます。暗号化をデフォルトで有効にするか、作成オペレーションごとに有効にするかにかかわらず、EBS 暗号化のデフォルトキーを上書きし、カスタマー管理の対称 CMK を選択できます。詳細については、「[Amazon EBS ボリュームの作成 \(p. 949\)](#)」および「[Amazon EBS スナップショットのコピー \(p. 977\)](#)」を参照してください。

スナップショットコピーをカスタマー管理の CMK に暗号化するには、[暗号化されていないスナップショットをコピーする \(デフォルトでの暗号化が有効になっていない場合\) \(p. 1020\)](#) に示すように、暗号化を有効にし、キーを指定する必要があります

### Important

Amazon EBS は非対称 CMK をサポートしていません。詳細については、AWS Key Management Service 開発者ガイドの「[対称キーと非対称キーの使用](#)」を参照してください。

EBS-Backed AMI からインスタンスを起動するときに新しい暗号化状態を適用することもできます。これは、EBS-backed AMI に、説明の通りに暗号化できる EBS ボリュームのスナップショットが含まれているためです。詳細については、「[EBS-Backed AMI での暗号化の利用 \(p. 151\)](#)」を参照してください。

## 暗号化シナリオ

暗号化された EBS リソースを作成すると、ボリューム作成パラメータまたは AMI やインスタンスのブロックデバイスマッピングで別のカスタマー管理の CMK を指定しない限り、アカウントの EBS 暗号化のデフォルトキーによって暗号化されます。詳細については、「[EBS 暗号化のデフォルトキー \(p. 1017\)](#)」を参照してください。

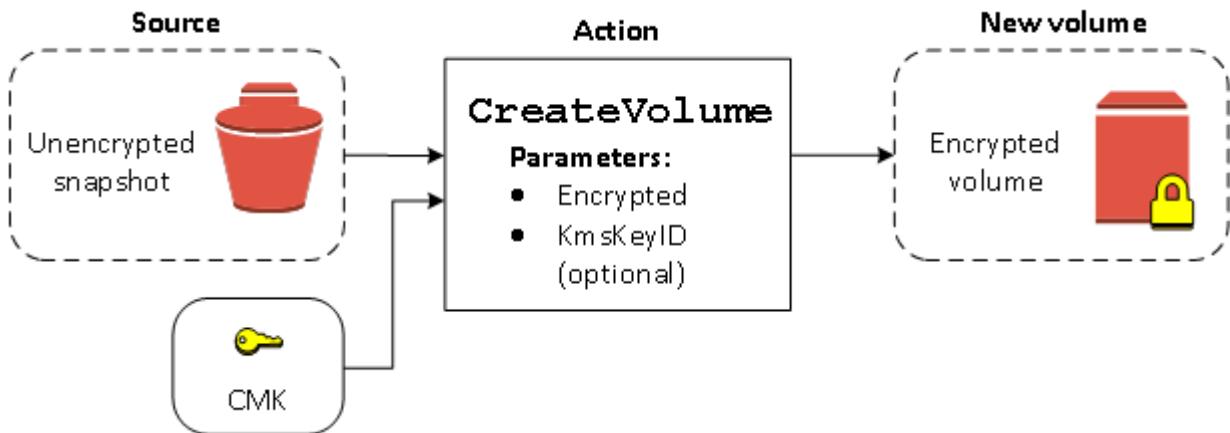
次の例では、ボリュームとスナップショットの暗号化状態を管理する方法を示します。暗号化のケースの完全なリストについては、「[暗号化の結果の表 \(p. 1022\)](#)」を参照してください。

### 例

- 暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっていない場合) (p. 1019)
- 暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっている場合) (p. 1020)
- 暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっていない場合) (p. 1020)
- 暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっている場合) (p. 1021)
- 暗号化ボリュームを再暗号化する (p. 1021)
- 暗号化スナップショットを再暗号化する (p. 1022)
- 暗号化されたボリュームと暗号化されていないボリュームとの間でデータを移行する (p. 1022)
- 暗号化の結果 (p. 1022)

### 暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっていない場合)

デフォルトでの暗号化を有効にしないと、暗号化されていないスナップショットから復元されたボリュームは、デフォルトで暗号化されません。ただし、Encrypted パラメータと、必要に応じて KmsKeyId パラメータを設定して、結果のボリュームを暗号化することができます。以下の図は、そのプロセスを示したものです。

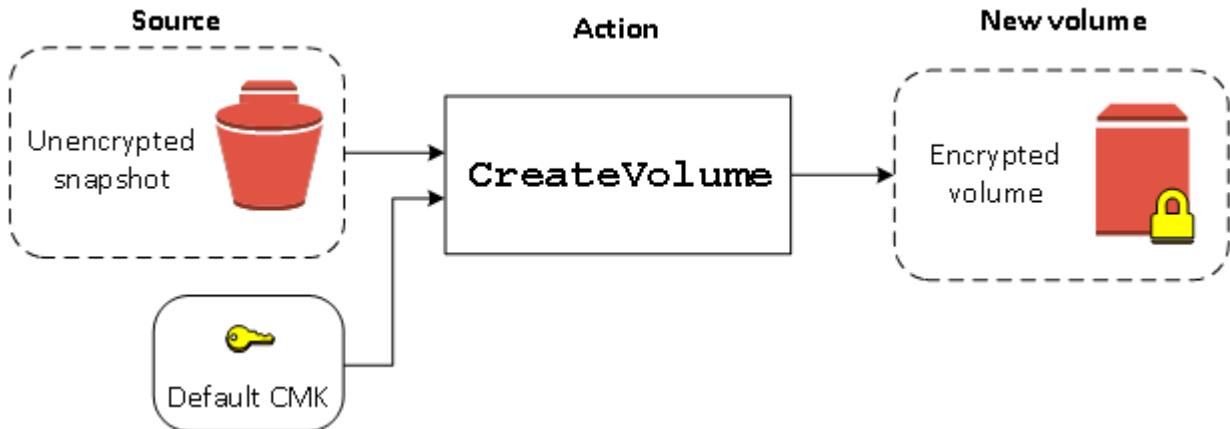


KmsKeyId パラメータを省略すると、結果のボリュームは EBS 暗号化のデフォルトキーを使用して暗号化されます。ボリュームを別の CMK に暗号化するには、キー ID を指定する必要があります。

詳細については、「[スナップショットからの Amazon EBS ボリュームの復元 \(p. 950\)](#)」を参照してください。

### 暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっている場合)

デフォルトでの暗号化を有効にした場合、暗号化されていないスナップショットから復元されたボリュームには暗号化が必須であり、デフォルトの CMK を使用するために暗号化パラメータは必要ありません。以下の図に、このデフォルトの簡単なケースを示しています。

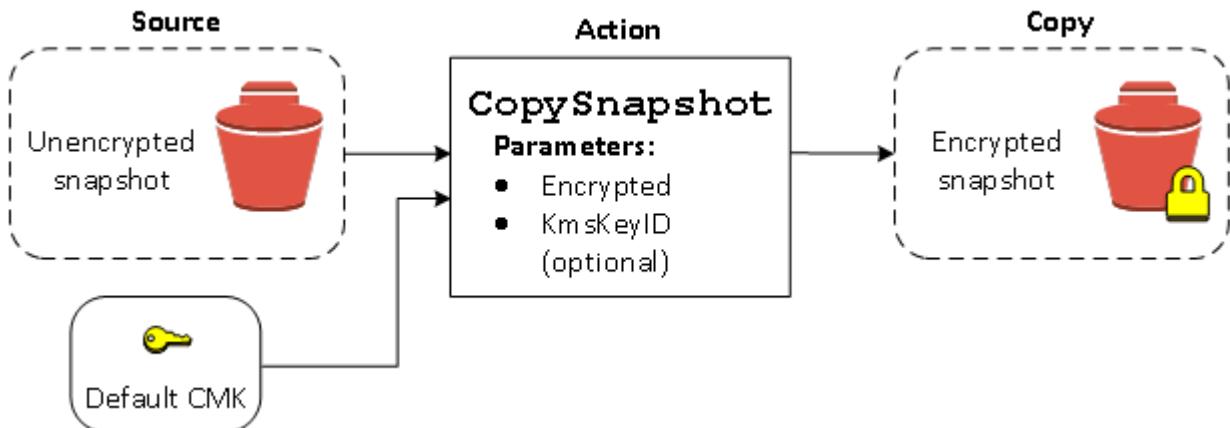


復元したボリュームをカスタマー管理の対称 CMK に暗号化する場合は、[暗号化されていないボリュームを復元する \(デフォルトでの暗号化が有効になっていない場合\) \(p. 1019\)](#) に示すように Encrypted と KmsKeyId の両方のパラメータを指定する必要があります。

### 暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっていない場合)

デフォルトでの暗号化を有効にしないと、暗号化されていないスナップショットのコピーは、デフォルトで暗号化されません。ただし、Encrypted パラメータと、必要に応じて KmsKeyId パラメータを設定して、結果のスナップショットを暗号化することができます。KmsKeyId を省略すると、結果のスナップショットはデフォルトの CMK に暗号化されます。ボリュームを別の対称 CMK に暗号化するには、キー ID を指定する必要があります。

以下の図は、そのプロセスを示したものです。



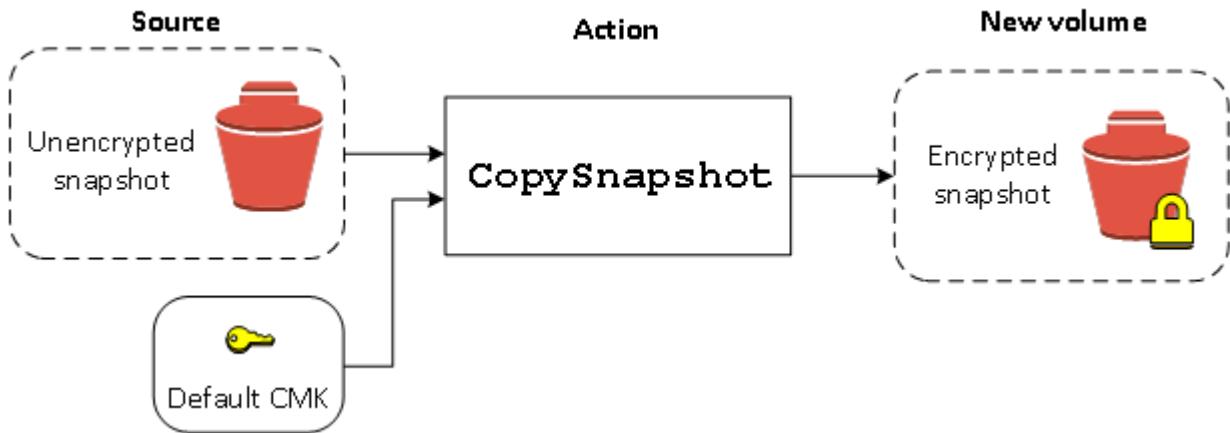
#### Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な (増分なし) コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

EBS ボリュームを暗号化するには、暗号化されていないスナップショットを暗号化されたスナップショットにコピーし、その暗号化されたスナップショットからボリュームを作成することができます。詳細については、「[Amazon EBS スナップショットのコピー \(p. 977\)](#)」を参照してください。

## 暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっている場合)

デフォルトでの暗号化を有効にした場合、暗号化されていないスナップショットのコピーには暗号化が必要であり、デフォルトの CMK を使用する場合は、暗号化パラメータは必要ありません。このデフォルトのケースを次の図に示します。

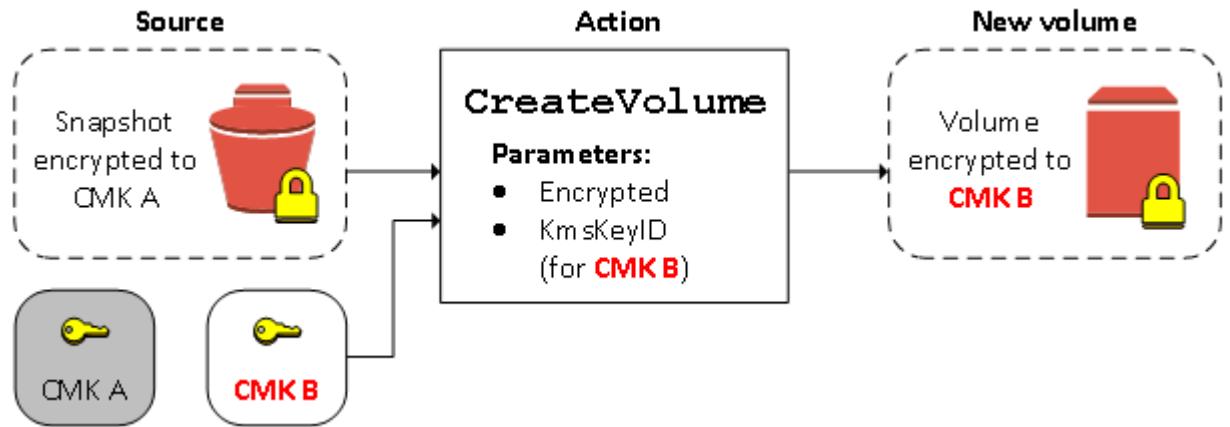


### Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な (増分なし) コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

## 暗号化ボリュームを再暗号化する

CreateVolume アクションが暗号化されたスナップショットに対して実行されるときは、別の CMK でそれを再暗号化することができます。以下の図は、そのプロセスを示したものです。この例では、CMK A と CMK B の 2 つの CMK を所有しています。ソーススナップショットは CMK A によって暗号化されています。ボリュームの作成中に、パラメータとして指定された CMK B のキー ID を使用して、ソースデータは自動的に復号され、次に CMK B によって再暗号化されます。



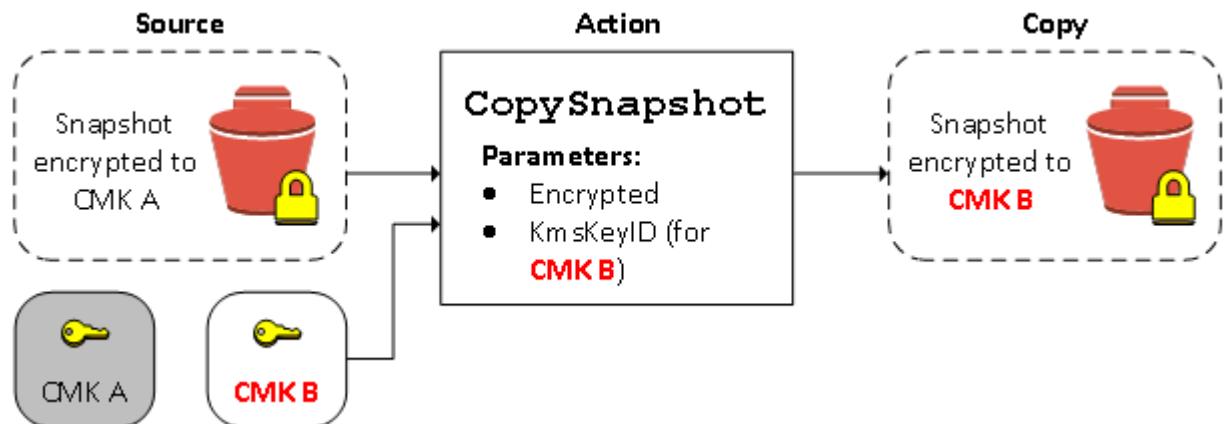
### Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な (増分なし) コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

詳細については、「[スナップショットからの Amazon EBS ボリュームの復元 \(p. 950\)](#)」を参照してください。

## 暗号化スナップショットを再暗号化する

スナップショットをコピー時に暗号化する機能により、すでに暗号化された自己所有のスナップショットに新しい対称 CMK を適用できます。結果として作成されたコピーから復元されたボリュームには、新しい CMK を使用してのみアクセスすることができます。以下の図は、そのプロセスを示したものです。この例では、CMK A と CMK B の 2 つの CMK を所有しています。ソーススナップショットは CMK A によって暗号化されています。コピー中に、パラメータとして指定された CMK B のキー ID を使用して、ソースデータは自動的に CMK B によって再暗号化されます。



### Note

スナップショットを新しい CMK にコピーし暗号化する場合、完全な(増分なし)コピーが常に作成されるため、遅延が、ストレージコストがさらに生じる原因になります。

関連するシナリオでは、共有されているスナップショットのコピーに新しい暗号化パラメータを適用するよう選択できます。デフォルトでは、コピーは、スナップショットの所有者によって共有された CMK を使用して暗号化されます。ただし、管理する別の CMK を使用して、共有スナップショットのコピーを作成することをお勧めします。これにより、元の CMK が侵害された場合や、所有者が何らかの理由で CMK を無効にした場合に、ボリュームへのアクセスが保護されます。詳細については、「[暗号化とスナップショットのコピー \(p. 979\)](#)」を参照してください。

## 暗号化されたボリュームと暗号化されていないボリュームとの間でデータを移行する

暗号化されているボリュームと暗号化されていないボリュームの両方に対してアクセス許可がある場合は、これらの間で自由にデータを転送できます。EC2 では、暗号化と復号化のオペレーションが透過的に実行されます。

たとえば、rsync コマンドを使用してデータをコピーします。次のコマンドでは、移行元のデータは /mnt/source にあり、移行先のボリュームは /mnt/destination にマウントされています。

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

## 暗号化の結果

次の表では、考えられる設定の組み合わせごとの暗号化の結果について説明しています。

| 暗号化は可能ですか？ | 暗号化はデフォルトで有効になっていますか？ | ボリュームのソース    | デフォルト(CMK が指定されていない) | カスタム(CMK が指定されている) |
|------------|-----------------------|--------------|----------------------|--------------------|
| いいえ        | いいえ                   | 新しい(空の)ボリューム | 暗号化されていない            | 該当なし               |

| 暗号化は可能ですか？ | 暗号化はデフォルトで有効になっていますか？ | ボリュームのソース                   | デフォルト (CMK が指定されていない) | カスタム (CMK が指定されている)   |
|------------|-----------------------|-----------------------------|-----------------------|-----------------------|
| いいえ        | いいえ                   | 所有する暗号化されていないスナップショット       | 暗号化されていない             |                       |
| いいえ        | いいえ                   | 所有する暗号化されたスナップショット          | 同じキーで暗号化されている         |                       |
| いいえ        | いいえ                   | 自分と共有されている暗号化されていないスナップショット | 暗号化されていない             |                       |
| いいえ        | いいえ                   | 自分と共有されている暗号化されたスナップショット    | デフォルト CMK* で暗号化されている  |                       |
| はい         | いいえ                   | 新しいボリューム                    | デフォルト CMK で暗号化されている   | 指定された CMK** で暗号化されている |
| はい         | いいえ                   | 所有する暗号化されていないスナップショット       | デフォルト CMK で暗号化されている   |                       |
| はい         | いいえ                   | 所有する暗号化されたスナップショット          | 同じキーで暗号化されている         |                       |
| はい         | いいえ                   | 自分と共有されている暗号化されていないスナップショット | デフォルト CMK で暗号化されている   |                       |
| はい         | いいえ                   | 自分と共有されている暗号化されたスナップショット    | デフォルト CMK で暗号化されている   |                       |
| いいえ        | はい                    | 新しい (空の) ボリューム              | デフォルト CMK で暗号化されている   |                       |
| いいえ        | はい                    | 所有する暗号化されていないスナップショット       | デフォルト CMK で暗号化されている   |                       |
| いいえ        | はい                    | 所有する暗号化されたスナップショット          | 同じキーで暗号化されている         |                       |
| いいえ        | はい                    | 自分と共有されている暗号化されていないスナップショット | デフォルト CMK で暗号化されている   |                       |
| いいえ        | はい                    | 自分と共有されている暗号化されたスナップショット    | デフォルト CMK で暗号化されている   |                       |
| はい         | はい                    | 新しいボリューム                    | デフォルト CMK で暗号化されている   | 指定された CMK で暗号化されている   |
| はい         | はい                    | 所有する暗号化されていないスナップショット       | デフォルト CMK で暗号化されている   |                       |
| はい         | はい                    | 所有する暗号化されたスナップショット          | 同じキーで暗号化されている         |                       |

| 暗号化は可能ですか？ | 暗号化はデフォルトで有効になっていますか？ | ボリュームのソース                   | デフォルト (CMK が指定されていない) | カスタム (CMK が指定されている) |
|------------|-----------------------|-----------------------------|-----------------------|---------------------|
| はい         | はい                    | 自分と共有されている暗号化されていないスナップショット | デフォルト CMK で暗号化されている   |                     |
| はい         | はい                    | 自分と共有されている暗号化されたスナップショット    | デフォルト CMK で暗号化されている   |                     |

\* これは、AWS アカウントおよびリージョンでの EBS 暗号化に使用されるデフォルトの CMK です。これはデフォルトでは、EBS 用の一意の AWS 管理の CMK です。または、カスタマー管理の CMK を指定できます。詳細については、「[EBS 暗号化のデフォルトキー \(p. 1017\)](#)」を参照してください。

\*\* これは、起動時にボリュームに対して指定されたカスタマー管理の CMK です。この CMK は、AWS アカウントおよびリージョンのデフォルトの CMK の代わりに使用されます。

## API と CLI を使用した暗号化のデフォルトの設定

以下の API アクションおよび CLI コマンドを使用して、デフォルトで暗号化およびデフォルトのカスタマースターキー (CMK) を管理できます。

| API アクション                                     | CLI コマンド                                       | 説明                                                          |
|-----------------------------------------------|------------------------------------------------|-------------------------------------------------------------|
| <a href="#">DisableEbsEncryptionByDefault</a> | <code>disable-ebs-encryption-by-default</code> | デフォルトでの暗号化を無効にします。                                          |
| <a href="#">EnableEbsEncryptionByDefault</a>  | <code>enable-ebs-encryption-by-default</code>  | デフォルトでの暗号化を有効にします。                                          |
| <a href="#">GetEbsDefaultKmsKeyId</a>         | <code>get-ebs-default-kms-key-id</code>        | デフォルトの CMK について説明します。                                       |
| <a href="#">GetEbsEncryptionByDefault</a>     | <code>get-ebs-encryption-by-default</code>     | デフォルトの暗号化が有効かどうかを示します。                                      |
| <a href="#">ModifyEbsDefaultKmsKeyId</a>      | <code>modify-ebs-default-kms-key-id</code>     | EBS ボリュームの暗号化に使用されるデフォルトの CMK を変更します。                       |
| <a href="#">ResetEbsDefaultKmsKeyId</a>       | <code>reset-ebs-default-kms-key-id</code>      | AWS 管理のデフォルト CMK を EBS ボリュームの暗号化に使用されるデフォルト CMK としてリセットします。 |

## Amazon EBS 高速スナップショット復元

Amazon EBS 高速スナップショット復元を使用するとスナップショットからボリュームを作成でき、このボリュームは作成時に完全に初期化された状態になります。これにより、ブロックの初回アクセス時における I/O オペレーションのレイテンシーがなくなります。高速スナップショット復元を使用して作成されたボリュームでは、プロビジョニングドパフォーマンスをすべて即座に提供できます。

開始するには、特定のアベイラビリティーゾーンで特定のスナップショットの高速スナップショット復元を有効にします。スナップショットとアベイラビリティーゾーンのペアごとに 1 つの高速スナップショット復元を参照します。リージョンごとに最大 50 の高速スナップショット復元を有効にできます。高速スナップショット復元が有効になっているアベイラビリティーゾーンの 1 つで、対応するスナップショットからボリュームを作成すると、ボリュームは高速スナップショット復元を使用して復元されます。

## 目次

- [高速スナップショット復元の状態 \(p. 1025\)](#)
- [ボリューム作成クレジット \(p. 1025\)](#)
- [高速スナップショット復元の管理 \(p. 1026\)](#)
- [高速スナップショット復元が有効になっているスナップショットの表示 \(p. 1026\)](#)
- [高速スナップショット復元を使用して復元したボリュームの表示 \(p. 1027\)](#)

## 高速スナップショット復元の状態

スナップショットに対して高速スナップショット復元を有効にすると、以下のいずれかの状態になります。

- `enabling` — 高速スナップショット復元の有効化がリクエストされました。
- `optimizing` — 高速スナップショット復元の有効化中です。スナップショットの最適化には TiB あたり 60 分を要します。
- `enabled` — 高速スナップショット復元は有効になっています。
- `disabling` — 高速スナップショット復元の無効化がリクエストされました。または、高速スナップショット復元の有効化のリクエストが失敗しました。
- `disabled` — 高速スナップショット復元は無効になっています。高速スナップショット復元は必要に応じて再度有効にすることができます。

## ボリューム作成クレジット

高速スナップショット復元のパフォーマンスの利点を全面的に享受するボリュームの数は、スナップショットのボリューム作成クレジットの数によって決まります。アベイラビリティーゾーンごとにスナップショットあたり 1 つのクレジットバケットがあります。高速スナップショット復元を有効にしてスナップショットから作成するボリュームごとにクレジットバケットの 1 つのクレジットが消費されます。

クレジットバケットのサイズは、スナップショットから作成したボリュームのサイズではなく、スナップショットのサイズによって決まります。スナップショットごとのクレジットバケットのサイズは、次のように計算されます。

```
MAX (1, MIN (10, FLOOR(1024/snapshot_size_gib)))
```

クレジットを消費すると、クレジットバケットは時間の経過に伴って補充されます。各クレジットバケットの補充レートは次のように計算されます。

```
MIN (10, 1024/snapshot_size_gib)
```

たとえば、サイズが 100 GiB のスナップショットに対して高速スナップショット復元を有効にすると、そのクレジットバケットの最大サイズは 10 クレジットとなり、補充レートは 1 時間あたり 10 クレジットになります。クレジットバケットが満杯である場合、このスナップショットからは同時に 10 個の初期化済みボリュームを作成できます。

Cloudwatch メトリクスを使用して、クレジットバケットのサイズと、各バケットで利用可能なクレジット数をモニタリングすることができます。詳細については、「[高速スナップショット復元メトリクス \(p. 1064\)](#)」を参照してください。

高速スナップショット復元を有効にしてスナップショットからボリュームを作成したら、[describe-volumes](#) を使用してボリュームを示し、高速スナップショット復元を使用して、ボリュームが初期化されたボリュームとして作成されたかどうかを、出力の fastRestored フィールドで確認することができます。

## 高速スナップショット復元の管理

スナップショットの高速スナップショット復元を有効にするには、次の手順に従います。スナップショットを所有している必要があります。他と共有しているスナップショットに対して高速スナップショット復元を有効にすることはできません。

高速スナップショット復元を有効または無効にするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Snapshots (スナップショット)] を選択します。
3. スナップショットを選択します。
4. [Actions (アクション)]、[Manage Fast Snapshot Restore (高速スナップショット復元の管理)] の順に選択します。.
5. アベイラビリティーゾーンを選択または選択解除し、[Save (保存)] を選択します。
6. 有効にした高速スナップショット復元の状態を追跡するには、[Description (説明)] タブの [Fast Snapshot Restore (高速スナップショット復元)] を確認します。

AWS CLI を使用して高速スナップショット復元を管理するには

- [enable-fast-snapshot-restores](#)
- [disable-fast-snapshot-restores](#)
- [describe-fast-snapshot-restores](#)

## 高速スナップショット復元が有効になっているスナップショットの表示

スナップショットの高速スナップショット復元の状態を表示するには、次の手順に従います。

コンソールを使用して高速スナップショット復元の状態を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Snapshots (スナップショット)] を選択します。
3. スナップショットを選択します。
4. [説明] タブの [スナップショットの高速復元] を参照します。高速スナップショット復元の状態が表示されます。たとえば、「2 Availability Zones optimizing (2 つのアベイラビリティーゾーンを最適化中)」や「2 Availability Zones enabled (2 つのアベイラビリティーゾーンを有効化)」のように表示されます。

AWS CLI を使用して高速スナップショット復元が有効になっているスナップショットを表示するには

[describe-fast-snapshot-restores](#) コマンドを使用して、高速スナップショット復元が有効になっているスナップショットを参照します。

```
aws ec2 describe-fast-snapshot-restores --filters Name=name,Values=enabled
```

出力例を次に示します。

```
{  
  "FastSnapshotRestores": [  
    {
```

```
"SnapshotId": "snap-0e946653493cb0447",
"AvailabilityZone": "us-east-2a",
"State": "enabled",
"StateTransitionReason": "Client.UserInitiated - Lifecycle state transition",
"OwnerId": "123456789012",
"EnablingTime": "2020-01-25T23:57:49.596Z",
"OptimizingTime": "2020-01-25T23:58:25.573Z",
"EnabledTime": "2020-01-25T23:59:29.852Z"
},
{
    "SnapshotId": "snap-0e946653493cb0447",
    "AvailabilityZone": "us-east-2b",
    "State": "enabled",
    "StateTransitionReason": "Client.UserInitiated - Lifecycle state transition",
    "OwnerId": "123456789012",
    "EnablingTime": "2020-01-25T23:57:49.596Z",
    "OptimizingTime": "2020-01-25T23:58:25.573Z",
    "EnabledTime": "2020-01-25T23:59:29.852Z"
}
]
```

## 高速スナップショット復元を使用して復元したボリュームの表示

該当するアベイラビリティゾーンで高速スナップショット復元が有効になっているスナップショットからボリュームを作成すると、ボリュームは高速スナップショット復元を使用して復元されます。

[describe-volumes](#) コマンドを使用して、高速スナップショット復元が有効になっているスナップショットから作成したボリュームを表示します。

```
aws ec2 describe-volumes --filters Name=fast-restored,Values=true
```

出力例を次に示します。

```
{
    "Volumes": [
        {
            "Attachments": [],
            "AvailabilityZone": "us-east-2a",
            "CreateTime": "2020-01-26T00:34:11.093Z",
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-a87a-5513e232e843",
            "Size": 20,
            "SnapshotId": "snap-0e946653493cb0447",
            "State": "available",
            "VolumeId": "vol-0d371921d4ca797b0",
            "Iops": 100,
            "VolumeType": "gp2",
            "FastRestored": true
        }
    ]
}
```

## Linux インスタンスの Amazon EBS および NVMe

[Nitro ベースのインスタンス \(p. 187\)](#)では、EBS ボリュームは NVMe ブロックデバイスとして公開されます。デバイス名は、/dev/nvme0n1、/dev/nvme1n1 などです。ブロックデバイスマッピングで指定したデバイス名は、NVMe デバイス名 (/dev/nvme[0-26]n1) を使用して名称変更されます。ブロックデバイスドライバでは、ブロックデバイスマッピングでボリュームに指定した順序とは異なる順序で NVMe デバイス名を割り当てることができます。

#### Note

「[Amazon EBS 製品の詳細](#)」に記載されている EBS のパフォーマンス安定性は、プロックデバイスインターフェイスに関係なく有効です。

#### コンテンツ

- NVMe ドライバのインストールまたはアップグレード (p. 1028)
- EBS デバイスの特定 (p. 1029)
- NVMe EBS ボリュームを操作する (p. 1030)
- I/O オペレーションタイムアウト (p. 1030)

## NVMe ドライバのインストールまたはアップグレード

NVMe ボリュームにアクセスするには、NVMe ドライバーをインストールする必要があります。インスタンスは、NVMe ボリュームの種類である NVMe EBS ボリュームや NVMe インスタンスマルチボリューム、または NVMe ボリュームではないものをサポートできます。詳細については、「[ネットワーキング機能とストレージ機能の概要 \(p. 188\)](#)」を参照してください。

以下の AMI には、必要な NVMe ドライバーが含まれています。

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (`linux-aws` カーネル) 以降
- Red Hat Enterprise Linux 7.4 以降
- SUSE Linux Enterprise Server 12 SP2 以降
- CentOS 7.4.1708 以降
- FreeBSD 11.1 以降
- Debian GNU/Linux 9 以降

Windows インスタンスの NVMe ドライバーの詳細については、Windows インスタンスの Amazon EC2 ユーザーガイドの「[Amazon EBS および Windows インスタンス](#)」を参照してください。

NVMe ドライバーが含まれない AMI を使用している場合は、次の手順でドライバーをインスタンスにインストールします。

#### NVMe ドライバをインストールするには

1. インスタンスに接続します。
2. パッケージのキャッシュを更新し、必要なパッケージの更新を次のように取得します。
  - Amazon Linux 2、Amazon Linux、CentOS、Red Hat Enterprise Linux の場合:

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

- Ubuntu と Debian の場合:

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

3. Ubuntu 16.04 以降には、`linux-aws` パッケージが含まれます。このパッケージには、Nitro ベースのインスタンスで必要な NVMe および ENA ドライバーが含まれます。最新バージョンにするには、次のように `linux-aws` パッケージにアップグレードします。

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

Ubuntu 14.04 の場合は、次のように最新の `linux-aws` パッケージをインストールできます。

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

4. インスタンスを再起動して、最新のカーネルバージョンを読み込みます。

```
sudo reboot
```

5. 再起動後にインスタンスに再接続します。

## EBS デバイスの特定

EBS では、シングルルート I/O 仮想化 (SR-IOV) を使用して、NVMe 規格を使用して Nitro ベースのインスタンスにボリュームをアタッチします。これらのデバイスは、オペレーティングシステムの標準 NVMe ドライバーに依存しています。これらのドライバーは、通常、インスタンスの起動時に PCI バスをスキャンしてアタッチされたデバイスを検出し、デバイスがブロックデバイスマッピングでどのように指定されているかではなく、デバイスが応答する順序に基づいてデバイスノードを作成します。Linux では、NVMe デバイス名は `/dev/nvme<x>n<y>` のパターンに従います。ここで、`<x>` は列挙順序、EBS の場合は `<y>` は 1 です。場合によっては、デバイスは後続のインスタンスの開始時に異なる順序で検出に応答することがあり、デバイス名が変更されます。

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

- Nitro ベースのインスタンスでは、ブロックデバイスマッピングは、EBS ボリュームをアタッチしているとき、または `AttachVolume` が `RunInstances` API コールが、NVMe コントローラー ID のベンダー固有のデータフィールドに取り込まれる際に Amazon EC2 コンソールで指定されます。バージョン 2017.09.01 以降の Amazon Linux AMI で、このデータを読み込んでブロックデバイスマッピングへのシンボリックリンクを作成する `udev` ルールを提供します。
- NVMe EBS ボリュームには、EBS ボリューム ID がデバイス ID のシリアル番号として設定されています。
- デバイスがフォーマットされると、ファイルシステムの存続期間中、存続する UUID が生成されます。デバイスラベルは同時に指定することができます。詳細については、「[Linux で Amazon EBS ボリュームを使用できるようにする \(p. 956\)](#)」および「[間違ったボリュームで起動する \(p. 1171\)](#)」を参照してください。

### Amazon Linux AMI

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

```
[ec2-user ~]$ sudo /sbin/ebsnvme-id /dev/nvme1n1
Volume ID: vol-01324f611e2463981
/dev/sdf
```

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

### その他の Linux AMI

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

次の例では、ボリューム ID やデバイス名を取得します。デバイス名は、NVMe コントローラベンダー固有の拡張子 (コントローラー ID のバイト 384:4095) を介して使用できます。

```
[ec2-user ~]$ sudo nvme id-cntl -v /dev/nvme1n1
NVME Identify Controller:
vid      : 0x1d0f
ssvid    : 0x1d0f
sn       : vol01234567890abcdef
mn       : Amazon Elastic Block Store
...
0000: 2f 64 65 76 2f 73 64 6a 20 20 20 20 20 20 20 20 "/dev/sdf..."
```

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

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

## NVMe EBS ボリュームを操作する

NVMe EBS ボリュームをフォーマットしてマウントするには、「[Linux で Amazon EBS ボリュームを使用できるようにする \(p. 956\)](#)」を参照してください。

Linux カーネル 4.2 以降を使用している場合は、NVMe EBS ボリュームのボリュームサイズを変更すると、自動的にインスタンスに反映されます。古い Linux カーネルの場合は、EBS ボリュームをデタッチしてアタッチするか、インスタンスを再起動してサイズ変更を反映させる必要があります。Linux カーネル 3.19 以降では、hdparm コマンドを次のように使用して NVMe デバイスの再スキャンを強制できます。

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

NVMe EBS ボリュームをデタッチすると、インスタンスには、ボリュームをデタッチする前に、ファイルシステムのキャッシュまたはメタデータをフラッシュする機会が失われます。したがって、NVMe EBS ボリュームをデタッチする前に、まずそのボリュームを同期およびアンマウントする必要があります。ボリュームのデタッチに失敗した場合は、「[インスタンスからの Amazon EBS ボリュームのデタッチ \(p. 967\)](#)」の説明に従って force-detach コマンドを試すことができます。

## I/O オペレーションタイムアウト

Nitro ベースのインスタンスに接続された EBS ボリュームは、オペレーティングシステムによって提供されるデフォルトの NVMe ドライバーを使用します。ほとんどのオペレーティングシステムは、NVMe デバイスに送信される I/O オペレーションのタイムアウトを指定します。デフォルトのタイムアウトは 30 秒で、nvme\_core.io\_timeout ブートパラメータを使用して変更できます。バージョン 4.6 以前の Linux カーネルでは、このパラメータは nvme.io\_timeout です。

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

Xen インスタンスに接続された EBS ボリュームに類似するエクスペリエンスのため、nvme\_core.io\_timeout を可能な限り最大値に設定することをお勧めします。現在のカーネルでは、最大値は 4294967295 ですが、以前のカーネルでは最大値は 255 です。Linux のバージョンに応じ

て、タイムアウトはすでにサポートされる最大値に設定されていることがあります。たとえば、Amazon Linux AMI 2017.09.01 以降では、デフォルトでタイムアウトが 4294967295 に設定されています。

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

## Amazon EBS – 最適化インスタンス

Amazon EBS– 最適化インスタンスは、最適化された設定スタックを使用し、Amazon EBS I/O に対して追加の専用の容量を提供します。この最適化は、Amazon EBS I/O とその他のインスタンスからのトラフィック間の競合を最小化することで、EBS ボリュームの高パフォーマンスを提供します。

EBS 最適化インスタンスは、Amazon EBS に専用の帯域幅を提供します。EBS 最適化インスタンスにアタッチした場合、汎用 SSD (gp2) ボリュームは、対応するベースラインパフォーマンスとバーストパフォーマンスを 99 % の時間で実現するように設計されています。また、プロビジョンド IOPS SSD (io1) ボリュームは、対応するプロビジョンドパフォーマンスを 99.9 % の時間で実現するように設計されています。スループット最適化 HDD (st1) と Cold HDD (sc1) の両方で、99% の時間はバーストスループットの 90% のパフォーマンス安定性が保証されます。毎時間、予測合計スループットの 99% 達成を目標に、準拠しない期間はほぼ均一に分散されています。詳細については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。

### 目次

- [サポートされるインスタンスタイプ \(p. 1031\)](#)
- [起動時に EBS 最適化を有効化する \(p. 1043\)](#)
- [実行中のインスタンスの EBS 最適化を有効にする \(p. 1043\)](#)

## サポートされるインスタンスタイプ

次の表は、EBS 最適化をサポートするインスタンスタイプを示しています。この表には、Amazon EBS の専用帯域幅、ストリーミング読み取りのワークロードと 128 KiB の I/O サイズでその接続において達成できる一般的な最大スループット、および 16 KiB の I/O を使用している場合にインスタンスがサポートできる IOPS の最大数などが含まれます。アプリケーションのニーズよりも多い専用 Amazon EBS スループットを提供する EBS – 最適化インスタンスを選択します。そうでないと、Amazon EBS と Amazon EC2 間の接続がパフォーマンスのボトルネックになる可能性があります。

### EBS 最適化 (デフォルト)

次の表は、EBS 最適化をサポートするインスタンスタイプを示します。EBS 最適化はデフォルトで有効になっています。EBS 最適化を有効にする必要はなく、EBS 最適化を無効にすると効果はなくなりません。

| インスタンスサイズ    | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|--------------|--------------|-----------------------------|----------------------|
| a1.medium *  | 3,500        | 437.5                       | 20,000               |
| a1.large *   | 3,500        | 437.5                       | 20,000               |
| a1.xlarge *  | 3,500        | 437.5                       | 20,000               |
| a1.2xlarge * | 3,500        | 437.5                       | 20,000               |
| a1.4xlarge   | 3,500        | 437.5                       | 20,000               |
| c4.large     | 500          | 62.5                        | 4,000                |
| c4.xlarge    | 750          | 93.75                       | 6,000                |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ     | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|---------------|--------------|-----------------------------|----------------------|
| c4.2xlarge    | 1,000        | 125                         | 8,000                |
| c4.4xlarge    | 2,000        | 250                         | 16,000               |
| c4.8xlarge    | 4,000        | 500                         | 32,000               |
| c5.large *    | 4,750        | 593.75                      | 20,000               |
| c5.xlarge *   | 4,750        | 593.75                      | 20,000               |
| c5.2xlarge *  | 4,750        | 593.75                      | 20,000               |
| c5.4xlarge    | 4,750        | 593.75                      | 20,000               |
| c5.9xlarge    | 9,500        | 1,187.5                     | 40,000               |
| c5.12xlarge   | 9,500        | 1,187.5                     | 40,000               |
| c5.18xlarge   | 19,000       | 2,375                       | 80,000               |
| c5.24xlarge   | 19,000       | 2,375                       | 80,000               |
| c5.metal      | 19,000       | 2,375                       | 80,000               |
| c5d.large *   | 4,750        | 593.75                      | 20,000               |
| c5d.xlarge *  | 4,750        | 593.75                      | 20,000               |
| c5d.2xlarge * | 4,750        | 593.75                      | 20,000               |
| c5d.4xlarge   | 4,750        | 593.75                      | 20,000               |
| c5d.9xlarge   | 9,500        | 1,187.5                     | 40,000               |
| c5d.12xlarge  | 9,500        | 1,187.5                     | 40,000               |
| c5d.18xlarge  | 19,000       | 2,375                       | 80,000               |
| c5d.24xlarge  | 19,000       | 2,375                       | 80,000               |
| c5d.metal     | 19,000       | 2,375                       | 80,000               |
| c5n.large *   | 4,750        | 593.75                      | 20,000               |
| c5n.xlarge *  | 4,750        | 593.75                      | 20,000               |
| c5n.2xlarge * | 4,750        | 593.75                      | 20,000               |
| c5n.4xlarge   | 4,750        | 593.75                      | 20,000               |
| c5n.9xlarge   | 9,500        | 1,187.5                     | 40,000               |
| c5n.18xlarge  | 19,000       | 2,375                       | 80,000               |
| c5n.metal     | 19,000       | 2,375                       | 80,000               |
| d2.xlarge     | 750          | 93.75                       | 6,000                |
| d2.2xlarge    | 1,000        | 125                         | 8,000                |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ      | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|----------------|--------------|-----------------------------|----------------------|
| d2.4xlarge     | 2,000        | 250                         | 16,000               |
| d2.8xlarge     | 4,000        | 500                         | 32,000               |
| f1.2xlarge     | 1,700        | 212.5                       | 12,000               |
| f1.4xlarge     | 3,500        | 437.5                       | 44,000               |
| f1.16xlarge    | 14,000       | 1,750                       | 75,000               |
| g3s.xlarge     | 850          | 106.25                      | 5,000                |
| g3.4xlarge     | 3,500        | 437.5                       | 20,000               |
| g3.8xlarge     | 7,000        | 875                         | 40,000               |
| g3.16xlarge    | 14,000       | 1,750                       | 80,000               |
| g4dn.xlarge *  | 3,500        | 437.5                       | 20,000               |
| g4dn.2xlarge * | 3,500        | 437.5                       | 20,000               |
| g4dn.4xlarge   | 4,750        | 593.75                      | 20,000               |
| g4dn.8xlarge   | 9,500        | 1,187.5                     | 40,000               |
| g4dn.12xlarge  | 9,500        | 1,187.5                     | 40,000               |
| g4dn.16xlarge  | 9,500        | 1,187.5                     | 40,000               |
| h1.2xlarge     | 1,750        | 218.75                      | 12,000               |
| h1.4xlarge     | 3,500        | 437.5                       | 20,000               |
| h1.8xlarge     | 7,000        | 875                         | 40,000               |
| h1.16xlarge    | 14,000       | 1,750                       | 80,000               |
| i3.large       | 425          | 53.13                       | 3000                 |
| i3.xlarge      | 850          | 106.25                      | 6000                 |
| i3.2xlarge     | 1,700        | 212.5                       | 12,000               |
| i3.4xlarge     | 3,500        | 437.5                       | 16,000               |
| i3.8xlarge     | 7,000        | 875                         | 32,500               |
| i3.16xlarge    | 14,000       | 1,750                       | 65,000               |
| i3.metal       | 19,000       | 2,375                       | 80,000               |
| i3en.large *   | 4,750        | 593.75                      | 20,000               |
| i3en.xlarge *  | 4,750        | 593.75                      | 20,000               |
| i3en.2xlarge * | 4,750        | 593.75                      | 20,000               |
| i3en.3xlarge * | 4,750        | 593.75                      | 20,000               |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ      | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|----------------|--------------|-----------------------------|----------------------|
| i3en.6xlarge   | 4,750        | 593.75                      | 20,000               |
| i3en.12xlarge  | 9,500        | 1,187.5                     | 40,000               |
| i3en.24xlarge  | 19,000       | 2,375                       | 80,000               |
| i3en.metal     | 19,000       | 2,375                       | 80,000               |
| m4.large       | 450          | 56.25                       | 3,600                |
| m4.xlarge      | 750          | 93.75                       | 6,000                |
| m4.2xlarge     | 1,000        | 125                         | 8,000                |
| m4.4xlarge     | 2,000        | 250                         | 16,000               |
| m4.10xlarge    | 4,000        | 500                         | 32,000               |
| m4.16xlarge    | 10,000       | 1,250                       | 65,000               |
| m5.large *     | 4,750        | 593.75                      | 18,750               |
| m5.xlarge *    | 4,750        | 593.75                      | 18,750               |
| m5.2xlarge *   | 4,750        | 593.75                      | 18,750               |
| m5.4xlarge     | 4,750        | 593.75                      | 18,750               |
| m5.8xlarge     | 6,800        | 850                         | 30,000               |
| m5.12xlarge    | 9,500        | 1,187.5                     | 40,000               |
| m5.16xlarge    | 13,600       | 1,700                       | 60,000               |
| m5.24xlarge    | 19,000       | 2,375                       | 80,000               |
| m5.metal       | 19,000       | 2,375                       | 80,000               |
| m5a.large *    | 2,880        | 360                         | 16,000               |
| m5a.xlarge *   | 2,880        | 360                         | 16,000               |
| m5a.2xlarge *  | 2,880        | 360                         | 16,000               |
| m5a.4xlarge    | 2,880        | 360                         | 16,000               |
| m5a.8xlarge    | 4,750        | 593.75                      | 20,000               |
| m5a.12xlarge   | 6,780        | 847.5                       | 30,000               |
| m5a.16xlarge   | 9,500        | 1,187.50                    | 40,000               |
| m5a.24xlarge   | 13,570       | 1,696.25                    | 60,000               |
| m5ad.large *   | 2,880        | 360                         | 16,000               |
| m5ad.xlarge *  | 2,880        | 360                         | 16,000               |
| m5ad.2xlarge * | 2,880        | 360                         | 16,000               |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ      | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|----------------|--------------|-----------------------------|----------------------|
| m5ad.4xlarge   | 2,880        | 360                         | 16,000               |
| m5ad.8xlarge   | 4,750        | 593.75                      | 20,000               |
| m5ad.12xlarge  | 6,780        | 847.5                       | 30,000               |
| m5ad.16xlarge  | 9,500        | 1,187.5                     | 40,000               |
| m5ad.24xlarge  | 13,570       | 1,696.25                    | 60,000               |
| m5d.large *    | 4,750        | 593.75                      | 18,750               |
| m5d.xlarge *   | 4,750        | 593.75                      | 18,750               |
| m5d.2xlarge *  | 4,750        | 593.75                      | 18,750               |
| m5d.4xlarge    | 4,750        | 593.75                      | 18,750               |
| m5d.8xlarge    | 6,800        | 850                         | 30,000               |
| m5d.12xlarge   | 9,500        | 1,187.5                     | 40,000               |
| m5d.16xlarge   | 13,600       | 1,700                       | 60,000               |
| m5d.24xlarge   | 19,000       | 2,375                       | 80,000               |
| m5d.metal      | 19,000       | 2,375                       | 80,000               |
| m5dn.large *   | 4,750        | 593.75                      | 18,750               |
| m5dn.xlarge *  | 4,750        | 593.75                      | 18,750               |
| m5dn.2xlarge * | 4,750        | 593.75                      | 18,750               |
| m5dn.4xlarge   | 4,750        | 593.75                      | 18,750               |
| m5dn.8xlarge   | 6,800        | 850                         | 30,000               |
| m5dn.12xlarge  | 9,500        | 1,187.5                     | 40,000               |
| m5dn.16xlarge  | 13,600       | 1,700                       | 60,000               |
| m5dn.24xlarge  | 19,000       | 2,375                       | 80,000               |
| m5n.large *    | 4,750        | 593.75                      | 18,750               |
| m5n.xlarge *   | 4,750        | 593.75                      | 18,750               |
| m5n.2xlarge *  | 4,750        | 593.75                      | 18,750               |
| m5n.4xlarge    | 4,750        | 593.75                      | 18,750               |
| m5n.8xlarge    | 6,800        | 850                         | 30,000               |
| m5n.12xlarge   | 9,500        | 1,187.5                     | 40,000               |
| m5n.16xlarge   | 13,600       | 1,700                       | 60,000               |
| m5n.24xlarge   | 19,000       | 2,375                       | 80,000               |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ     | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|---------------|--------------|-----------------------------|----------------------|
| p2.xlarge     | 750          | 93.75                       | 6,000                |
| p2.8xlarge    | 5,000        | 625                         | 32,500               |
| p2.16xlarge   | 10,000       | 1,250                       | 65,000               |
| p3.2xlarge    | 1,750        | 218.75                      | 10,000               |
| p3.8xlarge    | 7,000        | 875                         | 40,000               |
| p3.16xlarge   | 14,000       | 1,750                       | 80,000               |
| p3dn.24xlarge | 19,000       | 2,375                       | 80,000               |
| r4.large      | 425          | 53.13                       | 3,000                |
| r4.xlarge     | 850          | 106.25                      | 6,000                |
| r4.2xlarge    | 1,700        | 212.5                       | 12,000               |
| r4.4xlarge    | 3,500        | 437.5                       | 18,750               |
| r4.8xlarge    | 7,000        | 875                         | 37,500               |
| r4.16xlarge   | 14,000       | 1,750                       | 75,000               |
| r5.large *    | 4,750        | 593.75                      | 18,750               |
| r5.xlarge *   | 4,750        | 593.75                      | 18,750               |
| r5.2xlarge *  | 4,750        | 593.75                      | 18,750               |
| r5.4xlarge    | 4,750        | 593.75                      | 18,750               |
| r5.8xlarge    | 6,800        | 850                         | 30,000               |
| r5.12xlarge   | 9,500        | 1,187.5                     | 40,000               |
| r5.16xlarge   | 13,600       | 1,700                       | 60,000               |
| r5.24xlarge   | 19,000       | 2,375                       | 80,000               |
| r5.metal      | 19,000       | 2,375                       | 80,000               |
| r5a.large *   | 2,880        | 360                         | 16,000               |
| r5a.xlarge *  | 2,880        | 360                         | 16,000               |
| r5a.2xlarge * | 2,880        | 360                         | 16,000               |
| r5a.4xlarge   | 2,880        | 360                         | 16,000               |
| r5a.8xlarge   | 4,750        | 593.75                      | 20,000               |
| r5a.12xlarge  | 6,780        | 847.5                       | 30,000               |
| r5a.16xlarge  | 9,500        | 1,187.5                     | 40,000               |
| r5a.24xlarge  | 13,570       | 1,696.25                    | 60,000               |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ      | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|----------------|--------------|-----------------------------|----------------------|
| r5ad.large *   | 2,880        | 360                         | 16,000               |
| r5ad.xlarge *  | 2,880        | 360                         | 16,000               |
| r5ad.2xlarge * | 2,880        | 360                         | 16,000               |
| r5ad.4xlarge   | 2,880        | 360                         | 16,000               |
| r5ad.8xlarge   | 4,750        | 593.75                      | 20,000               |
| r5ad.12xlarge  | 6,780        | 847.5                       | 30,000               |
| r5ad.16xlarge  | 9,500        | 1,187.5                     | 40,000               |
| r5ad.24xlarge  | 13,570       | 1,696.25                    | 60,000               |
| r5d.large *    | 4,750        | 593.75                      | 18,750               |
| r5d.xlarge *   | 4,750        | 593.75                      | 18,750               |
| r5d.2xlarge *  | 4,750        | 593.75                      | 18,750               |
| r5d.4xlarge    | 4,750        | 593.75                      | 18,750               |
| r5d.8xlarge    | 6,800        | 850                         | 30,000               |
| r5d.12xlarge   | 9,500        | 1,187.5                     | 40,000               |
| r5d.16xlarge   | 13,600       | 1,700                       | 60,000               |
| r5d.24xlarge   | 19,000       | 2,375                       | 80,000               |
| r5d.metal      | 19,000       | 2,375                       | 80,000               |
| r5dn.large *   | 4,750        | 593.75                      | 18,750               |
| r5dn.xlarge *  | 4,750        | 593.75                      | 18,750               |
| r5dn.2xlarge * | 4,750        | 593.75                      | 18,750               |
| r5dn.4xlarge   | 4,750        | 593.75                      | 18,750               |
| r5dn.8xlarge   | 6,800        | 850                         | 30,000               |
| r5dn.12xlarge  | 9,500        | 1,187.5                     | 40,000               |
| r5dn.16xlarge  | 13,600       | 1,700                       | 60,000               |
| r5dn.24xlarge  | 19,000       | 2,375                       | 80,000               |
| r5n.large *    | 4,750        | 593.75                      | 18,750               |
| r5n.xlarge *   | 4,750        | 593.75                      | 18,750               |
| r5n.2xlarge *  | 4,750        | 593.75                      | 18,750               |
| r5n.4xlarge    | 4,750        | 593.75                      | 18,750               |
| r5n.8xlarge    | 6,800        | 850                         | 30,000               |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ     | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|---------------|--------------|-----------------------------|----------------------|
| r5n.12xlarge  | 9,500        | 1,187.5                     | 40,000               |
| r5n.16xlarge  | 13,600       | 1,700                       | 60,000               |
| r5n.24xlarge  | 19,000       | 2,375                       | 80,000               |
| t3.nano *     | 2,085        | 260.57                      | 11,800               |
| t3.micro *    | 2,085        | 260.57                      | 11,800               |
| t3.small *    | 2,085        | 260.57                      | 11,800               |
| t3.medium *   | 2,085        | 260.57                      | 11,800               |
| t3.large *    | 2,780        | 347.5                       | 15,700               |
| t3.xlarge *   | 2,780        | 347.5                       | 15,700               |
| t3.2xlarge *  | 2,780        | 347.5                       | 15,700               |
| t3a.nano *    | 2,085        | 260.57                      | 11,800               |
| t3a.micro *   | 2,085        | 260.57                      | 11,800               |
| t3a.small *   | 2,085        | 260.57                      | 11,800               |
| t3a.medium *  | 2,085        | 260.57                      | 11,800               |
| t3a.large *   | 2,780        | 347.5                       | 15,700               |
| t3a.xlarge *  | 2,780        | 347.5                       | 15,700               |
| t3a.2xlarge * | 2,780        | 347.5                       | 15,700               |
| u-6tb1.metal  | 19,000       | 2,375                       | 80,000               |
| u-9tb1.metal  | 19,000       | 2,375                       | 80,000               |
| u-12tb1.metal | 19,000       | 2,375                       | 80,000               |
| u-18tb1.metal | 28,000       | 3,500                       | 160,000              |
| u-24tb1.metal | 28,000       | 3,500                       | 160,000              |
| x1.16xlarge   | 7,000        | 875                         | 40,000               |
| x1.32xlarge   | 14,000       | 1,750                       | 80,000               |
| x1e.xlarge    | 500          | 62.5                        | 3,700                |
| x1e.2xlarge   | 1,000        | 125                         | 7,400                |
| x1e.4xlarge   | 1,750        | 218.75                      | 10,000               |
| x1e.8xlarge   | 3,500        | 437.5                       | 20,000               |
| x1e.16xlarge  | 7,000        | 875                         | 40,000               |
| x1e.32xlarge  | 14,000       | 1,750                       | 80,000               |

| インスタンスサイズ                 | 最大帯域幅 (Mbps) | 最大スループット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|---------------------------|--------------|-----------------------------|----------------------|
| <code>z1d.large</code> *  | 3,170        | 396.25                      | 13,333               |
| <code>z1d.xlarge</code> * | 3,170        | 396.25                      | 13,333               |
| <code>z1d.2xlarge</code>  | 3,170        | 396.25                      | 13,333               |
| <code>z1d.3xlarge</code>  | 4,750        | 593.75                      | 20,000               |
| <code>z1d.6xlarge</code>  | 9,500        | 1,187.5                     | 40,000               |
| <code>z1d.12xlarge</code> | 19,000       | 2,375                       | 80,000               |
| <code>z1d.metal</code>    | 19,000       | 2,375                       | 80,000               |

2020 年 2 月 26 日以降に起動される G4dn、I3en、Inf1、M5a、M5ad、R5a、R5ad、T3、T3a、および Z1d インスタンスは、デフォルトで上記の最大パフォーマンスをサポートします。2020 年 2 月 26 日より前に起動されたインスタンスのパフォーマンスを最大にするには、インスタンスを停止してから起動します。

#### 2019 年 12 月 3 日以降に起動される

C5、C5d、C5n、M5、M5d、M5n、M5dn、R5、R5d、R5n、R5dn、P3dn、u-6tb1.metal、u-9tb1.metal、および u-12tb1.metal インスタンスは、デフォルトで上記の最大パフォーマンスをサポートします。2019 年 12 月 3 日より前に起動されたインスタンスのパフォーマンスを最大にするには、インスタンスを停止してから起動します。

\* これらのインスタンスタイプでは、最大パフォーマンスを 24 時間につき少なくとも 30 分間維持することができます。ワークロードの最大パフォーマンスを 30 分以上維持する必要がある場合は、以下に示されているようにベースラインパフォーマンスに基づいて、インスタンスタイプを選択します。

| インスタンスサイズ                | ベースラインの帯域幅 (Mbps) | ベースラインスループット (MB/秒、128 KiB I/O) | ベースライン IOPS (16 KiB I/O) |
|--------------------------|-------------------|---------------------------------|--------------------------|
| <code>a1.medium</code>   | 300               | 37.5                            | 2,500                    |
| <code>a1.large</code>    | 525               | 65.625                          | 4,000                    |
| <code>a1.xlarge</code>   | 800               | 100                             | 6,000                    |
| <code>a1.2xlarge</code>  | 1,750             | 218.75                          | 10,000                   |
| <code>c5.large</code>    | 650               | 81.25                           | 4,000                    |
| <code>c5.xlarge</code>   | 1,150             | 143.75                          | 6,000                    |
| <code>c5.2xlarge</code>  | 2,300             | 287.5                           | 10,000                   |
| <code>c5d.large</code>   | 650               | 81.25                           | 4,000                    |
| <code>c5d.xlarge</code>  | 1,150             | 143.75                          | 6,000                    |
| <code>c5d.2xlarge</code> | 2,300             | 287.5                           | 10,000                   |
| <code>c5n.large</code>   | 650               | 81.25                           | 4,000                    |
| <code>c5n.xlarge</code>  | 1,150             | 143.75                          | 6,000                    |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ    | ベースラインの帯域幅<br>(Mbps) | ベースラインスルーパット (MB/秒、128 KiB I/O) | ベースライン IOPS (16 KiB I/O) |
|--------------|----------------------|---------------------------------|--------------------------|
| c5n.2xlarge  | 2,300                | 287.5                           | 10,000                   |
| g4dn.xlarge  | 950                  | 118.75                          | 3,000                    |
| g4dn.2xlarge | 1,150                | 143.75                          | 6,000                    |
| i3en.large   | 577                  | 72.1                            | 3,000                    |
| i3en.xlarge  | 1,154                | 144.2                           | 6,000                    |
| i3en.2xlarge | 2,307                | 288.39                          | 12,000                   |
| i3en.3xlarge | 3,800                | 475                             | 15,000                   |
| inf1.xlarge  | 1,190                | 148.75                          | 4,000                    |
| inf1.2xlarge | 1,190                | 148.75                          | 6,000                    |
| m5.large     | 650                  | 81.25                           | 3,600                    |
| m5.xlarge    | 1,150                | 143.75                          | 6,000                    |
| m5.2xlarge   | 2,300                | 287.5                           | 12,000                   |
| m5a.large    | 650                  | 81.25                           | 3,600                    |
| m5a.xlarge   | 1,085                | 135.63                          | 6,000                    |
| m5a.2xlarge  | 1,580                | 197.5                           | 8,333                    |
| m5ad.large   | 650                  | 81.25                           | 3,600                    |
| m5ad.xlarge  | 1,085                | 135.63                          | 6,000                    |
| m5ad.2xlarge | 1,580                | 197.5                           | 8,333                    |
| m5d.large    | 650                  | 81.25                           | 3,600                    |
| m5d.xlarge   | 1,150                | 143.75                          | 6,000                    |
| m5d.2xlarge  | 2,300                | 287.5                           | 12,000                   |
| m5dn.large   | 650                  | 81.25                           | 3,600                    |
| m5dn.xlarge  | 1,150                | 143.75                          | 6,000                    |
| m5dn.2xlarge | 2,300                | 287.5                           | 12,000                   |
| m5n.large    | 650                  | 81.25                           | 3,600                    |
| m5n.xlarge   | 1,150                | 143.75                          | 6,000                    |
| m5n.2xlarge  | 2,300                | 287.5                           | 12,000                   |
| r5.large     | 650                  | 81.25                           | 3,600                    |
| r5.xlarge    | 1,150                | 143.75                          | 6,000                    |
| r5.2xlarge   | 2,300                | 287.5                           | 12,000                   |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
EBS 最適化

| インスタンスサイズ    | ベースラインの帯域幅<br>(Mbps) | ベースラインスルーパット (MB/秒、128 KiB I/O) | ベースライン IOPS (16 KiB I/O) |
|--------------|----------------------|---------------------------------|--------------------------|
| r5a.large    | 650                  | 81.25                           | 3,600                    |
| r5a.xlarge   | 1,085                | 135.63                          | 6,000                    |
| r5a.2xlarge  | 1,580                | 197.5                           | 8,333                    |
| r5ad.large   | 650                  | 81.25                           | 3,600                    |
| r5ad.xlarge  | 1,085                | 135.63                          | 6,000                    |
| r5ad.2xlarge | 1,580                | 197.5                           | 8,333                    |
| r5d.large    | 650                  | 81.25                           | 3,600                    |
| r5d.xlarge   | 1,150                | 143.75                          | 6,000                    |
| r5d.2xlarge  | 2,300                | 287.5                           | 12,000                   |
| r5dn.large   | 650                  | 81.25                           | 3,600                    |
| r5dn.xlarge  | 1,150                | 143.75                          | 6,000                    |
| r5dn.2xlarge | 2,300                | 287.5                           | 12,000                   |
| r5n.large    | 650                  | 81.25                           | 3,600                    |
| r5n.xlarge   | 1,150                | 143.75                          | 6,000                    |
| r5n.2xlarge  | 2,300                | 287.5                           | 12,000                   |
| t3.nano      | 43                   | 5.43                            | 250                      |
| t3.micro     | 87                   | 10.86                           | 500                      |
| t3.small     | 174                  | 21.71                           | 1,000                    |
| t3.medium    | 347                  | 43.43                           | 2,000                    |
| t3.large     | 695                  | 86.86                           | 4,000                    |
| t3.xlarge    | 695                  | 86.86                           | 4,000                    |
| t3.2xlarge   | 695                  | 86.86                           | 4,000                    |
| t3a.nano     | 45                   | 5.63                            | 250                      |
| t3a.micro    | 90                   | 11.25                           | 500                      |
| t3a.small    | 175                  | 21.88                           | 1,000                    |
| t3a.medium   | 350                  | 43.75                           | 2,000                    |
| t3a.large    | 695                  | 86.86                           | 4,000                    |
| t3a.xlarge   | 695                  | 86.86                           | 4,000                    |
| t3a.2xlarge  | 695                  | 86.86                           | 4,000                    |
| z1d.large    | 800                  | 100                             | 3,333                    |

| インスタンスサイズ  | ベースラインの帯域幅 (Mbps) | ベースラインスルーパット (MB/秒、128 KiB I/O) | ベースライン IOPS (16 KiB I/O) |
|------------|-------------------|---------------------------------|--------------------------|
| z1d.xlarge | 1,580             | 197.5                           | 6,667                    |

EBSIOPBalance% メトリクスおよび EBSByteBalance% メトリクスは、インスタンスのサイズが正しいかどうかを判断するのに役立ちます。これらのメトリクスを CloudWatch コンソールで表示して、指定したしきい値に基づいてトリガーされるアラームを設定することができます。これらのメトリクスは、割合(%)で表されます。バランスの割合が常に低いインスタンスのサイズは、拡大する必要があります。バランスの割合が 100% を下回ることのないインスタンスは縮小する必要があります。詳細については、「[CloudWatch を使用したインスタンスのモニタリング \(p. 642\)](#)」を参照してください。

## EBS 最適化をサポート

次の表は、EBS 最適化をサポートするインスタンスタイプを示します。EBS 最適化はデフォルトでは有効になっていません。EBS 最適化は、これらのインスタンスの起動時または実行後に有効にすることができます。前述のパフォーマンスレベルを達成するには、インスタンスで EBS 最適化を有効にする必要があります。デフォルトで EBS 最適化が行われないインスタンスに対して EBS 最適化を有効にするとときは、専用の容量について安価な時間単位の料金を追加でお支払いいただきます。料金については、[Amazon EC2 料金ページのオンデマンドインスタンス](#)で EBS 最適化インスタンスを参照してください。

| インスタンスサイズ  | 最大帯域幅 (Mbps) | 最大スルーパット (MB/秒、128 KiB I/O) | 最大 IOPS (16 KiB I/O) |
|------------|--------------|-----------------------------|----------------------|
| c1.xlarge  | 1,000        | 125                         | 8,000                |
| c3.xlarge  | 500          | 62.5                        | 4,000                |
| c3.2xlarge | 1,000        | 125                         | 8,000                |
| c3.4xlarge | 2,000        | 250                         | 16,000               |
| g2.2xlarge | 1,000        | 125                         | 8,000                |
| i2.xlarge  | 500          | 62.5                        | 4,000                |
| i2.2xlarge | 1,000        | 125                         | 8,000                |
| i2.4xlarge | 2,000        | 250                         | 16,000               |
| m1.large   | 500          | 62.5                        | 4,000                |
| m1.xlarge  | 1,000        | 125                         | 8,000                |
| m2.2xlarge | 500          | 62.5                        | 4,000                |
| m2.4xlarge | 1,000        | 125                         | 8,000                |
| m3.xlarge  | 500          | 62.5                        | 4,000                |
| m3.2xlarge | 1,000        | 125                         | 8,000                |
| r3.xlarge  | 500          | 62.5                        | 4,000                |
| r3.2xlarge | 1,000        | 125                         | 8,000                |
| r3.4xlarge | 2,000        | 250                         | 16,000               |

`i2.8xlarge`、`c3.8xlarge`、および `r3.8xlarge` インスタンスには専用の EBS 帯域幅がないため、EBS 最適化を提供しません。これらのインスタンスでは、ネットワークトライフィックと Amazon EBS トライフィックは同じ 10 ギガビットネットワークインターフェイスで共有されます。

## 起動時に EBS 最適化を有効化する

インスタンスの最適化を有効にするには、EBS 最適化の属性を設定します。

コンソールを使用してインスタンスを起動するときに Amazon EBS 最適化を有効にするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. [Step 1: Choose an Amazon Machine Image (AMI)] で、AMI を選択します。
4. [Step 2: Choose an Instance Type] で、サポート対象の Amazon EBS 最適化として一覧表示されているインスタンスタイプを選択します。
5. [Step 3: Configure Instance Details] で必要なフィールドに入力し、[Launch as EBS-optimized instance] を選択します。前のステップで選択したインスタンスタイプが Amazon EBS 最適化をサポートしていない場合、このオプションは存在しません。選択したインスタンスタイプがデフォルトで Amazon EBS – 最適化される場合、このオプションが選択されており、選択を解除することはできません。
6. 指示に従ってウィザードを完了し、インスタンスを起動します。

コマンドラインを使用してインスタンスを起動するときに EBS 最適化を有効にするには

次のいずれかのオプションを対応するコマンドで使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `--ebs-optimized` と [run-instances \(AWS CLI\)](#)
- `-EbsOptimized` と [New-EC2Instance \(AWS Tools for Windows PowerShell\)](#)

## 実行中のインスタンスの EBS 最適化を有効にする

実行中のインスタンスの最適化を有効または無効にするには、Amazon EBS – 最適化インスタンス属性を変更します。

コンソールを使用して、実行中のインスタンスで EBS 最適化を有効にするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Instances] をクリックし、インスタンスを変更します。
3. [Actions] をクリックして [Instance State] を選択し、[Stop] をクリックします。

### Warning

インスタンスを停止すると、インスタンストアボリューム上のデータは消去されます。インスタンストアボリュームのデータを保持するには、このデータを永続的ストレージに必ずバックアップしてください。

4. 確認ダイアログボックスで [Yes, Stop] をクリックします。インスタンスが停止するまで、数分かかる場合があります。
5. インスタンスが選択されたままの状態で [Actions] をクリックし、[Instance Settings] を選択して [Change Instance Type] をクリックします。
6. [Change Instance Type] ダイアログボックスで、次のいずれかを実行します。

- 目的のインスタンスのインスタンスタイプがデフォルトで Amazon EBS – 最適化される場合、[EBS-optimized] が選択されており、選択解除できません。そのインスタンスでは Amazon EBS 最適化がすでに有効であるため、[Cancel] をクリックします。
  - 目的のインスタンスのインスタンスタイプが Amazon EBS 最適化をサポートしている場合は、[EBS 最適化]、[Apply] を選択します。
  - 目的のインスタンスのインスタンスタイプが Amazon EBS 最適化をサポートしていない場合は、[EBS 最適化] を選択することはできません。[Instance Type] から、Amazon EBS 最適化をサポートするインスタンスタイプを選択し、[EBS 最適化]、[Apply] を選択します。
7. [Actions]、[Instance State]、[Start] の順に選択します。

コマンドラインを使用して、実行中のインスタンスで EBS 最適化を有効にするには

次のいずれかのオプションを対応するコマンドで使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- ebs-optimized と [modify-instance-attribute \(AWS CLI\)](#)
- EbsOptimized と [Edit-EC2InstanceAttribute \(AWS Tools for Windows PowerShell\)](#)

## Linux インスタンスの Amazon EBS ボリュームのパフォーマンス

Amazon EBS のパフォーマンスは、I/O 特性やインスタンスとボリュームの設定などを含むいくつかの要因に左右されます。一般的に、Amazon EBS 製品および Amazon EC2 製品の詳細ページに記載されているガイドに従うだけで、良好なパフォーマンスを実現することができます。ただし、そのプラットフォームにおけるピークパフォーマンスを達成するには、多少のチューニングを行う必要のあるケースもあります。このトピックでは、一般的なベストプラクティスや、特定のユースケースに固有のパフォーマンスチューニングについて説明します。最適な設定を決定するには、ベンチマークに加えて、実際のワークロードからの情報でパフォーマンスをチューニングすることをお勧めします。EBS ボリュームの基本操作について理解したら、必要とする I/O パフォーマンスと、Amazon EBS パフォーマンスを向上させるオプションを確認し、そのパフォーマンス要件に対応できるようにすることをお勧めします。

### Note

AWS が EBS ボリュームタイプのパフォーマンスを更新しても、既存のボリュームにすぐには反映されない場合があります。古いボリュームで完全なパフォーマンスを確認するためには、最初に `ModifyVolume` アクションの実行が必要になる場合があります。詳細については、「[Linux の EBS ボリュームのサイズ、IOPS、またはタイプの変更](#)」を参照してください。

### コンテンツ

- [Amazon EBS パフォーマンスのヒント \(p. 1044\)](#)
- [I/O の特性とモニタリング \(p. 1047\)](#)
- [Amazon EBS ボリュームの初期化 \(p. 1049\)](#)
- [Linux での RAID 構成 \(p. 1051\)](#)
- [EBS ボリュームのベンチマーク \(p. 1055\)](#)

## Amazon EBS パフォーマンスのヒント

ここに示すヒントは、さまざまなユーザーシナリオで、EBS ボリュームから最適なパフォーマンスを得るために必要なベストプラクティスを表しています。

## EBS 最適化インスタンスを使用する

EBS 最適化スループットがサポートされていないインスタンスでは、インスタンスと EBS ボリュームの間のトラフィックが、ネットワークトラフィックと競合する場合があります。EBS 最適化インスタンスでは、これら 2 種類のトラフィックを分離した状態が維持されます。EBS 最適化インスタンスの設定によっては、追加コストが発生する場合 (C3, R3, M3 など) と、追加コストなしで常に EBS 最適化状態になる場合 (M4, C4, C5, D2 など) があります。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

## パフォーマンスの計算方法を理解する

EBS ボリュームのパフォーマンスを測定する場合、関連する測定単位と、パフォーマンスの計算方法を理解することが重要です。詳細については、「[I/O の特性とモニタリング \(p. 1047\)](#)」を参照してください。

## ワークロードを理解する

EBS ボリュームの最大パフォーマンス、I/O 操作の数およびサイズ、各アクションが完了するまでの所要時間は、互いに関連しています。これらの各要因 (パフォーマンス、I/O、レイテンシー) は相互に影響を与えます。また、アプリケーションが異なると、影響を受ける要因もさまざまになります。詳細については、「[EBS ボリュームのベンチマーク \(p. 1055\)](#)」を参照してください。

## スナップショットからボリュームを初期化する際のパフォーマンス低下を理解する

スナップショットから復元された新しい EBS ボリュームの各データブロックに初めてアクセスするときは、レイテンシーが著しく増加します。このパフォーマンスヒットは、以下のいずれかの方法で回避できます。

- 各ブロックへのアクセスが、ボリュームの本番環境への移行前に起こるようにする。このプロセスは、初期化と呼ばれます (以前は事前ウォーミングと呼ばれていました)。詳細については、「[Amazon EBS ボリュームの初期化 \(p. 1049\)](#)」を参照してください。
- スナップショットの高速スナップショット復元を有効化して、スナップショットから作成される EBS ボリュームが作成時に完全に初期化され、各ボリュームのあらゆるプロビジョンドパフォーマンスが即座に発揮されるようにします。詳細については、「[Amazon EBS 高速スナップショット復元 \(p. 1024\)](#)」を参照してください。

## HDD パフォーマンスが低下する要因

スループット最適化 HDD (st1) ボリュームまたは Cold HDD (sc1) ボリュームのスナップショットを作成すると、スナップショットの進行中はボリュームのベースライン値までパフォーマンスが低下します。この動作は、これらのボリュームタイプに固有です。パフォーマンスが制限される他の要因としては、インスタンスでのサポート範囲を超えるスループットの強要、スナップショットから復元したボリュームの初期化中のパフォーマンス低下、ボリュームに対する大量の小さなランダム I/O などがあります。HDD ボリュームのスループットを計算する方法については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。

アプリケーションから送られる I/O リクエスト数が十分でない場合も、パフォーマンスに影響します。これは、ボリュームのキュー長や I/O サイズを確認することで監視できます。このキュー長とは、アプリケーションからボリュームへの I/O リクエストのうち処理待ちのものの数です。最大限の安定性を確保するために、HDD-Backed ボリュームで 1 MiB のシーケンシャル I/O を実行する際には、キュー長 (整数に四捨五入) を 4 以上に保つ必要があります。ボリュームの安定したパフォーマンスを確保する方法については、「[I/O の特性とモニタリング \(p. 1047\)](#)」を参照してください。

## st1 および sc1 で読み取りの多い高スループットワークロードを先読みする

一部のワークロードでは読み取りが多く、オペレーティングシステムのページキャッシュを通じて (たとえば、ファイルシステムから) ブロックデバイスへのアクセスが行われます。この場合、最大スループッ

トを実現するには、先読みを 1 MiB に設定することをお勧めします。これは HDD ボリュームにのみ適用されるブロックデバイス単位の設定です。

ブロックデバイスに対する現在の先読み値を調べるには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo blockdev --report /dev/<device>
```

ブロックデバイス情報は次の形式で返されます。

| RO | RA  | SSZ | BSZ  | StartSec | Size       | Device        |
|----|-----|-----|------|----------|------------|---------------|
| rw | 256 | 512 | 4096 | 4096     | 8587820544 | /dev/<device> |

表示されているデバイスについては、先読み値として 256 (デフォルト値) が報告されています。この数値にセクターサイズ (512 バイト) を乗算すると、先読みバッファのサイズ (この場合は 128 KiB) を得ることができます。バッファ値を 1 MiB に設定するには、次のコマンドを使用します。

```
[ec2-user ~]$ sudo blockdev --setra 2048 /dev/<device>
```

最初のコマンドをもう一度実行して、先読み設定が 2,048 になったことを確認します。

この設定は、ワーカロードがサイズの大きなシーケンシャル I/O で構成される場合にのみ使用してください。ワーカロードの内容として、サイズの小さなランダム I/O がほとんどであれば、この設定を使用すると逆にパフォーマンスが低下します。一般的に、サイズの小さい I/O やランダム I/O が大部分を占めるワーカロードの場合は、`st1` や `sc1` ではなく、汎用 SSD (`gp2`) ボリュームの使用を検討してください。

## 最新の Linux カーネルを使用する

間接記述子がサポートされている最新の Linux カーネルを使用します。Linux カーネル 3.8 以降には、すべてこのサポートがあり、現行世代の EC2 インスタンスも同様です。平均 I/O サイズが 44 KiB 前後であれば、間接記述子がサポートされていないインスタンスやカーネルを使用している可能性があります。Amazon CloudWatch のメトリクスから平均 I/O サイズを得る方法については、「[I/O の特性とモニタリング \(p. 1047\)](#)」を参照してください。

`st1` または `sc1` ボリュームで最大スループットを達成するには、256 の値を、`xen_blkfront.max` パラメータ (Linux カーネルバージョン 4.6 未満の場合) または `xen_blkfront.max_indirect_segments` パラメータ (Linux カーネルバージョン 4.6 以降の場合) に適用することを推奨します。適切なパラメータは、OS の起動コマンドラインで設定できます。

たとえば、Amazon Linux AMI では、`/boot/grub/menu.lst` に記述されている GRUB 設定で `kernel` 行の末尾に追加できます。

```
kernel /boot/vmlinuz-4.4.5-15.26.amzn1.x86_64 root=LABEL=/ console=ttyS0
xen_blkfront.max=256
```

後のカーネルの場合、コマンドは次のようにになります。

```
kernel /boot/vmlinuz-4.9.20-11.31.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
xen_blkfront.max_indirect_segments=256
```

この設定を有効にするには、インスタンスを再起動する必要があります。

詳細については、「[GRUB の設定 \(p. 178\)](#)」を参照してください。他の Linux ディストリビューションでは (特に GRUB ブートローダーが使用されていない場合)、カーネル パラメータの調整に別のアプローチが必要になることがあります。

EBS I/O の特性の詳細については、このトピックの「[Amazon EBS: パフォーマンスを考慮した設計](#)」プレゼンテーションを参照してください。

## RAID 0 を使用してインスタンスのリソース使用率を最大化する

一部のインスタンスタイプでは、単一の EBS ボリュームをプロビジョニングする場合よりも I/O スループットを増やすことができます。複数の gp2、io1、st1、または sc1 ボリュームを RAID 0 設定で結合し、これらのインスタンス用の利用可能な帯域幅を使用できます。詳細については、「[Linux での RAID 構成 \(p. 1051\)](#)」を参照してください。

## Amazon CloudWatch を使用してパフォーマンスを追跡する

Amazon Web Services は、Amazon CloudWatch による分析と表示が可能な Amazon EBS のパフォーマンスマトリクスと、ボリュームの健全性の監視に使用できるステータスチェックを提供します。詳細については、「[ボリュームのステータスのモニタリング \(p. 960\)](#)」を参照してください。

## I/O の特性とモニタリング

ボリューム設定が同じであっても、特定の I/O 特性により EBS ボリュームのパフォーマンス動作が向上します。SSD-Backed ボリューム – 汎用 SSD (gp2) および プロビジョンド IOPS SSD (io1) – I/O 操作がランダムでもシーケンシャルでも、安定したパフォーマンスが提供されます。HDD-Backed ボリューム – スループット最適化 HDD (st1) および Cold HDD (sc1) – サイズが大きくシーケンシャルな I/O の場合のみ、最適なパフォーマンスが提供されます。アプリケーションにおける SSD ボリュームおよび HDD ボリュームのパフォーマンスについて理解するには、ボリュームに対するデマンド、ボリュームに対して使用可能な IOPS の量、I/O 操作が完了するまでにかかる時間、およびボリュームのスループット制限の間のつながりについて知ることが重要です。

### IOPS

IOPS とは、1 秒あたりの入出力操作数を表す測定単位です。操作は KiB 単位で測定され、基礎となるドライブテクノロジーは、ボリュームタイプが 1 つの I/O としてカウントするデータの最大量を決定します。SSD ボリュームは HDD ボリュームよりもはるかに効率的に小規模またはランダム I/O を処理するため、I/O サイズは SSD ボリュームで 256 KiB、HDD ボリュームで 1,024 KiB に制限されています。

小さな I/O 操作が物理的に連続している場合、Amazon EBS ではできる限りこれらを最大サイズになるまで単一の I/O 操作にマージして処理します。たとえば、SSD ボリュームでは、1 つの 1,024 KiB の I/O 操作は 4 つのオペレーション ( $1,024 \div 256 = 4$ ) としてカウントされ、それぞれが 32 KiB の連続する 8 つの I/O オペレーションは、1 つのオペレーション ( $8 \times 32 = 256$ ) としてカウントされます。ただし、それが 32 KiB の 8 つのランダム I/O 操作は、8 つの操作としてカウントされます。32 KiB 未満の各 I/O 操作は 1 つの操作としてカウントされます。

同様に、HDD-Backed ボリュームの場合、1,024 KiB の単一 I/O 操作も、8 つの連続する 128 KiB の操作も、1 つの操作としてカウントされます。ただし、128 KiB のランダム I/O 操作 8 つは、8 つの操作としてカウントされます。

このため、3,000 IOPS をサポートする SSD-Backed ボリュームを (io1 ボリュームを 3,000 IOPS でプロビジョニングするか、gp2 ボリュームを 1000 GiB にサイズ設定することによって) 作成し、十分な帯域幅を提供できる EBS 最適化インスタンスにアタッチした場合、1 秒あたり最大 3,000 件の I/O 操作分のデータを転送できます (スループットは I/O サイズで決まります)。

### ボリュームのキュー長とレイテンシー

ボリュームのキュー長とは、デバイスに対する保留中の I/O リクエストの数です。レイテンシーとは、実際に I/O 操作にかかるエンドツーエンドのクライアント時間です。つまり、I/O を EBS に送信してから、読み取りまたは書き込みの I/O が完了したという確認を EBS から受信するまでの時間ということになります。ゲストオペレーティングシステムまたは EBS へのネットワークリンクでのボトルネックを回避するには、I/O サイズとレイテンシーに合わせて正しくキュー長を調整する必要があります。

最適なキュー長は、アプリケーションがどの程度 IOPS およびレイテンシーの影響を受けるかによってワークロードごとに異なります。EBS ボリュームで利用可能なパフォーマンスをフル活用するための十分な I/O リクエストがワークロードから提供されないと、プロビジョニングどおりの IOPS またはスループットをボリュームで実現できないことがあります。

トランザクション量の多いアプリケーションは、I/O レイテンシーの上昇の影響を受けるため、`io1` および `gp2` の SSD-Backed ボリュームが適しています。キュー長を小さく抑え、ボリュームで利用可能な限り高い IOPS を維持することにより、低いレイテンシーと高い IOPS を実現できます。ボリュームで利用可能な IOPS を超える IOPS を継続的に強制すると、I/O レイテンシーが上昇する可能性があります。

スループットが高いアプリケーションは I/O レイテンシーの上昇による影響を受けにくいため、`st1` および `sc1` の HDD-Backed ボリュームが適しています。HDD-Backed ボリュームに対する高いスループットを維持するには、サイズの大きなシーケンシャル I/O を実行するときにキュー長を大きくします。

#### I/O サイズとボリュームのスループット制限

SSD-Backed ボリュームで I/O サイズが非常に大きい場合は、ボリュームのスループット制限に達することにより、IOPS 値がプロビジョニングした値よりも小さくなることがあります。たとえば、利用可能なバーストクレジットを持つ 1000 GiB 未満の `gp2` ボリュームの IOPS 制限は 3,000 で、ボリュームスループット制限は 250 MiB/s です。256 KiB の I/O サイズを使用している場合、ボリュームは 1000 IOPS ( $1000 \times 256 \text{ KiB} = 250 \text{ MiB}$ ) でスループット制限に達します。より小さい I/O サイズ (16 KiB など) では、スループットが 250 MiB/s を大幅に下回っているため、同じボリュームで 3,000 IOPS を維持できます。(これらの例では、ボリュームの I/O がインスタンスのスループット限界に達していないと想定しています)。各 EBS ボリュームタイプのスループット制限については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。

サイズの小さな I/O 操作では、インスタンス内で測定した IOPS がプロビジョニングの値より高くなることがあります。この状況は、インスタンスのオペレーティングシステムが、小さな I/O 操作を Amazon EBS に渡す前に、大きな操作にマージした場合に生じます。

ワークロードが HDD バックアップの `st1` および `sc1` ボリュームでシーケンシャル I/O を使用する場合、ワークロードで使用している I/O がシーケンシャルであれば、インスタンス内で測定した IOPS が予測値より高くなることがあります。この状況は、インスタンスのオペレーティングシステムが、シーケンシャル I/O をマージし、1,024 KiB サイズ単位でカウントすることによって生じます。ワークロードで小さな I/O またはランダム I/O を使用している場合は、スループットが予測値より低くなることがあります。これは、非シーケンシャルの各ランダム I/O をカウントして合計の IOPS カウントを求める過程で、予測より早くボリュームの IOPS 制限に達する場合があるためです。

EBS ボリュームタイプに関係なく、設定したはずの IOPS またはスループットを得られない場合は、EC2 インスタンスの帯域幅が制限要因になっていないか確認してください。最適なパフォーマンスを得るには、常に現行世代の EBS 最適化インスタンス (または、10 Gb/s のネットワーク接続を確保できるインスタンス) を使用してください。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。予想された IOPS が得られない別の原因として、EBS ボリュームに対して十分な I/O を提供していないことが考えられます。

#### CloudWatch で I/O 特性を監視する

これらの I/O 特性は、各ボリュームの [CloudWatch メトリクス \(p. 960\)](#) を使用して監視できます。考慮する必要のある重要なメトリクスには、次のようなものがあります。

- `BurstBalance`
- `VolumeReadBytes`
- `VolumeWriteBytes`
- `VolumeReadOps`
- `VolumeWriteOps`
- `VolumeQueueLength`

`BurstBalance` は `gp2`、`st1`、および `sc1` ボリュームのバーストバケットバランスを残りのバランスの割合として表示します。バーストバケットが減ると、ボリューム I/O (`gp2` ボリューム用) またはボリュームスループット (`st1` および `sc1` ボリューム用) はベースラインにスロットリングされます。この理由でボリュームに制限が適用されているかどうかを確認するには、`BurstBalance` の値を調べてください。

st1 および sc1 の HDD-Back ボリュームは、1,024 KiB の最大 I/O サイズを活用するワークロードに最適な設定になっています。ボリュームの平均 I/O サイズを求めるには、VolumeWriteBytes を VolumeWriteOps で除算します。読み取り操作にも同じ計算を適用できます。平均 I/O サイズが 64 KiB を下回る場合は、st1 または sc1 のボリュームに送る I/O 操作のサイズを大きくすると、パフォーマンスが向上します。

#### Note

平均 I/O サイズが 44 KiB 前後であれば、間接記述子がサポートされていないインスタンスやカーネルを使用している可能性があります。Linux カーネル 3.8 以降には、すべてこのサポートがあり、現行世代のインスタンスも同様です。

I/O レイテンシーが必要な値よりも高い場合、VolumeQueueLength をチェックして、アプリケーションがプロビジョニングした IOPS 以上の処理を実行しようとしていることを確認します。ボリュームが提供できる以上の IOPS を必要とするアプリケーションの場合は、より高いレベルのベースパフォーマンスを実現するサイズの gp2 ボリュームを使用するか、より多くの IOPS をプロビジョニングできる io1 を使用して、レイテンシーを低く抑えることを検討してください。

Amazon EBS I/O の特性の詳細については、このトピックの「[Amazon EBS: パフォーマンスを考慮した設計](#)」プレゼンテーションを参照してください。

## Amazon EBS ボリュームの初期化

新しい EBS ボリュームは、利用可能になるとすぐに最大のパフォーマンスを発揮し、初期化(以前は事前ウォーミングと呼ばれました)を必要としません。

スナップショットから復元されたボリュームへのアクセスは、ストレージブロックが Amazon S3 からプルダウンされてボリュームに書き込まれると可能になります。この事前処理には一定の時間がかかるため、各ブロックへの初回アクセス時には、I/O 操作のレイテンシーが著しく増加する可能性があります。ボリュームのパフォーマンスは、すべてのブロックがダウンロードされてボリュームに書き込まれると正常値に達します。

#### Important

スナップショットから復元された io1 ボリュームを初期化している間は、ボリュームのパフォーマンスが想定レベルの 50% を下回る場合があります。このため、ボリュームの [I/O Performance] ステータスチェックでは warning 状態が表示されます。これは想定の動作です。初期化中の io1 ボリュームの warning 状態は無視してかまいません。詳細については、「[EBS ボリュームステータスチェック \(p. 960\)](#)」を参照してください。

ほとんどのアプリケーションにとって、ボリュームの存続期間全体で初期化コストを割り当てることは、許容範囲内です。本番環境におけるこの初期パフォーマンスヒットは、以下のいずれかの方法で回避できます。

- ボリューム全体の即時初期化を強制する。詳細については、「[Amazon EBS ボリュームの初期化 \(p. 1049\)](#)」を参照してください。
- スナップショットの高速スナップショット復元を有効化して、スナップショットから作成される EBS ボリュームが作成時に完全に初期化され、各ボリュームのあらゆるプロビジョンドパフォーマンスが即座に発揮されるようにします。詳細については、「[Amazon EBS 高速スナップショット復元 \(p. 1024\)](#)」を参照してください。

## Linux の Amazon EBS ボリュームの初期化

新しい EBS ボリュームは、利用可能になるとすぐに最大のパフォーマンスを発揮し、初期化(以前は事前ウォーミングと呼ばれました)を必要としません。スナップショットから復元されたボリュームの場合、ボリュームのすべてのブロックから読み取りを行うには、dd ユーティリティまたは fio ユーティリティを使用します。ボリュームのすべてのデータが保持されます。

## スナップショットから復元された Linux のボリュームを初期化するには

- 新しく復元されたボリュームを Linux インスタンスにアタッチします。
- インスタンスのプロックデバイスを一覧表示するには、lsblk コマンドを使用します。

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0 30G  0 disk
xvda1 202:1    0   8G  0 disk /
```

ここでは、新しいボリューム/dev/xvdf がアタッチされていますが、マウントされていないことがわかります (MOUNTPOINT 列の下にリストされているパスがないため)。

- デバイスのすべてのブロックを読み取るには、dd ユーティリティまたは fio ユーティリティを使用します。Linux システムにデフォルトでインストールされているのは dd コマンドですが、マルチスレッドの読み取りが可能な fio の方が、処理速度が大幅に速くなります。

### Note

このステップは、使用している EC2 インスタンスの帯域幅、ボリュームに対してプロビジョニングされている IOPS、そしてボリュームのサイズに応じて、数分から数時間かかることがあります。

[dd] if (入力ファイル) パラメータは、初期化するドライブに設定します。of (出力ファイル) パラメータは、Linux ヌル仮想デバイス /dev/null に設定する必要があります。bs パラメータは、読み取り操作のブロックサイズに設定します。最適なパフォーマンスを得るには、この値を 1 MB に設定します。

### Important

dd を誤って使用すると、ボリュームのデータが失われる場合があります。以下のコマンド例に正確に従ってください。if=/dev/**xvdf** パラメータのみ、読み出しているデバイスの名前によって異なります。

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

[fio] システムに fio がインストールされている場合、ボリュームを初期化するには次のコマンドを使用します。--filename (入力ファイル) パラメータは、初期化するドライブに設定します。

```
[ec2-user ~]$ sudo fio --filename=/dev/xvdf --rw=read --bs=128k --iodepth=32 --
ioengine=libaio --direct=1 --name=volume-initialize
```

Amazon Linux に fio をインストールするには、次のコマンドを使用します。

```
sudo yum install -y fio
```

Ubuntu に fio インストールするには、次のコマンドを使用します。

```
sudo apt-get install -y fio
```

この操作が終了すると、読み取り操作のレポートが表示されます。これでボリュームを使用する準備ができました。詳細については、「[Linux で Amazon EBS ボリュームを使用できるようにする \(p. 956\)](#)」を参照してください。

## Linux での RAID 構成

Amazon EBS では、従来のベアメタルサーバーで使用できる標準的な RAID 設定はすべて使用できます。ただしその RAID 設定が、お使いのインスタンスのオペレーティングシステムでサポートされている必要があります。これは、RAID がすべてソフトウェアレベルで実現されるためです。単一のボリュームで実現できる以上の I/O パフォーマンスを実現するため、RAID 0 では複数のボリュームをともにストライピングできます。インスタンスでの冗長性確保のため、RAID 1 では 2 つのボリュームを同時にミラーリングできます。

Amazon EBS ボリュームのデータは、同じアベイラビリティーボーン内の複数のサーバーにレプリケートされます。これは、コンポーネントの 1 つに障害が発生したことが原因でデータが失われるのを防ぐためです。このレプリケーションにより、一般的なコモディティディスクドライブに比べて Amazon EBS ボリュームの信頼性が 10 倍に高まります。詳細については、Amazon EBS 製品の詳細ページの「[Amazon EBS の可用性と耐久性](#)」を参照してください。

### Note

RAID ボリュームからの起動は避ける必要があります。GRUB は通常 RAID アレイの 1 つのデバイスのみにインストールされ、ミラーリングされたデバイスの 1 つに障害が発生した場合、オペレーティングシステムを起動できなくなる場合があります。

Windows インスタンスで RAID アレイを作成する必要がある場合は、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows での RAID 設定](#)」を参照してください。

### コンテンツ

- [RAID 設定オプション \(p. 1051\)](#)
- [Linux での RAID アレイの作成 \(p. 1052\)](#)
- [RAID 配列でボリュームのスナップショットを作成する \(p. 1055\)](#)

## RAID 設定オプション

次の表では、一般的な RAID 0 と RAID 1 のオプションを比較しています。

| 設定     | 使用アイテム                                                                  | 利点                                                                  | 欠点                                                                                                       |
|--------|-------------------------------------------------------------------------|---------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| RAID 0 | (データレプリケーションが既に別にセットアップされている) 使用頻度が高いデータベースなど、I/O パフォーマンスが耐障害性よりも重要な場合。 | I/O がストライプ内のボリュームにわたって分散されます。ボリュームを追加すると、スループットと IOPS を追加したことになります。 | ストライプのパフォーマンスは、セット内で最もパフォーマンスが良くないボリュームに制限されます。単一ボリュームの損失により、アレイのデータが完全に失われます。                           |
| RAID 1 | クリティカルなアプリケーション内など、耐障害性が I/O パフォーマンスより重要な場合。                            | データ堅牢性の観点から見て、より安全です。                                               | 書き込みパフォーマンスの向上は得られません。データが同時に複数のボリュームに書き込まれるため、非 RAID 構成と比較して Amazon EC2 と Amazon EBS の間に大きな帯域幅が必要となります。 |

### Important

RAID 5 と RAID 6 ではボリュームに使用できる IOPS の一部がパリティ書き込み操作によって消費されるため、Amazon EBS にはこれらの RAID モードをお勧めしません。RAID アレイの構

成によっては、これらの RAID モードで使用できる IOPS が RAID 0 構成と比較して 20 ~ 30% 少なくなる場合があります。これらの RAID モードにはコストの増加も伴います。ボリュームサイズとスピードが同じ 2 ボリュームの RAID 0 アレイの方が、コストが 2 倍の 4 ボリュームの RAID 6 アレイよりも優れたパフォーマンスが得られる場合があります。

RAID 0 アレイを作成すると、単一の Amazon EBS ボリュームでプロビジョニングする場合よりも、ファイルシステムで高レベルのパフォーマンスが実現されます。RAID 1 アレイは、冗長性を向上させるため、データのミラーリングを可能にします。この手順を実行する前に、RAID アレイで必要となるサイズと、プロビジョニングする IOPS の数を決定してください。

RAID 0 アレイの最終的なサイズは、アレイ内のボリュームサイズの合計です。帯域幅は、アレイ内のボリュームで利用可能な帯域幅の合計です。RAID 1 アレイの最終的なサイズと帯域幅は、アレイ内にあるボリュームのサイズと帯域幅に等しくなります。たとえば、それぞれ 4,000 のプロビジョンド IOPS が設定された 2 つの 500 GiB Amazon EBS io1 ボリュームがある場合、使用可能な帯域幅が 8,000 IOPS、スループットが 1,000 MiB/秒の 500 GiB の RAID 0 アレイか、または使用可能な帯域幅が 4,000 IOPS、スループットが 500 MiB/秒の 500 GiB の RAID 1 アレイを構築できます。

このドキュメントでは、基本的な RAID のセットアップの例を紹介します。RAID 設定、パフォーマンス、および復旧の詳細については、Linux RAID Wiki ([https://raid.wiki.kernel.org/index.php/Linux\\_Raid](https://raid.wiki.kernel.org/index.php/Linux_Raid)) を参照してください。

## Linux での RAID アレイの作成

次の手順に従って RAID アレイを作成します。Windows インスタンスに関する手順は、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows での RAID アレイの作成](#)」で入手できます。

Linux で RAID アレイを作成するには

- アレイに Amazon EBS ボリュームを作成します。詳細については、「[Amazon EBS ボリュームの作成 \(p. 949\)](#)」を参照してください。

### Important

アレイに作成するボリュームのサイズと IOPS パフォーマンス値は同一にしてください。EC2 インスタンスで利用可能な帯域幅を超えるアレイを作成しないよう注意してください。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

- アレイをホストするインスタンスに Amazon EBS ボリュームをアタッチします。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。
- mdadm コマンドを使用して、新しくアタッチした Amazon EBS ボリュームから論理 RAID デバイスを作成します。`[number_of_volumes]` に、構成するアレイ内のボリュームの数を入れ、`device_name` に、アレイ内の各ボリュームのデバイス名 (`/dev/xvdf` など) を入れます。`MY_RAID` を、配列の一意の名前で置き換えることもできます。

### Note

インスタンスのデバイス名を見つけるには、`lsblk` コマンドを使用してデバイスのリストを表示します。

(RAID 0 のみ) RAID 0 アレイを作成するには、次のコマンドを実行します (アレイをストライプ化するには `--level=0` オプションをメモします)。

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=0 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

(RAID 1 のみ) RAID 1 アレイを作成するには、次のコマンドを実行します (アレイをミラーリングするには `--level=1` オプションをメモします)。

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=1 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

4. RAID アレイでの初期化と同期に許可された時間です。これらのオペレーションの進行状況は、次のコマンドを使用して追跡できます。

```
[ec2-user ~]$ sudo cat /proc/mdstat
```

出力例を次に示します。

```
Personalities : [raid1]
md0 : active raid1 xvdf[1] xvdf[0]
      20955008 blocks super 1.2 [2/2] [UU]
      [======>.....]  resync = 46.8% (9826112/20955008) finish=2.9min
speed=63016K/sec
```

一般的に、次のコマンドで RAID アレイに関する詳細情報を表示できます。

```
[ec2-user ~]$ sudo mdadm --detail /dev/md0
```

出力例を次に示します。

```
/dev/md0:
      Version : 1.2
      Creation Time : Mon Jun 27 11:31:28 2016
      Raid Level : raid1
      Array Size : 20955008 (19.98 GiB 21.46 GB)
      Used Dev Size : 20955008 (19.98 GiB 21.46 GB)
      Raid Devices : 2
      Total Devices : 2
      Persistence : Superblock is persistent

      Update Time : Mon Jun 27 11:37:02 2016
      State : clean
      ...
      ...
      ...

      Number  Major  Minor  RaidDevice State
          0      202     80        0    active sync   /dev/sdf
          1      202     96        1    active sync   /dev/sdg
```

5. RAID アレイにファイルシステムを作成し、それを後でマウントするときに、使用するラベルをそのファイルシステムに提供します。たとえば、ext4 ファイルシステムのラベル **MY\_RAID** で作成するには、次のコマンドを実行します。

```
[ec2-user ~]$ sudo mkfs.ext4 -L MY_RAID /dev/md0
```

アプリケーションの要件またはオペレーティングシステムの制限によって、ext3 や XFS などの異なるファイルシステムタイプを使用できます(対応するファイルシステム作成コマンドについては、ファイルシステムの資料を参照してください)。

6. RAID アレイがブート時に自動的に再編成されることを確認するには、RAID 情報を含むように設定ファイルを作成します。

```
[ec2-user ~]$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
```

Note

Amazon Linux 以外の Linux ディストリビューションを使用している場合、このファイルは別の場所に配置する必要があります。詳細については、Linux システムの man mdadm.conf を参照してください。

7. 新しい RAID 設定のブロックデバイスマジュールを適切に事前ロードする新しいラムディスクイメージを作成する:

```
[ec2-user ~]$ sudo dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

8. RAID アレイのマウントポイントを作成します。

```
[ec2-user ~]$ sudo mkdir -p /mnt/raid
```

9. 最後に、作成したマウントポイントに RAID デバイスをマウントします。

```
[ec2-user ~]$ sudo mount LABEL=MY_RAID /mnt/raid
```

これで RAID デバイスを使用する準備ができました。

10. (オプション) システムブート時に常に、この Amazon EBS ボリュームをマウントするには、/etc/fstab ファイルにデバイス用のエントリを追加します。

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

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

- b. お好みのテキストエディタ (nano や vim など) を使用して、/etc/fstab ファイルを開きます。
- c. 「UUID=」で始まる行にコメントして、ファイルの最後に次の形式で RAID ボリュームの新しい行を追加します。

```
device_label mount_point file_system_type fs_mntops fs_freq fs_passno
```

この行の最後の 3 つのフィールドは、ファイルシステムのマウントオプション、ファイルシステムのダンプ頻度、ブート時に実行されるファイルシステムチェックの順番です。これらの値がわからない場合、以下の例の値を使用してください (defaults,nofail 0 2))。/etc/fstab エントリの詳細については、fstab マニュアルページを参照してください (コマンドラインで man fstab と入力します)。たとえば、マウントポイント /mnt/raid にラベル MY\_RAID を持つデバイスに ext4 ファイルシステムをマウントするには、/etc/fstab に次のエントリを追加します。

Note

このボリュームをアタッチしないでインスタンスを起動することを目的としている場合(たとえば、このボリュームが異なるインスタンス間で移動される可能性がある場合)、nofail マウントオプションを追加し、ボリュームのマウントでエラーが発生してもインスタンスが起動できるようにしてください。Debian から派生した OS (Ubuntu など) では、nobootwait マウントオプションも追加する必要があります。

|               |           |      |                 |   |   |
|---------------|-----------|------|-----------------|---|---|
| LABEL=MY_RAID | /mnt/raid | ext4 | defaults,nofail | 0 | 2 |
|---------------|-----------|------|-----------------|---|---|

- d. 新しいエントリを /etc/fstab に追加した後、エントリが正しく動作するかを確認する必要があります。sudo mount -a コマンドを使用して、すべてのファイルシステムを /etc/fstab にマウントします。

```
[ec2-user ~]$ sudo mount -a
```

前のコマンドを実行してもエラーが発生しない場合、/etc/fstab ファイルに問題はあります。次回ブート時にファイルシステムは自動的にマウントされます。このコマンドを実行してエラーが発生した場合、エラーを調べて、/etc/fstab を修正してください。

#### Warning

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

- e. (オプション) /etc/fstab のエラーの修正方法が不明な場合、次のコマンドを使って、いつでもバックアップの /etc/fstab ファイルを復元することができます。

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

## RAID 配列でボリュームのスナップショットを作成する

スナップショットを使用して、RAID 配列で EBS ボリュームのデータをバックアップする場合には、そのスナップショットが一貫していることを確認する必要があります。これは、ボリュームのスナップショットが個別に作成されるためです。同期されていないスナップショットから RAID 配列の EBS ボリュームを復元すると、配列の整合性は低下します。

RAID 配列の一貫性のあるスナップショットを作成するには、「[EBS マルチボリュームスナップショット](#)」を使用します。マルチボリュームスナップショットを使用すると、EC2 インスタンスにアタッチされている複数の EBS ボリュームにわたって、ポイントインタイムで、データ調整済みのクラッシュ整合性スナップショットを取得できます。スナップショットは複数の EBS ボリュームにわたって自動的に作成されるため、一貫性を保証できるように、ボリューム間で調整してインスタンスを停止する必要はありません。詳細については、「[Amazon EBS スナップショットの作成](#)」のマルチボリューム EBS スナップショットを作成するステップを参照してください。

## EBS ボリュームのベンチマーク

I/O ワークロードをシミュレートすることで、Amazon EBS ボリュームのパフォーマンスをテストできます。手順は次のとおりです。

1. EBS 最適化インスタンスを作成する。
2. 新しい EBS ボリュームを作成します。
3. EBS 最適化インスタンスにボリュームをアタッチする。
4. ブロックデバイスを設定およびマウントします。
5. ツールをインストールし、I/O パフォーマンスを評価する。
6. ボリュームの I/O パフォーマンスを評価する。
7. ボリュームを削除し、料金が発生しないようにインスタンスを終了する。

#### Important

手順の一部を実行すると、ベンチマークを実行する EBS ボリューム上の既存のデータが破壊されます。ベンチマーク手順は、本番ボリュームではなく、テスト目的で特別に作成されたボリュームで使用するために用意されています。

## インスタンスのセットアップ

EBS ボリュームで最適なパフォーマンスを実現するには、EBS 最適化インスタンスを使用することをお勧めします。EBS 最適化インスタンスは、Amazon EC2 および Amazon EBS 間の専用スループットとインスタンスを提供します。EBS 最適化インスタンスは、Amazon EC2 と Amazon EBS の間で所定の帯域幅を実現するものであり、インスタンスタイプに応じて仕様で選択できます。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

EBS 最適化インスタンスを作成するには、Amazon EC2 コンソールを使用してインスタンスを起動するときに [Launch as an EBS-Optimized instance (EBS 最適化インスタンスとして起動する)] を選択するか、コマンドラインを使用して `--ebs-optimized` を指定します。必ずこのオプションをサポートする現行世代のインスタンスタイプを起動してください。詳細については、「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」を参照してください。

## プロビジョンド IOPS SSD (io1) ボリュームのセットアップ

io1 ボリュームを作成するには、Amazon EC2 コンソールを使用してボリュームを作成するときに、[プロビジョンド IOPS SSD] を選択するか、コマンドラインで `--type io1 --iops n` (`n` は 100 ~ 64,000 の整数) を指定します。EBS ボリュームの仕様の詳細については、「[Amazon EBS ボリュームの種類 \(p. 933\)](#)」を参照してください。EBS ボリュームの作成の詳細については、「[Amazon EBS ボリュームの作成 \(p. 949\)](#)」を参照してください。インスタンスへのボリュームのアタッチについては、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。

テストの例では、高いレベルのパフォーマンスを提供する、6 ボリュームを持つ RAID アレイを作成することをお勧めします。ボリューム数ではなく、プロビジョニングしたギガバイト（および io1 ボリュームに対してプロビジョニングした IOPS 数）に対して料金が発生するので、複数の小さなボリュームを作成し、そのボリュームを使用してストライプセットを使用するための追加コストはありません。Oracle Orion を使用してボリュームを評価する場合は、Oracle ASM と同じ方法でストライピングをシミュレートできます。したがって、Orion でストライピングを行えるようにすることをお勧めします。別のベンチマークツールを使用する場合は、ボリュームのストライピングを自分で行う必要があります。

6 ボリュームのストライプセットを Amazon Linux で作成するには、次のようなコマンドを使用します。

```
[ec2-user ~]$ sudo mdadm --create /dev/md0 --level=0 --chunk=64 --raid-devices=6 /dev/sdf /dev/sdg /dev/sdh /dev/sdi /dev/sdj /dev/sdk
```

この例では、XFS ファイルシステムを使用します。要件を満たすファイルシステムを使用する必要があります。次のコマンドを使用して XFS ファイルシステムサポートをインストールします。

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

さらに、次のコマンドを使用して、XFS ファイルシステムの作成、マウント、および所有権の割り当てを行います。

```
[ec2-user ~]$ sudo mkdir -p /mnt/p_iops_vo10 && sudo mkfs.xfs /dev/md0
[ec2-user ~]$ sudo mount -t xfs /dev/md0 /mnt/p_iops_vo10
[ec2-user ~]$ sudo chown ec2-user:ec2-user /mnt/p_iops_vo10/
```

## スループット最適化 HDD (st1) または Cold HDD (sc1) ボリュームのセットアップ

st1 ボリュームを作成するには、Amazon EC2 コンソールを使用してボリュームを作成するときに [スループット最適化 HDD] を選択するか、コマンドラインを使用して `--type st1` を指定します。sc1 ボリュームを作成するには、Amazon EC2 コンソールを使用してボリュームを作成するときに [-type sc1] を選択するか、コマンドラインを使用して Cold HDD を指定します。EBS ボリュームの作成の詳細については、「[Amazon EBS ボリュームの作成 \(p. 949\)](#)」を参照してください。インスタンスへのこれらのボリュームのアタッチについては、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。

AWS では、AWS CloudFormation と共に JSON テンプレートを使用して、このセットアップ手順を簡単にすることができます。これには、[テンプレート](#)にアクセスして JSON ファイルとして保存します。AWS CloudFormation では、独自の SSH キーを設定して、st1 ボリュームを評価するためのパフォーマンステスト環境を簡単にセットアップすることができます。テンプレートを使用すると、現行世代のインスタンスと 2 TiB の st1 ボリュームが作成され、このボリュームが /dev/xvdf のインスタンスにアタッチされます。

#### テンプレートを使用して HDD ボリュームを作成するには

1. AWS CloudFormation コンソール (<https://console.aws.amazon.com/cloudformation>) を開きます。
2. [Create Stack] を選択します。
3. [Upload a Template to Amazon S3] を選択し、さきほど入手した JSON テンプレートを選択します。
4. スタックに、"ebs-perf-testing" のような名前を付け、インスタンスタイプ (デフォルトは r3.8xlarge) および SSH キーを選択します。
5. [Next] を 2 回選択し、[Create Stack] を選択します。
6. 新しいスタックのステータスが [CREATE\_IN\_PROGRESS] から [COMPLETE] に移行したら、[Outputs (出力)] を選択して新しいインスタンスのパブリック DNS エントリを取得します。このインスタンスには、2 TiB の st1 ボリュームがアタッチされます。
7. ユーザー **ec2-user** として、前のステップで DNS エントリから取得したホスト名を使用し、新しいスタックに SSH を使用して接続します。
8. [ベンチマークツールをインストールする \(p. 1057\)](#) に進みます。

#### ベンチマークツールをインストールする

EBS ボリュームのパフォーマンスを評価するために使用できるツールの一部を、次の表に示します。

| ツール                          | 説明                                                                                                                                                                                                                                                       |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fio                          | <p>I/O パフォーマンスを評価します(fio は libaio-devel に依存することに注意してください。)</p> <p>fio を Amazon Linux にインストールするには、次のコマンドを実行します。</p> <pre>[ec2-user ~]\$ sudo yum install -y fio</pre> <p>Ubuntu に fio インストールするには、次のコマンドを実行します。</p> <pre>sudo apt-get install -y fio</pre> |
| Oracle Orion Calibration ツール | Oracle データベースで使用するストレージシステムの I/O パフォーマンスを調整します。                                                                                                                                                                                                          |

これらのベンチマークツールは、さまざまなテストパラメータをサポートしています。使用するのは、ボリュームがサポートするワークロードを見積もるためのコマンドです。評価に必要な基本的なコマンドの例を以下に示します。

#### ボリュームキュー長の選択

ワークロードとボリュームタイプに基づいて最適なボリュームキュー長を選択します。

##### SSD-Backed ボリュームのキュー長

SSD-Backed ボリュームでワークロードに最適なキュー長を決定するには、使用可能な 1000 IOPS ごとにキュー長 1 を指定するようにお勧めします (gp2 ボリュームのベースライン、io1 ボリュームにプロビ

ジョギングする値)。その後、アプリケーションのパフォーマンスを監視して、アプリケーション要件に応じて値を調整することができます。

プロビジョニングした IOPS、スループット、または最適なシステムキュー長(現在は 32 に設定)に達するまでは、キュー長を大きくする方が有益です。たとえば、IOPS として 3,000 がプロビジョニングされたボリュームでは、キュー長 3 を設定します。アプリケーションに最適な値を確認するには、これらの値を増減して調整してください。

### HDD-Backed ボリュームのキュー長

HDD-Backed ボリュームのワークロードに対する最適なキュー長を決定するには、1 MiB のシーケンシャル I/O の実行時に 4 以上のキュー長を設定しておくようお勧めします。その後、アプリケーションのパフォーマンスを監視して、アプリケーション要件に応じて値を調整することができます。たとえば、2 TiB の st1 ボリュームで、バーストスループットが 500 MiB/秒、IOPS が 500 の場合は、1,024 KiB、512 KiB、または 256 KiB のシーケンシャル I/O を実行する際に、キュー長をそれぞれ 4、8、または 16 に設定します。アプリケーションに最適な値を確認するには、これらの値を増減して調整してください。

### C ステートを無効にします

ベンチマークを実行する前に、プロセッサの C ステートを無効にする必要があります。サポートされている CPU の一時的にアイドリング状態のコアは、電力を節約するために C ステートに入ることができます。コアが処理を再開するために呼び出されると、コアが再び完全に動作するまで一定の時間が経過します。このレインジャーは、プロセッサのベンチマークルーチンを妨げる可能性があります。C ステートとその EC2 インスタンスタイプでサポートされるインスタンスの詳細については、「[EC2 インスタンスタイプのプロセッサのステート制御](#)」を参照してください。

#### Linux オペレーティングシステムで C ステートを無効にする

Amazon Linux、RHEL、および CentOS で C ステートを無効にするには、次のようにします。

1. C ステートの数を取得します。

```
$ cpupower idle-info | grep "Number of idle states:"
```

2. C ステート c1 から cN にして無効にします。理想的には、コアは c0 ステートにある必要があります。

```
$ for i in `seq 1 $((N-1))` ; do cpupower idle-set -d $i; done
```

### ベンチマークテストを実行する

次の手順では、さまざまな EBS ボリュームタイプに対するベンチマークコマンドについて説明します。

EBS ボリュームがアタッチされている EBS 最適化インスタンスで、次のコマンドを実行します。EBS ボリュームをスナップショットから復元した場合は、ベンチマークテストを実行する前に、必ず初期化してください。詳細については、「[Amazon EBS ボリュームの初期化 \(p. 1049\)](#)」を参照してください。

ボリュームのテストが完了したら、クリーンアップに関するトピック「[Amazon EBS ボリュームの削除 \(p. 969\)](#)」および「[インスタンスの終了 \(p. 545\)](#)」を参照してください。

#### io1 ボリュームの評価

作成したストライプセットで fio を実行します。

次のコマンドは、16 KB のランダム書き込みオペレーションを実行します。

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_volo --name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G --numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

次のコマンドは、16 KB のランダム読み取りオペレーションを実行します。

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_volo --name fio_test_file --direct=1 --rw=randread --bs=16k --size=1G --numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
```

結果の読み方については、チュートリアル「[fio のディスク IO パフォーマンスの確認](#)」を参照してください。

### st1 ボリュームおよび sc1 ボリュームの評価

st1 ボリュームまたは sc1 ボリュームで fio を実行します。

#### Note

これらのテストを実行する前に、「[st1 および sc1 で読み取りの多い高スループットワークロードを先読みする \(p. 1045\)](#)」の説明に従って、バッファ付き I/O をインスタンスに設定してください。

次のコマンドでは、アタッチされた st1 ブロックデバイス (/dev/xvdf など) に対して、1 MiB のシーケンシャル読み取り操作を実行します。

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=read --randrepeat=0 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 --name=fio_direct_read_test
```

次のコマンドでは、アタッチされた st1 ブロックデバイスに対して、1 MiB のシーケンシャル書き込み操作を実行します。

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=write --randrepeat=0 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 --name=fio_direct_write_test
```

ワークロードによっては、ブロックデバイスの異なる部分に対してシーケンシャル読み取りとシーケンシャル書き込みの組み合わせを実行するケースがあります。このようなワークロードを評価する場合は、読み取りと書き込みに対して別々の fio ジョブを同時に実行し、fio offset\_increment オプションを使用して、ブロックデバイスの別々の場所を各ジョブに割り当てるをお勧めします。

このワークロードの実行は、シーケンシャル書き込みまたはシーケンシャル読み取りのワークロードの場合より、少し複雑になります。テキストエディターを使用して、次の内容を含む fio ジョブファイル (この例では fio\_rw\_mix.cfg) を作成します。

```
[global]
clocksource=clock_gettime
randrepeat=0
runtime=180
offset_increment=100g

[sequential-write]
bs=1M
ioengine=libaio
direct=1
iodepth=8
filename=/dev/<device>
do_verify=0
rw=write
rwmixread=0
rwmixwrite=100

[sequential-read]
```

```
bs=1M
ioengine=libaio
direct=1
iodepth=8
filename=/dev/<device>
do_verify=0
rw=read
rwmixread=100
rwmixwrite=0
```

次に、以下のコマンドを実行します。

```
[ec2-user ~]$ sudo fio fio_rw_mix.cfg
```

結果の読み方については、チュートリアル「[fio のディスク IO パフォーマンスの確認](#)」を参照してください。

シケンシャルの読み取りまたは書き込みの操作を使用しても、ダイレクト I/O の fio ジョブを複数実行した場合は、st1 および sc1 ボリュームで予測を下回るスループットになります。単一のダイレクト I/O ジョブを使用し、iodepth パラメータを指定して、I/O 操作の同時実行数を制御することをお勧めします。

## Amazon EBS の Amazon CloudWatch メトリクス

CloudWatch メトリクスは、ボリュームの実行動作を表示または分析したり、それらの実行動作についてのアラームを設定したりするために使用できる統計データです。

次の表は、Amazon EBS ボリュームで利用可能なモニタリングデータのタイプをまとめたものです。

| タイプ | 説明                                                                       |
|-----|--------------------------------------------------------------------------|
| 基本  | データは自動的に 5 分間無料で取得できます。このデータには、EBS-backed インスタンスのルートデバイスピリュームのデータが含まれます。 |
| 詳細  | プロビジョンド IOPS SSD (io1) ボリュームは、1 分間のメトリクスを CloudWatch に自動的に送信します。         |

CloudWatch からデータを取得したときに、Period リクエストパラメータを含めて、返されるデータの詳細程度を指定できます。これは、データの収集に使用する期間（5 分間）とは異なります。有効なデータが確実に返されるように、リクエストには収集期間以上の期間を指定することをお勧めします。

データは、CloudWatch API または Amazon EC2 コンソールのいずれかを使用して取得できます。コンソールは CloudWatch API から未加工データを取得し、そのデータに基づいて一連のグラフを表示します。必要に応じて、API のデータまたはコンソールのグラフのいずれかを使用できます。

## Amazon EBS のメトリクス

Amazon Elastic Block Store (Amazon EBS) は、複数のメトリクスのデータポイントを CloudWatch に送信します。Amazon EBS 汎用 SSD (gp2)、スループット最適化 HDD (st1)、Cold HDD (sc1)、および磁気 (スタンダード) ボリュームは自動的に 5 分間のメトリクスを CloudWatch に送信します。プロビジョンド IOPS SSD (io1) ボリュームは、1 分間のメトリクスを CloudWatch に自動的に送信します。データは、ボリュームがインスタンスにアタッチされている場合にのみ CloudWatch に報告されます。

これらのメトリクスの一部は、Nitro ベースのインスタンスに違いがあります。Nitro システムに基づくインスタンスタイプのリストについては、「[Nitro ベースのインスタンス \(p. 187\)](#)」を参照してください。

AWS/EBS 名前空間には、次のメトリクスが含まれます。

## メトリクス

- ボリュームメトリクス (p. 1061)
- 高速スナップショット復元メトリクス (p. 1064)

## ボリュームメトリクス

| メトリクス            | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VolumeReadBytes  | <p>指定された期間の読み取りオペレーションに関する情報を提供します。Sum 統計は、期間内に転送されたバイトの総数をレポートします。Average 統計は、期間中の各読み取りオペレーションの平均サイズを報告します。ただし、平均が指定された期間にわたる平均を表す Nitro ベースのインスタンスにアタッチされたボリュームを除きます。SampleCount 統計は期間中に読み取りオペレーションの合計数を報告します。ただし、サンプル数が統計的計算で使用されるデータポイントの数を表す、Nitro ベースのインスタンスにアタッチされたボリュームを除きます。Xen インスタンスでは、ボリュームに読み取りアクティビティがある場合にのみデータが報告されます。</p> <p>このメトリクスの Minimum および Maximum 統計は、Nitro ベースのインスタンスにアタッチされたボリュームでのみサポートされます。</p> <p>単位: バイト</p> |
| VolumeWriteBytes | <p>指定された期間の書き込みオペレーションに関する情報を提供します。Sum 統計は、期間内に転送されたバイトの総数をレポートします。Average 統計は、期間中の各書き込みオペレーションの平均サイズを報告します。ただし、平均が指定された期間にわたる平均を表す Nitro ベースのインスタンスにアタッチされたボリュームを除きます。SampleCount 統計は期間中に書き込みオペレーションの合計数を報告します。ただし、サンプル数が統計的計算で使用されるデータポイントの数を表す、Nitro ベースのインスタンスにアタッチされたボリュームを除きます。Xen インスタンスでは、ボリュームに書き込みアクティビティがある場合にのみデータが報告されます。</p> <p>このメトリクスの Minimum および Maximum 統計は、Nitro ベースのインスタンスにアタッチされたボリュームでのみサポートされます。</p> <p>単位: バイト</p> |
| VolumeReadOps    | <p>指定期間内の読み取りオペレーションの総数。</p> <p>その期間の 1 秒あたりの読み込み I/O 操作回数 (読み取り IOPS) の平均を算出するには、その期間の読み取りオペレーション回数の合計をその期間の秒数で割ります。</p> <p>このメトリクスの Minimum および Maximum 統計は、Nitro ベースのインスタンスにアタッチされたボリュームでのみサポートされます。</p> <p>単位: カウント</p>                                                                                                                                                                                                                 |
| VolumeWriteOps   | 指定期間内の書き込みオペレーションの総数。                                                                                                                                                                                                                                                                                                                                                                                                                       |

| メトリクス                             | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                   | <p>その期間の 1 秒あたりの書き込み I/O 操作回数 (書き込み IOPS) の平均を算出するには、その期間の書き込みオペレーション回数の合計をその期間の秒数で割ります。</p> <p>このメトリクスの <code>Minimum</code> および <code>Maximum</code> 統計は、Nitro ベースのインスタンスにアタッチされたボリュームでのみサポートされます。</p> <p>単位: カウント</p>                                                                                                                                                                                                                                                                                  |
| <code>VolumeTotalReadTime</code>  | <p><b>Note</b></p> <p>このメトリクスは、マルチアタッチが有効なボリュームではサポートされません。</p> <p>指定期間内に完了した操作すべての読み取りオペレーションに要した時間 (秒) の合計。複数のリクエストが同時に送信された場合は、この合計が期間の長さを超えることがあります。たとえば、期間が 5 分間 (300 秒) で、その期間内に完了した操作の数が 700 あり、1 つの操作に 1 秒かかるとすれば、この値は 700 秒となります。Xen インスタンスでは、ボリュームに読み取りアクティビティがある場合にのみデータが報告されます。</p> <p>このメトリクスの <code>Average</code> 統計は、Nitro ベースのインスタンスにアタッチされたボリュームには該当しません。</p> <p>このメトリクスの <code>Minimum</code> および <code>Maximum</code> 統計は、Nitro ベースのインスタンスにアタッチされたボリュームでのみサポートされます。</p> <p>単位: 秒</p>  |
| <code>VolumeTotalWriteTime</code> | <p><b>Note</b></p> <p>このメトリクスは、マルチアタッチが有効なボリュームではサポートされません。</p> <p>指定期間内に完了した操作すべての、書き込みオペレーションに要した時間 (秒) の合計。複数のリクエストが同時に送信された場合は、この合計が期間の長さを超えることがあります。たとえば、期間が 5 分間 (300 秒) で、その期間内に完了した操作の数が 700 あり、1 つの操作に 1 秒かかるとすれば、この値は 700 秒となります。Xen インスタンスでは、ボリュームに書き込みアクティビティがある場合にのみデータが報告されます。</p> <p>このメトリクスの <code>Average</code> 統計は、Nitro ベースのインスタンスにアタッチされたボリュームには該当しません。</p> <p>このメトリクスの <code>Minimum</code> および <code>Maximum</code> 統計は、Nitro ベースのインスタンスにアタッチされたボリュームでのみサポートされます。</p> <p>単位: 秒</p> |

| メトリクス                      | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VolumeIdleTime             | <p><b>Note</b></p> <p>このメトリクスは、マルチアタッチが有効なボリュームではサポートされません。</p> <p>指定期間内に、読み取りと書き込みのどちらの操作も行われなかつた時間(秒)の合計。</p> <p>このメトリクスの Average 統計は、Nitro ベースのインスタンスにアタッチされたボリュームには該当しません。</p> <p>このメトリクスの Minimum および Maximum 統計は、Nitro ベースのインスタンスにアタッチされたボリュームでのみサポートされます。</p> <p>単位: 秒</p>                                                                                                                                                                                                                   |
| VolumeQueueLength          | <p>指定期間内に完了を待っていた読み取りおよび書き込みの操作リクエストの数。</p> <p>このメトリクスの Sum 統計は、Nitro ベースのインスタンスにアタッチされたボリュームには該当しません。</p> <p>このメトリクスの Minimum および Maximum 統計は、Nitro ベースのインスタンスにアタッチされたボリュームでのみサポートされます。</p> <p>単位: カウント</p>                                                                                                                                                                                                                                                                                        |
| VolumeThroughputPercentage | <p><b>Note</b></p> <p>このメトリクスは、マルチアタッチが有効なボリュームではサポートされません。</p> <p>プロビジョンド IOPS SSD ボリュームでのみ使用されます。Amazon EBS ボリュームにプロビジョニングされた合計 IOPS (1 秒間あたりの I/O 操作回数) に対する、提供された IOPS の割合(パーセント)。プロビジョンド IOPS SSD ボリュームは、1 年の 99.9 パーセントにわたり、プロビジョニンド IOPS の 10 パーセント以内のパフォーマンスを提供します。</p> <p>書き込みの間、他に保留中の I/O リクエストが 1 分以内になれば、メトリクス値は 100% となります。また、お客様が行ったアクション(たとえば使用率ピーク時にボリュームのスナップショットを作成する、EBS 最適化インスタンス以外でボリュームを実行する、そのボリュームのデータに初めてアクセスするなど)によってボリュームの I/O 性能が一時的に低下する場合があります。</p> <p>単位: パーセント</p> |

| メトリクス                      | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VolumeConsumedReadWriteOps | <p>プロビジョンド IOPS SSD ボリュームでのみ使用されます。指定された期間内に消費された読み書き操作の合計数 (256K キャパシティーユニットに標準化)。</p> <p>それぞれ 256K より小さい I/O 操作は、1 消費 IOPS とカウントされます。256K より大きい I/O 操作は、256K キャパシティーユニットでカウントされます。たとえば、1024K I/O は 4 消費 IOPS としてカウントされます。</p> <p>単位: カウント</p>                                                                                                                                                                                                                                                                              |
| BurstBalance               | <p>汎用 SSD (gp2)、スループット最適化 HDD (st1)、および Cold HDD (sc1) ボリュームでのみ使用されます。バーストバケットに残っている I/O クレジット (gp2 用) またはスループットクレジット (st1 と sc1 用) の割合に関する情報を提供します。データは、ボリュームがアクティブな場合にのみ CloudWatch に報告されます。ボリュームがアタッチされていない場合、データは報告されません。</p> <p>このメトリクスの Sum 統計は、Nitro ベースのインスタンスにアタッチされたボリュームには該当しません。</p> <p>ボリュームのベースラインパフォーマンスが最大バーストパフォーマンスを超えない場合、クレジットは消費されません。バーストバランスは 0% (Nitro ベースのインスタンス) または 100% (Nitro ベースのインスタンス以外) として報告されます。詳細については、「<a href="#">I/O クレジットおよびバーストパフォーマンス (p. 936)</a>」を参照してください。</p> <p>単位: パーセント</p> |

## 高速スナップショット復元メトリクス

| メトリクス                               | 説明                                                                                                                                                         |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FastSnapshotRestoreCreditsUsed      | <p>蓄積できるボリューム作成クレジットの最大数。このメトリクスは、アベイラビリティーボリュームごとにスナップショット単位で報告されます。</p> <p>最も有益な統計は Average です。Minimum 統計と Maximum 統計の結果は、Average と同じであり、代わりに使用できます。</p> |
| FastSnapshotRestoreCreditsAvailable | <p>使用可能なボリューム作成クレジットの数。このメトリクスは、アベイラビリティーボリュームごとにスナップショット単位で報告されます。</p> <p>最も有益な統計は Average です。Minimum 統計と Maximum 統計の結果は、Average と同じであり、代わりに使用できます。</p>   |

## Amazon EBS メトリクスのディメンション

サポートされているディメンションはボリューム ID (VolumeId) です。すべての使用可能な統計がボリューム ID でフィルタリングされます。

[ボリュームメトリクス \(p. 1061\)](#)の場合、サポートされているディメンションはボリューム ID (VolumeId) です。すべての使用可能な統計がボリューム ID でフィルタリングされます。

[高速スナップショット復元メトリクス \(p. 1064\)](#)の場合、サポートされているディメンションはスナップショット ID (SnapshotId) とアベイラビリティゾーン (AvailabilityZone) です。

## Amazon EC2 コンソールのグラフ

ボリュームを作成したら、Amazon EC2 コンソールでボリュームのモニタリンググラフを確認できます。コンソールの [Volumes] ページでボリュームを選択し、[Monitoring] を選択します。次の表は、表示されるグラフをまとめたものです。右側の欄は、各グラフを作成するために CloudWatch API の未加工データメトリクスがどのように使用されるかを示しています。すべてのグラフの期間は5分です。

| グラフ                     | 未加工メトリクスの使用に関する説明                                                                                                                                                                                                                                                                                                                            |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 読み込み帯域幅 (KiB/s)         | $\text{Sum}(\text{VolumeReadBytes}) / \text{Period} / 1024$                                                                                                                                                                                                                                                                                  |
| 書き込み帯域幅 (KiB/s)         | $\text{Sum}(\text{VolumeWriteBytes}) / \text{Period} / 1024$                                                                                                                                                                                                                                                                                 |
| 読み込みスループット (IOPS)       | $\text{Sum}(\text{VolumeReadOps}) / \text{Period}$                                                                                                                                                                                                                                                                                           |
| 書き込みスループット (IOPS)       | $\text{Sum}(\text{VolumeWriteOps}) / \text{Period}$                                                                                                                                                                                                                                                                                          |
| 平均キュー長 (ops) (オペレーション)  | $\text{Avg}(\text{VolumeQueueLength})$                                                                                                                                                                                                                                                                                                       |
| % アイドル時間                | $\text{Sum}(\text{VolumeIdleTime}) / \text{Period} \times 100$                                                                                                                                                                                                                                                                               |
| 平均読み込みサイズ (KiB/オペレーション) | <p><math>\text{Avg}(\text{VolumeReadBytes}) / 1024</math></p> <p>Nitro ベースのインスタンスの場合、<a href="#">CloudWatch Metric Math</a> での公式を使用して平均読み込みサイズを算出します。</p> $(\text{Sum}(\text{VolumeReadBytes}) / \text{Sum}(\text{VolumeReadOps})) / 1024$ <p>VolumeReadBytes および VolumeReadOps メトリクスは EBS CloudWatch コンソールで使用できます。</p>                      |
| 平均書き込みサイズ (KiB/オペレーション) | <p><math>\text{Avg}(\text{VolumeWriteBytes}) / 1024</math></p> <p>Nitro ベースのインスタンスの場合、<a href="#">CloudWatch Metric Math</a> での公式を使用して平均書き込みサイズを算出します。</p> $(\text{Sum}(\text{VolumeWriteBytes}) / \text{Sum}(\text{VolumeWriteOps})) / 1024$ <p>VolumeWriteBytes および VolumeWriteOps メトリクスは EBS CloudWatch コンソールで使用できます。</p>                 |
| 平均読み込み待ち時間 (ms/オペレーション) | <p><math>\text{Avg}(\text{VolumeTotalReadTime}) \times 1000</math></p> <p>Nitro ベースのインスタンスの場合、<a href="#">CloudWatch Metric Math</a> での公式を使用して平均レイテンシーを算出します。</p> $(\text{Sum}(\text{VolumeTotalReadTime}) / \text{Sum}(\text{VolumeReadOps})) \times 1000$ <p>VolumeTotalReadTime および VolumeReadOps メトリクスは EBS CloudWatch コンソールで使用できます。</p> |

| グラフ                     | 未加工メトリクスの使用に関する説明                                                                                                                                                                                                                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 平均書き込み待ち時間 (ms/オペレーション) | $\text{Avg}(\text{VolumeTotalWriteTime}) \times 1000$<br>Nitro ベースのインスタンスの場合、CloudWatch Metric Math で次の公式を使用して平均書き込み待ち時間を算出します。<br>$(\text{Sum}(\text{VolumeTotalWriteTime}) / \text{Sum}(\text{VolumeWriteOps})) * 1000$<br>VolumeTotalWriteTime および VolumeWriteOps メトリクスは EBS CloudWatch コンソールで使用できます。 |

平均レイテンシーグラフおよび平均サイズグラフでは、期間中に完了したオペレーション(読み込みまたは書き込みのうち、いずれかグラフに該当する方)の合計数に基づいて平均が計算されます。

## Amazon EBS での Amazon CloudWatch Events

Amazon EBS は、Amazon CloudWatch Events での様々なボリューム、スナップショット、および暗号化ステータスの変更に基づいて通知を出力します。CloudWatch イベントでは、ボリューム、スナップショット、または暗号化キーの状態の変化に応じてプログラムによるアクションをトリガーするルールを設定できます。たとえば、スナップショットが作成された場合、AWS Lambda 関数をトリガーして、完了したスナップショットを他のアカウントと共有したり、それを災害対策の目的で別のリージョンにコピーしたりできます。

CloudWatch でのイベントは、JSON オブジェクトとして表されます。イベント固有のフィールドは、JSON オブジェクトの「detail(詳細)」セクションに表示されます。“event” フィールドにはイベント名が入ります。“result” フィールドには、イベントをトリガーしたアクションの完了したステータスが入ります。詳細については、『Amazon CloudWatch Events ユーザーガイド』の「CloudWatch イベント の イベントパターン」を参照してください。

詳細については、Amazon CloudWatch ユーザーガイドの「[イベントの使用](#)」を参照してください。

### 目次

- [EBS ボリュームイベント \(p. 1066\)](#)
- [EBS スナップショットイベント \(p. 1069\)](#)
- [EBS ボリュームの変更イベント \(p. 1073\)](#)
- [EBS 高速スナップショット復元イベント \(p. 1073\)](#)
- [AWS Lambda による CloudWatch イベント の処理 \(p. 1074\)](#)

## EBS ボリュームイベント

Amazon EBS は、次のボリュームイベントが発生したときに、CloudWatch イベント にイベントを送信します。

### イベント

- [ボリュームを作成 \(createVolume\) \(p. 1067\)](#)
- [ボリュームを削除 \(deleteVolume\) \(p. 1068\)](#)
- [ボリュームのアタッチまたは再アタッチ \(attachVolume、reattachVolume\) \(p. 1068\)](#)

## ボリュームを作成 (createVolume)

createVolume イベントは、ボリュームを作成するアクションが完了したときに、AWS アカウントに送信されます。ただし、保存、ログ作成、アーカイブはされません。このイベントの結果は、available または failed のいずれかです。次の例に示すように無効な KMS キーが指定された場合、作成は失敗します。

### イベントデータ

以下に示すのは、createVolume イベントが正常に完了したときに EBS から出力される JSON オブジェクトの例です。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"  
    ],  
    "detail": {  
        "result": "available",  
        "cause": "",  
        "event": "createVolume",  
        "request-id": "01234567-0123-0123-0123-0123456789ab"  
    }  
}
```

以下に示すのは、createVolume が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因是無効な KMS キーです。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "sa-east-1",  
    "resources": [  
        "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567"  
    ],  
    "detail": {  
        "event": "createVolume",  
        "result": "failed",  
        "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab  
is disabled.",  
        "request-id": "01234567-0123-0123-0123-0123456789ab",  
    }  
}
```

以下に示すのは、createVolume イベントが失敗した後で EBS から出力される JSON オブジェクトの例です。失敗の原因是、KMS キーの保留中のインポートです。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-0123456789ab",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
}
```

```
"account": "012345678901",
"time": "yyyy-mm-ddThh:mm:ssZ",
"region": "sa-east-1",
"resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
],
"detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
is pending import.",
    "request-id": "01234567-0123-0123-0123-0123456789ab",
}
}
```

## ボリュームを削除 (deleteVolume)

deleteVolume イベントは、ボリュームを削除するアクションが完了したときに、AWS アカウントに送信されます。ただし、保存、ログ作成、アーカイブはされません。このイベントの結果は deleted です。削除が完了しない場合、イベントは送信されません。

### イベントデータ

以下に示すのは、deleteVolume イベントが正常に完了したときに EBS から出力される JSON オブジェクトの例です。

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-012345678901",
    "detail-type": "EBS Volume Notification",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"
    ],
    "detail": {
        "result": "deleted",
        "cause": "",
        "event": "deleteVolume",
        "request-id": "01234567-0123-0123-0123-0123456789ab"
    }
}
```

## ボリュームのアタッチまたは再アタッチ (attachVolume、reattachVolume)

インスタンスにボリュームをアタッチまたは再アタッチできない場合、attachVolume または reattachVolume イベントが AWS アカウントに送信されます。ただし、保存、ログ作成、アーカイブはされません。次の例に示すように、KMS キーを使用して EBS ボリュームを暗号化し、キーが無効になつた場合、インスタンスへのアタッチまたは再アタッチにそのキーが後で使用されると、EBS はイベントを出力します。

### イベントデータ

以下に示すのは、attachVolume が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因是、KMS キーの保留中の削除です。

#### Note

AWS は、サーバーの定期メンテナンスの後でボリュームに再アタッチを試みる場合があります。

```
{
```

```
"version": "0",
"id": "01234567-0123-0123-0123-0123456789ab",
"detail-type": "EBS Volume Notification",
"source": "aws.ec2",
"account": "012345678901",
"time": "yyyy-mm-ddThh:mm:ssZ",
"region": "us-east-1",
"resources": [
    "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
    "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
],
"detail": {
    "event": "attachVolume",
    "result": "failed",
    "cause": "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
is pending deletion.",
    "request-id": ""
}
}
```

以下に示すのは、reattachVolume が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因是、KMS キーの保留中の削除です。

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "EBS Volume Notification",
    "source": "aws.ec2",
    "account": "012345678901",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
        "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
    ],
    "detail": {
        "event": "reattachVolume",
        "result": "failed",
        "cause": "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab
is pending deletion.",
        "request-id": ""
    }
}
```

## EBS スナップショットイベント

Amazon EBS は、次のボリュームイベントが発生したときに、CloudWatch イベントにイベントを送信します。

### イベント

- スナップショットを作成 (createSnapshot) (p. 1069)
- スナップショットを作成 (createSnapshots) (p. 1070)
- スナップショット のコピー (copySnapshot) (p. 1071)
- スナップショットの共有 (shareSnapshot) (p. 1072)

### スナップショットを作成 (createSnapshot)

createSnapshot イベントは、スナップショットを作成するアクションが完了したときに、AWS アカウントに送信されます。ただし、保存、ログ作成、アーカイブはされません。このイベントの結果は、succeeded または failed のいずれかです。

## イベントデータ

以下に示すのは、`createSnapshot` イベントが正常に完了したときに EBS から出力される JSON オブジェクトの例です。`detail` セクションで、`source` フィールドにはソースボリュームの ARN が入ります。`StartTime` フィールドと `EndTime` フィールドは、スナップショット作成の開始時間と終了時間を示します。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-west-2::snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "createSnapshot",  
        "result": "succeeded",  
        "cause": "",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",  
        "source": "arn:aws:ec2:us-west-2::volume/vol-01234567",  
        "StartTime": "yyyy-mm-ddThh:mm:ssZ",  
        "EndTime": "yyyy-mm-ddThh:mm:ssZ"    }  
}
```

## スナップショットを作成 (createSnapshots)

`createSnapshots` イベントは、マルチボリュームスナップショットを作成するアクションが完了したときに、AWS アカウントに送信されます。このイベントの結果は、`succeeded` または `failed` のいずれかです。

## イベントデータ

以下に示すのは、`createSnapshots` イベントが正常に完了したときに EBS から出力される JSON オブジェクトの例です。`detail` セクションで、`source` フィールドには、マルチボリュームスナップショットセットのソースボリュームの ARN が入ります。`StartTime` フィールドと `EndTime` フィールドは、スナップショット作成の開始時間と終了時間を示します。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Multi-Volume Snapshots Completion Status",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:snapshot/snap-01234567",  
        "arn:aws:ec2:us-east-1:snapshot/snap-012345678"  
    ],  
    "detail": {  
        "event": "createSnapshots",  
        "result": "succeeded",  
        "cause": "",  
        "request-id": "",  
        "startTime": "yyyy-mm-ddThh:mm:ssZ",  
        "endTime": "yyyy-mm-ddThh:mm:ssZ",  
        "snapshots": [  
            "snap-01234567",  
            "snap-012345678"  
        ]  
    }  
}
```

```
{  
    "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",  
    "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",  
    "status": "completed"  
},  
{  
    "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",  
    "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",  
    "status": "completed"  
}  
]  
}  
}
```

以下に示すのは、`createSnapshots` が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因は、1つ以上のスナップショットが完了できなかつたことです。`snapshot_id` の値は、失敗したスナップショットの ARN です。`StartTime` と `EndTime` は、スナップショットを作成するアクションの開始時間と終了時間を表します。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Multi-Volume Snapshots Completion Status",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2::us-east-1:snapshot/snap-01234567",  
        "arn:aws:ec2::us-east-1:snapshot/snap-012345678"  
    ],  
    "detail": {  
        "event": "createSnapshots",  
        "result": "failed",  
        "cause": "Snapshot snap-01234567 is in status deleted",  
        "request-id": "",  
        "startTime": "yyyy-mm-ddThh:mm:ssZ",  
        "endTime": "yyyy-mm-ddThh:mm:ssZ",  
        "snapshots": [  
            {  
                "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",  
                "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",  
                "status": "error"  
            },  
            {  
                "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",  
                "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",  
                "status": "deleted"  
            }  
        ]  
    }  
}
```

## スナップショット のコピー (copySnapshot)

`copySnapshot` イベントは、スナップショットをコピーするアクションが完了したときに、AWS アカウントに送信されます。ただし、保存、ログ作成、アーカイブはされません。このイベントの結果は、`succeeded` または `failed` のいずれかです。

### イベントデータ

以下に示すのは、`copySnapshot` イベントが成功したときに EBS から出力される JSON オブジェクトの例です。`snapshot_id` の値は、新しく作成されたスナップショットの ARN です。`detail` セクション

ンで、source の値は、ソーススナップショットの ARN です。StartTime とEndTime は、スナップショットをコピーするアクションの開始時間と終了時間を示します。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "123456789012",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-west-2::snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "copySnapshot",  
        "result": "succeeded",  
        "cause": "",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",  
        "source": "arn:aws:ec2:eu-west-1::snapshot/snap-76543210",  
        "StartTime": "yyyy-mm-ddThh:mm:ssZ",  
        "EndTime": "yyyy-mm-ddThh:mm:ssZ",  
        "Incremental": "True"  
    }  
}
```

以下に示すのは、copySnapshot が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因是無効なソーススナップショット ID です。snapshot\_id の値は、失敗したスナップショットの ARN です。detail セクションで、source の値は、ソーススナップショットの ARN です。StartTime とEndTime は、スナップショットをコピーするアクションの開始時間と終了時間を示します。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "123456789012",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-west-2::snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "copySnapshot",  
        "result": "failed",  
        "cause": "Source snapshot ID is not valid",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",  
        "source": "arn:aws:ec2:eu-west-1::snapshot/snap-76543210",  
        "StartTime": "yyyy-mm-ddThh:mm:ssZ",  
        "EndTime": "yyyy-mm-ddThh:mm:ssZ"  
    }  
}
```

## スナップショットの共有 (shareSnapshot)

shareSnapshot イベントは、別のアカウントがスナップショットを共有したときに AWS アカウントに送信されます。ただし、保存、ログ作成、アーカイブはされません。結果は常に succeeded です。

### イベントデータ

shareSnapshot イベントが完了したときに EBS から出力される JSON オブジェクトの例を以下に示します。detail セクションの source の値は、スナップショットを共有したユーザーの AWS アカウント番号です。StartTime と EndTime は、スナップショットを共有するアクションの開始時間と終了時間を示します。shareSnapshot イベントは、プライベートスナップショットが別のユーザーと共有された場合にのみ出力されます。パブリックスナップショットを共有しても、イベントはトリガーされません。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Snapshot Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-west-2::snapshot/snap-01234567"  
    ],  
    "detail": {  
        "event": "shareSnapshot",  
        "result": "succeeded",  
        "cause": "",  
        "request-id": "",  
        "snapshot_id": "arn:aws:ec2:us-west-2::snapshot/snap-01234567",  
        "source": "012345678901",  
        "StartTime": "yyyy-mm-ddThh:mm:ssZ",  
        "EndTime": "yyyy-mm-ddThh:mm:ssZ"  
    }  
}
```

## EBS ボリュームの変更イベント

Amazon EBS は、ボリュームが変更されると、CloudWatch イベントに modifyVolume イベントを送信します。ただし、保存、ログ作成、アーカイブはされません。

```
{  
    "version": "0",  
    "id": "01234567-0123-0123-0123-012345678901",  
    "detail-type": "EBS Volume Notification",  
    "source": "aws.ec2",  
    "account": "012345678901",  
    "time": "yyyy-mm-ddThh:mm:ssZ",  
    "region": "us-east-1",  
    "resources": [  
        "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"  
    ],  
    "detail": {  
        "result": "optimizing",  
        "cause": "",  
        "event": "modifyVolume",  
        "request-id": "01234567-0123-0123-0123-0123456789ab"  
    }  
}
```

## EBS 高速スナップショット復元イベント

スナップショットの高速スナップショット復元の状態が変わると、Amazon EBS はイベントを CloudWatch イベントに送信します。

以下はこのイベントのサンプルデータです。

```
{
```

```
"version": "0",
"id": "01234567-0123-0123-0123-012345678901",
"detail-type": "EBS Fast Snapshot Restore State-change Notification",
"source": "aws.ec2",
"account": "123456789012",
"time": "yyyy-mm-ddThh:mm:ssZ",
"region": "us-east-1",
"resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-03a55cf56513fa1b6"
],
"detail": {
    "snapshot-id": "snap-1234567890abcdef0",
    "state": "optimizing",
    "zone": "us-east-1a",
    "message": "Client.UserInitiated - Lifecycle state transition",
}
}
```

state の想定される値は、enabling、optimizing、enabled、disabling、および disabled です。

message の想定される値は次のとおりです。

**Client.InvalidSnapshot.InvalidState** – The requested snapshot transitioned to an invalid state (Error)

高速スナップショット復元を有効にするリクエストが失敗し、状態は disabling または disabled に移行しました。このスナップショットに対しては、高速スナップショット復元を有効にすることができません。

**Client.UserInitiated**

状態は、正常に enabling または disabling に移行しました。

**Client.UserInitiated - Lifecycle state transition**

状態は、正常に optimizing、enabled、または disabled に移行しました。

**Server.InsufficientCapacity** – There was insufficient capacity available to satisfy the request

高速スナップショット復元を有効にするリクエストが容量不足のために失敗し、状態は disabling または disabled に移行しました。しばらく待ってから、もう一度試してください。

**Server.InternalError** – An internal error caused the operation to fail

高速スナップショット復元を有効にするリクエストが内部エラーのために失敗し、状態は disabling または disabled に移行しました。しばらく待ってから、もう一度試してください。

## AWS Lambda による CloudWatch イベントの処理

Amazon EBS と CloudWatch イベントを使用して、データのバックアップのワークフローを自動化できます。そのためには、IAM ポリシー、イベントを処理する AWS Lambda 関数、および受信イベントを照合して Lambda 関数にルーティングする Amazon CloudWatch Events ルールを作成する必要があります。

次の手順では、`createSnapshot` イベントを使用して完成したスナップショットを災害対策の目的で自動的に別のリージョンにコピーします。

完了したスナップショットを別のリージョンにコピーするには

1. 次の例に示すような IAM ポリシーを作成し、`CopySnapshot` アクションを実行して CloudWatch イベント ログに書き込むためのアクセス許可を提供します。このポリシーを CloudWatch イベントを処理する IAM ユーザーに割り当てます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs>CreateLogGroup",  
                "logs>CreateLogStream",  
                "logs:PutLogEvents"  
            ],  
            "Resource": "arn:aws:logs:*:*:  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CopySnapshot"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

2. CloudWatch コンソールから利用できる関数を Lambda で定義します。以下の Lambda 関数は、Node.js で記述したサンプルであり、該当する `createSnapshot` イベント(スナップショットの完成を示す)が Amazon EBS から出力されたときに CloudWatch から呼び出されます。この関数は、呼び出されると、スナップショットを `us-east-2` から `us-east-1` にコピーします。

```
// Sample Lambda function to copy an EBS snapshot to a different region  
  
var AWS = require('aws-sdk');  
var ec2 = new AWS.EC2();  
  
// define variables  
var destinationRegion = 'us-east-1';  
var sourceRegion = 'us-east-2';  
console.log ('Loading function');  
  
//main function  
exports.handler = (event, context, callback) => {  
  
    // Get the EBS snapshot ID from the CloudWatch event details  
    var snapshotArn = event.detail.snapshot_id.split('/');  
    const snapshotId = snapshotArn[1];  
    const description = `Snapshot copy from ${snapshotId} in ${sourceRegion}.`;  
    console.log ("snapshotId:", snapshotId);  
  
    // Load EC2 class and update the configuration to use destination region to  
    // initiate the snapshot.  
    AWS.config.update({region: destinationRegion});  
    var ec2 = new AWS.EC2();  
  
    // Prepare variables for ec2.modifySnapshotAttribute call  
    const copySnapshotParams = {  
        Description: description,  
        DestinationRegion: destinationRegion,  
        SourceRegion: sourceRegion,  
        SourceSnapshotId: snapshotId  
    };  
  
    // Execute the copy snapshot and log any errors  
    ec2.copySnapshot(copySnapshotParams, (err, data) => {  
        if (err) {  
            console.error(`Error copying snapshot: ${err.message}`);  
            callback(err);  
        } else {  
            console.log(`Snapshot copied successfully to ${destinationRegion}`);  
            callback(null, data);  
        }  
    });  
};
```

```
        const errorMessage = `Error copying snapshot ${snapshotId} to region
${destinationRegion}.`;
        console.log(errorMessage);
        console.log(err);
        callback(errorMessage);
    } else {
        const successMessage = `Successfully started copy of snapshot ${snapshotId}
to region ${destinationRegion}.`;
        console.log(successMessage);
        console.log(data);
        callback(null, successMessage);
    }
});
```

Lambda 関数は、CloudWatch コンソールから確実に利用できるように、CloudWatch イベントが発生するリージョンで作成します。詳細については、「[AWS Lambda 開発者ガイド](#)」を参照してください。

3. <https://console.aws.amazon.com/cloudwatch/> にある CloudWatch コンソールを開きます。
4. [Events]、[Create rule]、[Select event source]、[Amazon EBS Snapshots] の順に選択します。
5. [Specific Event(s)] で [createSnapshot] を選択し、[Specific Result(s)] で [succeeded] を選択します。
6. [Rule target] で、前に作成したサンプル関数を見つけて選択します。
7. [Target]、[Add Target] の順に選択します。
8. [Lambda function] で、前に作成した Lambda 関数を選択し、[Configure details] を選択します。
9. [Configure rule details] ページで、[Name] と [Description] の値を入力します。[State] チェックボックスをオンにして、関数をアクティブにします ([Enabled] に設定します)。
10. [Create rule] を選択します。

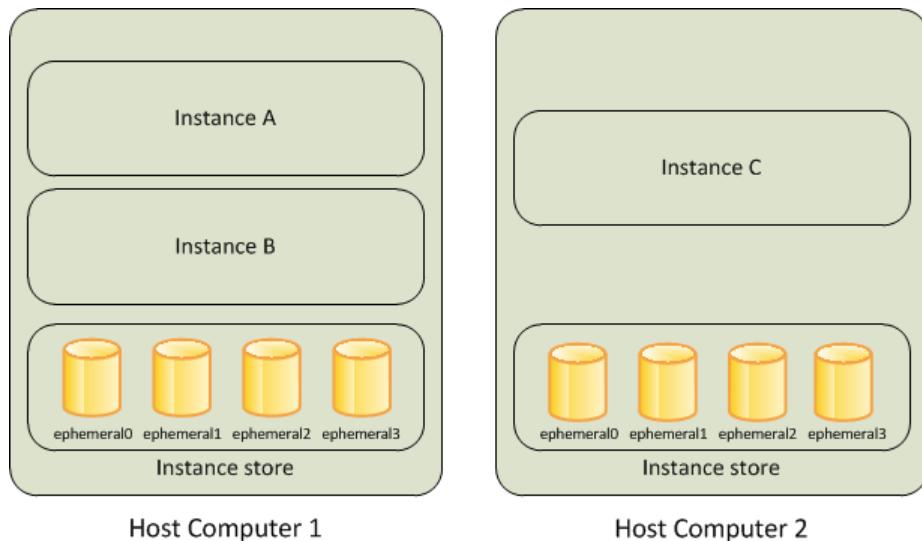
作成したルールが、[Rules] タブに表示されます。上の例で、設定したイベントは次回にスナップショットをコピーすると EBS から出力されます。

## Amazon EC2 インスタンスストア

インスタンスストアは、インスタンス用のブロックレベルの一時ストレージを提供します。このストレージは、ホストコンピュータに物理的にアタッチされたディスク上にあります。インスタンスストアは、頻繁に変更される情報 (パッファ、キャッシュ、スクラッチデータ、その他の一時コンテンツなど) の一時ストレージに最適です。また、インスタンスのフリート全体でレプリケートされるデータ (負荷分散されたウェブサーバーパークなど) にも適しています。

インスタンスストアは、ブロックデバイスとして表示される 1 つ以上のインスタンスストアボリュームで構成されます。インスタンスストアのサイズと、利用可能なデバイスの数は、インスタンスタイプによって異なります。

インスタンスストアボリュームの仮想デバイスは `ephemeral[0-23]` です。1 つのインスタンスストアボリュームをサポートするインスタンスタイプには、`ephemeral0` があります。2 つのインスタンスストアボリュームをサポートするインスタンスタイプは、`ephemeral0`、`ephemeral1` などを持ちます。



## コンテンツ

- [インスタンスストアの存続期間 \(p. 1077\)](#)
- [インスタンスストアボリューム \(p. 1078\)](#)
- [EC2 インスタンスにインスタンスストアボリュームを追加する \(p. 1083\)](#)
- [SSD インスタンスストアボリューム \(p. 1086\)](#)
- [インスタンスストアスワップボリューム \(p. 1087\)](#)
- [インスタンスストアボリュームのディスクパフォーマンスの最適化 \(p. 1090\)](#)

## インスタンスストアの存続期間

インスタンスを起動する場合にのみ、インスタンス用にインスタンスストアボリュームを指定できます。1つのインスタンスからインスタンスストアをデタッチして別のインスタンスにアタッチすることはできません。

インスタンスストア上のデータは、関連付けられたインスタンスの運用中のみ維持されます。インスタンスが再ブートされた場合、その再ブートが意図的なものでも、意図せずに行われたとしても、インスタンスストアのデータは維持されます。ただし、次のいずれの状況でも、インスタンスストアのデータは失われます。

- 基盤となるディスクドライブで障害が発生した
- インスタンスが停止した
- インスタンスが終了した

したがって、長期的に使用する重要なデータがある場合は、インスタンスストアに頼りすぎないようにしてください。代わりに、Amazon S3、Amazon EBS、または Amazon EFS などのより堅牢なデータストレージを使用してください。

インスタンスを停止または終了するとき、インスタンスストアのストレージの各ブロックはリセットされます。そのため、別のインスタンスのインスタンスストアを通じてデータにアクセスすることはできません。

インスタンスから AMI を作成すると、そのインスタンスストアボリューム上のデータは保持されず、AMI から起動するインスタンスのインスタンスストアボリュームに存在しません。

インスタンスタイプを変更すると、インスタンスストアは新しいインスタンスタイプにアタッチされません。詳細については、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。

## インスタンスストアボリューム

インスタンスタイプにより、使用できるインスタンスストアのサイズ、およびインスタンスストアボリュームで使用されるハードウェアの種類が決まります。インスタンスストアボリュームは、インスタンスの使用料に含まれます。インスタンス（デフォルトで利用可能な NVMe インスタンスストアボリュームを除く）を起動するときに使用するインスタンスストアボリュームを指定する必要があります。指定したインスタンスストアボリュームは、使用する前にフォーマットおよびマウントしておきます。インスタンスの起動後に、インスタンスストアボリュームを使用できるようにすることはできません。詳細については、「[EC2 インスタンスにインスタンスストアボリュームを追加する \(p. 1083\)](#)」を参照してください。

インスタンスタイプの中には、NVMe または SATA ベースのシリッドステートドライブ (SSD) を使用して、高いランダム I/O パフォーマンスを実現するものがあります。レイテンシーが非常に短いストレージが必要だが、インスタンスの削除時にデータを保持する必要がない場合、または耐障害性を備えたアーキテクチャーを使用できる場合は、このオプションを使用することをお勧めします。詳細については、「[SSD インスタンスストアボリューム \(p. 1086\)](#)」を参照してください。

次の表は、サポートされている各インスタンスタイプで使用できるインスタンスストアボリュームの数量、サイズ、タイプ、パフォーマンス最適化を示しています。EBS 専用タイプを含めたインスタンスタイプの一覧については、「[Amazon EC2 インスタンスタイプ](#)」を参照してください。

| インスタンスタイプ    | インスタンスストアボリューム       | タイプ      | 初期化が必要* | TRIM サポート** |
|--------------|----------------------|----------|---------|-------------|
| c1.medium    | 1 x 350 GB†          | HDD      | ✓       |             |
| c1.xlarge    | 4 x 420 GB (1.6 TB)  | HDD      | ✓       |             |
| c3.large     | 2 x 16 GB (32 GB)    | SSD      | ✓       |             |
| c3.xlarge    | 2 x 40 GB (80 GB)    | SSD      | ✓       |             |
| c3.2xlarge   | 2 x 80 GB (160 GB)   | SSD      | ✓       |             |
| c3.4xlarge   | 2 x 160 GB (320 GB)  | SSD      | ✓       |             |
| c3.8xlarge   | 2 x 320 GB (640 GB)  | SSD      | ✓       |             |
| c5d.large    | 1 x 50 GB            | NVMe SSD |         | ✓           |
| c5d.xlarge   | 1 x 100 GB           | NVMe SSD |         | ✓           |
| c5d.2xlarge  | 1 x 200 GB           | NVMe SSD |         | ✓           |
| c5d.4xlarge  | 1 x 400 GB           | NVMe SSD |         | ✓           |
| c5d.9xlarge  | 1 x 900 GB           | NVMe SSD |         | ✓           |
| c5d.12xlarge | 2 x 900 GB (1.8 TB)  | NVMe SSD |         | ✓           |
| c5d.18xlarge | 2 x 900 GB (1.8 TB)  | NVMe SSD |         | ✓           |
| c5d.24xlarge | 4 x 900 GB (3.6 TB)  | NVMe SSD |         | ✓           |
| c5d.metal    | 4 x 900 GB (3.6 TB)  | NVMe SSD |         | ✓           |
| cc2.8xlarge  | 4 x 840 GB (3.36 TB) | HDD      | ✓       |             |
| cr1.8xlarge  | 2 x 120 GB (240 GB)  | SSD      | ✓       |             |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
インスタンスストアボリューム

| インスタンスタイプ     | インスタンスストアボリューム         | タイプ      | 初期化が必要* | TRIM サポート** |
|---------------|------------------------|----------|---------|-------------|
| d2.xlarge     | 3 x 2,000 GB (6 TB)    | HDD      |         |             |
| d2.2xlarge    | 6 x 2,000 GB (12 TB)   | HDD      |         |             |
| d2.4xlarge    | 12 x 2,000 GB (24 TB)  | HDD      |         |             |
| d2.8xlarge    | 24 x 2,000 GB (48 TB)  | HDD      |         |             |
| f1.2xlarge    | 1 x 470 GB             | NVMe SSD |         | ✓           |
| f1.4xlarge    | 1 x 940 GB             | NVMe SSD |         | ✓           |
| f1.16xlarge   | 4 x 940 GB (3.76 TB)   | NVMe SSD |         | ✓           |
| g2.2xlarge    | 1 x 60 GB              | SSD      | ✓       |             |
| g2.8xlarge    | 2 x 120 GB (240 GB)    | SSD      | ✓       |             |
| g4dn.xlarge   | 1 x 125 GB             | NVMe SSD |         | ✓           |
| g4dn.2xlarge  | 1 x 225 GB             | NVMe SSD |         | ✓           |
| g4dn.4xlarge  | 1 x 225 GB             | NVMe SSD |         | ✓           |
| g4dn.8xlarge  | 1 x 900 GB             | NVMe SSD |         | ✓           |
| g4dn.12xlarge | 1 x 900 GB             | NVMe SSD |         | ✓           |
| g4dn.16xlarge | 1 x 900 GB             | NVMe SSD |         | ✓           |
| h1.2xlarge    | 1 x 2000 GB (2 TB)     | HDD      |         |             |
| h1.4xlarge    | 2 x 2000 GB (4 TB)     | HDD      |         |             |
| h1.8xlarge    | 4 x 2000 GB (8 TB)     | HDD      |         |             |
| h1.16xlarge   | 8 x 2000 GB (16 TB)    | HDD      |         |             |
| hs1.8xlarge   | 24 x 2,000 GB (48 TB)  | HDD      | ✓       |             |
| i2.xlarge     | 1 x 800 GB             | SSD      |         | ✓           |
| i2.2xlarge    | 2 x 800 GB (1.6 TB)    | SSD      |         | ✓           |
| i2.4xlarge    | 4 x 800 GB (3.2 TB)    | SSD      |         | ✓           |
| i2.8xlarge    | 8 x 800 GB (6.4 TB)    | SSD      |         | ✓           |
| i3.large      | 1 x 475 GB             | NVMe SSD |         | ✓           |
| i3.xlarge     | 1 x 950 GB             | NVMe SSD |         | ✓           |
| i3.2xlarge    | 1 x 1,900 GB           | NVMe SSD |         | ✓           |
| i3.4xlarge    | 2 x 1,900 GB (3.8 TB)  | NVMe SSD |         | ✓           |
| i3.8xlarge    | 4 x 1,900 GB (7.6 TB)  | NVMe SSD |         | ✓           |
| i3.16xlarge   | 8 x 1,900 GB (15.2 TB) | NVMe SSD |         | ✓           |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
インスタンスストアボリューム

| インスタンスタイプ     | インスタンスストアボリューム         | タイプ      | 初期化が必要* | TRIM サポート** |
|---------------|------------------------|----------|---------|-------------|
| i3.metal      | 8 x 1,900 GB (15.2 TB) | NVMe SSD |         | ✓           |
| i3en.large    | 1 x 1,250 GB           | NVMe SSD |         | ✓           |
| i3en.xlarge   | 1 x 2,500 GB           | NVMe SSD |         | ✓           |
| i3en.2xlarge  | 2 x 2,500 GB (5 TB)    | NVMe SSD |         | ✓           |
| i3en.3xlarge  | 1 x 7,500 GB           | NVMe SSD |         | ✓           |
| i3en.6xlarge  | 2 x 7,500 GB (15 TB)   | NVMe SSD |         | ✓           |
| i3en.12xlarge | 4 x 7,500 GB (30 TB)   | NVMe SSD |         | ✓           |
| i3en.24xlarge | 8 x 7,500 GB (60 TB)   | NVMe SSD |         | ✓           |
| i3en.metal    | 8 x 7,500 GB (60 TB)   | NVMe SSD |         | ✓           |
| m1.small      | 1 x 160 GB†            | HDD      | ✓       |             |
| m1.medium     | 1 x 410 GB             | HDD      | ✓       |             |
| m1.large      | 2 x 420 GB (840 GB)    | HDD      | ✓       |             |
| m1.xlarge     | 4 x 420 GB (1.6 TB)    | HDD      | ✓       |             |
| m2.xlarge     | 1 x 420 GB             | HDD      | ✓       |             |
| m2.2xlarge    | 1 x 850 GB             | HDD      | ✓       |             |
| m2.4xlarge    | 2 x 840 GB (1.68 TB)   | HDD      | ✓       |             |
| m3.medium     | 1 x 4 GB               | SSD      | ✓       |             |
| m3.large      | 1 x 32 GB              | SSD      | ✓       |             |
| m3.xlarge     | 2 x 40 GB (80 GB)      | SSD      | ✓       |             |
| m3.2xlarge    | 2 x 80 GB (160 GB)     | SSD      | ✓       |             |
| m5ad.large    | 1 x 75 GB              | NVMe SSD |         | ✓           |
| m5ad.xlarge   | 1 x 150 GB             | NVMe SSD |         | ✓           |
| m5ad.2xlarge  | 1 x 300 GB             | NVMe SSD |         | ✓           |
| m5ad.4xlarge  | 2 x 300 GB (600 GB)    | NVMe SSD |         | ✓           |
| m5ad.12xlarge | 2 x 900 GB (1.8 TB)    | NVMe SSD |         | ✓           |
| m5ad.24xlarge | 4 x 900 GB (3.6 TB)    | NVMe SSD |         | ✓           |
| m5d.large     | 1 x 75 GB              | NVMe SSD |         | ✓           |
| m5d.xlarge    | 1 x 150 GB             | NVMe SSD |         | ✓           |
| m5d.2xlarge   | 1 x 300 GB             | NVMe SSD |         | ✓           |
| m5d.4xlarge   | 2 x 300 GB (600 GB)    | NVMe SSD |         | ✓           |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
インスタンスストアボリューム

| インスタンスタイプ     | インスタンスストアボリューム      | タイプ*     | 初期化が必要* | TRIM サポート** |
|---------------|---------------------|----------|---------|-------------|
| m5d.8xlarge   | 2 x 600 GB (1.2 TB) | NVMe SSD |         | ✓           |
| m5d.12xlarge  | 2 x 900 GB (1.8 TB) | NVMe SSD |         | ✓           |
| m5d.16xlarge  | 4 x 600 GB (2.4 TB) | NVMe SSD |         | ✓           |
| m5d.24xlarge  | 4 x 900 GB (3.6 TB) | NVMe SSD |         | ✓           |
| m5d.metal     | 4 x 900 GB (3.6 TB) | NVMe SSD |         | ✓           |
| m5dn.large    | 1 x 75 GB           | NVMe SSD |         | ✓           |
| m5dn.xlarge   | 1 x 150 GB          | NVMe SSD |         | ✓           |
| m5dn.2xlarge  | 1 x 300 GB          | NVMe SSD |         | ✓           |
| m5dn.4xlarge  | 2 x 300 GB (600 GB) | NVMe SSD |         | ✓           |
| m5dn.8xlarge  | 2 x 600 GB (1.2 TB) | NVMe SSD |         | ✓           |
| m5dn.12xlarge | 2 x 900 GB (1.8 TB) | NVMe SSD |         | ✓           |
| m5dn.16xlarge | 4 x 600 GB (2.4 TB) | NVMe SSD |         | ✓           |
| m5dn.24xlarge | 4 x 900 GB (3.6 TB) | NVMe SSD |         | ✓           |
| p3dn.24xlarge | 2 x 900 GB (1.8 TB) | NVMe SSD |         | ✓           |
| r3.large      | 1 x 32 GB           | SSD      |         | ✓           |
| r3.xlarge     | 1 x 80 GB           | SSD      |         | ✓           |
| r3.2xlarge    | 1 x 160 GB          | SSD      |         | ✓           |
| r3.4xlarge    | 1 x 320 GB          | SSD      |         | ✓           |
| r3.8xlarge    | 2 x 320 GB (640 GB) | SSD      |         | ✓           |
| r5ad.large    | 1 x 75 GB           | NVMe SSD |         | ✓           |
| r5ad.xlarge   | 1 x 150 GB          | NVMe SSD |         | ✓           |
| r5ad.2xlarge  | 1 x 300 GB          | NVMe SSD |         | ✓           |
| r5ad.4xlarge  | 2 x 300 GB (600 GB) | NVMe SSD |         | ✓           |
| r5ad.12xlarge | 2 x 900 GB (1.8 TB) | NVMe SSD |         | ✓           |
| r5ad.24xlarge | 4 x 900 GB (3.6 TB) | NVMe SSD |         | ✓           |
| r5d.large     | 1 x 75 GB           | NVMe SSD |         | ✓           |
| r5d.xlarge    | 1 x 150 GB          | NVMe SSD |         | ✓           |
| r5d.2xlarge   | 1 x 300 GB          | NVMe SSD |         | ✓           |
| r5d.4xlarge   | 2 x 300 GB (600 GB) | NVMe SSD |         | ✓           |
| r5d.8xlarge   | 2 x 600 GB (1.2 TB) | NVMe SSD |         | ✓           |

| インスタンスタイプ     | インスタンスストアボリューム         | タイプ      | 初期化が必要* | TRIM サポート** |
|---------------|------------------------|----------|---------|-------------|
| r5d.12xlarge  | 2 x 900 GB (1.8 TB)    | NVMe SSD |         | ✓           |
| r5d.16xlarge  | 4 x 600 GB (2.4 TB)    | NVMe SSD |         | ✓           |
| r5d.24xlarge  | 4 x 900 GB (3.6 TB)    | NVMe SSD |         | ✓           |
| r5d.metal     | 4 x 900 GB (3.6 TB)    | NVMe SSD |         | ✓           |
| r5dn.large    | 1 x 75 GB              | NVMe SSD |         | ✓           |
| r5dn.xlarge   | 1 x 150 GB             | NVMe SSD |         | ✓           |
| r5dn.2xlarge  | 1 x 300 GB             | NVMe SSD |         | ✓           |
| r5dn.4xlarge  | 2 x 300 GB (600 GB)    | NVMe SSD |         | ✓           |
| r5dn.8xlarge  | 2 x 600 GB (1.2 TB)    | NVMe SSD |         | ✓           |
| r5dn.12xlarge | 2 x 900 GB (1.8 TB)    | NVMe SSD |         | ✓           |
| r5dn.16xlarge | 4 x 600 GB (2.4 TB)    | NVMe SSD |         | ✓           |
| r5dn.24xlarge | 4 x 900 GB (3.6 TB)    | NVMe SSD |         | ✓           |
| x1.16xlarge   | 1 x 1,920 GB           | SSD      |         |             |
| x1.32xlarge   | 2 x 1,920 GB (3.84 TB) | SSD      |         |             |
| x1e.xlarge    | 1 x 120 GB             | SSD      |         |             |
| x1e.2xlarge   | 1 x 240 GB             | SSD      |         |             |
| x1e.4xlarge   | 1 x 480 GB             | SSD      |         |             |
| x1e.8xlarge   | 1 x 960 GB             | SSD      |         |             |
| x1e.16xlarge  | 1 x 1,920 GB           | SSD      |         |             |
| x1e.32xlarge  | 2 x 1,920 GB (3.84 TB) | SSD      |         |             |
| z1d.large     | 1 x 75 GB              | NVMe SSD |         | ✓           |
| z1d.xlarge    | 1 x 150 GB             | NVMe SSD |         | ✓           |
| z1d.2xlarge   | 1 x 300 GB             | NVMe SSD |         | ✓           |
| z1d.3xlarge   | 1 x 450 GB             | NVMe SSD |         | ✓           |
| z1d.6xlarge   | 1 x 900 GB             | NVMe SSD |         | ✓           |
| z1d.12xlarge  | 2 x 900 GB (1.8 TB)    | NVMe SSD |         | ✓           |
| z1d.metal     | 2 x 900 GB (1.8 TB)    | NVMe SSD |         | ✓           |

\* 特定のインスタンスにアタッチされたボリュームは、初期化されないと初回書き込み時のパフォーマンスが低下します。詳細については、「[インスタンスストアボリュームのディスクパフォーマンスの最適化 \(p. 1090\)](#)」を参照してください。

\*\* 詳細については、「[インスタンスストアボリュームの TRIM のサポート \(p. 1087\)](#)」を参照してください。

<sup>†</sup>c1.medium および m1.small インスタンスタイプには、900 MB のインスタンスストアスワップボリュームも含まれます。これは起動時に自動的に有効にされない場合があります。詳細については、「[インスタンスストアスワップボリューム \(p. 1087\)](#)」を参照してください。

## EC2 インスタンスにインスタンスストアボリュームを追加する

ブロックデバイスマッピングを使用して、インスタンスの EBS ボリュームとインスタンスストアボリュームを指定します。ブロックデバイスマッピングの各エントリには、デバイス名とそれがマッピングされたボリュームが含まれます。デフォルトのブロックデバイスマッピングは、使用する AMI によって指定されます。または、起動するときにインスタンスのブロックデバイスマッピングを指定できます。

インスタンスタイプによってサポートされるすべての NVMe インスタンスストアボリュームが自動的に列挙され、インスタンスの起動時にデバイス名が割り当てられます。それらを AMI のブロックデバイスマッピングに含めます。含めないとインスタンスは効果がありません。詳細については、「[ブロックデバイスマッピング \(p. 1100\)](#)」を参照してください。

ブロックデバイスマッピングでは、常にインスタンスのルートボリュームを指定します。ルートボリュームは、Amazon EBS ボリュームまたはインスタンスストアボリュームのいずれかです。詳細については、「[ルートデバイスのストレージ \(p. 96\)](#)」を参照してください。ルートボリュームは自動的にマウントされます。ルートボリュームのインスタンスストアボリュームを持つインスタンスの場合、このボリュームのサイズは AMI によって異なりますが、最大サイズは 10 GB です。

インスタンスを実行するときに、ブロックデバイスマッピングを使用して追加の EBS ボリュームを指定するか、インスタンスの実行後に追加の EBS ボリュームをアタッチすることができます。詳細については、「[Amazon EBS ボリューム \(p. 931\)](#)」を参照してください。

インスタンスを起動する場合にのみ、インスタンスのインスタンスストアボリュームを指定できます。また、起動後のインスタンスにインスタンスストアボリュームをアタッチすることはできません。

インスタンスタイプを変更すると、インスタンスストアは新しいインスタンスタイプにアタッチされません。詳細については、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。

インスタンスで使用できるインスタンスストアボリュームの数とサイズは、インスタンスタイプによって異なります。インスタンスタイプによっては、インスタンスストアボリュームをサポートしていないものがあります。ブロックデバイスマッピングのインスタンスストアボリュームの数が、インスタンスに利用できるインスタンスストアボリュームの数を超える場合は、追加のボリュームは無視されます。インスタンスタイプごとのインスタンスストアボリュームのサポートの詳細については、「[インスタンスストアボリューム \(p. 1078\)](#)」を参照してください。

インスタンスに選択するインスタンスタイプが非 NVMe のインスタンスストアボリュームをサポートしている場合は、起動するときにインスタンスのブロックデバイスマッピングに追加する必要があります。NVMe インスタンスストアボリュームは、デフォルトで利用できます。インスタンスを起動したら、使用する前に、インスタンスのインスタンスストアボリュームがフォーマットされ、マウントされていることを確認する必要があります。instance store-backed インスタンスのルートボリュームは自動的にマウントされます。

### コンテンツ

- [AMI にインスタンスストアボリュームを追加する \(p. 1084\)](#)
- [インスタンスにインスタンスストアボリュームを追加する \(p. 1084\)](#)
- [インスタンスでインスタンスストアボリュームを使用できるようにする \(p. 1085\)](#)

## AMI にインスタンスストアボリュームを追加する

インスタンスストアボリュームが含まれる、ブロックデバイスマッピングを持つ AMI を作成できます。インスタンスストアボリュームをサポートするインスタンスタイプと、ブロックデバイスマッピングでインスタンスストアボリュームを指定する AMI を持つインスタンスを起動する場合、インスタンスには、これらのインスタンスストアボリュームが含まれます。ブロックデバイスマッピングのインスタンスストアボリュームの数がインスタンスに利用できるインスタンスストアボリュームの数を超える場合、追加のインスタンスストアボリュームは無視されます。

### 考慮事項

- M3 インスタンスの場合は、AMI ではなく、インスタンスのブロックデバイスマッピングでインスタンスストアボリュームを指定します。Amazon EC2 は、AMI のブロックデバイスマッピングでのみ指定されたインスタンスストアボリュームを無視することができます。
- インスタンスを起動する際に、AMI ブロックデバイスマッピングで指定された 非 NVMe インスタンスストアボリュームを省略したり、インスタンスストアボリュームを追加したりできます。

コンソールを使用して Amazon EBS-backed AMI にインスタンスストアボリュームを追加するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. [Actions]、[Image]、[Create Image] の順に選択します。
4. [Create Image] ダイアログボックスで、イメージの意味のある名前と説明を入力します。
5. 追加する各インスタンスストアボリュームについて、[Add New Volume] を選択し、[Volume Type] からインスタンスストアボリュームを選択して、[Device] からデバイス名を選択します。（詳しくは、[Linux インスタンスでのデバイスの名前付け \(p. 1098\)](#) を参照してください）。使用できるインスタンスストアボリュームの数は、インスタンスタイプによって異なります。NVMe インスタンスストアボリュームを持つインスタンスの場合、これらのボリュームのデバイスマッピングは、オペレーティングシステムがこれらのボリュームを列挙する順序によって決まります。
6. [Create Image] を選択します。

コマンドラインを使用して AMI にインスタンスストアボリュームを追加するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `create-image` または `register-image` (AWS CLI)
- `New-EC2Image` および `Register-EC2Image` (AWS Tools for Windows PowerShell)

## インスタンスにインスタンスストアボリュームを追加する

インスタンスを起動するときに、デフォルトのブロックデバイスマッピングが、指定した AMI によって提供されます。追加のインスタンスストアボリュームが必要な場合は、起動時にインスタンスに追加する必要があります。AMI ブロックデバイスマッピングで指定されたデバイスを省略することもできます。

### 考慮事項

- M3 インスタンスの場合は、インスタンスのブロックデバイスマッピングで指定しなくても、インスタンスストアボリュームを受け取る可能性があります。
- HS1 インスタンスの場合は、AMI のブロックデバイスマッピングで指定するインスタンスストアボリュームの数にかかわらず、AMI から起動されるインスタンスのブロックデバイスマッピングには、サポートされたインスタンスストアボリュームの最大数が自動的に含まれます。起動する前に、インスタン

ンスのブロックデバイスマッピングから、必要としないインスタンスストアボリュームを明示的に削除する必要があります。

コンソールを使用して、インスタンスのブロックデバイスマッピングを更新するには

1. Amazon EC2 コンソールを開きます。
2. ダッシュボードから、[Launch Instance] を選択します。
3. [Step 1: Choose an Amazon Machine Image (AMI)] で、使用する AMI を選択し、[Select] を選択します。
4. ウィザードに従って [Step 1: Choose an Amazon Machine Image (AMI)]、[Step 2: Choose an Instance Type]、および [Step 3: Configure Instance Details] を完了します。
5. [Step 4: Add Storage] で、必要に応じて既存のエントリを変更します。追加する各インスタンスストアボリュームについて、[Add New Volume] を選択し、[Volume Type] からインスタンスストアボリュームを選択して、[Device] からデバイス名を選択します。使用できるインスタンスストアボリュームの数は、インスタンスタイプによって異なります。
6. ウィザードを終了してインスタンスを起動します。
7. (オプション) インスタンスで利用可能なインスタンスストアボリュームを表示するには、lsblk コマンドを実行します。

コマンドラインを使用してインスタンスのブロックデバイスマッピングを更新するには

次のいずれかのオプションコマンドを対応するコマンドで使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2へのアクセス \(p. 3\)](#) を参照してください。

- --block-device-mappings と [run-instances](#) (AWS CLI)
- -BlockDeviceMapping と [New-EC2Instance](#) (AWS Tools for Windows PowerShell)

## インスタンスでインスタンスストアボリュームを使用できるようにする

インスタンスの起動後に、インスタンスストアボリュームはインスタンスで使用できますが、ボリュームがマウントされるまでアクセスすることはできません。Linux インスタンスの場合は、インスタンスタイプによって、どのインスタンスストアボリュームがマウントされるのか、またご自分でマウントできるインスタンスが何かが決まります。Windows インスタンスの場合は、EC2Config サービスが、インスタンス用にインスタンスストアボリュームをマウントします。インスタンスのブロックデバイスドライバーは、ボリュームのマウント時に実際のボリューム名を割り当てますが、この割り当てられた名前は、Amazon EC2 が推奨する名前とは異なる可能性があります。

多くのインスタンスストアボリュームは ext3 ファイルシステムを使用して事前にフォーマットされています。TRIM コマンドをサポートする SSD ベースのインスタンスストアボリュームは、ファイルシステムを使用して事前にフォーマットされていません。ただし、インスタンスを起動してから、選択したファイルシステムでボリュームをフォーマットすることもできます。詳細については、「[インスタンスストアボリュームの TRIM のサポート \(p. 1087\)](#)」を参照してください。Windows インスタンスの場合は、EC2Config サービスが NTFS ファイルシステムでインスタンスストアボリュームをフォーマットします。

インスタンスストアデバイスが使用できるかどうかは、インスタンスマタデータを使用してインスタンスの内部から確認できます。詳細については、「[インスタンスストアボリュームのインスタンスブロックデバイスマッピングの表示 \(p. 1108\)](#)」を参照してください。

Windows インスタンスの場合は、Windows Disk Management を使用してインスタンスストアボリュームを表示することもできます。詳細については、「[Windows Disk Management を使用したディスクの一覧表示](#)」を参照してください。

Linux インスタンスの場合、次の手順で説明されているように、インスタンスストアボリュームを表示してマウントできます。

Linux でインスタンスストアボリュームを使用できるようにするには

1. SSH クライアントを使用してインスタンスに接続します。
2. `df -h` コマンドを使用して、フォーマットおよびマウントされたボリュームを表示します。`lsblk` を使用して、起動時にマッピングされたが、フォーマットおよびマウントされていないボリュームを表示します。
3. マッピングされたのみのインスタンスストアボリュームをフォーマットしてマウントするには、以下の作業を行います。
  - a. `mkfs` コマンドを使用してデバイスでファイルシステムを作成します。
  - b. `mkdir` コマンドを使用してデバイスをマウントするディレクトリを作成します。
  - c. `mount` コマンドを使用して、新しく作成されたディレクトリにデバイスをマウントします。

## SSD インスタンスストアボリューム

C、G2、I2、I3、M3、R3、および X1 の各インスタンスは、ソリッドステートドライブ (SSD) を使用するインスタンスストアボリュームをサポートすることで、高いランダム I/O パフォーマンスを実現しています。インスタンスタイプごとのインスタンスストアボリュームのサポートの詳細については、「[インスタンスストアボリューム \(p. 1078\)](#)」を参照してください。

Linux で SSD インスタンスストアボリュームから最高の IOPS 性能を得るには、最新バージョンの Amazon Linux、またはカーネルバージョン 3.8 以降の別の Linux AMI を使用することをお勧めします。カーネルバージョン 3.8 以降の Linux AMI を使用しないと、これらのインスタンスタイプで最大可能な IOPS パフォーマンスはインスタンスで実現されません。

他のインスタンスストアボリュームと同様に、インスタンスの SSD インスタンスストアボリュームを起動するときにマップする必要があります。SSD インスタンスボリューム上のデータは、関連するインスタンスの存続期間中のみ維持されます。詳細については、「[EC2 インスタンスにインスタンスストアボリュームを追加する \(p. 1083\)](#)」を参照してください。

## NVMe SSD ボリューム

C5d、G4、I3、I3en、F1、M5ad、M5d、`p3dn.24xlarge`、R5ad、R5d、および z1d インスタンスは、Non-Volatile Memory Express (NVMe) SSD インスタンスストアボリュームを提供します。NVMe ボリュームにアクセスするには、[NVMe ドライバー \(p. 1028\)](#)をインストールする必要があります。以下の AMI はこの要件を満たしています。

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 (`linux-aws` カーネル) 以降
- Red Hat Enterprise Linux 7.4 以降
- SUSE Linux Enterprise Server 12 SP2 以降
- CentOS 7.4.1708 以降
- FreeBSD 11.1 以降
- Debian GNU/Linux 9 以降

インスタンスに接続したら、`lspci` コマンドを使用して NVMe デバイスをリストできます。次に示すのは、4 つの NVMe デバイスをサポートする `i3.8xlarge` インスタンスの出力例です。

```
[ec2-user ~]$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
```

```
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 01)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Device 1d0f:ec20
00:17.0 Non-Volatile memory controller: Device 1d0f:cd01
00:18.0 Non-Volatile memory controller: Device 1d0f:cd01
00:19.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1a.0 Non-Volatile memory controller: Device 1d0f:cd01
00:1f.0 Unassigned class [ff80]: XenSource, Inc. Xen Platform Device (rev 01)
```

サポートされているオペレーティングシステムを使用しているが、NVMe デバイスが表示されない場合は、次のコマンドを使用して NVMe モジュールが読み込まれていることを確認します。

- Amazon Linux、Amazon Linux 2、Ubuntu 14/16、Red Hat Enterprise Linux、SUSE Linux Enterprise Server、CentOS 7

```
$ lsmod | grep nvme
nvme              48813   0
```

- Ubuntu 18

```
$ cat /lib/modules/$(uname -r)/modules.builtin | grep nvme
s/nvme/host/nvme-core.ko
kernel/drivers/nvme/host/nvme.ko
kernel/drivers/nvmem/nvmem_core.ko
```

NVMe ボリュームは NVMe 1.0e 仕様に準拠しています。NVMe コマンドは NVMe ボリュームで使用できます。Amazon Linux では、yum install コマンドを使用して repo から nvme-cli パッケージをインストールできます。サポートされているバージョンの Linux では、イメージで利用可能でない場合は nvme-cli パッケージをダウンロードできます。

NVMe インスタンスストレージのデータは、インスタンスのハードウェアモジュールに実装されている XTS-AES-256 ブロック暗号を使用して暗号化されます。暗号化キーは、ハードウェアモジュールで作成され、NVMe インスタンスストレージデバイスごとに固有です。すべての暗号化キーは、インスタンスが停止または終了して復元できないときに破棄されます。この暗号化を無効にしたり、独自の暗号キーを指定したりすることはできません。

## インスタンスストアボリュームの TRIM のサポート

C5d、F1、G4、I2、I3、I3en、M5ad、M5d、p3dn.24xlarge、R3、R5ad、R5d、および z1d インスタンスでは、TRIM を持つ SSD ボリュームがサポートされます。

TRIM をサポートしているインスタンスストアボリュームは、インスタンスに割り当てられる前に完全に TRIM が実行されます。これらのボリュームは、インスタンスの起動時にファイルシステムを使用してフォーマットされないため、マウントして使用する前にボリュームをフォーマットする必要があります。これらのボリュームを迅速に使用できるようにするには、ボリュームをフォーマットするときに、TRIM 操作をスキップします。

TRIM をサポートするインスタンスストアボリュームでは、TRIM コマンドを使用して、書き込んだデータが不要になったときに SSD コントローラーに通知することができます。これにより、より多くの空き領域がコントローラーに与えられ、その結果書き込み増幅が減り、パフォーマンスが向上します。Linux では、fstrim コマンドを使用して定期的な TRIM を有効にします。

## インスタンスストアスワップボリューム

Linux のスワップ空間は、システムで物理的に割り当てられたよりも多くのメモリを必要とする場合に使用できます。スワップ空間を有効にすると、Linux システムは頻繁に使用されないメモリページを物理メ

モリからスワップ空間(既存のファイルシステムの専用パーティションまたはスワップファイル)にスワッピし、高速なアクセスを必要とするメモリページのためにその空間を解放します。

Note

スワップ空間をメモリページングに使用しても、RAM 使用時ほど高速でも効率的でもありません。スワップ空間に定期的にメモリをページングするワークロードの場合は、RAM が多くサイズの大きいインスタンスタイプに移行することを検討してください。詳細については、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。

c1.medium および m1.small インスタンスタイプの物理メモリ容量は限られており、起動時には Linux AMI の仮想メモリとして機能する 900 MiB スワップボリュームが与えられます。Linux カーネルはこのスワップ領域をルートデバイス上のパーティションとして認識しますが、ルートデバイスのタイプに関係なく、実際には別のインスタンスストアボリュームです。

Amazon Linux は自動的にこのスワップ空間を有効にして使用しますが、AMI では、このスワップ空間を認識して使用するために、追加のステップが必要になる場合があります。インスタンスがスワップ空間を使用しているかどうか確認するには、swapon -s コマンドを使用できます。

```
[ec2-user ~]$ swapon -s
Filename           Type      Size   Used   Priority
/dev/xvda3        partition 917500  0      -1
```

上記のインスタンスには、900 MiB のスワップボリュームがアタッチされ、有効になっています。このコマンドでスワップボリュームが表示されない場合は、そのデバイスに対してスワップ空間を有効しなければならない可能性があります。利用可能なディスクは、lsblk コマンドを使用して確認します。

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0   8G  0 disk /
xvda3 202:3    0  896M 0 disk
```

ここで、スワップボリューム xvda3 はインスタンスで利用できますが、有効になっていません (MOUNTPOINT フィールドが空です)。スワップボリュームは swapon コマンドを使って有効にできます。

Note

lsblk でリストされるデバイス名の先頭に /dev/ を付加する必要があります。デバイスは、sda3、sde3、xvde3 など、異なる名前になる場合があります。システムのデバイス名は、次のコマンドで使用します。

```
[ec2-user ~]$ sudo swapon /dev/xvda3
```

これで、スワップ空間が lsblk および swapon -s 出力に表示されます。

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1    0   8G  0 disk /
xvda3 202:3    0  896M 0 disk [SWAP]
[ec2-user ~]$ swapon -s
Filename           Type      Size   Used   Priority
/dev/xvda3        partition 917500  0      -1
```

また、システムを起動する度にこのスワップ空間が自動的に有効になるように、/etc/fstab ファイルを編集する必要があります。

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

(システムの swap デバイス名を使用して) 次の行を /etc/fstab ファイルに追加します。

```
/dev/xvda3      none    swap    sw    0     0
```

### インスタンスストアボリュームをスワップ空間として使用するには

どのインスタンスストアボリュームもスワップ空間として使用できます。たとえば、m3.medium インスタンスタイプは、スワップ空間に適した 4 GB の SSD インスタンスストアボリュームを含みます。インスタンスストアボリュームがはるかに大きい場合(たとえば、350 GB)、ボリュームに 4~8 GB の小さいスワップパーティションを作成し、残りをデータボリュームにすることもできます。

#### Note

この手順は、インスタンストレージをサポートするインスタンスタイプのみに適用されます。サポートされているインスタンスタイプについては、「[インスタンスストアボリューム \(p. 1078\)](#)」を参照してください。

1. インスタンスにアタッチされたブロックデバイスの一覧を表示して、インスタンスストアボリュームのデバイス名を取得します。

```
[ec2-user ~]$ lsblk -p
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
/dev/xvdb 202:16   0 4G  0 disk /media/ephemeral0
/dev/xvda1 202:1   0 8G  0 disk /
```

この例では、インスタンスストアボリュームは /dev/xvdb です。これは Amazon Linux インスタンスであるため、インスタンスストアボリュームはフォーマットされ、/media/ephemeral0 にマウントされます。すべての Linux オペレーティングシステムでこれが自動的に実行されるわけではありません。

2. (省略可能) インスタンスストアボリュームがマウントされている場合 (lsblk コマンドの出力に MOUNTPOINT が表示されます)、次のコマンドを使ってアンマウントする必要があります。

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

3. mkswap コマンドを使って、デバイスに Linux スワップ領域をセットアップします。

```
[ec2-user ~]$ sudo mkswap /dev/xvdb
mkswap: /dev/xvdb: warning: wiping old ext3 signature.
Setting up swap space version 1, size = 4188668 KiB
no label, UUID=b4f63d28-67ed-46f0-b5e5-6928319e620b
```

4. 新しいスワップ空間を有効にします。

```
[ec2-user ~]$ sudo swapon /dev/xvdb
```

5. 新しいスワップ空間が使用されていることを確認します。

```
[ec2-user ~]$ swapon -s
Filename  Type  Size Used Priority
/dev/xvdb          partition 4188668 0 -1
```

6. システムを起動する度にこのスワップ空間が自動的に有効になるように、/etc/fstab ファイルを編集します。

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

/etc/fstab ファイルに /dev/xvdb (または /dev/sdb) 用の項目がある場合は、それを以下の行に合わせて変更します。このデバイス用の項目がない場合は、/etc/fstab ファイルに以下の行を追加します(システムのスワップデバイス名を使用します)。

|           |      |      |    |   |   |
|-----------|------|------|----|---|---|
| /dev/xvdb | none | swap | sw | 0 | 0 |
|-----------|------|------|----|---|---|

### Important

インスタンスが停止すると、インスタンストアボリュームデータが失われます。これは、「[Step 3 \(p. 1089\)](#)」で作成したインスタンストアスワップ領域のフォーマットが含まれます。インスタンストアのスワップ領域を使用するように設定されたインスタンスを停止および再起動した場合、新しいインスタンストアボリュームで「[Step 1 \(p. 1089\)](#)」から「[Step 5 \(p. 1089\)](#)」を繰り返す必要があります。

## インスタンストアボリュームのディスクパフォーマンスの最適化

Amazon EC2 でのディスクの仮想化方法が原因となり、一部のインスタンストアボリュームに対する最初の書き込みは、書き込みの場所にかかわらず、それ以降の書き込みより速度が遅くなります。ほとんどのアプリケーションでは、インスタンスの存続期間全体でこのコストを負担することは、許容範囲内です。ただし、高いディスクパフォーマンスを必要とする場合は、本稼働環境での使用の前に、ドライブのすべての場所に一度書き込みを行うことで初期化することをお勧めします。

### Note

インスタンスタイプの中には、初期化を行わずに、起動時に最大限のパフォーマンスを発揮する直接アタッチされた Solid State Drive (SSD) および TRIM サポートを使用するものがあります。各インスタンスタイプのインスタンストアについては、「[インスタンストアボリューム \(p. 1078\)](#)」を参照してください。

レイテンシーやスループットに関してさらに柔軟性が必要な場合は、Amazon EBS を使用することをお勧めします。

インスタンストアボリュームを初期化するには、初期化するストア (例: dd または /dev/sdb) に応じて、次の /dev/nvme1n1 コマンドを使用します。

### Note

必ずドライブをアンマウントしてから、このコマンドを実行してください。  
初期化には長い時間がかかる場合があります (エクストララージのインスタンスで約 8 時間)。

インスタンストアボリュームを初期化するには、m1.large、m1.xlarge、c1.xlarge、m2.xlarge、m2.2xlarge、m2.4xlarge インスタンスタイプで次のコマンドを使用します。

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

すべてのインスタンストアボリュームに対して同時に初期化を実行するには、次のコマンドを使用します。

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

ドライブを RAID 用に構成すると、ドライブのすべての場所に書き込みを行うことで、ドライブが初期化されます。ソフトウェアベースの RAID を構成するときは、再構築の最低速度を必ず変更してください。

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

## ファイルストレージ

クラウドファイルストレージは、クラウド上にデータを保存する方法で、サーバーとアプリケーションは共有ファイルシステムを通してデータにアクセスできます。クラウドファイルストレージが持つこの互換性は共有ファイルシステムに依存するワークロードに最適で、コードを変更することなく統合を容易に行えます。

スケーラビリティを持たない、またはデータを保護するための冗長性がほとんどないブロックストレージを基盤として使用する、コンピューティングインスタンス上の単一ノードのファイルサーバーから、独自のクラスター化されたソリューション、[Amazon Elastic File System \(Amazon EFS\) \(p. 1091\)](#) または [Amazon FSx for Windows ファイルサーバー \(p. 1095\)](#) などの完全マネージド型のソリューションまで、さまざまなファイルストレージソリューションがあります。

### Amazon Elastic File System (Amazon EFS)

Amazon EFS は、Amazon EC2 と併用できるスケーラブルなファイルストレージを提供します。EFS ファイルシステムを作成し、ファイルシステムをマウントするためにインスタンスを設定できます。複数のインスタンスで実行している作業負荷やアプリケーションの一般的なデータソースとして EFS ファイルシステムを使用できます。詳細については、「[Amazon Elastic File System 製品ページ](#)」を参照してください。

このチュートリアルでは、EFS ファイルシステム、およびそのファイルシステムを使ってデータを共有できる 2 つの Linux インスタンスを作成します。

#### Important

Amazon EFS は Windows インスタンスではサポートされていません。

#### タスク

- 前提条件 (p. 1091)
- ステップ 1: EFS ファイルシステムの作成 (p. 1092)
- ステップ2: ファイルシステムをマウントします。 (p. 1092)
- ステップ3: ファイルシステムをテストする (p. 1094)
- ステップ 4: クリーンアップする (p. 1094)

## 前提条件

- EC2 インスタンスと EFS マウントターゲットに関するセキュリティグループ (efs-sg など) を作成し、以下のルールを追加します。
  - コンピュータから EC2 インスタンスへのインバウンド SSH 接続を許可する (ソースはネットワークの CIDR ブロックです)。
  - このセキュリティグループに関する EC2 インスタンスから EFS マウントターゲットを介したファイルシステムへのインバウンド NFS 接続を許可する (ソースはセキュリティグループ自体です)。詳細については、「[Amazon EFS ルール \(p. 924\)](#)」と「[Amazon EC2 インスタンスとマウントターゲットのセキュリティグループ](#)」(Amazon Elastic File System ユーザーガイド) を参照してください。
- キーペアの作成。インスタンスの作成時にキーペアを指定しないと、それに接続できません。詳細については、「[キーペアを作成する \(p. 22\)](#)」を参照してください。

## ステップ 1: EFS ファイルシステムの作成

Amazon EFS では、複数のインスタンスが同時にマウントおよびアクセスできるファイルシステムを作成することができます。詳細については、『Amazon Elastic File System ユーザーガイド』の「[Creating Resources for Amazon EFS](#)」を参照してください。

ファイルシステムを作成するには

1. Amazon Elastic File System コンソール (<https://console.aws.amazon.com/efs/>) を開きます。
2. [Create file system] を選択します。
3. [Configure network access] ページで、次の操作を行います。
  - a. [VPC] で、インスタンスで使用する VPC を選択します。
  - b. [Create mount targets] で、すべてのアベイラビリティーゾーンを選択します。
  - c. 各アベイラビリティーゾーンで、[Security group] の値が [前提条件 \(p. 1091\)](#) で作成したセキュリティグループであることを確認します。
  - d. [Next Step] を選択します。
4. [Configure file system settings] ページで、次の操作を行います。
  - a. Key=Name のタグ用に、[Value] でファイルシステムの名前を入力します。
  - b. [Choose throughput mode] で、[Bursting] オプションをデフォルトのままにします。
  - c. [Choose performance mode] では、デフォルトオプション [汎用] を維持します。
  - d. [Next Step] を選択します。
5. [Configure client access] ページで、デフォルトの設定を維持し、[Next Step] を選択します。
6. [Review and create] ページで、[Create File System] を選択します。
7. ファイルシステムを作成した後、このチュートリアルで後ほど使用するため、ファイルシステム ID をメモします。

## ステップ2: ファイルシステムをマウントします。

次の手順を使用して 2 つの t2.micro インスタンスを起動します。ユーザーデータスクリプトは起動時にファイルシステムを両方のインスタンスにマウントし、/etc/fstab を更新してインスタンスの再起動後にファイルシステムが再度マウントされるようにします。T2 インスタンスは、サブネットで起動する必要があることにご注意ください。デフォルト VPC またはデフォルト以外の VPC を使用することができます。

### Note

他の方法でもボリュームをマウントできます（たとえば、すでに実行しているインスタンスなど）。詳細については、『Amazon Elastic File System ユーザーガイド』の「[ファイルシステムのマウント](#)」を参照してください。

### 2 つのインスタンスを起動して EFS ファイルシステムをマウントする

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. [インスタンスの作成] を選択します。
3. [Choose an Amazon Machine Image] ページで、HVM 仮想化タイプで Amazon Linux AMI を選択します。
4. [Choose an Instance Type] ページで、デフォルトのインスタンスタイプ t2.micro を維持し、[Next: Configure Instance Details] を選択します。
5. [Configure Instance Details] ページで以下の操作を実行します。
  - a. [Number of instances] に「2」と入力します。

- b. [デフォルト VPC] デフォルト VPC がある場合、[Network] のデフォルト値です。デフォルト VPC と [Subnet] のデフォルト値を維持し、インスタンス用に Amazon EC2 が選択するアベイラビリティゾーンでデフォルトのサブネットを使用します。
- [デフォルト以外の VPC] [Network] の VPC と [Subnet] のパブリックサブネットを選択します。
- c. [デフォルト以外のVPC] [Auto-assign Public IP] で、[Enable] を選択します。それ以外の場合、インスタンスではパブリック IP アドレスまたはパブリック DNS 名を取得しません。
- d. [Advanced Details (詳細情報)] で [テキストで] を選択し、以下のスクリプトを [ユーザーデータ] に貼り付けます。[FILE\_SYSTEM\_ID] をファイルシステムの ID で更新します。またオプションで、[MOUNT\_POINT] をマウント済みのファイルシステムのディレクトリで更新できます。

#### IMDSv2

```
#!/bin/bash
yum update -y
yum install -y nfs-utils
FILE_SYSTEM_ID=fs-XXXXXXX
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"`
AVAILABILITY_ZONE=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" -s
http://169.254.169.254/latest/meta-data/placement/availability-zone)
REGION=${AVAILABILITY_ZONE:0:-1}
MOUNT_POINT=/mnt/efs
mkdir -p ${MOUNT_POINT}
chown ec2-user:ec2-user ${MOUNT_POINT}
echo ${FILE_SYSTEM_ID}.efs.${REGION}.amazonaws.com:/ ${MOUNT_POINT} nfs4
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,_netdev 0 0
>> /etc/fstab
mount -a -t nfs4
```

#### IMDSv1

```
#!/bin/bash
yum update -y
yum install -y nfs-utils
FILE_SYSTEM_ID=fs-XXXXXXX
AVAILABILITY_ZONE=$(curl -s http://169.254.169.254/latest/meta-data/placement/
availability-zone )
REGION=${AVAILABILITY_ZONE:0:-1}
MOUNT_POINT=/mnt/efs
mkdir -p ${MOUNT_POINT}
chown ec2-user:ec2-user ${MOUNT_POINT}
echo ${FILE_SYSTEM_ID}.efs.${REGION}.amazonaws.com:/ ${MOUNT_POINT} nfs4
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,_netdev 0 0
>> /etc/fstab
mount -a -t nfs4
```

- e. ウィザードのステップ 6 に進みます。
6. [Configure Security Group (セキュリティグループの設定)] ページで、[Select an existing security group (既存のセキュリティグループの選択)] を選択し、「[前提条件 \(p. 1091\)](#)」で作成したセキュリティグループを選択したら、[Review and Launch (確認して起動)] を選択します。
7. [Review Instance Launch] ページで、[Launch] を選択します。
8. [Select an existing key pair or create a new key pair] ダイアログボックスで、[Choose an existing key pair] を選択し、キーペアを選択します。確認のチェックボックスを選択し、[Launch Instances] を選択します。
9. ナビゲーションペインの [Instances] を選択して、インスタンスのステータスを表示します。最初、ステータスは pending です。ステータスが running に変わったら、インスタンスは使用できる状態です。

## ステップ3: ファイルシステムをテストする

インスタンスに接続し、ファイルシステムが指定したディレクトリにマウントされていることを確認します（たとえば、/mnt/efs）。

ファイルシステムがマウントされていることを確認するには

1. インスタンスに接続します。詳細については、「[Linux インスタンスへの接続 \(p. 505\)](#)」を参照してください。
2. 各インスタンスのターミナルウインドウから、df -T コマンドを実行して、EFS ファイルシステムがマウントされていることを確認します。

```
$ df -T
Filesystem      Type            1K-blocks   Used       Available Use% Mounted on
/dev/xvda1      ext4           8123812    1949800   6073764   25% /
devtmpfs        devtmpfs        4078468     56        4078412   1% /dev
tmpfs           tmpfs          4089312     0         4089312   0% /dev/shm
efs-dns         nfs4           9007199254740992   0      9007199254740992   0% /mnt/efs
```

なお、ファイルシステムの名前（サンプル出力では `efs-dns` として表示）は次の形式になります。

```
file-system-id.efs.aws-region.amazonaws.com:/
```

3. (オプション) 1 つのインスタンスからファイルシステムでファイルを作成し、他のインスタンスからファイルを表示できることを確認します。

- a. 最初のインスタンスから、次のコマンドを実行してファイルを作成します。

```
$ sudo touch /mnt/efs/test-file.txt
```

- b. 2 つ目のインスタンスから、次のコマンドを実行してファイルを表示します。

```
$ ls /mnt/efs
test-file.txt
```

## ステップ 4: クリーンアップする

このチュートリアルを完了した後、インスタンスを終了して、ファイルシステムを削除できます。

インスタンスを終了するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. 終了するインスタンスを選択します。
4. [アクション]、[インスタンスの状態]、[終了] の順に選択します。
5. 確認を求めるメッセージが表示されたら、[Yes, Terminate] を選択します。

ファイルシステムを削除するには

1. Amazon Elastic File System コンソール (<https://console.aws.amazon.com/efs/>) を開きます。
2. 削除するファイルシステムを選択します。
3. [Actions]、[Delete file system] の順に選択します。
4. 確認の入力を求められたら、ファイルシステムの ID を入力し、[Delete File System] を選択します。

## Amazon FSx for Windows ファイルサーバー

Amazon FSx for Windows ファイルサーバーは、エンタープライズアプリケーションを簡単に AWS に移行および移行するための機能、パフォーマンス、および互換性を備えた完全ネイティブの Windows ファイルシステムを基盤とする、フルマネージド型の Windows ファイルサーバーを提供します。

Amazon FSx は、Microsoft Windows Server 上に構築されたフルマネージド型ファイルストレージを使用して、幅広いエンタープライズ Windows ワークロードをサポートします。Amazon FSx は、Windows ファイルシステム機能、およびネットワーク経由でファイルストレージにアクセスするための業界標準のサーバーメッセージブロック (SMB) プロトコルをネイティブでサポートしています。Amazon FSx は、ネイティブの Windows 互換性、エンタープライズのパフォーマンスと機能、一貫した 1 ミリ秒未満のレイテンシーで、AWS クラウドのエンタープライズアプリケーションに最適化されています。

Amazon FSx のファイルストレージを使用すると、Windows の開発者や管理者が今日使用しているコード、アプリケーション、およびツールを変更することなく引き続き使用できます。Amazon FSx に最適な Windows アプリケーションとワーカロードには、ビジネスアプリケーション、ホームディレクトリ、ウェブ配信、コンテンツ管理、データ分析、ソフトウェアビルド設定、およびメディア処理ワーカロードが含まれます。

フルマネージド型サービスとして、Amazon FSx for Windows ファイルサーバーはファイルサーバーとストレージボリュームの設定とプロビジョニングの管理オーバーヘッドを排除します。さらに、Amazon FSx は Windows ソフトウェアを最新の状態に保ち、ハードウェア障害を検出して対処し、バックアップを実行します。また、Microsoft Active Directory 用の AWS Directory Service、Amazon WorkSpaces、AWS Key Management Service、AWS CloudTrail など、他の AWS サービスとの高度な統合も提供します。

詳細については、[Amazon FSx for Windows ファイルサーバー ユーザーガイド](#) を参照してください。

## Amazon Simple Storage Service (Amazon S3)

Amazon S3 は、インターネットデータのリポジトリです。また、Amazon S3 は高速かつ低コストで信頼性に優れたデータストレージインフラストラクチャを実現します。ウェブスケールのコンピューティングを簡単に行えるように設計されており、Amazon EC2 の内部やウェブ上のどこからでも、いつでも必要な量だけデータを格納および取得できます。Amazon S3 では、複数の施設にまたがる複数のデバイスにデータオブジェクトが冗長的に保存されるので、多数のさまざまなクライアントやアプリケーションスレッドからデータオブジェクトに対する読み込み/書き込みの同時アクセスが可能になります。Amazon S3 に格納された冗長データを使用すれば、インスタンスまたはアプリケーションの障害から迅速かつ確実に復旧できます。

Amazon EC2 は Amazon マシンイメージ (AMI) を格納するために Amazon S3 を使用します。AMI は EC2 インスタンスを起動するために使用します。インスタンスに障害が発生した場合は、格納済みの AMI を使用して別のインスタンスを即座に起動できるので、高速復旧と事業継続が可能になります。

Amazon EC2 はデータボリュームのスナップショット (バックアップコピー) の格納にも Amazon S3 を使用します。アプリケーションまたはシステムで障害が発生した場合、スナップショットを使用すれば、データをしばらく確実に回復できます。またスナップショットは、複数の新しいデータボリュームの作成、既存のデータボリュームのサイズ拡張、アベイラビリティゾーン間でのデータボリュームの移動を行うためのベースラインとして使用することもできます。このような機能により、データの使い方を大幅に拡張できます。データボリュームとスナップショットの使用については、「[Amazon Elastic Block Store \(p. 929\)](#)」を参照してください。

オブジェクトは、Amazon S3 に格納される基本エンティティです。Amazon S3 に格納されるすべてのオブジェクトは、バケットに保管されます。バケットは Amazon S3 名前空間の最上位レベルを構成し、個々のストレージを所有するアカウントを識別します。Amazon S3 のバケットはインターネットのドメイン名に似ています。バケットに格納されたオブジェクトは一意のキーを持ち、HTTP URL アドレスを使用して取得されます。たとえば、キー (/photos/mygarden.jpg) を持つオブジェクト

が `aws-s3-bucket1` バケットに格納されている場合、このオブジェクトは URL (`http://aws-s3-bucket1.s3.amazonaws.com/photos/mygarden.jpg`) を使用してアドレス解決できます。

Amazon S3 の機能の詳細については、「[Amazon S3 の製品ページ](#)」を参照してください。

## Amazon S3 および Amazon EC2

Amazon S3 にはストレージとしての利点があるので、このサービスを使用して、EC2 インスタンスで使用するためにファイルおよびデータセットを格納する場合があります。Amazon S3 とインスタンスとの間でデータを移動するには、いくつかの方法があります。以下に説明する例以外にも、コンピュータやインスタンスから Amazon S3 のデータにアクセスできるさまざまなツールが、他のユーザーによって作成されています。一般的な一部のツールについては、AWS フォーラムで取り上げられています。

アクセス許可がある場合は、以下の方法を使用して、Amazon S3 とインスタンスとの間でファイルをコピーできます。

GET または wget

wget ユーティリティは、Amazon S3 からパブリックオブジェクトをダウンロードできる HTTP および FTP のクライアントです。これは、Amazon Linux やその他のほとんどのディストリビューションにデフォルトでインストールされ、Windows ではダウンロード可能です。Amazon S3 オブジェクトをダウンロードするには、次のコマンドを入力し、ダウンロードするオブジェクトの URL に置き換えます。

```
[ec2-user ~]$ wget https://my_bucket.s3.amazonaws.com/path-to-file
```

この方法では、要求するオブジェクトがパブリックである必要があります。オブジェクトがパブリックではない場合、[ERROR 403: Forbidden] メッセージを受け取ります。このエラーを受け取った場合は、Amazon S3 コンソールを開き、オブジェクトのアクセス許可をパブリックに変更します。詳細については、「[Amazon Simple Storage Service 開発者ガイド](#)」を参照してください。

AWS Command Line Interface

AWS Command Line Interface (AWS CLI) は、AWS のサービスを管理するための統合ツールです。AWS CLI を使用すると、ユーザーは自分自身を認証し、限定された項目を Amazon S3 からダウンロードしたり、項目をアップロードしたりできます。ツールのインストールおよび設定方法などの詳細については、「[AWS Command Line Interface の詳細ページ](#)」を参照してください。

aws s3 cp コマンドは、Unix cp コマンドと似ています。ファイルを Amazon S3 からインスタンスにコピーしたり、ファイルをインスタンスから Amazon S3 にコピーしたりできるほか、ファイルを Amazon S3 の 1 つの場所から別の場所にコピーすることもできます。

オブジェクトを Amazon S3 からインスタンスにコピーするには、次のコマンドを使用します。

```
[ec2-user ~]$ aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

オブジェクトをインスタンスから Amazon S3 にコピーして戻すには、次のコマンドを使用します。

```
[ec2-user ~]$ aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

aws s3 sync コマンドは、Amazon S3 バケット全体をローカルディレクトリの場所に同期できます。この機能は、データセットをダウンロードし、リモートセットでローカルコピーを最新の状態に保つ際に役立ちます。Amazon S3 バケットに対して適切なアクセス許可がある場合は、コマンドで送信元と送信先の場所を入れ替えることで、終了時にローカルディレクトリバックアップをクラウドにプッシュできます。

Amazon S3 バケット全体をインスタンスのローカルディレクトリにダウンロードするには、次のコマンドを使用します。

```
[ec2-user ~]$ aws s3 sync s3://remote_s3_bucket local_directory
```

#### Amazon S3 API

開発者は API を使用して、Amazon S3 のデータにアクセスできます。詳細については、「[Amazon Simple Storage Service 開発者ガイド](#)」を参照してください。この API およびその例を使用すると、アプリケーションを開発し、boto Python インターフェイスなどの他の API および SDK と統合するのに役立ちます。

## インスタンスボリューム数の制限

インスタンスで使用できるボリュームの最大数は、オペレーティングシステムやインスタンスタイプによって異なります。インスタンスに追加するボリューム数を検討する際の考慮事項は、I/O 帯域幅の増加とストレージ容量の増加のどちらが必要かという点です。

#### コンテンツ

- [Linux 固有のボリュームの制限 \(p. 1097\)](#)
- [Windows 固有のボリュームの制限 \(p. 1097\)](#)
- [インスタンスタイプの制限 \(p. 1098\)](#)
- [帯域幅と容量 \(p. 1098\)](#)

## Linux 固有のボリュームの制限

40 以上のボリュームをアタッチすると、起動に失敗する可能性があります。この数値には、ルートボリュームと、アタッチされたインスタンスストアボリュームや EBS ボリュームが含まれます。多数のボリュームがアタッチされているインスタンスで起動の問題が発生した場合は、インスタンスを停止し、起動プロセスに不可欠ではないボリュームをデタッチし、インスタンスの実行後にそれらのボリュームを再度アタッチします。

#### Important

Linux インスタンスに 40 より多くのボリュームをアタッチすることは、ベストエフォートベースでのみサポートされ、保証されません。

## Windows 固有のボリュームの制限

次の表は、使用中のドライバーに基づく Windows インスタンスのボリュームの制限を示しています。これらの数値には、ルートボリュームと、アタッチされたインスタンスストアボリュームや EBS ボリュームが含まれます。

#### Important

Windows インスタンスに次の数よりも多くのボリュームをアタッチすることは、ベストエフォートベースでのみサポートされ、保証されません。

| ドライバー     | ボリュームの制限 |
|-----------|----------|
| AWS PV    | 26       |
| Citrix PV | 26       |

| ドライバー      | ボリュームの制限 |
|------------|----------|
| Red Hat PV | 17       |

パフォーマンスに問題が発生する可能性が高いため、Windows インスタンスに AWS PV または Citrix PV ドライバーを持つ 26 を超えるボリュームを設定することはお勧めしません。

インスタンスで使用する PV ドライバーを決定する場合、または Windows インスタンスで Red Hat PV ドライバーから Citrix PV ドライバーにアップグレードする場合は、「[Windows インスタンスの PV ドライバーのアップグレード](#)」を参照してください。

デバイス名とボリュームの関連の詳細については、「[Windows インスタンスの Amazon EC2 ユーザーガイド](#)」の「[Windows EC2 インスタンスのボリュームへのディスクのマッピング](#)」を参照してください。

## インスタンスタイプの制限

A1、C5、C5d、C5n、G4、I3en、Inf1、M5、M5a、M5ad、M5d、M5dn、M5n、p3dn.24xlarge、R5、R5a、R5ad、および z1d インスタンスは、ネットワークインターフェイス、EBS ボリューム、および NVMe インスタンスストアボリュームを含め、最大 28 のアタッチをサポートしています。インスタンスごとに 1 つ以上のネットワークインターフェイスのアタッチメントがあります。NVMe インスタンスストアボリュームは自動的にアタッチされます。たとえば、EBS のみのインスタンスに追加のネットワークインターフェイスのアタッチがない場合は、そのインスタンスに最大 27 個の EBS ボリュームをアタッチできます。2 つの NVMe インスタンスストアボリュームを持つインスタンスに追加の 1 つのネットワークインターフェイスがある場合、そのインスタンスには 24 の EBS ボリュームをアタッチできます。詳細については、「[Elastic Network Interface \(p. 713\)](#)」および「[インスタンスストアボリューム \(p. 1078\)](#)」を参照してください。

c5.metal、c5d.metal、i3.metal、m5.metal、m5d.metal、r5.metal、r5d.metal、および z1d.metal インスタンスは、最大 31 の EBS ボリュームをサポートしています。

u-6tb1.metal、u-9tb1.metal、および u-12tb1.metal の各インスタンスは、最大 13 の EBS ボリュームをサポートしています。u-18tb1.metal および u-24tb1.metal の各インスタンスは最大 19 の EBS ボリュームをサポートしています。

## 帯域幅と容量

整合性がとれており予測可能な帯域幅のユースケースでは、EBS 最適化インスタンスまたは 10 ギガビットのネットワーク接続インスタンスと、汎用 SSD ボリュームまたはプロビジョント IOPS SSD ボリュームを使用します。「[Amazon EBS – 最適化インスタンス \(p. 1031\)](#)」のガイダンスに従い、最大のパフォーマンスを引き出すために、ボリュームに対してプロビジョニングした IOPS とインスタンスから利用できる帯域幅を一致させてください。RAID 構成では、多くの管理者は、8 つのボリュームよりも大きなアレイでパフォーマンスが低下したことを確認します。これは、I/O オーバーヘッドの増加が原因です。個々のアプリケーションのパフォーマンスをテストし、必要に応じて調整してください。

## Linux インスタンスでのデバイスの名前付け

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

インスタンスがサポートできるボリュームの数は、オペレーティングシステムによって決まります。詳細については、「[インスタンスボリューム数の制限 \(p. 1097\)](#)」を参照してください。

コンテンツ

- [使用できるデバイス名 \(p. 1099\)](#)
- [デバイス名に関する考慮事項 \(p. 1099\)](#)

Windows インスタンスのデバイス名の詳細については、『Windows インスタンスの Amazon EC2 ユーザーガイド』の「[Windows インスタンスでのデバイスの名前付け](#)」を参照してください。

## 使用できるデバイス名

Linux インスタンスでは、準仮想化 (PV) とハードウェア仮想マシン (HVM) の 2 種類の仮想化を使用できます。インスタンスの仮想化タイプは、インスタンスの起動に使用される AMI によって決まります。すべてのインスタンスタイプが HVM AMI をサポートしています。一部の前世代のインスタンスタイプは PV AMI をサポートしています。使用できる推奨のデバイス名はインスタンスの仮想化タイプによって異なるため、必ず AMI の仮想化タイプを確認してください。詳細については、「[Linux AMI 仮想化タイプ \(p. 98\)](#)」を参照してください。

次の表に、ブロックデバイスマッピングまたは EBS ボリュームの接続時に指定できる使用可能なデバイス名を示します。

| 仮想化タイプ | 使用可能                                                                                 | ルート用に予約済み                               | EBS ボリュームとして推奨                                                                                                     | インスタンストアボリューム                                                                                                              |
|--------|--------------------------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| 準仮想化   | /dev/sd[a-z]<br><br>/dev/sd[a-z][1-15]<br><br>/dev/hd[a-z]<br><br>/dev/hd[a-z][1-15] | /dev/sda1                               | /dev/sd[f-p]<br><br>/dev/sd[f-p][1-6]                                                                              | /dev/sd[b-e]                                                                                                               |
| HVM    | /dev/sd[a-z]<br><br>/dev/xvd[b-c][a-z]                                               | AMI による違い<br><br>/dev/sda1 or /dev/xvda | /dev/sd[f-p] *<br><br>/dev/sd[b-h] (h1.16xlarge)<br><br>/dev/sd[b-y] (d2.8xlarge)<br><br>/dev/sd[b-i] (i2.8xlarge) | /dev/sd[b-e]<br><br>/dev/sd[b-h] (h1.16xlarge)<br><br>/dev/sd[b-y] (d2.8xlarge)<br><br>/dev/sd[b-i] (i2.8xlarge)<br><br>** |

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

\*\* NVMe インスタンストアボリュームは自動的に列挙され、NVMe デバイス名が割り当てられます。

インスタンストアボリュームの詳細については、[Amazon EC2 インスタンストア \(p. 1076\)](#) を参照してください。NVMe EBS ボリュームについては、「[Linux インスタンスの Amazon EBS および NVMe \(p. 1027\)](#)」を参照してください。

## デバイス名に関する考慮事項

デバイス名を選択するときは、以下の点を常に考慮する必要があります。

- インスタンスストアボリュームをアタッチするために使用されたデバイス名を使用して EBS ボリュームをアタッチすることはできますが、動作を予測できない場合があるため、この方法は使用しないことを強くお勧めします。
- インスタンスの NVMe インスタンスストアボリュームの数は、インスタンスのサイズによって異なります。NVMe インスタンスストアボリュームは自動的に列挙され、NVMe デバイス名 (`/dev/nvme[0-26]n1`)
- カーネルのブロックデバイスドライバによっては、指定した名前とは異なる名前でデバイスがアタッチされる可能性があります。たとえば、デバイス名 (`/dev/sdh`) を指定した場合、デバイスの名前が `/dev/xvdh` や `/dev/hdh` に変更される場合があります。ほとんどの場合、末尾の文字は変更されません。Red Hat Enterprise Linux (および CentOS などのバリエント) の一部バージョンでは、末尾の文字が変更されることがあります (例: `/dev/sda` が `/dev/xvde`)。このような場合、各デバイス名の末尾の文字は同じ規則で変更されます。たとえば、`/dev/sdb` が `/dev/xvdf` に変更された場合、`/dev/sdc` は `/dev/xvdg` に変更されます。Amazon Linux は、名前が変更されたデバイスに対して指定した名前のシンボリックデバイスを作成します。他のオペレーティングシステムの動作は異なる場合があります。
- HVM AMI では、ルートデバイス用に予約されているデバイス名 (`/dev/sda1`) や `/dev/sda2` を除き、デバイス名の末尾に数字を使用することをサポートしていません。`/dev/sda2` は使用できますが、HVM インスタンスでこのデバイスマッピングを使用することは推奨していません。
- PV AMI を使用する場合は、末尾の番号の有無にかかわらず、同じデバイス文字を共有するボリュームをアタッチすることはできません。たとえば、あるボリュームを `/dev/sdc` としてアタッチし、別のボリュームを `/dev/sdc1` としてアタッチした場合、インスタンスは `/dev/sdc` のみを認識します。デバイス名の末尾に数字を使用するには、同じベース文字を共有するすべてのデバイス名の末尾に数字を使用する必要があります (例: `/dev/sdc1`、`/dev/sdc2`、`/dev/sdc3`)。
- 一部のカスタムカーネルでは、使用できるデバイス名が `/dev/sd[f-p]` や `/dev/sd[f-p][1-6]` に制限されている場合があります。`/dev/sd[q-z]` または `/dev/sd[q-z][1-6]` の使用に関して問題がある場合は、`/dev/sd[f-p]` または `/dev/sd[f-p][1-6]` に切り替えてみてください。

## ブロックデバイスマッピング

起動する各インスタンスには、Amazon EBS ボリュームまたはインスタンスストアボリュームという、どちらかのルートデバイスボリュームが関連付けられています。ブロックデバイスマッピングを使用すると、インスタンスの起動時にそのインスタンスにアタッチする追加の EBS ボリュームまたはインスタンスストアボリュームを指定できます。追加する EBS ボリュームは、実行中のインスタンスにアタッチすることができます。「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」をご参照ください。ただし、インスタンスストアボリュームについては、ブロックデバイスマッピングを使用して、インスタンスの起動時にアタッチする以外方法はありません。

ルートデバイスボリュームの詳細については、「[永続的ルートデバイスボリュームへの変更 \(p. 19\)](#)」を参照してください。

### コンテンツ

- [ブロックデバイスマッピングの概念 \(p. 1100\)](#)
- [AMI ブロックデバイスマッピング \(p. 1103\)](#)
- [インスタンスブロックデバイスマッピング \(p. 1105\)](#)

## ブロックデバイスマッピングの概念

ブロックデバイスは、一連のバイトまたはビット (ブロック) でデータを移動するストレージデバイスです。これらのデバイスはランダムアクセスをサポートし、通常は、バッファされた I/O を使用します。たとえば、ハードディスク、CD-ROM ドライブ、フラッシュドライブなどがブロックデバイスに含まれます。ブロックデバイスは物理的にコンピュータにアタッチできます。また、コンピュータに物理的にアタッチされているかのように、リモートでアクセスすることもできます。Amazon EC2 は、2 種類のブロックデバイスをサポートしています。

- インスタンスストアボリューム (基盤となるハードウェアがインスタンスのホストコンピュータに物理的にアタッチされている仮想デバイス)
- EBS ボリューム (リモートストレージデバイス)

ブロックデバイスマッピングでは、インスタンスにアタッチするブロックデバイス (インスタンスストアボリュームと EBS ボリューム) を定義します。ブロックデバイスマッピングは、AMI 作成プロセスの一環として、AMI から起動されるすべてのインスタンスによって使用されるように指定できます。また、インスタンスの起動時にブロックデバイスマッピングを指定することもできます。起動したインスタンスの AMI すでに指定されているマッピングは、このマッピングによって上書きされます。インスタンスタイプによってサポートされるすべての NVMe インスタンスストアボリュームが自動的に列挙され、インスタンスの起動時にデバイス名が割り当てられることに注意してください。それらをブロックデバイスマッピングに含めます。含めないとインスタンスは効果がありません。

#### コンテンツ

- [ブロックデバイスマッピングのエントリ \(p. 1101\)](#)
- [ブロックデバイスマッピングのインスタンスストアの警告 \(p. 1101\)](#)
- [ブロックデバイスマッピングの例 \(p. 1102\)](#)
- [オペレーティングシステムでデバイスを使用できるようにする方法 \(p. 1103\)](#)

## ブロックデバイスマッピングのエントリ

ブロックデバイスマッピングを作成するとき、インスタンスにアタッチする必要があるブロックデバイスごとに以下の情報を指定します。

- Amazon EC2 内で使用されるデバイス名。インスタンスのブロックデバイスドライバーは、ボリュームをマウントするときに実際のボリューム名を割り当てます。この割り当てられた名前は、Amazon EC2 が推奨する名前とは異なる可能性があります。詳細については、「[Linux インスタンスでのデバイスの名前付け \(p. 1098\)](#)」を参照してください。
- [インスタンスストアボリューム] 仮想デバイス: `ephemeral[0-23]`。インスタンスで使用できるインスタンスストアボリュームの数とサイズは、インスタンスタイプによって異なります。
- [NVMe インスタンスストアボリューム] これらのボリュームが自動的に列挙され、デバイス名が割り当てられます。それらをブロックデバイスマッピングに含めます。含めないとインスタンスは効果がありません。
- [EBS ボリューム] ブロックデバイスを作成するときに使用するスナップショットの ID (`snap-xxxxxxxx`)。ボリュームサイズを指定する場合、この値はオプションです。
- [EBS ボリューム] ボリュームのサイズ (GiB 単位)。指定されたサイズは、指定されたスナップショットのサイズ以上である必要があります。
- [EBS ボリューム] インスタンス終了時にボリュームを削除するかどうか (`true` または `false`)。デフォルト値は、ルートデバイスボリュームでは `true`、アタッチされたボリュームでは `false` です。AMI を作成するときは、そのブロックデバイスマッピングがインスタンスからこの設定を継承します。インスタンスを起動するときに、AMI からこの設定を継承します。
- [EBS ボリューム] ボリュームタイプとして、`gp2` (汎用 SSD)、`io1` (プロビジョンド IOPS SSD)、`st1` (スループット最適化 HDD)、`sc1` (Cold HDD)、または `standard` (マグネティック) を指定します。デフォルト値は `gp2` です。
- [EBS ボリューム] ボリュームがサポートする 1 秒あたりの入力/出力オペレーションの数 (IOPS)。  
(`gp2`、`st1`、`sc1`、`standard` ボリュームの場合は使用されません。)

## ブロックデバイスマッピングのインスタンスストアの警告

ブロックデバイスマッピングでインスタンスストアボリュームがある場合は、インスタンスを AMI から起動すると、いくつかの警告が表示されます。

- インスタンスタイプによって中に含まれるインスタンストアボリューム数が異なり、インスタンストアボリュームがまったく含まれないインスタンスタイプもあります。単一インスタンストアボリュームのみをサポートするインスタンスタイプで、AMI が 2 つのインスタンストアボリュームにマッピングされている場合、インスタンスは単一のインスタンストアボリュームのみで起動します。
- インスタンストアボリュームをマッピングできるのは、起動時のみに限られます。インスタンストアボリュームのないインスタンスを停止することはできません (t2.micro など)。インスタンストアボリュームをサポートするインスタンスに変更し、インスタンストアボリュームを含めて再起動します。ただし、AMI をインスタンスから作成し、インスタンストアボリュームをサポートするインスタンスタイプで起動して、インスタンストアボリュームをインスタンスにマッピングすることは可能です。
- インスタンストアボリュームをマッピングしたインスタンスを起動し、インスタンスを停止して、インスタンストアボリュームの少ないインスタンスタイプに変更して再開すれば、最初の起動からマッピングしたインスタンストアボリュームもインスタンスのメタデータに表示されます。ただし、インスタンスに使用できるのは、そのインスタンスタイプでサポートされているインスタンストアボリュームの最大数までです。

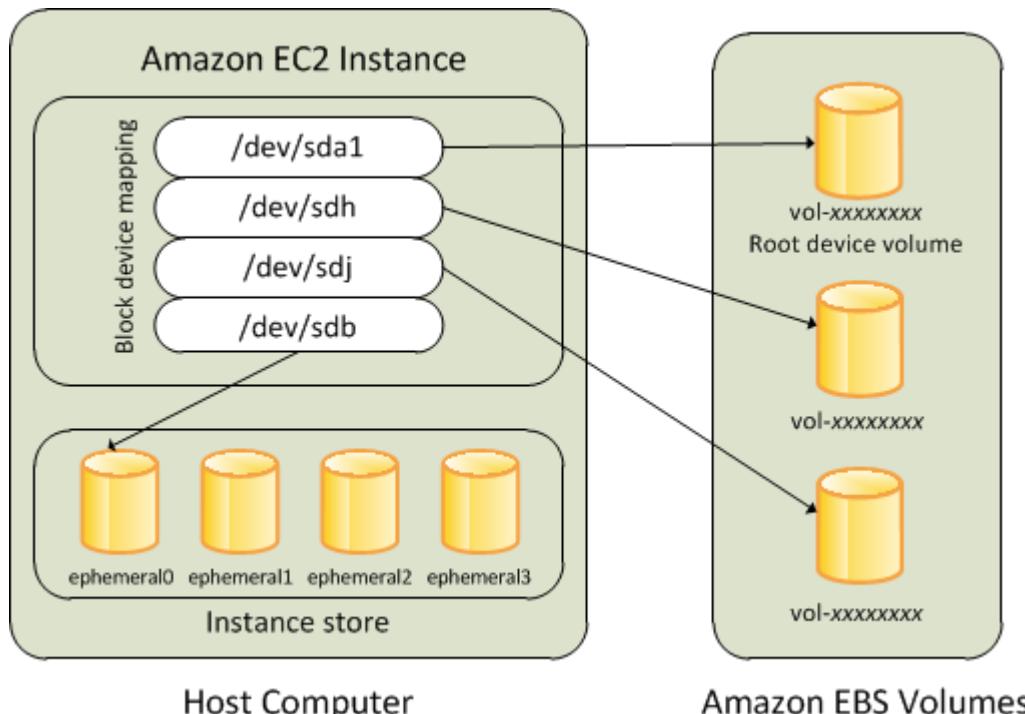
Note

インスタンスが停止されると、インスタンストアボリュームのデータはすべて失われます。

- 起動時のインスタンストア容量によっては、M3 インスタンスが AMI インスタンストアブロックデバイスのマッピングを (起動時に指定されていない限り) 無視します。インスタンスの起動時にインスタンストアボリュームを使用するには、起動する AMI ボリュームに AMI でインスタンストアボリュームがマッピングされていたとしても、起動時にインスタンストアブロックデバイスのマッピングを指定する必要があります。

## ブロックデバイスマッピングの例

この図は、EBS-backed インスタンスのブロックデバイスマッピングの例を示しています。この例では、/dev/sdb を ephemeral0 にマッピングし、2 つの EBS ボリュームを 1 つは /dev/sdh に、もう 1 つは /dev/sdj にマッピングします。また、ルートデバイスピリュームである EBS ボリューム、/dev/sda1 も示しています。



このブロックデバイスマッピングの例は、このトピックのコマンドおよび API の例で使用されています。ブロックデバイスマッピングを作成するコマンドおよび API の例については、「[AMI 用のブロックデバイスマッピングの指定 \(p. 1103\)](#)」および「[インスタンスの起動時にブロックデバイスマッピングを更新する \(p. 1106\)](#)」を参照してください。

## オペレーティングシステムでデバイスを使用できるようにする方法

Amazon EC2 では、ブロックデバイスの記述に、/dev/sdh や xvdfh などのデバイス名が使われます。また、Amazon EC2 では、EC2 インスタンスにアタッチするブロックデバイスを、ブロックデバイスマッピングで指定します。ストレージデバイスにアクセスするには、インスタンスにアタッチしたブロックデバイスが、オペレーティングシステムによって事前にマウントされていなければなりません。ブロックデバイスがインスタンスからデタッチされると、そのデバイスはオペレーティングシステムによってアンマウントされ、ストレージデバイスにアクセスできなくなります。

Linux インスタンスの場合、ブロックデバイスマッピングで指定されたデバイス名は、インスタンスの初回起動時に対応するブロックデバイスにマッピングされます。デフォルトでフォーマットおよびマウントされるインスタンスストアボリュームは、インスタンスタイプによって決まります。インスタンスタイプで使用できるインスタンスストアボリューム数を超えていない場合は、起動時に追加のインスタンスストアボリュームをマウントできます。詳細については、「[Amazon EC2 インスタンスストア \(p. 1076\)](#)」を参照してください。ボリュームがフォーマットおよびマウントされるときに使用されるデバイスは、インスタンスのブロックデバイスドライバーによって決まります。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。

## AMI ブロックデバイスマッピング

各 AMI にブロックデバイスマッピングがあります。このブロックデバイスマッピングは、AMI からのインスタンスの起動時にそのインスタンスにアタッチするブロックデバイスを指定します。Amazon が提供する AMI には、ルートデバイスのみが含まれます。追加のブロックデバイスを AMI に追加するには、独自の AMI を作成する必要があります。

### コンテンツ

- [AMI 用のブロックデバイスマッピングの指定 \(p. 1103\)](#)
- [AMI ブロックデバイスマッピングの EBS ボリュームの表示 \(p. 1105\)](#)

## AMI 用のブロックデバイスマッピングの指定

AMI を作成する場合に、ルートボリュームに加えて、ボリュームを指定するには、2つの方法があります。インスタンスから AMI を作成する前に、実行中のインスタンスにすでにボリュームをアタッチしている場合、AMI のブロックデバイスマッピングにそれらの同じボリュームが含まれます。EBS ボリュームの場合、既存のデータが新しいスナップショットに保存され、それがブロックデバイスマッピングで指定される新しいスナップショットになります。インスタンスストアボリュームの場合、データは維持されません。

EBS-backed AMI の場合、ブロックデバイスマッピングを使用して、EBS ボリュームとインスタンスストアボリュームを追加できます。instance store-backed AMI の場合、イメージの登録時にイメージマニフェストファイルでブロックデバイスマッピングエントリを変更して、インスタンスストアボリュームのみを追加できます。

### Note

M3 インスタンスの場合、インスタンスのブロックデバイスマッピングで起動時にインスタンスストアボリュームを指定する必要があります。AMI のブロックデバイスマッピングで指定したインスタンスストアボリュームは、インスタンスブロックデバイスマッピングの一部として指定されていない場合、M3 インスタンスを起動した際に無視される可能性があります。

## コンソールを使用してボリュームを AMI に追加するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択し、[Actions]、[Image]、[Create Image] の順に選択します。
4. [Create Image] ダイアログボックスで、[Add New Volume] を選択します。
5. [Type] リストからボリュームタイプを選択し、[Device] リストからデバイス名を選択します。ボリュームの場合、オプションで、スナップショット、ボリュームサイズ、および EBS ボリュームタイプを指定できます。
6. [Create Image] を選択します。

## コマンドラインを使用して AMI にボリュームを追加するには

EBS-Backed AMI のプロックデバイスマッピングを指定するには、[create-image](#) AWS CLI コマンドを使用します。Instance Store-Backed AMI のプロックデバイスマッピングを指定するには、[register-image](#) AWS CLI コマンドを使用します。

--block-device-mappings パラメータを使用してプロックデバイスマッピングを指定します。JSON でエンコードされた引数は、コマンドラインで直接指定することも、ファイルを参照して指定することもできます。

```
--block-device-mappings [mapping, ...]  
--block-device-mappings [file://mapping.json]
```

インスタンストアボリュームを追加するには、次のマッピングを使用します。

```
{  
    "DeviceName": "/dev/sdf",  
    "VirtualName": "ephemeral0"  
}
```

空の 100 GiB gp2 ボリュームを追加するには、次のマッピングを使用します。

```
{  
    "DeviceName": "/dev/sdg",  
    "Ebs": {  
        "VolumeSize": 100  
    }  
}
```

スナップショットに基づいた EBS ボリュームを追加するには、次のマッピングを使用します。

```
{  
    "DeviceName": "/dev/sdh",  
    "Ebs": {  
        "SnapshotId": "snap-xxxxxxxx"  
    }  
}
```

デバイスのマッピングを省略するには、次のマッピングを使用します。

```
{  
    "DeviceName": "/dev/sdj",  
    "NoDevice": ""  
}
```

}

または、次のコマンド (AWS Tools for Windows PowerShell) で `-BlockDeviceMapping` パラメータを使用することもできます。

- [New-EC2Image](#)
- [Register-EC2Image](#)

## AMI ブロックデバイスマッピングの EBS ボリュームの表示

AMI のブロックデバイスマッピングの EBS ボリュームを簡単に列挙できます。

コンソールを使用して AMI の EBS ボリュームを表示するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [AMIs] を選択します。
3. [Filter] リストから [EBS images] を選択して、EBS-Backed AMI のリストを取得します。
4. ご希望の AMI を選択し、[Details] タブを確認します。少なくとも、ルートデバイスでは次の情報を使用できます。
  - ルートデバイスタイプ (ebs)
  - [Root Device Name] (例: /dev/sda1)
  - [Block Devices] (例: /dev/sda1=snap-1234567890abcdef0:8:true)

AMI がブロックデバイスマッピングを使用して追加の EBS ボリュームで作成された場合、[Block Devices] フィールドには、その追加の EBS ボリュームのマッピングも表示されます。(この画面には、インスタンスストアボリュームは表示されません)。

コマンドラインを使用して AMI の EBS ボリュームを表示するには

[describe-images](#) (AWS CLI) コマンドまたは [Get-EC2Image](#) (AWS Tools for Windows PowerShell) コマンドを使用して、AMI のブロックデバイスマッピング内の EBS ボリュームを列挙します。

## インスタンスブロックデバイスマッピング

デフォルトでは、起動するインスタンスには、そのインスタンスを起動した AMI のブロックデバイスマッピングで指定されたストレージデバイスが含まれます。インスタンスを起動するときに、インスタンスのブロックデバイスマッピングへの変更を指定できます。この変更は AMI のブロックデバイスマッピングを上書きするか、このブロックデバイスマッピングに統合されます。

### 制限

- ルートボリュームの場合、変更できるのはボリュームサイズ、ボリュームタイプ、および [合わせて削除] フラグのみです。
- EBS ボリュームを変更する場合、そのサイズを小さくすることはできません。そのため、指定するスナップショットのサイズは、AMI のブロックデバイスマッピングで指定されたスナップショットのサイズ以上であることが必要です。

### コンテンツ

- [インスタンスの起動時にブロックデバイスマッピングを更新する \(p. 1106\)](#)
- [実行中のインスタンスのブロックデバイスマッピングの更新 \(p. 1107\)](#)
- [インスタンスブロックデバイスマッピングの EBS ボリュームの表示 \(p. 1107\)](#)

- インスタンスストアボリュームのインスタンスロックデバイスマッピングの表示 (p. 1108)

## インスタンスの起動時にロックデバイスマッピングを更新する

インスタンスの起動時に、EBS ボリュームとインスタンスストアボリュームをインスタンスに追加できます。インスタンスのロックデバイスマッピングを更新しても、そのインスタンスが起動された AMI のロックデバイスマッピングは完全には変更されないことに注意してください。

コンソールを使用してボリュームをインスタンスに追加するには

1. Amazon EC2 コンソールを開きます。
2. ダッシュボードから、[Launch Instance] を選択します。
3. [Choose an Amazon Machine Image (AMI)] ページで、使用する AMI を選択し、[Select] を選択します。
4. ウィザードにしたがって [Choose an Instance Type] ページと [Configure Instance Details] ページを設定します。
5. [Add Storage] ページで、以下のようにルートボリューム、EBS ボリューム、およびインスタンスストアボリュームを変更できます。
  - ルートボリュームのサイズを変更するには、[Type] 列で [Root] ボリュームを見つけて、[Size] フィールドを変更します。
  - インスタンスの起動に使用された AMI のロックデバイスマッピングで指定された EBS ボリュームを削除するには、ボリュームを見つけて、[Delete] アイコンをクリックします。
  - EBS ボリュームを追加するには、[新しいボリュームの追加] を選択し、[Type] リストから [EBS] を選択して、各フィールド ([Device]、[Snapshot] など) に入力します。
  - インスタンスの起動に使用された AMI のロックデバイスマッピングで指定されたインスタンスストアボリュームを削除するには、ボリュームを見つけて、[Delete] アイコンを選択します。
  - インスタンスストアボリュームを追加するには、[新しいボリュームの追加] を選択し、[Type] リストから [インスタンスストア] を選択して、[Device] からデバイス名を選択します。
6. ウィザードの残りのページを完了した後、[起動] を選択します。

コマンドラインを使用してボリュームをインスタンスに追加するには

`run-instances` AWS CLI コマンドを使用して、インスタンスのロックデバイスマッピングを指定します。

次のパラメータを使用してロックデバイスマッピングを指定します。

```
--block-device-mappings [mapping, ...]
```

たとえば、EBS-backed AMI が、次のロックデバイスマッピングを指定するとします。

- `/dev/sdb=ephemeral0`
- `/dev/sdh=snap-1234567890abcdef0`
- `/dev/sdj=:100`

この AMI から起動したインスタンスに `/dev/sdj` がアタッチされないようにするには、次のマッピングを使用します。

```
{
    "DeviceName": "/dev/sdj",
    "NoDevice": ""}
```

}

/dev/sdh のサイズを 300 GiB に増やすには、次のマッピングを指定します。デバイス名を指定することでボリュームを特定できるため、/dev/sdh のスナップショット ID を指定する必要はありません。

```
{  
    "DeviceName": "/dev/sdh",  
    "Ebs": {  
        "VolumeSize": 300  
    }  
}
```

追加インスタンスストアボリューム /dev/sdc をアタッチするには、次のマッピングを指定します。インスタンスタイプが複数のインスタンスストアボリュームをサポートしていない場合、このマッピングは効果がありません。

```
{  
    "DeviceName": "/dev/sdc",  
    "VirtualName": "ephemeral1"  
}
```

または、[New-EC2Instance](#) コマンド (AWS Tools for Windows PowerShell) で `-BlockDeviceMapping` パラメータを使用することもできます。

## 実行中のインスタンスのロックデバイスマッピングの更新

次の [modify-instance-attribute](#) AWS CLI コマンドを使用して、実行中のインスタンスのロックデバイスマッピングを更新できます。この属性を変更する前に、インスタンスを停止する必要はありません。

```
aws ec2 modify-instance-attribute --instance-id i-1a2b3c4d --block-device-mappings file://mapping.json
```

たとえば、インスタンスの削除時にルートボリュームを保持するには、`mapping.json` で以下を指定します。

```
[  
    {  
        "DeviceName": "/dev/sda1",  
        "Ebs": {  
            "DeleteOnTermination": false  
        }  
    }  
]
```

または、[Edit-EC2InstanceAttribute](#) コマンド (AWS Tools for Windows PowerShell) で `-BlockDeviceMapping` パラメータを使用することもできます。

## インスタンスロックデバイスマッピングの EBS ボリュームの表示

インスタンスにマッピングされた EBS ボリュームを簡単に列挙できます。

### Note

2009 年 10 月 31 日 API のリリース以前に起動されたインスタンスについては、AWS では、ロックデバイスマッピングを表示できません。AWS がロックデバイスマッピングを表示できるようにするには、ボリュームをデタッチしてから再アタッチする必要があります。

## コンソールを使用してインスタンスの EBS ボリュームを表示するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. 検索バーに「Root Device Type」と入力し、[EBS] を選択します。これにより、EBS-backed インスタンスのリストが表示されます。
4. 目的のインスタンスを選択し、[Description] タブに表示された詳細を確認します。少なくとも、ルートデバイスでは次の情報を使用できます。
  - ルートデバイスタイプ (ebs)
  - [Root device] (例: /dev/sda1)
  - [Block devices] (例: /dev/sda1, /dev/sdh, /dev/sdj)

インスタンスがロックデバイスマッピングを使用して追加の EBS ボリュームで起動された場合、[Block Devices] フィールドには、その追加のボリュームと、ルートデバイスが表示されます。(このダイアログボックスには、インスタンスストアボリュームが表示されないことに注意してください)。

|                         |                       |
|-------------------------|-----------------------|
| <b>Root device type</b> | ebs                   |
| <b>Root device</b>      | /dev/sda1             |
| <b>Block devices</b>    | /dev/sda1<br>/dev/sdf |

5. ブロックデバイスの追加情報を表示するには、[ブロックデバイス] の横でそのエントリを選択します。これにより、ブロックデバイスに関する次の情報が表示されます。
  - [EBS ID] (vol-xxxxxxxx)
  - ルートデバイスタイプ (ebs)
  - [アタッチ時刻] (yyyy-mmThh:mm:ss.ssTZD)
  - ブロックデバイスマステータス (attaching, attached, detaching, detached)
  - 終了時に削除 (Yes, No)

## コマンドラインを使用してインスタンスの EBS ボリュームを表示するには

`describe-instances` (AWS CLI) コマンドまたは `Get-EC2Instance` (AWS Tools for Windows PowerShell) コマンドを使用して、インスタンスのブロックデバイスマッピングで EBS ボリュームを列挙します。

## インスタンスストアボリュームのインスタンスロックデバイスマッピングの表示

インスタンスのブロックデバイスマッピングを表示した場合、EBS ボリュームのみが表示され、インスタンスストアボリュームは表示されません。ブロックデバイスマッピングで非 NVMe インスタンスストアボリュームを照会するには、インスタンスマタデータを使用します。NVMe インスタンスストアボリュームは含まれていません。

インスタンスマタデータのすべてのリクエストの基本 URI は `http://169.254.169.254/latest/` です。詳細については、「[インスタンスマタデータとユーザーデータ \(p. 593\)](#)」を参照してください。

まず、実行中にインスタンスに接続します。インスタンスからこのクエリを使用して、そのブロックデバイスマッピングを取得します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/block-device-mapping/
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/
```

レスポンスには、インスタンスのロックデバイスの名前が含まれます。たとえば、instance store – Backed m1.small インスタンスの出力は次のようにになります。

```
ami  
ephemeral0  
root  
swap
```

ami デバイスは、インスタンスによって判断されるルートデバイスです。インスタンストアボリュームの名前は ephemeral[0-23] です。swap デバイスはページファイル用です。EBS ボリュームもマップした場合、そのボリュームは、ebs1、ebs2 のように表示されます。

ロックデバイスマッピングの個別のロックデバイスの詳細を確認するには、ここで示すように、前のクエリにロックデバイスの名前を追加します。

IMDSv2

```
[ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

IMDSv1

```
[ec2-user ~]$ curl http://169.254.169.254/latest/meta-data/block-device-mapping/  
ephemeral0
```

インスタンスタイプは、インスタンスに利用できるインスタンストアボリュームの数を決定します。ロックデバイスマッピングのインスタンストアボリュームの数が、インスタンスに利用できるインスタンストアボリュームの数を超える場合は、追加のボリュームは無視されます。インスタンスにインスタンストアボリュームを表示するには、lsblk コマンドを実行します。各インスタンスタイプがサポートするインスタンストアボリュームの数は、「[インスタンストアボリューム \(p. 1078\)](#)」を参照してください。

# リソースとタグ

Amazon EC2 にはさまざまなリソースが用意されており、それらを作成して利用することができます。これらのリソースには、イメージ、インスタンス、ボリューム、スナップショットなどがあります。リソースを作成すると、リソースに一意のリソース ID が割り当てられます。

一部のリソースには、それらの整理と識別に役立つように、ユーザーが定義できる値で値にタグを付けることができます。

以下のトピックでは、リソースとタグ、およびそれらの使用方法について説明します。

## コンテンツ

- [リソースの場所 \(p. 1110\)](#)
- [リソース ID \(p. 1111\)](#)
- [リソースのリスト表示とフィルタリング \(p. 1116\)](#)
- [Amazon EC2 リソースにタグを付ける \(p. 1120\)](#)
- [Amazon EC2 サービスの制限 \(p. 1130\)](#)
- [Amazon EC2 使用状況レポート \(p. 1132\)](#)

## リソースの場所

リソースには、すべてのリージョン（グローバル）で使用できるものと、そのリソースが存在するリージョンまたはアベイラビリティーゾーンに固有のものがあります。

| リソース               | タイプ            | Description                                                                                                                                                                                                       |
|--------------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AWS アカウント          | グローバル          | すべてのリージョンで同じ AWS アカウントを使用できます。                                                                                                                                                                                    |
| キーペア               | グローバルまたはリージョン別 | Amazon EC2 を使用して作成したキーペアは、そのペアを作成したリージョンに関連付けられます。独自の RSA キーペアを作成し、それを使用するリージョンにアップロードできます。したがって、各リージョンにアップロードすることで、キーペアをグローバルに利用可能にすることができます。<br>詳細については、「 <a href="#">Amazon EC2 のキーペア (p. 899)</a> 」を参照してください。 |
| Amazon EC2 リソース識別子 | リージョン別         | 各リソース識別子 (AMI ID、インスタンス ID、EBS ボリューム ID、EBS スナップショット ID など) はリージョンに固定され、そのリソースを作成したリージョンでのみ使用できます。                                                                                                                |
| ユーザーが指定したリソース名     | リージョン別         | 各リソース名 (セキュリティグループ名前、キーペア名など) はリージョンに固定され、そのリソースを作成したリージョンでのみ使用できます。複数の地域で同じ名前のリソースを作成することはできますが、それぞれが相互に関連付けられることはできません。                                                                                         |
| AMI                | リージョン別         | AMI は、Amazon S3 内でそのファイルが置かれているリージョンに固定されます。AMI は、別のリージョン                                                                                                                                                         |

| リソース            | タイプ         | Description                                                                                                                                                  |
|-----------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |             | にコピーできます。詳細については、「 <a href="#">AMI のコピー (p. 155)</a> 」を参照してください。                                                                                             |
| Elastic IP アドレス | リージョン別      | Elastic IP アドレスはリージョンに固定されており、同じリージョンのインスタンスにのみ関連付けることができます。                                                                                                 |
| セキュリティグループ      | リージョン別      | セキュリティグループはリージョンに固定されており、同じリージョンのインスタンスにのみ割り当てることができます。インスタンスが、セキュリティグループルールを使用するリージョン以外のインスタンスと通信できるようにすることはできません。別のリージョン内のインスタンスからのトラフィックは、WAN 帯域幅とみなされます。 |
| EBS スナップショット    | リージョン別      | EBS スナップショットはリージョンに固定されており、同じリージョン内のボリュームの作成にのみ使用できます。スナップショットは、別のリージョンにコピーできます。詳細については、「 <a href="#">Amazon EBS スナップショットのコピー (p. 977)</a> 」を参照してください。       |
| EBS ボリューム       | アベイラビリティゾーン | Amazon EBS ボリュームはアベイラビリティゾーンに固定されており、同じアベイラビリティゾーンのインスタンスにのみアタッチできます。                                                                                        |
| インスタンス          | アベイラビリティゾーン | インスタンスは、そのインスタンスを起動したアベイラビリティゾーンに固定されています。ただし、インスタンス ID はリージョンに固定されています。                                                                                     |

## リソース ID

リソースが作成されると、各リソースに一意のリソース ID が割り当てられます。リソース ID を使用して、Amazon EC2 コンソールでリソースを見つけることができます。コマンドラインツールまたは Amazon EC2 API を使用して Amazon EC2 を操作している場合、特定のコマンドにはリソース ID が必要になります。たとえば、インスタンスを停止するために `stop-instances` AWS CLI コマンドを使用している場合、コマンドでインスタンス ID を指定する必要があります。

### リソース ID の長さ

リソース ID は、リソース ID (スナップショットの `snap` など) にハイフンと英数字の一意の組み合わせが続く形式です。2016 年 1 月から、Amazon EC2 および Amazon EBS リソースタイプには段階的に長い ID を導入しています。英数字の組み合わせの長さは 8 文字でした。新しい ID は 17 文字形式 (たとえば、インスタンス ID の場合は `i-1234567890abcdef0`) です。

サポートされるリソースタイプにはオプトイン期間と期限日が設定されており、その期間中はリソース ID 形式を選択できます。期限日を過ぎると、リソースにはデフォルトで長い ID 形式が使用されます。特定のリソースタイプについては、期限日を過ぎると長い ID 形式を無効にすることはできなくなります。

リソースタイプごとに異なるオプトイン期間と期限日が設定されています。次の表に、サポートされるリソースタイプとそれぞれのオプトイン期間および期限日をまとめています。

| リソースタイプ                                                 | オプトイン期間 | 期限日              |
|---------------------------------------------------------|---------|------------------|
| <code>instance   snapshot   reservation   volume</code> | 使用不可    | 2016 年 12 月 15 日 |

| リソースタイプ                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | オプトイン期間                        | 期限日             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|-----------------|
| bundle   conversion-task   customer-gateway   dhcp-options   elastic-ip-allocation   elastic-ip-association   export-task   flow-log   image   import-task   internet-gateway   network-acl   network-acl-association   network-interface   network-interface-attachment   prefix-list   route-table   route-table-association   security-group   subnet   subnet-cidr-block-association   vpc   vpc-cidr-block-association   vpc-endpoint   vpc-peering-connection   vpn-connection   vpn-gateway | 2018 年 2 月 9 日～2018 年 6 月 30 日 | 2018 年 6 月 30 日 |

#### オプトイン期間中

オプトイン期間中はいつでも、リソースに対して長い ID 形式を有効にしたり無効にしたりできます。リソースタイプに対して長い ID 形式を有効にすると、新しいリソースはすべて長い ID 形式で作成されます。

##### Note

リソース ID が作成後に変更することはありません。したがって、オプトイン期間中に長い ID の有効化または無効化を実施した場合、既存のリソース ID には影響しません。

AWS アカウントの作成時期によっては、デフォルトで、サポートされるリソースタイプでより長い ID が使用されます。ただし、そのリソースタイプの期限日まではより長い ID を使用しないことを選択できます。詳細については、『Amazon EC2 のよくある質問』の「[長い EC2 および EBS リソース ID](#)」を参照してください。

#### 期限日の経過後

期限日が経過した後は、リソースタイプに対して長い ID を無効にすることはできません。新しいリソースはすべて長い ID で作成されます。

## 長い ID の使用

IAM ユーザーおよび IAM ロールごとに長い ID を有効または無効にすることができます。デフォルトでは、IAM ユーザーまたはロールは root ユーザーと同じ設定になります。

### トピック

- ・ [長い ID の設定の表示 \(p. 1112\)](#)
- ・ [長い ID の設定の変更 \(p. 1114\)](#)

## 長い ID の設定の表示

コンソールとコマンドラインツールを使用して、長い ID をサポートするリソースタイプを表示できます。

## コンソールを使用して長い ID の設定を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 画面の上のナビゲーションバーで、長い ID の設定を表示するリージョンを選択します。
3. ダッシュボードで、[Account Attributes] の [Resource ID length management] を選択します。
4. [Advanced Resource ID Management (詳細リソース ID 管理)] を展開すると、長い ID に対応するリソースタイプとそれぞれの期限日が表示されます。

## コマンドラインを使用して長い ID の設定を表示するには

以下のいずれかのコマンドを使用します。

- [describe-id-format \(AWS CLI\)](#)

```
aws ec2 describe-id-format --region region
```

- [Get-EC2IdFormat \(AWS Tools for Windows PowerShell\)](#)

```
Get-EC2IdFormat -Region region
```

## コマンドラインを使用して特定の IAM ユーザーまたは IAM ロールの長い ID の設定を表示するには

以下のいずれかのコマンドを使用して、リクエストで IAM ユーザー、IAM ロール、またはルートアカウントユーザーの ARN を指定します。

- [describe-identity-id-format \(AWS CLI\)](#)

```
aws ec2 describe-identity-id-format --principal-arn arn-of-iam-principal --region region
```

- [Get-EC2IdentityIdFormat \(AWS Tools for Windows PowerShell\)](#)

```
Get-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Region region
```

## コマンドラインを使用して特定のリージョンに対する長い ID の設定を一括表示するには

[describe-aggregate-id-format](#) AWS CLI コマンドを使用して、リージョン全体について長い ID の設定を一括表示したり、各リソースタイプのすべての ARN に対する長い ID を一括表示したりできます。このコマンドは、特定のリージョンにおいてすべて長い ID が選択されているかどうかを短時間で調査する場合に役立ちます。

```
aws ec2 describe-aggregate-id-format --region region
```

## カスタムの長い ID 形式を明示的に定義したユーザーを特定するには

[describe-principal-id-format](#) AWS CLI コマンドを使用して、長い ID の設定を明示的に指定したルートユーザーおよびすべての IAM ロールと IAM ユーザーについて、長い ID 形式の設定を表示できます。このコマンドは、デフォルトの長い ID 設定をオーバーライドした IAM ユーザーと IAM ロールを識別する場合に役立ちます。

```
aws ec2 describe-principal-id-format --region region
```

## 長い ID の設定の変更

コンソールやコマンドラインツールを使用して、オプトイン期間中のリソースタイプについて長い ID の設定を変更できます。

### Note

このセクションの AWS CLI および AWS Tools for Windows PowerShell コマンドはリージョンごとにのみ使用できます。特に指定がない限り、これらはデフォルトのリージョンに適用されます。他のリージョンの設定を変更するには、コマンドに `region` パラメータを含めます。

コンソールを使用して長い ID の設定を変更するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 画面の上のナビゲーションバーで、長い ID の設定を変更するリージョンを選択します。
3. ダッシュボードで、[Account Attributes] の [Resource ID length management] を選択します。
4. 次のいずれかを行ってください。
  - すべてのリージョンのすべての IAM ユーザーについて、サポートされるリソースタイプに対して長い ID を有効にするには、[Switch to longer IDs (長い ID への切り替え)]、[Yes, switch to longer IDs (はい、長い ID に切り替えます)] の順に選択します。

### Important

IAM ユーザーと IAM ロールでこのアクションを実行するには `ec2:ModifyIdentityIdFormat` アクセス許可が必要です。

- ご利用の IAM ユーザーアカウントにおいて、特定のリソースタイプに対して長い ID の設定を変更するには、[Advanced Resource ID Management (詳細リソース ID 管理)] を展開し、[My IAM Role/User (IAM ロール/ユーザー)] 列で該当するチェックボックスをオンにして長い ID を有効にするか、チェックボックスをオフにして長い ID を無効にします。
- すべての IAM ユーザーにおいて、特定のリソースタイプに対して長い ID の設定を変更するには、[Advanced Resource ID Management (詳細リソース ID 管理)] を展開し、[すべての IAM ロール/ユーザー] 列で該当するチェックボックスをオンにして長い ID を有効にするか、チェックボックスをオフにして長い ID を無効にします。

コマンドラインを使用してご利用の IAM ユーザーアカウントの長い ID の設定を変更するには

以下のいずれかのコマンドを使用します。

### Note

これらのコマンドをルートユーザーとして使用している場合、IAM ユーザーまたはロールでこれらの設定を自分用に明示的に上書きしていかなければ、変更は AWS アカウント全体に適用されます。

- [modify-id-format \(AWS CLI\)](#)

```
aws ec2 modify-id-format --resource resource_type --use-long-ids
```

このコマンドを使用して、サポートされるすべてのリソースタイプに対して長い ID の設定を変更することもできます。これを行うには、`resource_type` パラメータを `all-current` に置き換えます。

```
aws ec2 modify-id-format --resource all-current --use-long-ids
```

### Note

長い ID を無効にするには、`use-long-ids` パラメータを `no-use-long-ids` に置き換えます。

- [Edit-EC2IdFormat \(AWS Tools for Windows PowerShell\)](#)

```
Edit-EC2IdFormat -Resource resource_type -UseLongId boolean
```

このコマンドを使用して、サポートされるすべてのリソースタイプに対して長い ID の設定を変更することもできます。これを行うには、`resource_type` パラメータを `all-current` に置き換えます。

```
Edit-EC2IdFormat -Resource all-current -UseLongId boolean
```

コマンドラインを使用して特定の IAM ユーザーまたは IAM ロールの長い ID の設定を変更するには

以下のいずれかのコマンドを使用して、リクエストで IAM ユーザー、IAM ロール、またはルートユーザーの ARN を指定します。

- [modify-identity-id-format \(AWS CLI\)](#)

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --  
resource resource_type --use-long-ids
```

このコマンドを使用して、サポートされるすべてのリソースタイプに対して長い ID の設定を変更することもできます。これを行うには、`all-current` を `--resource` パラメータに指定します。

```
aws ec2 modify-identity-id-format --principal-arn arn-of-iam-principal --resource all-  
current --use-long-ids
```

#### Note

長い ID を無効にするには、`use-long-ids` パラメータを `no-use-long-ids` に置き換えます。

- [Edit-EC2IdentityIdFormat \(AWS Tools for Windows PowerShell\)](#)

```
Edit-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Resource resource_type -  
UseLongId boolean
```

このコマンドを使用して、サポートされるすべてのリソースタイプに対して長い ID の設定を変更することもできます。これを行うには、`all-current` を `-Resource` パラメータに指定します。

```
Edit-EC2IdentityIdFormat -PrincipalArn arn-of-iam-principal -Resource all-current -  
UseLongId boolean
```

## 長い ID 設定に対するアクセスの制御

デフォルトでは、IAM ユーザーおよびロールは、関連付けられた IAM ポリシーを通じて明示的にアクセス許可が付与されていない限り、以下のアクションを使用するアクセス許可がありません。

- `ec2:DescribeIdFormat`
- `ec2:DescribeIdentityIdFormat`
- `ec2:DescribeAggregateIdFormat`
- `ec2:DescribePrincipalIdFormat`
- `ec2:ModifyIdFormat`

- `ec2:ModifyIdentityIdFormat`

たとえば、IAM ロールは、ポリシーステートメントの "Action": "ec2:\*" エレメントを通じてすべての Amazon EC2 アクションを使用するアクセス許可を持っている場合があります。

IAM ユーザーおよびロールがアカウントの長いリソース ID 設定を自分用またはアカウントの他のユーザー/ロール用に表示または変更できないようにするには、IAM ポリシーに次のステートメントを含めます。

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": [  
                "ec2:ModifyIdFormat",  
                "ec2:DescribeIdFormat",  
                "ec2:ModifyIdentityIdFormat",  
                "ec2:DescribeIdentityIdFormat",  
                "ec2:DescribeAggregateIdFormat",  
                "ec2:DescribePrincipalIdFormat"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

以下のアクションに対するリソースレベルのアクセス権限はサポートしていません。

- `ec2:DescribeIdFormat`
- `ec2:DescribeIdentityIdFormat`
- `ec2:DescribeAggregateIdFormat`
- `ec2:DescribePrincipalIdFormat`
- `ec2:ModifyIdFormat`
- `ec2:ModifyIdentityIdFormat`

## リソースのリスト表示とフィルタリング

Amazon EC2 コンソールを使用して、一部のタイプのリソースのリストを取得できます。対応するコマンドまたは API アクションを使用して、各タイプのリソースのリストを取得できます。リソースが多い場合は、所定の条件に一致するリソースのみが表示されるように結果をフィルタリングすることができます。

### コンテンツ

- [高度な検索 \(p. 1116\)](#)
- [コンソールを使用してリソースをリスト表示する \(p. 1117\)](#)
- [コンソールを使用してリソースをフィルタリングする \(p. 1118\)](#)
- [CLI および API を使用した一覧表示とフィルタリング \(p. 1119\)](#)

## 高度な検索

高度な検索を使用すると、フィルタを組み合わせて正確な検索結果を得ることができます。キーワード、ユーザー定義のタグキー、事前定義のリソース属性でフィルタリングできます。

使用できる検索タイプは次のとおりです。

- キーワードによる検索

キーワードで検索するには、検索ボックスに検索したいキーワードを入力するか貼り付けて、Enter を選択します。たとえば、インスタンス ID を入力すれば、特定のインスタンスを検索することができます。

- フィールドによる検索

リソースに関連付けられたフィールド、タグ、属性で検索することもできます。たとえば、すべての停止状態のインスタンスを見つけるには、

- 検索ボックスで「**Instance State**」という文字列の入力を開始します。入力するにつれて、フィールドの候補リストが表示されます。
- リストから [Instance State] を選択します。
- 値の候補リストから [Stopped] を選択します。
- さらにリストを絞り込むには、検索ボックスを選択して、より多くの検索オプションを指定します。

- 高度な検索

複数のフィルタを組み合わせることで、高度なクエリを作成することができます。たとえば、タグで検索して本稼働用スタックで実行中の Flying Mountain プロジェクトのインスタンスを絞り込み、次に属性で検索して、すべての t2.micro インスタンスか us-west-2a 内にあるすべてのインスタンスか、またはその両方のインスタンスを絞り込みます。

- 逆順検索

特定の値と一致しないリソースを検索することができます。たとえば、すべての削除されていないインスタンスを一覧表示するには、[Instance State] フィールドを、削除した値の前に感嘆符 (!) をつけて検索します。

- 部分検索

フィールドで検索する場合、部分的な文字列を入力すれば、その文字列が含まれたすべてのリソースをフィールドから検索することができます。たとえば、[インスタンスタイプ] を検索し、「t2」と入力すれば、すべての t2.micro、t2.small、t2.medium インスタンスを見つけることができます。

- 正規表現

フィールド内の値を特定のパターンと一致させる場合、正規表現が役立ちます。たとえば、Name タグで検索し、「^s.\*」と入力すると、Name タグが「s」で始まるすべてのインスタンスを見つけることができます。正規表現の検索には、大文字と小文字は区別されません。

正確な検索結果が得られたら、簡単に参照できるようにその URL をブックマークできます。大量のインスタンスがある場合は、フィルタとブックマークを利用すると大幅に時間を節約できます。検索を繰り返し実行する必要はありません。

#### 検索フィルタの結合

通常、同じキーフィールド（たとえば、tag:Name、検索、インスタンスの状態）を持つ複数のフィルタは、OR で自動的に結合されます。これは意図的な処理で、AND で結合すると、ほとんどのフィルタが論理的でなくなることがその理由です。たとえば、Instance State=running AND Instance State=stopped で検索した場合、0 件の検索が返されます。ほとんどの場合、詳細検索を行うには、異なるキーフィールドでは補完的な検索用語を使用します。ここでは、AND ルールが自動的に適用されます。tag: Name:=All values and tag: Instance State=running を検索した場合、これらの両方の条件を満たす検索結果が取得されます。検索結果を調整するには、結果が要件と一致するまで、単に文字列内のフィルタを 1 つずつ削除します。

## コンソールを使用してリソースをリスト表示する

コンソールを使用して、最も一般的に使用される Amazon EC2 リソースタイプを表示できます。その他のリソースを表示するには、コマンドラインインターフェイスまたは API アクションを使用します。

## コンソールを使用して EC2 リソースをリスト表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、リソースに対応するオプションを選択します ([AMIs] や [Instances] など)。

### EC2 Dashboard New

Events New

Tags

Reports

Limits

#### ▼ INSTANCES

Instances

Instance Types

Launch Templates New

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts New

Scheduled Instances

Capacity Reservations

#### ▼ IMAGES

AMIs

Bundle Tasks

#### ▼ ELASTIC BLOCK STORE

Volumes

3. このページには、すべての利用可能リソースが表示されます。

## コンソールを使用してリソースをフィルタリングする

Amazon EC2 コンソールを使用して、最も一般的に使用されるリソースタイプについてフィルタリングとソートを実行できます。たとえば、インスタンスページの検索バーを使用して、タグ、属性、キーワードでインスタンスをソートすることができます。

また、各ページの検索フィールドを使用して、特定の属性または値を持つリソースを検索できます。文字列の一部または複数の文字列に基づいて検索するために、正規表現を使用できます。たとえば、MySGセキュリティグループを使用するすべてのインスタンスを検索するには、検索フィールドに「MySG」と入力します。その結果には、文字列の一部に MySG を含むすべての値が含まれます。たとえば、MySG2 や MySG3 が該当します。結果を MySG だけに絞り込むには、検索フィールドに「\bMySG\b」と入力します。タイプが m1.small または m1.large であるすべてのインスタンスを一覧表示するには、検索フィールドに「m1.small|m1.large」と入力します。

**us-east-1b** アベイラビリティーゾーンでステータスが **available** のボリュームをリスト表示するには

1. ナビゲーションペインの [Volumes] を選択します。
2. 検索ボックスをクリックして、メニューの [アタッチメントのステータス] を選択し、[デタッチ済み] を選択します。(デタッチされたボリュームは、同じアベイラビリティーゾーン内のインスタンスにアタッチすることができます)。
3. 検索ボックスを再びクリックして、[State] を選択し、[Available] を選択します。
4. 検索ボックスを再びクリックして、[アベイラビリティーゾーン] を選択し、[us-east-1b] を選択します。
5. この基準に一致するボリュームが表示されます。

Amazon EBS-Backed のパブリック 64 ビット Linux AMI をリストするには

1. ナビゲーションペインで [AMIs] を選択します。
2. [Filter] ペインの [Filter] リストで、[Public images]、[EBS images]、[Windows] の順にクリックします。
3. 検索フィールドに `x86_64` と入力します。
4. この基準に一致する AMI が表示されます。

## CLI および API を使用した一覧表示とフィルタリング

リソースタイプごとに、そのタイプのリソースの一覧表示に使用する CLI コマンドまたは API リクエストが用意されています。たとえば、`ec2-describe-images` または `DescribeImages` を使用すると、Amazon マシンイメージ (AMI) を一覧表示できます。レスポンスには、すべてのリソースに関する情報が含まれます。

リソースのリストが長くなる場合には、特定の条件と一致するリソースのみが表示されるように結果をフィルタリングすることができます。フィルタ値は複数指定できます。また、複数のフィルタを指定することもできます。たとえば、タイプが `m1.small` または `m1.large` で、インスタンスの削除時に削除するように設定された EBS ボリュームがアタッチされた、すべてのインスタンスをリスト表示することができます。表示結果に含まれるには、インスタンスがすべてのフィルタと一致する必要があります。

フィルタの値には、ワイルドカードを使用することもできます。アスタリスク (\*) は 0 個以上の文字、クエスチョンマーク (?) は 0 文字または 1 文字にマッチングします。

たとえば、フィルタの値として `database` を使用すると、説明が `database` と等しい EBS スナップショットのみ取得することができます。`*database*` と指定した場合、説明に `database` が含まれるすべてのスナップショットが返されます。`database?` を指定した場合、説明が特定のパターン (`database` と等しい、または `database + 1 文字` と等しい) に一致するスナップショットのみ返されます。

疑問符の数によって、結果に含める文字の最大数が決まります。たとえば、`database????` を指定した場合、説明が `database + 最大 4 文字` と等しいスナップショットのみが返されます。5 文字以上 + `database` の説明は検索結果から除外されます。

フィルタの値は大文字と小文字が区別されます。正確な文字列の一致、または部分文字列の一致 (ワイルドカードを使用) のみをサポートしています。リソースの結果リストが長い場合は、正確な文字列フィルタを使用すると応答が速くなることがあります。

検索には、ワイルドカード文字のリテラル値を含めることができます。ただ、文字の前にバックスラッシュを使用してエスケープする必要があります。たとえば、`\*amazon\?\\"` という値では、リテラル文字列 `*amazon?\\"` が検索されます。

Amazon EC2 リソースごとにサポートされるフィルタのリストについては、該当するドキュメントを参照してください。

- AWS CLI については、「[AWS CLI Command Reference](#)」の該当する `describe` コマンドを参照してください。
- Windows PowerShell については、「[AWS Tools for PowerShell Cmdlet Reference](#)」の該当する `Get` コマンドを参照してください。
- Query API については、「[Amazon EC2 API Reference](#)」の該当する `Describe API` アクションを参照してください。

## Amazon EC2 リソースにタグを付ける

インスタンス、イメージ、その他の Amazon EC2 リソースの管理を支援するため、各リソースにはタグという形式で独自のメタデータをオプションで割り当てることができます。ここでは、タグとその作成方法について説明します。

### Warning

タグのキーと値は、多くの異なる API コールから返されます。`DescribeTags` へのアクセスを拒否しても、他の API から返されるタグへのアクセスは自動的に拒否されません。ベストプラクティスとして、機密データをタグに含めないようお勧めします。

### 目次

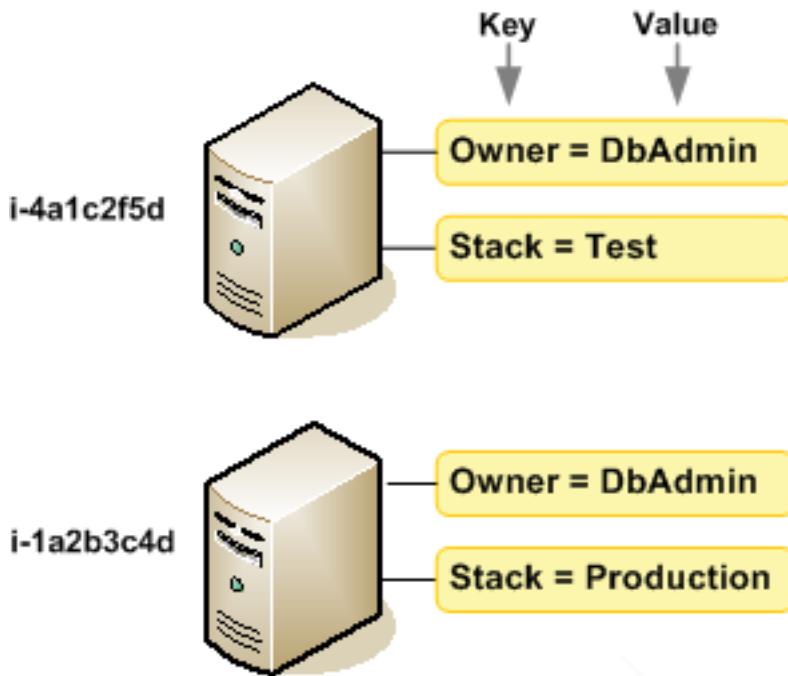
- [タグの基本 \(p. 1120\)](#)
- [リソースにタグを付ける \(p. 1121\)](#)
- [タグの制限 \(p. 1124\)](#)
- [請求用のリソースにタグを付ける \(p. 1125\)](#)
- [コンソールでのタグの処理 \(p. 1125\)](#)
- [CLI または API でのタグの操作 \(p. 1128\)](#)

## タグの基本

タグとは、AWS リソースに付けるラベルです。タグはそれぞれ、1 つのキーとオプションの 1 つの値で構成されており、どちらもお客様側が定義します。

タグを使用すると、AWS リソースを目的、所有者、環境などさまざまな方法で分類することができます。同じ型のリソースが多い場合に役立ちます—割り当てたタグに基づいて特定のリソースをすばやく識別できます。たとえば、アカウントの各インスタンスの所有者とスタックレベルを追跡しやすくするために、Amazon EC2 インスタンスに対して一連のタグを定義できます。

次の図は、タグの機能を示しています。図の中では、インスタンスのそれぞれに 2 つのタグを割り当てています。1 つは `Owner` のキーを使用、もう 1 つは `stack` キーを使用します。各タグには値も関連付けられています。



ニーズを満たす一連のタグキーをリソースタイプごとに考案されることをお勧めします。一貫性のあるタグキーを用いることで、リソースの管理が容易になります。追加したタグに基づいてリソースを検索およびフィルタリングできます。効果的なリソースのタグ付け戦略を実装する方法の詳細については、AWS ホワイトペーパー「[Tagging Best Practices \(タグ付けのベストプラクティス\)](#)」を参照してください。

タグには、Amazon EC2 に関する意味はなく、完全に文字列として解釈されます。また、タグは自動的にリソースに割り当てられます。タグのキーと値は編集でき、タグはリソースからいつでも削除できます。タグの値を空の文字列に設定することはできますが、タグの値を null に設定することはできません。特定のリソースについて既存のタグと同じキーを持つタグを追加した場合、古い値は新しい値によって上書きされます。リソースを削除すると、リソースのタグも削除されます。

AWS マネジメントコンソール、AWS CLI、および Amazon EC2 API を使用して、タグで作業できます。

AWS Identity and Access Management (IAM) を使用している場合は、AWS アカウント内のユーザーに対してタグを作成、編集、削除するアクセス許可を割り当てることができます。詳細については、「[Amazon EC2 の Identity and Access Management \(p. 839\)](#)」を参照してください。

## リソースにタグを付ける

アカウントにすでに存在するほとんどの Amazon EC2 リソースにタグ付けできます。以下の表 (p. 1122) に、タグ付けをサポートするリソースを示します。

Amazon EC2 コンソールを使用している場合、関連するリソース画面の [タグ] タブを使用してリソースにタグを適用するか、[タグ] 画面を使用することができます。一部のリソースの画面では、リソースの作成時にリソースのタグを指定できます。たとえば、Name のキーと指定した値をタグ付けします。ほとんどの場合、リソースの作成後すぐに (リソースの作成時ではなく) コンソールによりタグが適用されます。コンソールではリソースを Name タグに応じて整理できますが、このタグには Amazon EC2 サービスに対する意味論的意味はありません。

Amazon EC2 API、AWS CLI、または AWS SDK を使用している場合、`CreateTags` EC2 API アクションを使用してタグを既存のリソースに適用できます。さらに、リソース作成アクションによっては、リソースの作成時にリソースのタグを指定できます。リソースの作成時にタグを適用できない場合は、リソース作成プロセスがロールバックされます。これにより、リソースがタグ付きで作成されるか、まったく作成されないようになるため、タグ付けされていないリソースが存在することがなくなります。作成時にリソースにタグ付けすることで、リソース作成後にカスタムタグ付けスクリプトを実行する必要がなくなります。

次の表では、タグ付け可能な Amazon EC2 リソースと、Amazon EC2 API、AWS CLI、または AWS SDK を使用した作成時にタグ付け可能なリソースについて説明します。

#### Amazon EC2 リソースのタグ付けのサポート

| リソース                      | タグをサポート    | 作成時のタグ付けをサポート |
|---------------------------|------------|---------------|
| AFI                       | はい         | はい            |
| AMI                       | あり         | なし            |
| バンドルタスク                   | なし         | いいえ           |
| キャパシティーの予約                | あり         | はい            |
| クライアント VPN エンドポイント        | はい         | はい            |
| クライアント VPN ルート            | いいえ        | [No (なし)]     |
| カスタマーゲートウェイ               | あり         | いいえ           |
| Dedicated Host            | [Yes (あり)] | はい            |
| Dedicated Host 予約         | あり         | なし            |
| DHCP オプション                | あり         | なし            |
| EBS スナップショット              | あり         | あり            |
| EBS ボリューム                 | あり         | あり            |
| EC2 フリート                  | あり         | [Yes (あり)]    |
| Egress-only インターネットゲートウェイ | はい         | いいえ           |
| Elastic IP アドレス           | あり         | いいえ           |
| Elastic Graphics アクセラレーター | はい         | なし            |
| インスタンス                    | あり         | あり            |
| インスタンスストアボリューム            | 該当なし       | 該当なし          |
| インターネットゲートウェイ             | はい         | いいえ           |
| IP アドレスプール (BYOIP)        | はい         | いいえ           |
| キーペア                      | はい         | いいえ           |
| 起動テンプレート                  | はい         | はい            |
| 起動テンプレートのバージョン            | いいえ        | いいえ           |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
リソースにタグを付ける

| リソース                                   | タグをサポート    | 作成時のタグ付けをサポート |
|----------------------------------------|------------|---------------|
| ローカルゲートウェイ                             | はい         | いいえ           |
| ローカルゲートウェイルート テーブル                     | はい         | いいえ           |
| ローカルゲートウェイ仮想インターフェイス                   | はい         | いいえ           |
| ローカルゲートウェイ仮想インターフェイスグループ               | はい         | いいえ           |
| ローカルゲートウェイルート テーブル VPC の関連付け           | はい         | いいえ           |
| ローカルゲートウェイルート テーブル仮想インターフェイス グループの関連付け | はい         | いいえ           |
| NAT ゲートウェイ                             | はい         | はい            |
| ネットワーク ACL                             | [Yes (あり)] | [No (なし)]     |
| ネットワークインターフェイス                         | [Yes (あり)] | [No (なし)]     |
| 配置グループ                                 | はい         | いいえ           |
| リザーブドインスタンス                            | [Yes (あり)] | [No (なし)]     |
| リザーブドインスタンス出品                          | [No (なし)]  | [No (なし)]     |
| ルートテーブル                                | あり         | いいえ           |
| スポットフリートのリクエスト                         | はい         | はい            |
| スポットインスタンスリクエスト                        | あり         | なし            |
| セキュリティグループ                             | あり         | なし            |
| サブネット                                  | あり         | いいえ           |
| Traffic Mirror フィルタ                    | はい         | はい            |
| Traffic Mirror セッション                   | はい         | はい            |
| Traffic Mirror ターゲット                   | はい         | はい            |
| 転送ゲートウェイ                               | あり         | あり            |
| 転送ゲートウェイルートテーブル                        | あり         | あり            |
| 転送ゲートウェイ VPC アタッチメント                   | あり         | あり            |
| 仮想プライベートゲートウェイ                         | あり         | なし            |
| VPC                                    | あり         | なし            |
| VPC エンドポイント                            | はい         | はい            |

| リソース              | タグをサポート | 作成時のタグ付けをサポート |
|-------------------|---------|---------------|
| VPC エンドポイントサービス   | はい      | はい            |
| VPC エンドポイントサービス設定 | はい      | はい            |
| VPC フローログ         | はい      | はい            |
| VPC ピア接続          | はい      | いいえ           |
| VPN 接続            | はい      | いいえ           |

Amazon EC2 コンソールで Amazon EC2 インスタンスの起動ウィザードを使用して、作成時にインスタンスとボリュームにタグを付けることができます。ボリューム画面を使用して作成時に EBS ボリュームにタグを付けたり、スナップショット画面を使用して EBS スナップショットにタグを付けたりすることができます。または、リソースを作成するときに、リソース作成 Amazon EC2 API ([RunInstances](#) など) を使用してタグを適用することもできます。

IAM ポリシーでタグベースのリソースレベルアクセス許可を、作成時のタグ付けをサポートする Amazon EC2 API アクションに適用し、作成時にリソースにタグ付けできるユーザーとグループを細かく制御できます。リソースは、作成時から適切に保護されます。タグはリソースに即座に適用されるため、リソースの使用を制御するタグベースのリソースレベルアクセス権限がただちに有効になります。リソースは、より正確に追跡および報告されます。新しいリソースにタグ付けの使用を適用し、リソースで設定されるタグキーと値を制御できます。

さらに、リソースレベルのアクセス許可を IAM ポリシーの `CreateTags` および `DeleteTags` Amazon EC2 API アクションに適用し、既存のリソースで設定されるタグキーと値を制御することもできます。詳細については、「[例: リソースのタグ付け \(p. 873\)](#)」を参照してください。

請求用のリソースへのタグ付けの詳細については、「AWS Billing and Cost Management ユーザーガイド」の「[コスト配分タグの使用](#)」を参照してください。

## タグの制限

タグには以下のような基本制限があります。

- リソースあたりのタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 – 128 文字 (Unicode) (UTF-8)
- 値の最大長 – 256 文字 (Unicode) (UTF-8)
- EC2 ではタグ内に任意の文字を使用できますが、他のサービスでは制限があります。すべてのサービスで使用できる文字は、UTF-8 で表現できる文字、数字、およびスペースに加えて、`+ - = . _ : / @` です。
- タグのキーと値は大文字と小文字が区別されます。
- `aws:` プレフィックスは AWS 専用として予約されています。タグにこのプレフィックスが付いたタグキーがある場合、タグのキーまたは値を編集、削除することはできません。`aws:` プレフィックスを持つタグは、リソースあたりのタグ数の制限時には計算されません。

タグのみに基づいてリソースを終了、停止、終了することはできません。リソース識別子を指定する必要があります。たとえば、`DeleteMe` というタグキーを使用してタグ付けしたスナップショットを削除するには、`DeleteSnapshots` のようなスナップショットのリソース識別子を指定して `snap-1234567890abcdef0` アクションを使用する必要があります。

パブリックリソースまたは共有リソースにタグを付けることはできますが、割り当てるタグはタグ付けを行った AWS アカウントだけが使用できるものであり、リソースを共有するその他のアカウントは使用できません。

すべてのリソースにタグ付けすることはできません。詳細については、「[Amazon EC2 リソースのタグ付けのサポート \(p. 1122\)](#)」を参照してください。

## 請求用のリソースにタグを付ける

タグを使用して AWS 請求書を整理し、自分のコスト構造を反映することができます。そのためには、AWS アカウントにサインアップして、タグキー値が含まれた AWS アカウントの請求書を取得する必要があります。タグによるコスト配分レポートの設定の詳細については、『AWS Billing and Cost Management ユーザーガイド』の「[毎月のコスト配分レポート](#)」を参照してください。リソースを組み合わせたコストを確認するには、同じタグキー値を持つリソースに基づいて、請求情報を整理します。たとえば、複数のリソースに特定のアプリケーション名のタグを付け、請求情報を整理することで、複数のサービスを利用しているアプリケーションの合計コストを確認することができます。詳細については、AWS Billing and Cost Management ユーザーガイドの「[コスト配分タグの使用](#)」を参照してください。

### Note

レポートを有効にすると、約 24 時間後に、今月のデータを表示できるようになります。

コスト割り当てタグは、どのリソースがコストに貢献しているかを示すことができますが、リソースを削除または非アクティブ化にしてもコストは必ずしも削減されるわけではありません。たとえば、元のデータを含むスナップショットが削除された場合でも、別のスナップショットによって参照されるスナップショットデータは維持されます。詳細については、『AWS Billing and Cost Management ユーザーガイド』の「[Amazon Elastic Block Store のボリュームおよびスナップショット](#)」を参照してください。

### Note

タグされている Elastic IP アドレスは、コスト配分レポートには表示されません。

## コンソールでのタグの処理

Amazon EC2 コンソールを使用して、同じリージョン内のすべての Amazon EC2 リソースで使用されているタグを表示できます。タグは、リソース別およびリソースタイプ別で表示し、指定したタグに関連付けられている各リソースタイプの項目数を表示することができます。また、Amazon EC2 コンソールを使用して、同時に 1 つまたは複数のリソースについてタグの適用またはタグの削除を行うことができます。

リソースをリスト表示するときのフィルタの使い方については、「[リソースのリスト表示とフィルタリング \(p. 1116\)](#)」を参照してください。

使いやすさと最適な結果を実現するために、AWS マネジメントコンソールで Tag Editor を使用してください。統一された方法で一元的にタグを作成および管理できます。オプションングループの操作方法の詳細については、『AWS マネジメントコンソールのご利用開始』の「[Tag Editor の使用](#)」を参照してください。

### 目次

- [タグを表示する \(p. 1126\)](#)
- [個々のリソースでのタグの追加と削除 \(p. 1126\)](#)
- [リソースグループへのタグの追加と削除 \(p. 1127\)](#)
- [インスタンスを起動するときにタグを追加する \(p. 1127\)](#)
- [タグを使用してリソースリストをフィルタリングする \(p. 1128\)](#)

## タグを表示する

Amazon EC2 コンソールでは、2 種類の方法でタグを表示できます。個々のリソースまたはすべてのリソースについて、タグを表示できます。

### 個々のリソースのタグを表示する

Amazon EC2 コンソールでリソース固有のページを選択すると、リソースリストが表示されます。たとえば、ナビゲーションペインの [インスタンス] を選択すると、Amazon EC2 インスタンスリストが表示されます。このようなリスト（インスタンスなど）からリソースを選択し、リソースがタグをサポートしている場合、タグを表示し、管理することができます。ほとんどのリソースページで、詳細ペインの [Tags] タブでタグを表示できます。

リソースリストには、キーが同じタグのすべての値を表示する列を追加できます。この列で、タグを使用してリソースリストの並べ替えやフィルタリングを行うことができます。新しい列をリソースリストに追加し、タグを表示するには、2 つの方法があります。

- [Tags] タブで、[Show Column] を選択します。新しい列がコンソールに追加されます。
- [Show/Hide Columns] (歯車型のアイコン) を選択し、[Show/Hide Columns] ダイアログボックスの [Your Tag Keys] のタグキーを選択します。

### すべてのリソースのタグを表示する

Amazon EC2 コンソールのナビゲーションペインの [Tags] を選択して、すべてのリソースのタグを表示できます。次の図は、リソースタイプごとに使用中のすべてのタグが表示された [Tags] ペインです。

The screenshot shows a table titled "Tags" with the following data:

| Tag Key    | Tag Value | Total              | Instances | AMIs | Volumes |
|------------|-----------|--------------------|-----------|------|---------|
| Manage Tag | Name      | DNS Server         | 1         | 1    | 0       |
| Manage Tag | Owner     | TeamB              | 2         | 0    | 2       |
| Manage Tag | Owner     | TeamA              | 2         | 0    | 2       |
| Manage Tag | Purpose   | Project2           | 1         | 0    | 0       |
| Manage Tag | Purpose   | Logs               | 1         | 0    | 1       |
| Manage Tag | Purpose   | Network Management | 1         | 1    | 0       |
| Manage Tag | Purpose   | Project1           | 2         | 0    | 2       |

## 個々のリソースでのタグの追加と削除

リソースのページから、個々のリソースのタグを直接管理できます。

### 個々のリソースにタグを追加するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、ニーズに合ったリージョンを選択します。一部の Amazon EC2 リソースはリージョン間で共有できるため、この選択は重要です。詳細については、「[リソースの場所 \(p. 1110\)](#)」を参照してください。
3. ナビゲーションペインで、リソースタイプを選択します ([Instances] など)。
4. リソースリストからリソースを選択し、[Tags]、[Add/Edit Tags] を選択します。
5. [Add/Edit Tags] ダイアログボックスで、各タグのキーと値を指定し、[Save] を選択します。

### 個々のリソースからタグを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、ニーズに合ったリージョンを選択します。一部の Amazon EC2 リソースはリージョン間で共有できるため、この選択は重要です。詳細については、「[リソースの場所 \(p. 1110\)](#)」を参照してください。
3. ナビゲーションペインで、リソースタイプを選択します ([Instances] など)。
4. リソースリストからリソースを選択し、[Tags] を選択します。
5. [Add/Edit Tags] を選択し、タグの [Delete] アイコンを選択して、[Save] を選択します。

## リソースグループへのタグの追加と削除

### リソースグループにタグを追加するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、ニーズに合ったリージョンを選択します。一部の Amazon EC2 リソースはリージョン間で共有できるため、この選択は重要です。詳細については、「[リソースの場所 \(p. 1110\)](#)」を参照してください。
3. ナビゲーションペインで、[Tags] を選択します。
4. コンテンツペインの上部にある [Manage Tags] を選択します。
5. [Filter] で、タグを追加するリソースのタイプ (インスタンスなど) を選択します。
6. リソースリストで、タグを追加する各リソースの横にあるチェックボックスをオンにします。
7. [Add Tag] の [Key] と [Value] に、タグキーと値を入力し、[Add Tag] を選択します。

#### Note

既存のタグとタグキーが同じ新しいタグを追加すると、既存のタグは新しいタグで上書きされます。

### リソースグループからタグを削除するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、ニーズに合ったリージョンを選択します。一部の Amazon EC2 リソースはリージョン間で共有できるため、この選択は重要です。詳細については、「[リソースの場所 \(p. 1110\)](#)」を参照してください。
3. ナビゲーションペインで、[Tags]、[Manage Tags] を選択します。
4. 使用中のタグを表示するには、[Show/Hide Columns] (歯車型のアイコン) を選択し、[Show/Hide Columns] ダイアログボックスで、表示するタグキーを選択して [Close] を選択します。
5. [Filter] で、タグを削除するリソースのタイプ (インスタンスなど) を選択します。
6. リソースリストで、タグを削除する各リソースの横にあるチェックボックスをオンにします。
7. [Remove Tag] の [Key] に、タグの名前を入力し、[Remove Tag] を選択します。

## インスタンスを起動するときにタグを追加する

### [Launch Wizard] を使用してタグを追加するには

1. ナビゲーションバーで、インスタンスを起動するリージョンを選択します。一部の Amazon EC2 リソースはリージョン間で共有できるため、この選択は重要です。ニーズに合ったリージョンを選択します。詳細については、「[リソースの場所 \(p. 1110\)](#)」を参照してください。
2. [インスタンスの作成] を選択します。

3. [Choose an Amazon Machine Image (AMI)] ページには、Amazon マシンイメージ (AMI) と呼ばれる基本設定リストが表示されます。使用する AMI を選択し、[Select] を選択します。AMI の選択の詳細については、「[Linux AMI の検索 \(p. 100\)](#)」を参照してください。
4. [Configure Instance Details] ページで、必要に応じてインスタンスの設定を行い、[Next: Add Storage] を選択します。
5. [Add Storage] ページで、インスタンスに追加のストレージボリュームを指定できます。完了したら、[Next: Add Tags] を選択します。
6. [Add Tags] ページで、インスタンス、ボリューム、またはその両方のタグを指定します。インスタンスに複数のタグを追加するには、[Add another tag] を選択します。完了したら、[次の手順: セキュリティグループの設定] を選択します。
7. [Configure Security Group] ページで、所有する既存のセキュリティグループから選択するか、wizard で新しいセキュリティグループを作成します。完了したら、[Review and Launch] を選択します。
8. 設定を確認します。選択した内容でよければ、[Launch] を選択します。既存のキーペアを選択するか、新しいキーペアを作成し、確認のチェックボックスを選択して、[Launch Instances] を選択します。

## タグを使用してリソースリストをフィルタリングする

1つまたは複数のタグキーとタグ値に基づいて、リソースリストをフィルタリングできます。

タグを使用してリソースリストをフィルタリングするには

1. 次の手順でタグの列を表示します。
  - a. リソースを選択します。
  - b. 詳細ペインで、[Tags] を選択します。
  - c. リスト内のタグを選択し、[Show Column] を選択します。
2. フィルタリストを表示するタグの列の右上にあるフィルタアイコンを選択します。
3. タグ値を選択し、[Apply Filter] を選択して結果リストをフィルタリングします。

Note

フィルタについての詳細は、「[リソースのリスト表示とフィルタリング \(p. 1116\)](#)」を参照してください。

## CLI または API でのタグの操作

リソースのタグの追加、更新、リスト表示、および削除には、次を使用します。対応するドキュメントに例が記載されています。

| タスク                    | AWS CLI                       | AWS Tools for Windows PowerShell | API アクション                    |
|------------------------|-------------------------------|----------------------------------|------------------------------|
| 1 つ以上のタグを追加、または上書きします。 | <a href="#">create-tags</a>   | <a href="#">New-EC2Tag</a>       | <a href="#">CreateTags</a>   |
| 1 つ以上のタグを削除します。        | <a href="#">delete-tags</a>   | <a href="#">Remove-EC2Tag</a>    | <a href="#">DeleteTags</a>   |
| 1 つ以上のタグを記述します。        | <a href="#">describe-tags</a> | <a href="#">Get-EC2Tag</a>       | <a href="#">DescribeTags</a> |

タグに応じて、リソースのリストをフィルタリングすることもできます。次の例では、[describe-instances](#) コマンドでタグを使用して、インスタンスをフィルタリングする方法を示しています。

Note

コマンドラインで JSON 形式のパラメータを入力する方法はオペレーティングシステムによって異なります。Linux、MacOS、または Unix と Windows PowerShell では、一重引用符 ('') を使用して JSON データ構造を囲みます。Windows コマンドラインでコマンドを使用するときは一重引用符を省略します。詳細については、「AWS Command Line Interface のパラメータ値の指定」を参照してください。

例 1: 特定のタグキーでインスタンスの詳細を示します。

次のコマンドは、タグの値にかかわらず Stack タグでインスタンスの詳細を示します。

```
aws ec2 describe-instances --filters Name=tag-key,Values=Stack
```

例 2: 特定のタグでインスタンスの詳細を示します。

次のコマンドは、Stack=production タグでインスタンスの詳細を示します。

```
aws ec2 describe-instances --filters Name=tag:Stack,Values=production
```

例 3: 特定のタグの値でインスタンスの詳細を示します。

次のコマンドは、タグキーにかかわらず値 production を持つタグでインスタンスの詳細を示します。

```
aws ec2 describe-instances --filters Name=tag-value,Values=production
```

一部のリソース作成アクションでは、リソースの作成時にタグを指定できます。以下のアクションでは、作成時のタグ付けがサポートされます。

| タスク                    | AWS CLI                       | AWS Tools for Windows PowerShell | API アクション                    |
|------------------------|-------------------------------|----------------------------------|------------------------------|
| 1 つまたは複数のインスタンスを起動します。 | <a href="#">run-instances</a> | <a href="#">New-EC2Instance</a>  | <a href="#">RunInstances</a> |
| EBS ボリュームを作成します。       | <a href="#">create-volume</a> | <a href="#">New-EC2Volume</a>    | <a href="#">CreateVolume</a> |

次の例は、リソースの作成時にタグを適用する方法を示しています。

例 4: インスタンスを起動し、インスタンスおよびボリュームにタグを適用する

次のコマンドは、インスタンスを起動し、webserver のキーと production の値を持つタグをインスタンスに適用します。さらに、cost-center キーと cc123 の値を持つタグを、作成された EBS ボリューム (この場合はルートボリューム) に適用します。

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-6e7f829e --tag-specifications 'ResourceType=instance,Tags=[{Key=webserver,Value=production}],' 'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

起動時にインスタンスとボリュームの両方に同じタグキーと値を適用できます。次のコマンドは、インスタンスを起動し、cost-center のキーと cc123 の値を持つタグを、作成されたインスタンスとすべての EBS ボリュームに適用します。

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --subnet-id subnet-6e7f829e --tag-
```

```
specifications 'ResourceType=instance,Tags=[{Key=cost-center,Value=cc123}]'  
'ResourceType=volume,Tags=[{Key=cost-center,Value=cc123}]'
```

#### 例 5: ボリュームを作成してタグを適用する

次のコマンドは、ボリュームを作成し、2つのタグ `purpose = production` および `cost-center = cc123` を適用します。

```
aws ec2 create-volume --availability-zone us-east-1a --volume-type gp2 --size 80 --  
tag-specifications 'ResourceType=volume,Tags=[{Key=purpose,Value=production},{Key=cost-  
center,Value=cc123}]'
```

#### 例 6: リソースにタグを追加する

この例では、タグ (`Stack=production`) を指定されたイメージに追加するか、タグキーが `Stack` の AMI 用に既存のタグを上書きします。コマンドが成功した場合、出力は返りません。

```
aws ec2 create-tags --resources ami-78a54011 --tags Key=Stack,Value=production
```

#### 例 7: タグを複数のリソースに追加する

この例では、2つのタグを AMI とインスタンス用に追加 (または上書き) します。一方のタグにはキー (`webserver`) のみ含まれており、値は設定されていません (値を空の文字列に設定)。もう1つのタグはキー (`stack`) と値 (`Production`) で構成されます。コマンドが成功した場合、出力は返りません。

```
aws ec2 create-tags --resources ami-1a2b3c4d i-1234567890abcdef0 --tags  
Key=webserver,Value= Key=stack,Value=Production
```

#### 例 8: 特殊文字のタグを追加する

この例では、タグ (`[Group]=test`) をインスタンスに追加します。角かっこ ([と]) は特殊文字であり、バックスラッシュ (\) でエスケープする必要があります。

```
aws ec2 create-tags --resources i-1234567890abcdef0 --tags Key=\[Group\],Value=test
```

Windows PowerShell を使用している場合は、バックスラッシュ (\) で文字を区切り、それらを二重引用符 ("") で囲み、キーと値の構造全体を一重引用符 ('') で囲みます。

```
aws ec2 create-tags --resources i-1234567890abcdef0 --tags 'Key=\"[Group]\",Value=test'
```

Linux or OS X を使用している場合は、キーと値の構造全体を一重引用符 ('') で囲んでから、特殊文字を含む要素を二重引用符 ("") で囲みます。

```
aws ec2 create-tags --resources i-1234567890abcdef0 --tags 'Key=\"[Group]\",Value=test'
```

## Amazon EC2 サービスの制限

Amazon EC2 にはさまざまなリソースが用意されており、それらを利用することができます。リソースにはイメージ、インスタンス、ボリューム、スナップショットなどがあります。AWS アカウントを作成するときに、リージョンごとにこれらのリソースに対してデフォルトの制限 (クォータとも呼ばれます) が設定されます。たとえば、リージョンで起動できるインスタンスの数に対する制限があります。この制限に

より、米国西部(オレゴン)リージョンでインスタンスを起動した場合、リクエストを行っても、インスタンスの使用数がそのリージョンにおける現在のインスタンスの制限を超えることはありません。

Amazon EC2 コンソールでは、Amazon EC2 および Amazon VPC コンソールで管理されるリソースについて、制限に関する情報が提供されます。これらの制限の多くは、リクエストによって引き上げることができます。提供される制限とに関する情報をを利用して、AWS インフラストラクチャを管理します。制限の引き上げに対するリクエストは、制限の引き上げが必要となる前に計画してください。

他のサービスの制限に関する詳細については、『Amazon ウェブ サービス全般のリファレンス』の「[AWS サービスの制限](#)」を参照してください。

## 現在の制限を表示する

Amazon EC2 コンソールの [EC2 Limits (EC2 の制限)] ページを使用して、Amazon EC2 や Amazon VPC から提供されるリソースに対するリージョンごとの現在の制限を表示できます。

現在の制限を表示するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、リージョンを選択します。



3. ナビゲーションペインで、[Limits] を選択します。
4. リストでリソースを探します。検索フィールドを使用して、リソース名またはリソースグループでリストをフィルタリングできます。[Current Limit (現在の制限)] 列には、アカウントにおけるリソースの現在の最大値が表示されます。

## 制限の引き上げのリクエスト

Amazon EC2 コンソールの [Limits] ページを使用して、Amazon EC2 や Amazon VPC から提供されるリソースに対するリージョンごとの制限の引き上げをリクエストします。

### 制限の拡大をリクエストするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、リージョンを選択します。
3. ナビゲーションペインで、[Limits] を選択します。
4. リストでリソースを選択し、リクエスト制限の引き上げを選択します。
5. 制限の引き上げのフォームにある必須フィールドを入力します。指定した連絡方法を使用して、お客様に応答が返されます。

## ポート 25 を使用した E メール送信に関する制限

デフォルトでは、Amazon EC2 はすべてのインスタンスで SMTP ポート 25 のトラフィックを絞ります。この調整の削除をリクエストできます。詳細については、AWS ナリッジセンターの「[EC2 インスタンスからポート 25 のスロットルを削除する方法を教えてください。](#)」を参照してください。

## Amazon EC2 使用状況レポート

AWS は、EC2 インスタンスの使用状況およびリザーブドインスタンスの使用量を分析できる、Cost Explorerと呼ばれる無料のレポートツールを提供します。

Cost Explorerは、使用量とコストの表を表示するために使用できる無料のツールです。過去 13 か月までのデータを表示でき、また次の 3 か月間にどのくらい使用する可能性があるかを予測します。時間の経過に伴う AWS リソースの使用量パターンを確認して、さらに照会が必要な分野を識別し、コストの把握に役立つ傾向を確認するには、Cost Explorerを使用します。データの時間範囲を指定したり、時間データを日または月ごとに表示することもできます。

以下に、Cost Explorerの使用により回答を得られる、いくつかの質問の例を示します。

- 各インスタンスタイプのインスタンスには、どのくらいのコストがかかりますか？
- 特定の部門でどのくらいのインスタンス時間が使用されていますか？
- 自分のインスタンスの使用は、複数のアベイラビリティゾーンにわたってどのように分散されていますか？
- 自分のインスタンスの使用は、複数の AWS アカウントにわたってどのように分散されていますか？
- 自分は リザーブドインスタンス をどのくらい適切に使用しているのでしょうか？
- リザーブドインスタンス によってコストを削減できているのでしょうか？

### Cost Explorerで Amazon EC2 レポートを表示

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Reports] をクリックし、表示するレポートを選択します。

レポートはCost Explorerで開きます。固定されたフィルタ設定に基づいて、使用量とコストの傾向についての情報を表示する構成済みのビューを提供します。

#### Note

Cost Explorer を使用するには、それをアカウントで有効にする必要があります。詳細については、「[Cost Explorer の有効化](#)」を参照してください。

レポートの保存を含む、Cost Explorerのレポートの使用に関する詳細な情報については、「[Cost Explorerによるコストの分析](#)」を参照してください。

# インスタンスのトラブルシューティング

次のドキュメントは、インスタンスに関する問題を解決するために役立ちます。

## コンテンツ

- インスタンスの起動に関する問題のトラブルシューティング (p. 1133)
- インスタンスへの接続に関するトラブルシューティング (p. 1135)
- インスタンスの停止に関するトラブルシューティング (p. 1143)
- インスタンスの削除 (シャットダウン) のトラブルシューティング (p. 1145)
- ステータスチェックに失敗したインスタンスのトラブルシューティング (p. 1146)
- 到達できないインスタンスのトラブルシューティング (p. 1168)
- 間違ったボリュームで起動する (p. 1171)
- Linux 用 EC2Rescue を使用する (p. 1172)
- 診断割り込みの送信 (上級ユーザーのみ) (p. 1182)

Windows インスタンスの詳細なヘルプについては、Windows インスタンスの Amazon EC2 ユーザーガイドの「[Windows インスタンスのトラブルシューティング](#)」を参照してください。

## インスタンスの起動に関する問題のトラブルシューティング

以下の問題が発生すると、インスタンスを起動できなくなります。

### 起動に関する問題

- インスタンス制限の超過 (p. 1133)
- インスタンス容量の不足 (p. 1134)
- インスタンスがすぐに削除される (p. 1134)

## インスタンス制限の超過

### 説明

新しいインスタンスを起動するか、停止したインスタンスを再起動しようとすると、`InstanceLimitExceeded` エラーが発生する。

### 原因

新しいインスタンスを起動するか、停止したインスタンスを再起動しようとすると `InstanceLimitExceeded` エラーが発生する場合、リージョンで起動できるインスタンス数の制限に達しています。AWS アカウントを作成するときに、リージョンごとに、実行できるインスタンスの数についてデフォルトの制限が設定されます。

## ソリューション

インスタンス制限の引き上げは、リージョンごとにリクエストできます。詳細については、「[Amazon EC2 サービスの制限 \(p. 1130\)](#)」を参照してください。

## インスタンス容量の不足

### 説明

新しいインスタンスを起動するか、停止したインスタンスを再起動しようすると、`InsufficientInstanceCapacity` エラーが発生する。

### 原因

インスタンスを起動したり、停止したインスタンスを再起動したりする際に `InsufficientInstanceCapacity` エラーが発生する場合、現在 AWS にはリクエストに対応するため必要とされる十分なオンデマンドキャパシティーがありません。

## ソリューション

この問題を解決するには、以下の手順を実行します。

- 数分間待ってからリクエストを再度送信します。容量は頻繁に変化します。
- インスタンス数を減らして新しいリクエストを送信します。たとえば、15 インスタンスを起動する 1 つのリクエストを行っている場合、代わりに 5 つのインスタンスに対する 3 つのリクエストを作成するか、1 つのインスタンスに対する 15 のリクエストを作成してみてください。
- インスタンスを起動する場合は、アベイラビリティーゾーンを指定しないで新しいリクエストを送信します。
- インスタンスを起動する場合は、別のインスタンスタイプを使用して新しいリクエストを送信します（これは後でサイズを変更できます）。詳細については、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。
- クラスターープレイスマントグループにインスタンスを起動すると、容量不足エラーが発生する場合があります。詳細については、「[プレイスマントグループのルールと制限 \(p. 794\)](#)」を参照してください。
- オンデマンド キャパシティーの予約を作成してみてください。これにより、いつでも Amazon EC2 容量を予約できます。詳細については、「[オンデマンドキャパシティー予約 \(p. 431\)](#)」を参照してください。
- 長期的なキャパシティーの予約であるリザーブドインスタンスの購入を試みます。詳細については、「[Amazon EC2 リザーブドインスタンス](#)」を参照してください。

## インスタンスがすぐに削除される

### 説明

インスタンスの状態が、再起動直後に `pending` から `terminated` に変わる。

### 原因

インスタンスがすぐに終了する理由を次にいくつか示します。

- EBS ボリューム制限に達した。
- EBS スナップショットが破損している。
- ルート EBS ボリュームは暗号化されていて、復号用の KMS キーにアクセスする権限がない。

- インスタンスを起動するために使用した Instance Store-Backed AMI で、必要な部分 (image.part.xx ファイル)。

## ソリューション

削除された理由は、Amazon EC2 コンソールまたは AWS Command Line Interface を使用して確認できます。

Amazon EC2 コンソールを使用して、削除された理由を確認するには

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
- [説明] タブで、[状態遷移の理由] ラベルの横にある理由を確認します。

AWS Command Line Interface を使用して、削除された理由を確認するには

- `describe-instances` コマンドを使用して、インスタンス ID を指定します。

```
aws ec2 describe-instances --instance-id instance_id
```

- コマンドによって返された JSON レスポンスで、`StateReason` レスポンス要素の値を確認します。

次のコードブロックは `StateReason` レスポンス要素の例を示しています。

```
"StateReason": {  
    "Message": "Client.VolumeLimitExceeded: Volume limit exceeded",  
    "Code": "Server.InternalError"  
},
```

問題に対処するには

確認した削除の理由に応じて、次のアクションの 1 つを実行します。

- 理由が `Client.VolumeLimitExceeded: Volume limit exceeded` である場合、EBS ボリュームの制限に達しました。詳細については、「[インスタンスピリューム数の制限 \(p. 1097\)](#)」を参照してください。Amazon EBS ボリュームの制限の引き上げリクエストを送信するには、AWS サポートセンターの[ケースの作成](#)フォームに入力してください。詳細については、「[Amazon EC2 サービスの制限 \(p. 1130\)](#)」を参照してください。
- 理由が `Client.InternalError: Client error on launch` である場合、通常はルートボリュームが暗号化されていて、復号用の KMS キーにアクセスする許可がないことを示します。必要な KMS キーにアクセスする権限を取得するには、IAM ユーザーに適切な KMS アクセス権限を追加します。詳細については、『AWS Key Management Service Developer Guide』の「[AWS KMS でのキー権限の使用](#)」を参照してください。

## インスタンスへの接続に関するトラブルシューティング

以下では、発生する可能性のある問題、およびインスタンスへの接続時に表示される可能性のあるメッセージを示します。

コンテンツ

- インスタンスへの接続エラー: 接続タイムアウト (p. 1136)
- エラー: キーをロードできません ... Expecting: ANY PRIVATE KEY (p. 1138)
- エラー: ユーザーキーがサーバーによって認識されない (p. 1138)
- エラー: Host key not found, Permission denied (publickey)、または Authentication failed, permission denied (ホストキーが見つかりません、権限の拒否 (publickey)、または認証失敗、権限の拒否) (p. 1140)
- エラー: Unprotected Private Key File (保護されていないプライベートキーファイル) (p. 1141)
- エラー: プライベートキーの先頭は「-----BEGIN RSA PRIVATE KEY-----」、末尾は「-----END RSA PRIVATE KEY-----」にする必要があります (p. 1142)
- エラー: Server refused our key または No supported authentication methods available (サーバーはキーを拒否しましたまたは利用可能なサポートされる認証方法はありません) (p. 1142)
- ブラウザを使用して接続できない (p. 1142)
- インスタンスに対して Ping を実行できない (p. 1143)
- エラー: サーバーによる予期しないネットワーク接続の閉鎖 (p. 1143)

Windows インスタンスの詳細なヘルプについては、Windows インスタンスの Amazon EC2 ユーザーガイドの「[Windows インスタンスのトラブルシューティング](#)」を参照してください。

## インスタンスへの接続エラー: 接続タイムアウト

インスタンスへ接続しようとして、エラーメッセージ Network error: Connection timed out または Error connecting to [instance], reason: -> Connection timed out: connect が表示される場合、次を実行します。

- セキュリティグループルールを調べます。適切なポートのパブリック IPv4 アドレスからのインバウンド トラフィックがセキュリティグループルールで許可されている必要があります。
  1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
  2. ナビゲーションペインで [インスタンス] を選択し、インスタンスを選択します。
  3. コンソールページの下部の [Description] タブで、[Security groups] の横にある [view inbound rules] を選択して、選択されたインスタンスに対して有効なループのリストを表示します。
  4. Linux インスタンスの場合: [ルールの表示] を選択すると、ウィンドウが開いて、トラフィックが許可されるポートが表示されます。ご使用のコンピュータからポート 22 (SSH) へのトラフィックを許可するルールがあることを確認します。

Windows インスタンスの場合: [ルールの表示] を選択すると、ウィンドウが開いて、トラフィックが許可されるポートが表示されます。ご使用のコンピュータからポート 3389 (RDP) へのトラフィックを許可するルールがあることを確認します。

インスタンスを再起動するたびに、新しい IP アドレス (およびホスト名) が割り当てられます。セキュリティグループに、単一の IP アドレスからのインバウンド トラフィックを許可するルールがあれば、コンピュータが企業ネットワークにある場合またはインターネットサービスプロバイダー (ISP) を通じて接続する場合は、このアドレスが静的ではない可能性があります。この場合は、クライアントコンピュータで使用されている IP アドレスの範囲を指定します。ご使用のセキュリティグループに、前の手順で説明したインバウンド トラフィックを許可するルールがない場合は、セキュリティグループにルールを追加します。詳細については、「[インスタンスへのネットワークアクセスの許可 \(p. 897\)](#)」を参照してください。

セキュリティグループのルールの詳細については、『Amazon VPC ユーザーガイド』の「[セキュリティグループのルール](#)」を参照してください。

- サブネットのルートテーブルを確認します。VPC の外部へのすべてのトラフィックを VPC のインターネットゲートウェイに送信するには、ルートが必要です。
  1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで [インスタンス] を選択し、インスタンスを選択します。
  3. [Description] タブで、[VPC ID] および [Subnet ID] の値を書き留めます。
  4. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
  5. ナビゲーションペインで、[Internet Gateways] を選択します。ご使用の VPC にアタッチされているインターネットゲートウェイがあることを確認します。それ外の場合は、[Create Internet Gateway] を選択してインターネットゲートウェイを作成します。インターネットゲートウェイを選択し、[VPC にアタッチ] を選択して指示どおりにインターネットゲートウェイを VPC にアタッチします。
  6. ナビゲーションペインで [Subnets] を選択し、サブネットを選択します。
  7. [Route Table] タブで、送信先として 0.0.0.0/0、ターゲットとして VPC のインターネットゲートウェイが指定されたルートがあることを確認します。IPv6 アドレスを使用してインスタンスに接続する場合は、インターネットゲートウェイを指しているすべての IPv6 トラフィック (::/0) 用のルートがあることを確認します。それ以外の場合は、以下の作業を行います。
    - a. ルートテーブルの ID (rtb-xxxxxxxx) を選択して、ルートテーブルに移動します。
    - b. [Routes] タブで、[Edit routes] を選択します。[Add route] を選択して、0.0.0.0/0 を宛先として追加し、インターネットゲートウェイをターゲットとして使用します。IPv6 の場合は、[Add route] を選択して、::/0 を宛先として追加し、インターネットゲートウェイをターゲットとして使用します。
    - c. [Save routes] を選択します。
- サブネットのネットワークアクセスコントロールリスト (ACL) を確認します。ネットワーク ACL が適切なポートのローカル IP アドレスからのインバウンドおよびアウトバウンドトラフィックを許可する必要があります。デフォルトのネットワーク ACL では、すべてのインバウンドトラフィックとアウトバウンドトラフィックを許可します。
    1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
    2. ナビゲーションペインで [サブネット] を選択し、任意のサブネットを選択します。
    3. [Description (説明)] タブで、[ネットワーク ACL] を探し、その ID (acl-xxxxxxxx) を選択します。
    4. ネットワーク ACL を選択します。[インバウンドルール] で、コンピュータからのトラフィックがルールで許可されていることを確認します。そうでない場合、コンピュータからのトラフィックをブロックしているルールを削除または変更します。
    5. [アウトバウンドルール] で、コンピュータへのトラフィックがルールで許可されていることを確認します。そうでない場合、コンピュータへのトラフィックをブロックしているルールを削除または変更します。
  - ご使用のコンピュータが社内ネットワークに接続されている場合は、社内ファイアウォールで、ご使用のコンピュータのインバウンドおよびアウトバウンドのトラフィックがポート 22 (Linux インスタンスの場合) またはポート 3389 (Windows インスタンスの場合) で許可されているかどうか、ネットワーク管理者に問い合わせてください。

ご使用のコンピュータにファイアウォールが設定されている場合、そのファイアウォールでコンピュータのインバウンドおよびアウトバウンドのトラフィックがポート 22 (Linux インスタンスの場合) またはポート 3389 (Windows インスタンスの場合) で許可されているかどうか確認します。

- インスタンスにパブリック IPv4 アドレスがあることを確認します。そうでない場合は、Elastic IP アドレスをインスタンスに関連付けることができます。詳細については、「[Elastic IP アドレス \(p. 705\)](#)」を参照してください。
- サーバーが過負荷になっている可能性のあるインスタンスの CPU 負荷を確認します。AWS は自動的に Amazon CloudWatch メトリクスおよびインスタンスステータスなどのデータを提供します。これを使用してインスタンスの CPU 負荷を確認でき、必要に応じて、負荷の処理方法を調整できます。詳細については、「[CloudWatch を使用したインスタンスのモニタリング \(p. 642\)](#)」を参照してください。
- 負荷が変化する場合、[Auto Scaling](#) および [Elastic Load Balancing](#) を使用して、インスタンスの増減を自動的に縮小・拡張できます。
- 負荷が常に増加している場合、大きなインスタンスタイプに移動できます。詳細については、「[インスタンスタイプを変更する \(p. 267\)](#)」を参照してください。

IPv6 アドレスを使用してインスタンスに接続するには、以下のことを確認します。

- サブネットはインターネットゲートウェイへの IPv6 トラフィック (`:::/0`) のルートを持つフルートテーブルに関連付けられている必要があります。
- セキュリティグループルールでは、適切なポート (Linux の場合は 22、Windows の場合は 3389) のローカル IPv6 アドレスからの着信トラフィックを許可する必要があります。
- ネットワーク ACL ルールでは、インバウンドおよびアウトバウンドの IPv6 トラフィックを許可する必要があります。
- 古い AMI からインスタンスを起動した場合、DHCPv6 用に設定されていない可能性があります (IPv6 アドレスはネットワークインターフェイスでは自動的に認識されません)。詳細については、「[インスタンスでの IPv6 の設定](#)」(Amazon VPC ユーザーガイド) を参照してください。
- ローカルコンピュータに IPv6 アドレスがあり、IPv6 を使用するように設定されている必要があります。

## エラー: キーをロードできません ... Expecting: ANY PRIVATE KEY

インスタンスに接続しようとして、エラーメッセージ、`unable to load key ... Expecting: ANY PRIVATE KEY` が表示される場合、プライベートキーが保存されているファイルが正しく設定されていません。プライベートキーファイルが `.pem` で終わる場合でも、正しく設定されていない可能性があります。プライベートキーファイルが正しく設定されていない原因として考えられるのは、証明書がないことです。

プライベートキーファイルが正しく設定されていない場合は、以下の手順に従ってエラーを解決する

- 新しいキーペアを作成します。詳細については、「[Amazon EC2 を使用してキーペアを作成する \(p. 901\)](#)」を参照してください。
- 新しいキーペアをインスタンスに追加します。詳細については、「[プライベートキーを紛失した場合の Linux インスタンスへの接続 \(p. 907\)](#)」を参照してください。
- 新しいキーペアを使用してインスタンスに接続します。

## エラー: ユーザーキーがサーバーによって認識されない

SSH を使用してインスタンスに接続している場合

- `ssh -vvv` を使用して、接続中に 3 倍詳細デバッグ情報を取得します。

```
ssh -vvv -i [your key name].pem ec2-user@[public DNS address of your instance].compute-1.amazonaws.com
```

次のサンプル出力は、サーバーが認識しないキーを使用してインスタンスに接続しようとした場合に表示される可能性があります。

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
```

```
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA 9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

## PuTTY を使用してインスタンスに接続している場合

- 秘密キー (.pem) ファイルが PuTTY によって認識される形式 (.ppk) に変換されていることを確認します。プライベートキーの変換の詳細については、「[PuTTY を使用した Windows から Linux インスタンスへの接続 \(p. 520\)](#)」を参照してください。

### Note

PuTTYgen でプライベートキーファイルをロードし、[Generate] ではなく [Save Private Key] を選択します。

- AMI 用の適切なユーザー名で接続していることを確認します。[PuTTY Configuration] ウィンドウの [Host name] ボックスにユーザー名を入力します。
  - Amazon Linux 2 または Amazon Linux AMI の場合は、ユーザー名は `ec2-user` です。
  - CentOS AMI の場合、ユーザー名は `centos` です。
  - Debian AMI の場合は、ユーザー名は `admin` または `root` です。
  - Fedora AMI の場合、ユーザー名は `ec2-user` または `fedora` です。
  - RHEL AMI の場合は、ユーザー名は `ec2-user` または `root` のどちらかです。
  - SUSE AMI の場合は、ユーザー名は `ec2-user` または `root` のどちらかです。
  - Ubuntu AMI の場合は、ユーザー名は `ubuntu` です。
- それ以外の場合で、`ec2-user` および `root` が機能しない場合は、AMI プロバイダーに確認してください。
- 適切なポートへのインバウンドトラフィックを許可しているインバウンドセキュリティグループがあることを確認します。詳細については、「[インスタンスへのネットワークアクセスの許可 \(p. 897\)](#)」を参照してください。

## エラー: Host key not found、Permission denied (publickey)、または Authentication failed, permission denied (ホストキーが見つかりません、権限の拒否) (publickey)、または認証失敗、権限の拒否

SSH を使用してインスタンスに接続し、Host key not found in [directory]、Permission denied (publickey)、または Authentication failed, permission denied のいずれかのエラーが発生した場合は、AMI 用の適切なユーザー名で接続していて、なおかつインスタンス用の適切なプライベートキー (.pem) ファイルを指定していることを確認します。

適切なユーザー名は以下のとおりです。

- Amazon Linux 2 または Amazon Linux AMI の場合は、ユーザー名は ec2-user です。
- CentOS AMI の場合、ユーザー名は centos です。
- Debian AMI の場合は、ユーザー名は admin または root です。
- Fedora AMI の場合、ユーザー名は ec2-user または fedora です。
- RHEL AMI の場合は、ユーザー名は ec2-user または root のどちらかです。
- SUSE AMI の場合は、ユーザー名は ec2-user または root のどちらかです。
- Ubuntu AMI の場合は、ユーザー名は ubuntu です。
- それ以外の場合で、ec2-user および root が機能しない場合は、AMI プロバイダーに確認してください。

たとえば、SSH クライアントを使用して Amazon Linux インスタンスに接続するには、次のコマンドを使用します。

```
ssh -i /path/my-key-pair.pem ec2-user@public-dns-hostname
```

使用しているプライベートキーファイルが、インスタンスの起動時に選択したキーペアに対応していることを確認します。

- <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
- インスタンスを選択します。[Description] タブで、[Key pair name] の値を確認します。
- インスタンスを起動したときにキーペアを指定しなかった場合は、キーペアを確実に指定するために、インスタンスを終了してから新しいインスタンスを起動します。それまで使用していたインスタンスで、キーペアに対する .pem ファイルがもう存在しない場合は、そのキーペアを新しいキーペアで置き換えることができます。詳細については、「[プライベートキーを紛失した場合の Linux インスタンスへの接続 \(p. 907\)](#)」を参照してください。

独自のキーペアを生成した場合は、キージェネレータが RSA キーを作成するように設定されていることを確認します。DSA キーは受け入れられません。

Permission denied (publickey) エラーが表示され、上のいずれも当てはまらない場合 (たとえば、以前は接続できていたなど)、インスタンスのホームディレクトリのアクセス権限が変更された可能性があります。/home/ec2-user/.ssh/authorized\_keys のアクセス権限は、所有者のみに制限する必要があります。

インスタンスのアクセス権限を検証するには

- インスタンスを停止し、ルートボリュームをデタッチします。詳細については、「[インスタンスの停止と起動 \(p. 529\)](#)」および「[インスタンスからの Amazon EBS ボリュームのデタッチ \(p. 967\)](#)」を参照してください。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
エラー: Unprotected Private Key File (保護  
されていないプライベートキーファイル)

- 
2. 同じアベイラビリティーゾーンにある一時インスタンスを現在のインスタンスとして起動し(現在のインスタンスに使用したのと同様または同じ AMI を使用)、ルートボリュームを一時インスタンスにアタッチします。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 952\)](#)」を参照してください。
  3. 一時インスタンスに接続してマウントポイントを作成し、アタッチしたボリュームをマウントします。詳細については、「[Linux で Amazon EBS ボリュームを使用できるようにする \(p. 956\)](#)」を参照してください。
  4. 一時インスタンスから、アタッチされたボリュームの /home/ec2-user/ ディレクトリのアクセス権限をチェックします。必要に応じて、次のようにアクセス権限を調整します。

```
[ec2-user ~]$ chmod 600 mount_point/home/ec2-user/.ssh/authorized_keys
```

```
[ec2-user ~]$ chmod 700 mount_point/home/ec2-user/.ssh
```

```
[ec2-user ~]$ chmod 700 mount_point/home/ec2-user
```

5. ボリュームをアンマウントして一時インスタンスからデタッチし、元のインスタンスに再アタッチします。ルートボリュームに適切なデバイス名を指定したことを確認します(/dev/xvda など)。
6. インスタンスを起動します。一時インスタンスが必要なくなった場合は、終了できます。

## エラー: Unprotected Private Key File (保護されていないプライベートキーファイル)

プライベートキーファイルはその他のすべてのユーザーの読み取りおよび書き込み操作から保護される必要があります。プライベートキーがお客様以外のユーザーによって読み取りまたは書き込みできる場合、SSH はキーを無視し、次の警告メッセージが表示されます。

```
@@@@@@@  
@       WARNING: UNPROTECTED PRIVATE KEY FILE!       @  
@  
Permissions 0777 for '.ssh/my_private_key.pem' are too open.  
It is required that your private key files are NOT accessible by others.  
This private key will be ignored.  
bad permissions: ignore key: .ssh/my_private_key.pem  
Permission denied (publickey).
```

インスタンスへのログインを試みたときに同様のメッセージが表示された場合は、エラーメッセージの最初の行を調べて、インスタンスに対して正しいパブリックキーを使用していることを確認します。上記の例では、プライベートキー .ssh/my\_private\_key.pem をファイル権限 0777 とともに使用します。これにより、任意のユーザーがこのファイルの読み取りまたは書き込みを行うことができます。この権限レベルの安全性は非常に低いので、SSH はこのキーを無視します。エラーを修正するには、次のコマンドを実行し、プライベートキーファイルのパスを置き換えます。

```
[ec2-user ~]$ chmod 0400 .ssh/my_private_key.pem
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
エラー: プライベートキーの先頭は「----BEGIN  
RSA PRIVATE KEY----」、末尾は「----END RSA  
PRIVATE KEY----」にする必要があります

## エラー: プライベートキーの先頭は「----BEGIN RSA PRIVATE KEY----」、末尾は「----END RSA PRIVATE KEY----」にする必要があります

カードパーティーツール(例: ssh-keygen)を使用して RSA キーペアを作成すると、プライベートキーが OpenSSH キー形式で生成されます。インスタンスに接続する際、OpenSSH 形式のプライベートキーを使用してパスワードを復号すると、エラー("Private key must begin with \"----BEGIN RSA PRIVATE KEY----\" and end with \"----END RSA PRIVATE KEY----\"")が発生する場合があります。

エラーを解消するには、プライベートキーは PEM 形式である必要があります。PEM 形式でプライベートキーを作成するには、次のコマンドを使用します。

```
ssh-keygen -m PEM
```

## エラー: Server refused our key または No supported authentication methods available (サーバーはキーを拒否しましたまたは利用可能なサポートされる認証方法はありません)

PuTTY を使用してインスタンスに接続し、[Error: Server refused our key] または [Error: No supported authentication methods available] エラーが発生した場合は、AMI の適切なユーザー名で接続していることを確認します。[PuTTY 設定] ウィンドウの [ユーザー名] にユーザー名を入力します。

適切なユーザー名は以下のとおりです。

- Amazon Linux 2 または Amazon Linux AMI の場合は、ユーザー名は ec2-user です。
- CentOS AMI の場合、ユーザー名は centos です。
- Debian AMI の場合は、ユーザー名は admin または root です。
- Fedora AMI の場合、ユーザー名は ec2-user または fedora です。
- RHEL AMI の場合は、ユーザー名は ec2-user または root のどちらかです。
- SUSE AMI の場合は、ユーザー名は ec2-user または root のどちらかです。
- Ubuntu AMI の場合は、ユーザー名は ubuntu です。
- それ以外の場合で、ec2-user および root が機能しない場合は、AMI プロバイダーに確認してください。

秘密キー (.pem) ファイルが PuTTY によって認識される形式 (.ppk) に正しく変換されていることも確認する必要があります。プライベートキーの変換の詳細については、「[PuTTY を使用した Windows から Linux インスタンスへの接続 \(p. 520\)](#)」を参照してください。

## ブラウザを使用して接続できない

Amazon EC2 コンソールには、Java SSH クライアントを使用してブラウザからインスタンスに直接接続するオプションがあります。ブラウザで NPAPI がサポートされていない場合は、接続すると Chrome での NPAPI の廃止エラーメッセージが表示されます。このメッセージは、別のブラウザを使用することを推奨しています。ただし、これらのブラウザの最新バージョンでも NPAPI をサポートしていないため、それらを使用してインスタンスに接続することはできません。別の方法を選択して、インスタンスに接続する必要があります。

詳細については、以下のリソースを参照してください。

- 全般: [NPAPI Wikipedia の記事](#)
- Chrome: [NPAPI の廃止の記事](#)
- Firefox: [NPAPI の廃止の記事](#)
- Safari: [NPAPI の廃止の記事](#)

## インスタンスに対して Ping を実行できない

ping コマンドは、ICMP トラフィックの一種です。インスタンスに対して Ping を実行できない場合は、インバウンドセキュリティグループのルールで、すべてのソースからの、あるいはコマンドを発行しているコンピュータまたはインスタンスからの Echo Request メッセージについて、ICMP トラフィックが許可されていることを確認します。インスタンスから ping コマンドを発行できない場合は、アウトバウンドセキュリティグループのルールで、すべての宛先への、または Ping の対象であるホストへの Echo Request メッセージについて、ICMP トラフィックが許可されていることを確認します。

## エラー: サーバーによる予期しないネットワーク接続の閉鎖

Putty を使用してインスタンスに接続中に「サーバーによる予期しないネットワーク接続の閉鎖」エラーを受け取った場合、Putty 設定の接続ページでキープアライブを有効化して、切断を回避していることを確認してください。一部のサーバーは、指定された時間内にデータが一切受信されない場合に、クライアントを切断します。キープアライブ間の秒数を 59 秒に設定します。

キープアライブを有効後にも問題が依然として発生する場合には、Putty 設定の接続ページで Nagle のアルゴリズムを無効にすることを試してください。

## インスタンスの停止に関するトラブルシューティング

Amazon EBS-Backed インスタンスを停止して `stopping` 状態のままスタッキングしているように見える場合、基になるホストコンピュータに問題がある可能性があります。

インスタンスが `running` 状態ではない場合は、インスタンスの使用に対してコストは発生しません。

コンソールまたは AWS CLI を使用してインスタンスを強制的に停止できます。

- コンソールを使用してインスタンスを強制的に停止するには、処理が止まってしまったインスタンスを選択し、[Actions (アクション)]、[Instance State (インスタンスの状態)]、[Stop (停止)]、[Yes, Forcefully Stop (強制的に停止する)] の順に選択します。
- AWS CLI を使用してインスタンスを強制的に停止するには、`stop-instances` コマンドと `--force` オプションを次のように使用します。

```
aws ec2 stop-instances --instance-ids i-0123ab456c789d01e --force
```

10 分が経過してもインスタンスが停止しない場合は、[Amazon EC2 forum](#) にヘルプリクエストを投稿してください。迅速な解決のために、インスタンス ID を含めて、既に行つた手順について説明してください。また、サポートプランを契約している場合は、[サポートセンター](#)でサポートケースを作成できます。

## 代わりのインスタンスの作成

Amazon EC2 forum またはサポートセンターからの支援を待っている間に問題解決を試みる方法として、代わりのインスタンスを作成できます。処理が止まってしまったインスタンスの AMI を作成し、新しい AMI を使用して新しいインスタンスを起動します。

コンソールを使用して代わりのインスタンスを作成するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances (インスタンス)] を選択し、処理が止まってしまったインスタンスを選択します。
3. [Actions]、[Image]、[Create Image] の順に選択します。
4. [Create Image (イメージの作成)] ダイアログボックスで、以下のフィールドに入力し、[Create Image (イメージの作成)] を選択します。
  - a. AMI の名前と説明を指定します。
  - b. [No reboot] を選択します。

詳細については、「[インスタンスからの Linux AMI の作成 \(p. 117\)](#)」を参照してください。

5. AMI から新しいインスタンスを起動し、その新しいインスタンスが動作していることを確認します。
6. 処理が止まってしまったインスタンスを選択し、[Actions (アクション)]、[Instance State (インスタンスの状態)]、[Terminate (終了)] の順に選択します。インスタンスの終了処理も止まってしまう場合は、Amazon EC2 は数時間以内に自動的にそのインスタンスを強制終了します。

CLI を使用して代わりのインスタンスを作成するには

1. [create-image](#) (AWS CLI) コマンドと --no-reboot オプションを次のように使用して、処理が止まってしまったインスタンスから AMI を作成します。

```
aws ec2 create-image --instance-id i-0123ab456c789d01e --name "AMI" --description "AMI for replacement instance" --no-reboot
```

2. [run-instances](#) (AWS CLI) コマンドを次のように使用し、作成した AMI から新しいインスタンスを起動します。

```
aws ec2 run-instances --image-id ami-1a2b3c4d --count 1 --instance-type c3.large --key-name MyKeyPair --security-groups MySecurityGroup
```

3. 新しいインスタンスが動作していることを確認します。
4. 次のように [terminate-instances](#) (AWS CLI) コマンドを使用し、処理が止まってしまったインスタンスを終了します。

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

前の手順で説明された方法でインスタンスから AMI を作成できない場合は、次のようにして代わりのインスタンスを設定できます。

(代替方法) コンソールを使用して代わりのインスタンスを作成するには

1. インスタンスを選択し、[Description (説明)]、[Block devices (ブロックデバイス)] の順に選択します。各ボリュームを選択し、そのボリューム ID を書き留めます。必ずどのボリュームがルートボリュームであるかメモしておきます。

2. ナビゲーションペインの [Volumes] を選択します。インスタンスの各ボリュームを選択し、[Actions]、[Create Snapshot] の順に選択します。
3. ナビゲーションペインで、[Snapshots] を選択します。作成したスナップショットを選択し、[Actions]、[Create Volume] の順に選択します。
4. 処理が止まってしまったインスタンスと同じオペレーティングシステムのインスタンスを起動します。そのルートボリュームのボリューム ID とデバイス名をメモしておきます。
5. ナビゲーションペインで、[Instances] を選択し、起動したインスタンスを選択した後で、[Actions]、[Instance State]、[Stop] の順に選択します。
6. ナビゲーションペインで [Volumes] を選択し、停止したインスタンスのルートボリュームを選択した後で、[Actions]、[Detach Volume] の順に選択します。
7. 処理が停止してしまったインスタンスから作成したルートボリュームを選択し、[Actions]、[Attach Volume] の順に選択して、そのルートボリュームとして新しいインスタンスにアタッチします(書き留めたデバイス名を使用)。その他の非ルートボリュームをインスタンスにアタッチします。
8. ナビゲーションペインで、[Instances] を選択し、代わりのインスタンスを選択します。[Actions]、[Instance State]、[Start] の順に選択します。インスタンスが動作していることを確認します。
9. 処理が止まってしまったインスタンスを選び、[Actions]、[Instance State]、[Terminate] の順に選択します。インスタンスの終了処理も止まってしまう場合は、Amazon EC2 は数時間以内に自動的にそのインスタンスを強制終了します。

## インスタンスの削除(シャットダウン)のトラブルシューティング

インスタンスが `running` 状態ではない場合は、インスタンスの使用に対して課金されません。つまり、インスタンスを終了させると、そのステータスが `shutting-down` に変わるとすぐに、そのインスタンスへの課金は停止します。

### インスタンスの削除の遅延

インスタンスの `shutting-down` 状態が数分以上続く場合は、インスタンスによって実行されるシャットダウンスクリプトが原因で遅れている可能性があります。

もう 1 つ考えられる原因として、基盤となるホストコンピュータの問題があります。インスタンスの `shutting-down` 状態が数時間以上続く場合、Amazon EC2 はそれを停止したインスタンスとして扱い、強制終了します。

インスタンスの終了処理が停止していると考えられ、すでに数時間以上経過している場合は、[Amazon EC2 forum](#) にヘルプリクエストを投稿してください。迅速な解決のために、インスタンス ID を含めて、既に行つた手順について説明してください。また、サポートプランを契約している場合は、[サポートセンター](#)でサポートケースを作成できます。

### 表示されているインスタンスを削除する

インスタンスの削除後、インスタンスはしばらくの間削除されずに表示されたままとなります。状態は `terminated` となります。このエントリが数時間経過しても削除されない場合には、サポートに連絡してください。

### インスタンスを自動的に起動または終了する

すべてのインスタンスを終了すると、Amazon が新しいインスタンスを起動する場合があります。インスタンスを起動すると、Amazon がいずれかのインスタンスを終了する場合があります。インスタンスを停止した場合、インスタンスを終了させて、新しいインスタンスを起動させることもあります。通常このよ

うな動作は、お客様が Amazon EC2 Auto Scaling または Elastic Beanstalk を使用して、お客様が定義した条件に基づいてコンピューティングリソースを自動的に拡大/縮小したために起こります。

詳細については、[Amazon EC2 Auto Scaling ユーザーガイド](#) または [AWS Elastic Beanstalk 開発者ガイド](#) を参照してください。

## ステータスチェックに失敗したインスタンスのトラブルシューティング

以下の情報は、インスタンスでステータスチェックに失敗した場合の問題のトラブルシューティングに役立ちます。まず、アプリケーションで問題が発生しているかどうかを確認します。インスタンスでアプリケーションが正常に実行されていないことを確認した場合は、ステータスチェック情報とシステムログを確認します。

### コンテンツ

- [ステータスチェック情報の確認 \(p. 1146\)](#)
- [システムログの取得 \(p. 1147\)](#)
- [Linux ベースのインスタンスに関するシステムログエラーのトラブルシューティング \(p. 1148\)](#)
- [メモリ不足: プロセスの終了 \(p. 1148\)](#)
- [エラー: mmu\\_update failed \(メモリ管理の更新に失敗しました\) \(p. 1149\)](#)
- [I/O エラー \(ロックデバイス障害\) \(p. 1150\)](#)
- [I/O エラー: ローカルでもリモートディスクでもありません \(破損した分散ロックデバイス\) \(p. 1151\)](#)
- [request\\_module: runaway loop modprobe \(古い Linux バージョンでレガシーカーネル modprobe がループしている\) \(p. 1152\)](#)
- 「FATAL: kernel too old」および「fsck: No such file or directory while trying to open /dev」(カーネルとAMIの不一致) (p. 1153)
- 「FATAL: Could not load /lib/modules」または「BusyBox」(カーネルモジュールの欠如) (p. 1153)
- [エラー: 無効のカーネル \(EC2と互換性のないカーネル\) \(p. 1155\)](#)
- [fsck: No such file or directory while trying to open...\(ファイルシステムが見つからない。\) \(p. 1156\)](#)
- [General error mounting filesystems \(マウント失敗\) \(p. 1157\)](#)
- [VFS: Unable to mount root fs on unknown-block \(ルートファイルシステム不一致\) \(p. 1159\)](#)
- [Error: Unable to determine major/minor number of root device...\(ルートファイルシステム/デバイス不一致\) \(p. 1160\)](#)
- [XENBUS: Device with no driver... \(p. 1161\)](#)
- ... days without being checked, check forced (ファイルシステムのチェックが必要です) (p. 1162)
- [fsck died with exit status... \(デバイスが見つからない\) \(p. 1162\)](#)
- [GRUB プロンプト \(grubdom>\) \(p. 1163\)](#)
- [Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring.\(ハードコードされた MAC アドレス\) \(p. 1165\)](#)
- [SELinux ポリシーを読み込めません。Machine is in enforcing mode.Halting now.\(SELinux の誤設定\) \(p. 1166\)](#)
- [XENBUS: Timeout connecting to devices \(Xenbus タイムアウト\) \(p. 1167\)](#)

## ステータスチェック情報の確認

Amazon EC2 コンソールを使用して、問題のあるインスタンスを調査するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。

2. ナビゲーションペインで [インスタンス] を選択し、インスタンスを選択します。
3. 詳細ペインの [Status Checks] タブを選択して、すべての [System Status Checks] と [Instance Status Checks] に関する個々の結果を表示します。

システムのステータスチェックに失敗した場合、次のいずれかの方法を試すことができます。

- インスタンスの復旧アラームを作成します。詳細については、「[インスタンスを停止、終了、再起動、または復旧するアラームを作成する \(p. 663\)](#)」を参照してください。
- インスタンスタイプを [Nitro ベースのインスタンス \(p. 187\)](#) に変更した場合、必要な ENS と NVMe ドライバがないインスタンスから移行するとステータスチェックは失敗します。詳細については、「[インスタンスのサイズ変更の互換性 \(p. 268\)](#)」を参照してください。
- Amazon EBS-Backed AMI を使用するインスタンスの場合、いったんインスタンスを停止してから再開します。
- instance-store backed AMI を使用するインスタンスの場合、インスタンスを終了し、代わりのインスタンスを起動します。
- Amazon EC2 が問題を解決するのを待ちます。
- 問題を [Amazon EC2 forum](#) に投稿します。
- インスタンスが Auto Scaling グループにある場合は、Amazon EC2 Auto Scaling サービスによって、代わりのインスタンスが自動的に起動されます。詳細については、『Amazon EC2 Auto Scaling ユーザーガイド』の「[Auto Scaling インスタンスのヘルスチェック](#)」を参照してください。
- システムログを取得し、エラーを探します。

## システムログの取得

インスタンスのステータスチェックに失敗した場合は、インスタンスを再起動してシステムログを取得できます。ログから判明したエラーが問題のトラブルシューティングに役立つ場合があります。再起動すると、ログから不要な情報が消去されます。

インスタンスを再起動してシステムログを取得するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. [Actions]、[Instance State]、[Reboot] の順に選択します。インスタンスが再起動するまでには数分かかることがあります。
4. 問題がまだ存在することを確認します。再起動によって、問題が解決することがあります。
5. インスタンスのステータスが `running` の場合は、[Actions]、[Instance Settings]、[Get System Log] の順に選択します。
6. 画面に表示されるログを確認し、下記の既知のシステムログエラー文のリストを使用して、問題のトラブルシューティングを行います。
7. ステータスチェックの結果が実際とは異なる場合、またはステータスチェックでは検出されない問題がインスタンスに発生している場合、[Status Checks] タブの [Submit feedback] を選択して、検出テストの改善にご協力ください。
8. 問題が解決されない場合は、問題を [Amazon EC2 forum](#) に投稿できます。

## Linux ベースのインスタンスに関するシステムログエラーのトラブルシューティング

インスタンスの接続性チェックなど、インスタンスのステータスチェックに失敗した Linux ベースのインスタンスの場合、上記の手順に従ってシステムログを取得したことを確認します。次のリストは、一般的なシステムログエラー、および各エラーの問題解決に対して推奨する対処を示しています。

### メモリエラー

- メモリ不足: プロセスの終了 (p. 1148)
- エラー: mmu\_update failed (メモリ管理の更新に失敗しました) (p. 1149)

### デバイスエラー

- I/O エラー (ブロックデバイス障害) (p. 1150)
- I/O エラー: ローカルでもリモートディスクでもありません (破損した分散ブロックデバイス) (p. 1151)

### カーネルエラー

- request\_module: runaway loop modprobe (古い Linux バージョンでレガシーカーネル modprobe がループしている) (p. 1152)
- 「FATAL: kernel too old」および「fsck: No such file or directory while trying to open /dev」(カーネルと AMI の不一致) (p. 1153)
- 「FATAL: Could not load /lib/modules」または「BusyBox」(カーネルモジュールの欠如) (p. 1153)
- エラー: 無効のカーネル (EC2 と互換性のないカーネル) (p. 1155)

### ファイルシステムエラー

- fsck: No such file or directory while trying to open...(ファイルシステムが見つからない。) (p. 1156)
- General error mounting filesystems (マウント失敗) (p. 1157)
- VFS: Unable to mount root fs on unknown-block (ルートファイルシステム不一致) (p. 1159)
- Error: Unable to determine major/minor number of root device...(ルートファイルシステム/デバイス不一致) (p. 1160)
- XENBUS: Device with no driver... (p. 1161)
- ... days without being checked, check forced (ファイルシステムのチェックが必要です) (p. 1162)
- fsck died with exit status...(デバイスが見つからない) (p. 1162)

### [オペレーティングシステムエラー]

- GRUB プロンプト (grubdom>) (p. 1163)
- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring.(ハードコードされた MAC アドレス) (p. 1165)
- SELinux ポリシーを読み込めません。Machine is in enforcing mode.Halting now.(SELinux の誤設定) (p. 1166)
- XENBUS: Timeout connecting to devices (Xenbus タイムアウト) (p. 1167)

## メモリ不足: プロセスの終了

メモリ不足エラーは、下記のようなシステムログで示されます。

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
エラー: mmu\_update failed (メモリ管理の更新に失敗しました)

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879
or a child
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-
rss:101196kB, file-rss:204kB
```

## 可能性のある原因

メモリの枯渇

## 推奨する対処

| インスタンスタイプ             | 操作                                                                                                                                                                                                                                               |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | <p>次のいずれかを行ってください。</p> <ul style="list-style-type: none"><li>インスタンスを停止し、異なるインスタンスタイプを使用するようにインスタンスを変更した後、インスタンスを再び起動します。たとえば、大きいインスタンスタイプやメモリ最適化インスタンスタイプです。</li><li>インスタンスを再起動して、障害のないステータスに戻します。インスタンスタイプを変更しない限り、問題が再び発生する可能性があります。</li></ul> |
| Instance store-Backed | <p>次のいずれかを行ってください。</p> <ul style="list-style-type: none"><li>インスタンスを終了し、別のインスタンスタイプを指定して、新しいインスタンスを起動します。たとえば、大きいインスタンスタイプやメモリ最適化インスタンスタイプです。</li><li>インスタンスを再起動して、障害のないステータスに戻します。インスタンスタイプを変更しない限り、問題が再び発生する可能性があります。</li></ul>                |

## エラー: mmu\_update failed (メモリ管理の更新に失敗しました)

メモリ管理更新失敗は、下記のようなシステムログで示されます。

```
...
Press `ESC' to enter the menu... 0  [H[J  Booting 'Amazon Linux 2011.09
(2.6.35.14-95.38.amzn1.i686)'

root (hd0)
Filesystem type is ext2fs, using whole disk
kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=
en_US.UTF-8 KEYTABLE=us
```

```
initrd /boot/initramfs-2.6.35.14-95.38.amzn1.1686.img
ERROR: mmu_update failed with rc=-22
```

## 可能性のある原因

Amazon Linux に関する問題

## 推奨する対処

問題を [開発者フォーラム](#)に投稿するか、[AWS サポート](#)にご連絡ください。

## I/O エラー (ブロックデバイス障害)

入力/出力エラーは、次の例のようなシステムログで示されます。

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

## 可能性のある原因

| インスタンスタイプ             | 可能性のある原因                 |
|-----------------------|--------------------------|
| Amazon EBS-Backed     | 障害が発生した Amazon EBS ボリューム |
| Instance store-Backed | 障害が発生した物理ドライブ            |

## 推奨する対処

| インスタンスタイプ         | 操作                                |
|-------------------|-----------------------------------|
| Amazon EBS-Backed | 次の手順に従ってください。<br>1. インスタンスを停止します。 |

| インスタンスタイプ             | 操作                                                                                                                                                                                                                       |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <p>2. ボリュームをデタッチします。<br/>     3. ボリュームの回復を試みます。</p> <p><b>Note</b></p> <p>Amazon EBS ボリュームのスナップショットを頻繁に作成することをお勧めします。これによって、エラーのためにデータを損失する危険性が大幅に減少します。</p> <p>4. ボリュームをインスタンスに再アタッチします。<br/>     5. ボリュームをデタッチします。</p> |
| Instance store-Backed | <p>インスタンスを終了し、新しいインスタンスを起動します。</p> <p><b>Note</b></p> <p>データを復旧できない。バックアップから復旧します。</p> <p><b>Note</b></p> <p>Amazon S3 または Amazon EBS をバックアップに使用することをお勧めします。インスタンストアボリュームは、单一のホストと単一のディスクエラーに直接結びついています。</p>             |

## I/O エラー: ローカルでもリモートディスクでもありません (破損した分散ロックデバイス)

デバイスでの入力/出力エラーは、次の例のようなシステムログで示されます。

```

...
block drbd1: Local IO failed in request_timer_fn. Detaching...
Aborting journal on device drbd1-8.

block drbd1: IO ERROR: neither local nor remote disk

Buffer I/O error on device drbd1, logical block 557056
lost page write due to I/O error on drbd1

JBD2: I/O error detected when updating journal superblock for drbd1-8.

```

### 可能性のある原因

| インスタンスタイプ         | 可能性のある原因                 |
|-------------------|--------------------------|
| Amazon EBS-Backed | 障害が発生した Amazon EBS ボリューム |

| インスタンスタイプ             | 可能性のある原因      |
|-----------------------|---------------|
| Instance store-Backed | 障害が発生した物理ドライブ |

## 推奨する対処

インスタンスを終了し、新しいインスタンスを起動します。

Amazon EBS-Backed インスタンスの場合、最新スナップショットからイメージを作成して、データを回復できます。スナップショットを作成した後に追加されたデータは回復できません。

## request\_module: runaway loop modprobe (古い Linux バージョンでレガシーカーネル modprobe がループしている)

下記のようなシステムログで、この状態が示されます。不安定であるか古い Linux カーネル（例: 2.6.16-xenU）を使用すると、起動時に無限ループが発生することがあります。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007

BIOS-provided physical RAM map:

xen: 0000000000000000 - 0000000026700000 (usable)

0MB HIGHMEM available.
...

request_module: runaway loop modprobe binfmt-464c
```

## 推奨する対処

| インスタンスタイプ         | 操作                                                                                                                                                                                                                                                                                        |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed | <p>次のいずれかのオプションを使用して、GRUB ベースまたは静的な新しいカーネルを使用します。</p> <p>オプション 1: インスタンスを終了し、<code>-kernel</code> および <code>-ramdisk</code> パラメータを指定して新しいインスタンスを起動します。</p> <p>オプション 2:</p> <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. 新しいカーネルを使用するようカーネルとラムディスク属性を変更します。</li></ol> |

| インスタンスタイプ             | 操作                                                         |
|-----------------------|------------------------------------------------------------|
|                       | 3. インスタンスを起動します。                                           |
| Instance store-Backed | インスタンスを終了し、-kernel および -ramdisk パラメータを指定して新しいインスタンスを起動します。 |

## 「FATAL: kernel too old」および「fsck: No such file or directory while trying to open /dev」(カーネルとAMI の不一致)

下記のようなシステムログで、この状態が示されます。

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST 2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

### 可能性のある原因

互換性のないカーネルとユーザーランド

### 推奨する対処

| インスタンスタイプ             | 操作                                                                                               |
|-----------------------|--------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | 次の手順に従ってください。<br>1. インスタンスを停止します。<br>2. 新しいカーネルを使用するよう設定を変更します。<br>3. インスタンスを起動します。              |
| Instance store-Backed | 次の手順に従ってください。<br>1. より新しいカーネルを使用するAMIを作成します。<br>2. インスタンスを終了します。<br>3. 作成したAMIから新しいインスタンスを起動します。 |

## 「FATAL: Could not load /lib/modules」または「BusyBox」(カーネルモジュールの欠如)

下記のようなシステムログで、この状態が示されます。

```
[ 0.370415] Freeing unused kernel memory: 1716k freed
```

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
「FATAL: Could not load /lib/modules」または  
は「BusyBox」(カーネルモジュールの欠如)

```

Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file or directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing: No such
      file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device. Common problems:
- Boot args (cat /proc/cmdline)
  - Check rootdelay= (did the system wait long enough?)
  - Check root= (did the system wait for the right device?)
- Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file or directory
ALERT! /dev/sdal does not exist. Dropping to a shell!

BusyBox v1.13.3 (Ubuntu 1:1.13.3-1ubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)

```

## 可能性のある原因

次の 1 つ以上の条件によって、この問題が発生する可能性があります。

- ・ラムディスクが見つからない
- ・ラムディスクに正しいモジュールが見つからない
- ・Amazon EBS ルートボリュームが /dev/sdal として正しくアタッチされていない

## 推奨する対処

| インスタンスタイプ             | 操作                                                                                                                                                                                                                                                                           |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | <p>次の手順に従ってください。</p> <ol style="list-style-type: none"> <li>1. Amazon EBS ボリュームに対して正しいラムディスクを選択します。</li> <li>2. インスタンスを停止します。</li> <li>3. ボリュームをデタッチし、修復します。</li> <li>4. ボリュームをインスタンスにアタッチします。</li> <li>5. インスタンスを起動します。</li> <li>6. 正しいラムディスクを使用するよう AMI を変更します。</li> </ol> |
| Instance store-Backed | 次の手順に従ってください。                                                                                                                                                                                                                                                                |

| インスタンスタイプ | 操作                                                                                                                                     |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------|
|           | <ol style="list-style-type: none"><li>1. インスタンスを終了してから、正しいラムディスクを使って新たなインスタンスを起動します。</li><li>2. 正しいラムディスクを使って新たな AMI を作成します。</li></ol> |

## エラー: 無効のカーネル (EC2 と互換性のないカーネル)

下記のようなシステムログで、この状態が示されます。

```
...
root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /vmlinuz root=/dev/sda1 ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

Booting 'Fallback'

root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

### 可能性のある原因

次の一方または両方の条件によって、この問題が発生する可能性があります。

- 指定されたカーネルは GRUB でサポートされていません
- フォールバックカーネルが存在しません

### 推奨する対処

| インスタンスタイプ         | 操作                                                                                                                                                 |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed | <p>次の手順に従ってください。</p> <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. 機能するカーネルに変更します。</li><li>3. フォールバックカーネルをインストールします。</li></ol> |

| インスタンスタイプ             | 操作                                                                                                                                                                                                                          |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | 4. カーネルを訂正して AMI を変更します。                                                                                                                                                                                                    |
| Instance store-Backed | 次の手順に従ってください。 <ol style="list-style-type: none"> <li>1. インスタンスを終了してから、正しいカーネルを使って新たなインスタンスを起動します。</li> <li>2. 正しいカーネルを使って AMI を作成します。</li> <li>3. (オプション) データ復旧の技術サポートについては、<a href="#">AWS サポート</a>にお問い合わせください。</li> </ol> |

## fsck: No such file or directory while trying to open... (ファイルシステムが見つからない。)

下記のようなシステムログで、この状態が示されます。

```
Welcome to Fedora
Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [ OK ]

Starting udev: [ OK ]

Setting hostname localhost: [ OK ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[ OK ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh

/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem. If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
  e2fsck -b 8193 <device>

[FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

### 可能性のある原因

- ・ラムディスクファイルシステム定義 /etc/fstab にバグがある
- ・/etc/fstab のファイルシステム定義の設定が不適切
- ・ドライブが見つからないかドライブにエラーがある

## 推奨する対処

| インスタンスタイプ             | 操作                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | <p>次の手順に従ってください。</p> <ol style="list-style-type: none"><li>1. インスタンスを停止し、ルートボリュームをデタッチし、/etc/fstab を修復または変更し、ボリュームをインスタンスにアタッチし、インスタンスを起動します。</li><li>2. ラムディスクを修正して、変更した /etc/fstab を含めます (適用可能な場合)。</li><li>3. 新しいラムディスクを使用するよう AMI を変更します。</li></ol> <p>fstab の 6 番目のフィールドではマウントの可用性要件を定義します。0 以外の値を指定した場合、そのボリュームに対して fsck を実行して成功しなければならないことを意味します。Amazon EC2 でこのフィールドを使用すると、問題が発生することがあります。これは、実行に失敗すると通常は対話的なコンソールプロンプトが表示されますが、このコンソールプロンプトは現在 Amazon EC2 で使用できないためです。この機能は慎重に使用してください。また、fstab の Linux マニュアルページを参照してください。</p> |
| Instance store-Backed | <p>次の手順に従ってください。</p> <ol style="list-style-type: none"><li>1. インスタンスを終了し、新しいインスタンスを起動します。</li><li>2. 障害のある Amazon EBS ボリュームをデタッチし、インスタンスを再起動します。</li><li>3. (オプション) データ復旧の技術サポートについては、<a href="#">AWS サポート</a>にお問い合わせください。</li></ol>                                                                                                                                                                                                                                                                                                   |

## General error mounting filesystems (マウント失敗)

下記のようなシステムログで、この状態が示されます。

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting. Commit interval 5 seconds
```

```
EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

## 可能性のある原因

| インスタンスタイプ             | 可能性のある原因                                                                                                                                                                                 |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | <ul style="list-style-type: none"><li>Amazon EBS ボリュームがデタッチされているか、ボリュームにエラーがあります。</li><li>ファイルシステムが破損している。</li><li>ラムディスクと AMI の組み合わせが一致していないません (例: Debian ラムディスクと SUSE AMI)。</li></ul> |
| Instance store-Backed | <ul style="list-style-type: none"><li>ドライブにエラーがあります。</li><li>ファイルシステムが破損しています。</li><li>ラムディスクと AMI の組み合わせが一致していないません (例: Debian ラムディスクと SUSE AMI)。</li></ul>                             |

## 推奨する対処

| インスタンスタイプ         | 操作                                                                                                                                                                                                                                                                                                                     |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed | <p>次の手順に従ってください。</p> <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. ルートボリュームをデタッチする。</li><li>3. ルートボリュームを既知の動作しているインスタンスにアタッチします。</li><li>4. ファイルシステムチェックを実行します (fsck -a /dev/...)。</li><li>5. エラーを修正します。</li><li>6. ボリュームを既知の動作しているインスタンスからデタッチします。</li><li>7. 停止したインスタンスにボリュームをアタッチします。</li></ol> |

| インスタンスタイプ             | 操作                                                                                                                                                             |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | 8. インスタンスを起動します。<br>9. インスタンスのステータスを再確認します。                                                                                                                    |
| Instance store-Backed | 以下のいずれかを行ってください。<br><ul style="list-style-type: none"> <li>新しいインスタンスを起動します。</li> <li>(オプション) データ復旧の技術サポートについては、<a href="#">AWS サポート</a>にお問い合わせください。</li> </ul> |

## VFS: Unable to mount root fs on unknown-block (ルートファイルシステム不一致)

下記のようなシステムログで、この状態が示されます。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050527 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line: root=/dev/sda1 ro 4
...
Registering block device major 8
...
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

### 可能性のある原因

| インスタンスタイプ             | 可能性のある原因                                                                                                                                                                                                                              |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | <ul style="list-style-type: none"> <li>デバイスが正しくアタッチされていない。</li> <li>ルートデバイスが正しいデバイスポイントにアタッチされていない。</li> <li>ファイルシステムのフォーマットが正しくありません。</li> <li>レガシーカーネルを使用している(たとえば、2.6.16-XenU)。</li> <li>インスタンスの最新のカーネル更新(更新エラーまたは更新バグ)</li> </ul> |
| Instance store-Backed | ハードウェアデバイスのエラー。                                                                                                                                                                                                                       |

### 推奨する対処

| インスタンスタイプ         | 操作                                                                                                                                                                            |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed | <p>次のいずれかを行ってください。</p> <ul style="list-style-type: none"> <li>インスタンスを停止し、再起動します。</li> <li>正しいデバイスポイントでアタッチするようルートボリュームを変更します。たとえば、/dev/sda の代わりに /dev/sda1 を使用します。</li> </ul> |

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 Error: Unable to determine major/minor number of root  
 device...(ルートファイルシステム/デバイス不一致)

| インスタンスタイプ             | 操作                                                                                                                                                          |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <ul style="list-style-type: none"> <li>停止し、新しいカーネルを使用するように変更します。</li> <li>既知の更新バグを確認するには、Linux ディストリビューションのドキュメントを参照してください。カーネルを変更または再インストールします。</li> </ul> |
| Instance store-Backed | インスタンスを終了し、新しいカーネルを使用して、新しいインスタンスを起動します。                                                                                                                    |

## Error: Unable to determine major/minor number of root device...(ルートファイルシステム/デバイス不一致)

下記のようなシステムログで、この状態が示されます。

```
...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

### 可能性のある原因

- 仮想ブロックデバイスドライバーが見つからないか、設定が間違っている
- デバイス列挙が競合している (sda と xvda または sda1 の代わりに sda)
- インスタンスカーネルが正しく選択されていない

### 推奨する対処

| インスタンスタイプ         | 操作                                                                                                                                                                |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed | <p>次の手順に従ってください。</p> <ol style="list-style-type: none"> <li>インスタンスを停止します。</li> <li>ボリュームをデタッチします。</li> <li>デバイスのマッピングの問題を解決します。</li> <li>インスタンスを起動します。</li> </ol> |

| インスタンスタイプ             | 操作                                                                                                                                                           |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | 5. AMI を変更して、デバイスのマッピングの問題に対処します。                                                                                                                            |
| Instance store-Backed | 次の手順に従ってください。 <ol style="list-style-type: none"><li>適切な修正を使用して新しい AMI を作成します（ブロックデバイスを正しくマッピングします）。</li><li>インスタンスを終了し、作成した AMI から新しいインスタンスを起動します。</li></ol> |

## XENBUS: Device with no driver...

下記のようなシステムログで、この状態が示されます。

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
::: Starting udevd...
done.
::: Running Hook [udev]
::: Triggering uevents...<30>udevd[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

## 可能性のある原因

- 仮想ブロックデバイスドライバーが見つからないか、設定が間違っている
- デバイス列挙が競合している (sda と xvda)。
- インスタンスカーネルが正しく選択されていない

## 推奨する対処

| インスタンスタイプ             | 操作                                                                                                                                                                                           |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | 次の手順に従ってください。 <ol style="list-style-type: none"><li>インスタンスを停止します。</li><li>ボリュームをデタッチします。</li><li>デバイスのマッピングの問題を解決します。</li><li>インスタンスを起動します。</li><li>AMI を変更して、デバイスのマッピングの問題に対処します。</li></ol> |
| Instance store-Backed | 次の手順に従ってください。 <ol style="list-style-type: none"><li>適切な修正を使用して新しい AMI を作成します（ブロックデバイスを正しくマッピングします）。</li><li>インスタンスを終了し、作成した AMI から新しいインスタンスを起動します。</li></ol>                                 |

| インスタンスタイプ | 操作                                                                                                                                             |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <ol style="list-style-type: none"><li>適切な修正を使用して AMI を作成します(ブロックデバイスを正しくマッピングします)。</li><li>インスタンスを終了し、作成した AMI を使用して新しいインスタンスを起動します。</li></ol> |

## ... days without being checked, check forced (ファイルシステムのチェックが必要です)

下記のようなシステムログで、この状態が示されます。

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

### 可能性のある原因

ファイルシステムのチェック期間が経過したため、ファイルシステムチェックが強制実行されている。

### 推奨する対処

- ファイルシステムチェックが完了するまで待ちます。ルートファイルシステムのサイズによっては、ファイルシステムチェックに時間がかかることがあります。
- tune2fs またはファイルシステムに適したツールを使用してファイルシステムを変更し、ファイルシステムチェック(fsck)の実行を削除します。

## fsck died with exit status...(デバイスが見つからない)

下記のようなシステムログで、この状態が示されます。

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

### 可能性のある原因

- 存在しないドライブをラムディスクが検索している
- ファイルシステムの整合性チェックが強制実行されている
- ドライブにエラーがあるか、デタッチされている

## 推奨する対処

| インスタンスタイプ             | 操作                                                                                                                                                                                                                                                                   |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | <p>問題を解決するため、次のいずれかを試します。</p> <ul style="list-style-type: none"><li>インスタンスを停止し、ボリュームを既存の実行中インスタンスにアタッチします。</li><li>整合性チェックを手動で実行します。</li><li>ラムディスクを修正して、関連するユーティリティを含めます。</li><li>ファイルシステム調整パラメータを変更して、整合性要件を削除します（お勧めしません）。</li></ul>                              |
| Instance store-Backed | <p>問題を解決するため、次のいずれかを試します。</p> <ul style="list-style-type: none"><li>ラムディスクに正しいツールをリバンドリングします。</li><li>ファイルシステム調整パラメータを変更して、整合性要件を削除します（お勧めしません）。</li><li>インスタンスを終了し、新しいインスタンスを起動します。</li><li>（オプション）データ復旧の技術サポートについては、<a href="#">AWS サポート</a>にお問い合わせください。</li></ul> |

## GRUB プロンプト (grubdom>)

下記のようなシステムログで、この状態が示されます。

```
GNU GRUB  version 0.97  (629760K lower / 0K upper memory)

[ Minimal BASH-like line editing is supported.  For
the first word, TAB lists possible command
completions.  Anywhere else TAB lists the possible
completions of a device/filename. ]

grubdom>
```

## 可能性のある原因

| インスタンスタイプ         | 可能性のある原因                                                                                                                                                                                             |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed | <ul style="list-style-type: none"><li>GRUB 設定ファイルがありません。</li><li>正しくない GRUB イメージが使用されています。別の場所に GRUB 設定ファイルが必要です。</li><li>GRUB 設定ファイルを保存するために使用されているファイルシステムがサポートされていません（たとえば、ルートファイルシステムを</li></ul> |

| インスタンスタイプ             | 可能性のある原因                                                                                                                                                                                                                                    |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | GRUB の以前のバージョンでサポートされていないタイプに変換した)                                                                                                                                                                                                          |
| Instance store-Backed | <ul style="list-style-type: none"> <li>GRUB 設定ファイルがありません。</li> <li>正しくない GRUB イメージが使用されています。別の場所に GRUB 設定ファイルが必要です。</li> <li>GRUB 設定ファイルを保存するために使用されているファイルシステムがサポートされていません（たとえば、ルートファイルシステムを GRUB の以前のバージョンでサポートされていないタイプに変換した）</li> </ul> |

## 推奨する対処

| インスタンスタイプ         | 操作                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed | <p>オプション 1: AMI を変更しインスタンスを再作成します。</p> <ol style="list-style-type: none"> <li>ソース AMI を変更して、標準の場所に GRUB 設定ファイルを作成します (/boot/grub/menu.lst)。</li> <li>GRUB のバージョンが、基になるファイルシステムのタイプをサポートしていることを確認し、必要に応じて GRUB をアップグレードします。</li> <li>適切な GRUB イメージを選択します (hd0 – 第 1 ドライブまたは hd00 – 第 1 ドライブ、第 1 パーティション)。</li> <li>インスタンスを終了し、作成した AMI を使用して新しいインスタンスを起動します。</li> </ol> <p>オプション2: 既存のインスタンスの修正:</p> <ol style="list-style-type: none"> <li>インスタンスを停止します。</li> <li>ルートファイルシステムをデタッチします。</li> <li>ルートファイルシステムを既知の動作しているインスタンスにアタッチします。</li> <li>ファイルシステムをマウントします。</li> <li>GRUB 設定ファイルを作成します。</li> <li>GRUB のバージョンが、基になるファイルシステムのタイプをサポートしていることを確認し、必要に応じて GRUB をアップグレードします。</li> <li>ファイルシステムをデタッチします。</li> <li>元のインスタンスにアタッチします。</li> <li>カーネル属性を変更して、適切な GRUB イメージを使用します (第 1 ディスクまたは第 1 ディスクの第 1 パーティション)。</li> </ol> |

Amazon Elastic Compute Cloud  
Linux インスタンス用ユーザーガイド  
Bringing up interface eth0: Device eth0  
has different MAC address than expected,

| インスタンスタイプ             | 操作                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | 10. インスタンスを起動します。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Instance store-Backed | <p>オプション 1: AMI を変更しインスタンスを再作成します。</p> <ol style="list-style-type: none"><li>GRUB 設定ファイルを使用して、標準の場所に新しい AMI を作成します (/boot/grub/menu.lst)。</li><li>適切な GRUB イメージを選択します (hd0 – 第 1 ドライブまたは hd00 – 第 1 ドライブ、第 1 パーティション)。</li><li>GRUB のバージョンが、基になるファイルシステムのタイプをサポートしていることを確認し、必要に応じて GRUB をアップグレードします。</li><li>インスタンスを終了し、作成した AMI を使用して新しいインスタンスを起動します。</li></ol> <p>オプション 2: インスタンスを終了し、正しいカーネルを指定して新しいインスタンスを起動します。</p> <p>Note</p> <p>既存のインスタンスからデータを回復するには、<a href="#">AWS サポート</a>にお問い合わせください。</p> |

## Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring.(ハードコードされた MAC アドレス)

下記のようなシステムログで、この状態が示されます。

```
...
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring.
[FAILED]
Starting auditd: [ OK ]
```

### 可能性のある原因

AMI 設定にハードコードされたインターフェイス MAC がある

### 推奨する対処

| インスタンスタイプ         | 操作              |
|-------------------|-----------------|
| Amazon EBS-Backed | 次のいずれかを行ってください。 |

Amazon Elastic Compute Cloud  
 Linux インスタンス用ユーザーガイド  
 SELinux ポリシーを読み込めません。Machine is  
 in enforcing mode. Halting now.(SELinux の誤設定)

| インスタンスタイプ             | 操作                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <ul style="list-style-type: none"> <li>AMI を変更してハードコードを削除し、インスタンスを再作成します。</li> <li>ハードコードされた MAC アドレスを削除するようインスタンスを変更します。</li> </ul> <p>または</p> <p>次の手順に従ってください。</p> <ol style="list-style-type: none"> <li>1. インスタンスを停止します。</li> <li>2. ルートボリュームをデタッチする。</li> <li>3. ボリュームを別のインスタンスにアタッチし、ハードコードされた MAC アドレスを削除するようにボリュームを変更します。</li> <li>4. 初期インスタンスにボリュームをアタッチします。</li> <li>5. インスタンスを起動します。</li> </ol> |
| Instance store-Backed | <p>次のいずれかを行ってください。</p> <ul style="list-style-type: none"> <li>ハードコードされた MAC アドレスを削除するようインスタンスを変更します。</li> <li>インスタンスを終了し、新しいインスタンスを起動します。</li> </ul>                                                                                                                                                                                                                                                       |

## SELinux ポリシーを読み込めません。Machine is in enforcing mode. Halting now.(SELinux の誤設定)

下記のようなシステムログで、この状態が示されます。

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

### 可能性のある原因

SELinux が誤って有効にされた。

- 指定されたカーネルは GRUB でサポートされていません
- フォールバックカーネルが存在しません

### 推奨する対処

| インスタンスタイプ         | 操作                                                                                           |
|-------------------|----------------------------------------------------------------------------------------------|
| Amazon EBS-Backed | <p>次の手順に従ってください。</p> <ol style="list-style-type: none"> <li>1. 障害のあるインスタンスを停止します。</li> </ol> |

| インスタンスタイプ             | 操作                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <ol style="list-style-type: none"> <li>2. 障害の起きたインスタンスのルートボリュームをデタッチします。</li> <li>3. 別の実行中の Linux インスタンスにルートボリュームをアタッチします (リカバリインスタンスとして扱われます)。</li> <li>4. リカバリインスタンスを接続し、障害の起きたインスタンスのルートボリュームをマウントします。</li> <li>5. マウントしたルートボリュームの SELinux を無効にします。このプロセスは Linux ディストリビューションによって異なります。詳細については各 OS のドキュメントを参照してください。</li> </ol> <p style="text-align: center;"><b>Note</b></p> <p style="margin-left: 20px;">一部のシステムでは、SELINUX=disabled ファイル (<code>/etc/sysconfig/selinux</code> はリカバリインスタンスにマウントしたボリュームの場所) で <code>mount_point</code> と設定して SELinux を無効にします。</p> <ol style="list-style-type: none"> <li>6. リカバリインスタンスからルートボリュームをアンマウントしてデタッチし、元のインスタンスに再アタッチします。</li> <li>7. インスタンスを起動します。</li> </ol> |
| Instance store-Backed | <p>次の手順に従ってください。</p> <ol style="list-style-type: none"> <li>1. インスタンスを終了し、新しいインスタンスを起動します。</li> <li>2. (オプション) データ復旧の技術サポートについては、<a href="#">AWS サポート</a>にお問い合わせください。</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

## XENBUS: Timeout connecting to devices (Xenbus タイムアウト)

下記のようなシステムログで、この状態が示されます。

```

Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found. Try passing init= option to kernel.

```

### 可能性のある原因

- ブロックデバイスがインスタンスに接続されていない
- このインスタンスは古いインスタンスカーネルを使用している

## 推奨する対処

| インスタンスタイプ             | 操作                                                                                                                                      |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Amazon EBS-Backed     | 次のいずれかを行ってください。 <ul style="list-style-type: none"><li>AMI とインスタンスを変更して新しいカーネルを使用し、インスタンスを再作成します。</li><li>インスタンスを再起動します。</li></ul>       |
| Instance store-Backed | 次のいずれかを行ってください。 <ul style="list-style-type: none"><li>インスタンスを終了します。</li><li>AMI を変更して新しいカーネルを使用し、その AMI を使用して新しいインスタンスを起動します。</li></ul> |

## 到達できないインスタンスのトラブルシューティング

到達できないインスタンスのトラブルシューティングには、次の方法を使用できます。

### コンテンツ

- インスタンスの再起動 (p. 1168)
- インスタンスコンソール出力 (p. 1168)
- 接続できないインスタンスのスクリーンショットの取得 (p. 1169)
- ホストコンピュータに障害が発生した場合のインスタンスの復旧 (p. 1170)

## インスタンスの再起動

トラブルシューティングにも一般的なインスタンス管理にも、到達できないインスタンスを再起動する方法が重要です。

リセットボタンを押してコンピュータをリセットするように、Amazon EC2 コンソール、CLI、または API を使用して EC2 インスタンスをリセットできます。詳細については、「[インスタンスの再起動 \(p. 542\)](#)」を参照してください。

### Warning

Windows インスタンスの場合、この動作によりハードウェアが再起動され、その結果データが破損する可能性があります。

## インスタンスコンソール出力

コンソール出力は問題を診断する際に役立つツールで、特に、カーネルの問題やサービス設定の問題のトラブルシューティングを行うときに便利です。これらの問題が発生すると、SSH デーモンの開始前にインスタンスが停止したり、インスタンスに到達不能になったりする可能性があります。

Linux/Unix の場合、マシンに接続されている物理的なモニターに通常表示されるようなコンソール出力がインスタンスコンソール出力に表示されます。コンソール出力は、インスタンス遷移状態(開始、停止、再起動、終了)の直後に投稿されたバッファされた情報を返します。表示される出力は、継続的には更新されず、更新する価値があると思われる場合にのみ更新されます。

Windows インスタンスの場合は、インスタンスコンソール出力に直近のシステムイベントログエラーが 3 つ表示されます。

オプションで、インスタンスのライフサイクル中に最新のシリアルコンソールの出力をいつでも取得できます。このオプションは [Nitro ベースのインスタンス \(p. 187\)](#) でのみサポートされています。Amazon EC2 コンソールではサポートされていません。

Note

表示される出力のうち、保存されるのは最新の 64 KB のみです。この出力は、出力の送信から少なくとも 1 時間使用可能です。

インスタンスの所有者のみがコンソール出力にアクセスできます。コンソールまたはコマンドラインを使用して、インスタンスのコンソール出力を取得できます。

コンソールを使用してコンソール出力を取得するには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 左ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. [Actions]、[Instance Settings]、[Get System Log] の順に選択します。

コマンドラインを使用してコンソール出力を取得するには

次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [get-console-output \(AWS CLI\)](#)
- [Get-EC2ConsoleOutput \(AWS Tools for Windows PowerShell\)](#)

一般的なシステムログエラーの詳細については、[Linux ベースのインスタンスに関するシステムログエラーのトラブルシューティング \(p. 1148\)](#) を参照してください。

## 接続できないインスタンスのスクリーンショットの取得

SSH または RDP を介してインスタンスに接続できない場合、インスタンスのスクリーンショットをキャプチャし、イメージとして表示できます。このイメージにより、インスタンスのステータスについて可視化されるため、迅速にトラブルシューティングすることができます。インスタンスの実行中またはクラッシュ後にスクリーンショットを生成できます。このスクリーンショットにはデータ転送コストがかかりません。イメージは JPG 形式で生成され、100 KB 未満です。この機能は、NVIDIA GRID ドライバーを使用しているインスタンスや、ペアメタルインスタンス (タイプが \*.meta1 のインスタンス) ではサポートされません。この機能は以下のリージョンで利用できます。

- 米国東部 (バージニア北部) リージョン
- 米国東部 (オハイオ) リージョン
- 米国西部 (オレゴン) リージョン
- 米国西部 (北カリフォルニア) リージョン
- 歐州 (アイルランド) リージョン
- 歐州 (ランクフルト) リージョン
- アジアパシフィック (東京) リージョン
- アジアパシフィック (ソウル) リージョン
- アジアパシフィック (シンガポール) リージョン
- アジアパシフィック (シドニー) リージョン

- 南米 (サンパウロ) リージョン
- アジアパシフィック (ムンバイ) リージョン
- カナダ (中部) リージョン
- 欧州 (ロンドン) リージョン
- 欧州 (パリ) リージョン

インスタンスコンソールにアクセスするには

1. <https://console.aws.amazon.com/ec2/> で Amazon EC2 コンソールを開きます。
2. 左のナビゲーションペインの [インスタンス] を選択します。
3. キャプチャするインスタンスを選択します。
4. [アクション]、[インスタンスの設定] の順に選択します。
5. [Get Instance Screenshot] を選択します。

イメージを右クリックし、ダウンロードして保存します。

コマンドラインを使用してスナップショットをキャプチャするには

次のコマンドの 1 つを使用できます。返されるコンテンツは base64 でエンコードされます。これらのコマンドラインインターフェイスの詳細については、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `get-console-screenshot` (AWS CLI)
- `GetConsoleScreenshot` (Amazon EC2 クエリ API)

## ホストコンピュータに障害が発生した場合のインスタンスの復旧

基になるホストコンピュータのハードウェアで復旧不可能な問題が発生した場合、AWS はインスタンスの停止イベントをスケジュールすることができます。このようなイベントは事前に E メールで通知されます。

障害が発生したホストコンピュータで実行されている Amazon EBS-backed インスタンスを復旧するには

1. インスタンスストアボリュームの重要なデータを Amazon EBS または Amazon S3 にバックアップします。
2. インスタンスを停止します。
3. インスタンスを起動します。
4. 重要なデータを復元します。

詳細については、「[インスタンスの停止と起動 \(p. 529\)](#)」を参照してください。

障害が発生したホストコンピュータで実行されている Instance-store Backed インスタンスを復旧するには

1. インスタンスから AMI を作成します。
2. イメージを Amazon S3 にアップロードします。
3. 重要なデータを Amazon EBS または Amazon S3 にバックアップします。
4. インスタンスを終了します。

5. AMI から新しいインスタンスを起動します。
6. 重要なデータを新しいインスタンスに復元します。

詳細については、「[Instance Store-Backed Linux AMI の作成 \(p. 119\)](#)」を参照してください。

## 間違ったボリュームで起動する

状況によっては、/dev/xvda または /dev/sda にアタッチしたボリューム以外のボリュームが、インスタンスのルートボリュームになっている場合があります。これは、別のインスタンスのルートボリュームや、ルートボリュームのスナップショットから作成されたボリュームを、既存のルートボリュームのインスタンスにアタッチした場合に起こります。

これは Linux の初期ラムディスクの挙動です。/ で /etc/fstab として定義されたボリュームを選択した場合でも、一部のディストリビューションでは、ボリュームパーティションにアタッチされたラベルによってボリュームが決定されます。たとえば、/etc/fstab の内容が次のとおりであったとします。

```
LABEL=/ / ext4 defaults,noatime 1 1
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

両方のボリュームのラベルを確認すれば、両方に / ラベルが含まれることが判ります。

```
[ec2-user ~]$ sudo e2label /dev/xvda1
/
[ec2-user ~]$ sudo e2label /dev/xvdf1
/
```

この例では、初期ラムディスクの実行後、意図していた /dev/xvdf1 ボリュームからの起動ではなく、/dev/xvda1 がインスタンスを起動するルートデバイスになる結果となりました。これを解決するためには、同じ e2label コマンドを使用して、起動ボリュームではないアタッチ済みのボリュームのラベルを変更できます。

場合によっては、/etc/fstab で UUID を指定することで、この問題を解決できます。ただし、両方のボリュームが同じスナップショットから作成された場合、またはセカンダリボリュームがプライマリボリュームのスナップショットから作成されている場合は、両ボリュームは UUID を共有します。

```
[ec2-user ~]$ sudo blkid
/dev/xvda1: LABEL="/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
/dev/xvdf1: LABEL="old/" UUID=73947a77-ddbe-4dc7-bd8f-3fe0bc840778 TYPE="ext4"
PARTLABEL="Linux" PARTUUID=d55925ee-72c8-41e7-b514-7084e28f7334
```

アタッチされた ext4 ボリュームのラベルを変更するには

1. e2label コマンドを使用して、ボリュームのラベルを / 以外のものに変更します。

```
[ec2-user ~]$ sudo e2label /dev/xvdf1 old/
```

2. ボリュームに新しいラベルがあることを確認します。

```
[ec2-user ~]$ sudo e2label /dev/xvdf1
old/
```

### アタッチされた xfs ボリュームのラベルを変更するには

- `xfs_admin` コマンドを使用して、ボリュームのラベルを / 以外のものに変更します。

```
[ec2-user ~]$ sudo xfs_admin -L old/ /dev/xvdf1
writing all SBs
new label = "old/"
```

図のようにボリュームラベルを変更した後で、インスタンスを再起動すると、インスタンスの起動時にラムディスクが正しいボリュームを選択するはずです。

#### Important

新しいラベルを持つボリュームをデタッチし、別のインスタンスに戻してルートボリュームとして使用する場合は、上記の手順をもう一度実行してラベルを元の値に戻す必要があります。これを行わない場合、ラムディスクがラベル / を持つボリュームを見つけることができないため、別のインスタンスが起動しません。

## Linux 用 EC2Rescue を使用する

Linux 用 EC2Rescue は、使いやすいオープンソースのツールであり、Amazon EC2 Linux インスタンスで実行し、100 を超えるモジュールのライブラリを使用して一般的な問題を診断およびトラブルシューティングできます。Linux 用 EC2Rescue の汎用ユースケースには、syslog およびパッケージマネージャーログの収集、リソース使用状況データの収集、問題のある既知のカーネルパラメーターと一般的な OpenSSH の問題の診断および修復などがあります。

#### Note

Windows インスタンスを使用する場合には、「[EC2Rescue for Windows Server](#)」を参照してください。

#### コンテンツ

- [Linux 用 EC2Rescue のインストール \(p. 1172\)](#)
- [Linux 用 EC2Rescue の使用 \(p. 1175\)](#)
- [EC2Rescue モジュールを開発する \(p. 1177\)](#)

## Linux 用 EC2Rescue のインストール

Linux 用 EC2Rescue ツールは、次の前提要件を満たす Amazon EC2 Linux インスタンスにインストールできます。

#### 前提条件

- サポートされるオペレーティングシステム
  - Amazon Linux 2
  - Amazon Linux 2016.09+
  - SLES 12+
  - RHEL 7+
  - Ubuntu 16.04+
- ソフトウェア要件
  - Python 2.7.9+ または 3.2+

システムに必要な Python バージョンがある場合は、標準ビルトをインストールできます。それ以外の場合は、Python の最小のコピーを含むバンドル済みのビルトをインストールできます。

標準ビルトをインストールするには

1. 作動している Linux インスタンスから、[Linux 用 EC2Rescue](#) ツールをダウンロードします。

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz
```

2. (オプション) 先に進む前に、オプションで Linux 用 EC2Rescue インストールファイルの署名を検証できます。詳細については、「[\(省略可能\) Linux 用 EC2Rescue の署名を検証します。\(p. 1173\)](#)」を参照してください。
3. sha256 ハッシュファイルをダウンロードします。

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sha256
```

4. tarball の整合性を確認します。

```
sha256sum -c ec2rl.tgz.sha256
```

5. Tarball を解凍します。

```
tar -xvf ec2rl.tgz
```

6. ヘルプファイルを表示してインストールを検証します。

```
cd ec2rl-<version_number>
./ec2rl help
```

バンドル済みのビルトをインストールするには

ダウンロードのリンクと制限については、github の「[Linux 用 EC2Rescue](#)」を参照してください。

## (省略可能) Linux 用 EC2Rescue の署名を検証します。

以下に、Linux ベースのオペレーティングシステム用の Linux 用 EC2Rescue パッケージの有効性を検証するための推奨されるプロセスを示します。

インターネットからアプリケーションをダウンロードする場合は、ソフトウェア発行元のアイデンティティを認証し、アプリケーションの発行後に改ざん、あるいは破損がないか確認することをお勧めします。これにより、ウイルスやマルウェアに感染したバージョンのアプリケーションをインストールせずに済みます。

このトピックのステップを実行した後に Linux 用 EC2Rescue のソフトウェアが変更または破損していることが判明した場合は、インストールファイルを実行しないでください。このような場合は Amazon Web Services にご連絡ください。

Linux ベースのオペレーティングシステム用の Linux 用 EC2Rescue ファイルの署名には、GnuPG が使用されています。これは安全なデジタル署名のための、オープンソース実装のプリティグッドプライバシー (OpenPGP) 標準です。GnuPG (GPG とも呼ばれます) は、デジタル署名を通じて認証と完全性チェックを行います。AWS は、ダウンロードした Linux 用 EC2Rescue パッケージの検証に使用できるパブリックキーと署名を公開します。PGP と GnuPG (GPG) の詳細については、「<http://www.gnupg.org>」を参照してください。

まず、ソフトウェア発行元との信頼を確立します。ソフトウェア発行元のパブリックキーをダウンロードし、キー所有者が一致していることを確認してから、キーリングに追加します。キーリングとは、既知のパブリックキーの集合です。真正性が確立されたパブリックキーは、アプリケーションの署名を確認するために使用できます。

#### タスク

- [GPG ツールのインストール \(p. 1174\)](#)
- [パブリックキーの認証とインポート \(p. 1174\)](#)
- [パッケージの署名の確認 \(p. 1175\)](#)

## GPG ツールのインストール

お使いのオペレーティングシステムが Linux または Unix の場合、GPG ツールが既にインストールされている場合があります。システムにツールがインストール済みかどうかをテストするには、コマンドラインプロンプトで `gpg2` と入力します。GPG ツールがインストールされている場合、GPG のコマンドプロンプトが表示されます。GPG ツールがインストールされていない場合、コマンドが見つからないというエラーが表示されます。GnuPG パッケージはリポジトリからインストールできます。

Debian ベースの Linux に GPG ツールをインストールするには

- ターミナルから、次のコマンドを実行します。

```
apt-get install gnupg2
```

Red Hat ベースの Linux に GPG ツールをインストールするには

- ターミナルから、次のコマンドを実行します。

```
yum install gnupg2
```

## パブリックキーの認証とインポート

次の手順では、Linux 用 EC2Rescue のパブリックキーを認証し、信頼されたキーとして GPG キーリングへ追加します。

Linux 用 EC2Rescue のパブリックキーを認証してインポートするには

1. コマンドプロンプトで、次のコマンドを使用して当社のパブリック GPG ビルドキーのコピーを取得します。

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.key
```

2. `ec2rl.key` を保存したディレクトリのコマンドプロンプトで、次のコマンドを使用して Linux 用 EC2Rescue のパブリックキーをキーリングにインポートします。

```
gpg2 --import ec2rl.key
```

コマンドで次のような結果が返されます。

```
gpg: /home/ec2-user/.gnupg/trustdb.gpg: trustdb created
gpg: key 2FAE2A1C: public key "ec2autodiag@amazon.com <EC2 Rescue for Linux>" imported
gpg: Total number processed: 1
```

```
gpg: imported: 1 (RSA: 1)
```

## パッケージの署名の確認

GPG ツールをインストール後、Linux 用 EC2Rescue パブリックキーを認証してインポートし、Linux 用 EC2Rescue パブリックキーが信頼済みであることを確認すると、Linux 用 EC2Rescue インストールスクリプトの署名を確認できるようになります。

Linux 用 EC2Rescue インストールスクリプトの署名を確認するには

1. コマンドプロンプトで次のコマンドを実行し、インストールスクリプトの署名ファイルをダウンロードします。

```
curl -O https://s3.amazonaws.com/ec2rescuelinux/ec2rl.tgz.sig
```

2. `ec2rl.tgz.sig` と Linux 用 EC2Rescue インストールファイルを保存したディレクトリのコマンドプロンプトで次のコマンドを実行し、署名を確認します。ファイルが2つとも存在している必要があります。

```
gpg2 --verify ./ec2rl.tgz.sig
```

出力は次のようになります。

```
gpg: Signature made Thu 12 Jul 2018 01:57:51 AM UTC using RSA key ID 6991ED45
gpg: Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                 There is no indication that the signature belongs to the owner.
Primary key fingerprint: E528 BCC9 0DBF 5AFA 0F6C C36A F780 4843 2FAE 2A1C
Subkey fingerprint: 966B 0D27 85E9 AEEC 1146 7A9D 8851 1153 6991 ED45
```

出力に「Good signature from "ec2autodiag@amazon.com <EC2 Rescue for Linux>"」という句が含まれる場合は、署名が正常に確認されており、Linux 用 EC2Rescue のインストールスクリプトを実行できることを意味しています。

出力結果に「BAD signature」という句が含まれる場合、手順が正しいことをもう一度確認してください。この応答が続く場合は、Amazon Web Services に連絡してください。以前にダウンロードしたインストールファイルを実行しないでください。

以下は、表示される可能性のある警告の詳細です。

- WARNING: This key is not certified with a trusted signature! There is no indication that the signature belongs to the owner. これは、Linux 用 EC2Rescue の認証済みパブリックキーを所有していると考えるユーザーの個人レベルの信頼を参照します。本来は、ユーザーが Amazon Web Services オフィスを訪問してキーを受け取ることが理想的です。しかし、キーは多くの場合ウェブサイトからダウンロードされます。この場合、ウェブサイトは Amazon Web Services ウェブサイトです。
- gpg2: no ultimately trusted keys found. これは、特定のキーがユーザー（またはユーザーが信頼する他のユーザー）によって「最終的に信頼された」キーでないことを意味します。

詳細については、「<http://www.gnupg.org>」を参照してください。

## Linux 用 EC2Rescue の使用

ここでは、このツールを使い始めるために実行できる一般的なタスクについて説明します。

## タスク

- [Linux 用 EC2Rescue の実行 \(p. 1176\)](#)
- [結果をアップロードする \(p. 1176\)](#)
- [バックアップの作成 \(p. 1177\)](#)
- [ヘルプの使用 \(p. 1177\)](#)

## Linux 用 EC2Rescue の実行

次の例に示すように Linux 用 EC2Rescue を実行できます。

Example 例: すべてのモジュールを実行します

すべてのモジュールを実行するには、Linux 用 EC2Rescue をオプションを指定せずに実行します。

```
./ec2rl run
```

一部のモジュールには、ルートアクセスが必要です。ルートユーザーではない場合は、以下のように sudo を使用してこれらのモジュールを実行します。

```
sudo ./ec2rl run
```

Example 例: 特定のモジュールの実行

特定のモジュールのみ実行するには、--only-modules パラメータを使用します。

```
./ec2rl run --only-modules=module_name --arguments
```

たとえば、このコマンドは、dig モジュールを実行して、amazon.com ドメインに対してクエリを実行します。

```
./ec2rl run --only-modules=dig --domain=amazon.com
```

Example 例: 結果の表示

結果を /var/tmp/ec2rl に表示できます。

```
cat /var/tmp/ec2rl/logfile_location
```

例: dig モジュールのログファイルを表示する場合

```
cat /var/tmp/ec2rl/2017-05-11T15_39_21.893145/mod_out/run/dig.log
```

## 結果をアップロードする

AWS サポートが S3 バケットから結果あるいは結果の共有をリクエストする場合、Linux 用 EC2Rescue CLI ツールを使用してそれをアップロードします。Linux 用 EC2Rescue コマンドの出力によって、使用する必要があるコマンドが提供されます。

Example 例: AWS サポートへの結果のアップロード

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/2017-05-11T15_39_21.893145 --support-url="URLProvidedByAWSSupport"
```

Example 例: S3 バケットに結果をアップロードする

```
./ec2rl upload --upload-directory=/var/tmp/ec2rl/2017-05-11T15_39_21.893145 --presigned-url="YourPresignedS3URL"
```

Amazon S3 に署名付きの URL を生成するための詳細については、「[署名付き URL を使用したオブジェクトのアップロード](#)」を参照してください。

## バックアップの作成

次のコマンドを使用して、インスタンス、1つ以上のボリューム、または特定のデバイス ID のバックアップを作成します。

Example 例: Amazon マシンイメージ (AMI) を使用したインスタンスのバックアップ

```
./ec2rl run --backup=ami
```

Example 例: インスタンスに関連付けられるすべてのボリュームのバックアップを作成する

```
./ec2rl run --backup=allvolumes
```

Example 例: 特定のボリュームをバックアップする

```
./ec2rl run --backup=volumeID
```

## ヘルプの使用

Linux 用 EC2Rescue には、詳細を説明したヘルプファイルと利用できる各コマンドの構文が含まれています。

Example 例: 全般的なヘルプの表示

```
./ec2rl help
```

Example 例: 利用できるモジュールを一覧表示する

```
./ec2rl list
```

Example 例: 特定のモジュールのヘルプを表示する

```
./ec2rl help module_name
```

たとえば、dig モジュールのヘルプファイルを表示するには、以下のコマンドを使用します。

```
./ec2rl help dig
```

## EC2Rescue モジュールを開発する

モジュールは、データシリアル化スタンダードである YAML デ書き込まれます。モジュールの YAML ファイルは、モジュールとその属性を示す単一のドキュメントで構成されます。

### モジュールの属性を追加する

次の表には、利用できるモジュールの属性が一覧表示されます。

| 属性        | 説明                                                                                                                                                                                                                                                                                                   |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name      | モジュールの名前。この名前は、長さが 18 文字以下である必要があります。                                                                                                                                                                                                                                                                |
| バージョン     | モジュールのバージョン番号。                                                                                                                                                                                                                                                                                       |
| タイトル      | モジュールの短い説明タイトルです。この値は、長さが 50 文字以下である必要があります。                                                                                                                                                                                                                                                         |
| helptext  | <p>モジュールの拡張された説明。各列は、長さが 75 文字以下である必要があります。必須あるいはオプションでモジュールが引数を消費する場合、helptext 値にこの引数を含めます。</p> <p>例:</p> <pre>helptext: !!str     Collect output from ps for system   analysis   Consumes --times= for number of times to   repeat   Consumes --period= for time period   between repetition</pre> |
| placement | <p>モジュールが実行されるべきステージ。サポートされる値。</p> <ul style="list-style-type: none"> <li>• prediagnostic</li> <li>• run</li> <li>• postdiagnostic</li> </ul>                                                                                                                                                        |
| language  | <p>モジュールコードが書き込まれている言語。サポートされる値。</p> <ul style="list-style-type: none"> <li>• bash</li> <li>• python</li> </ul> <p style="text-align: center;"><b>Note</b></p> <p>Python コードは、Python 2.7.9+ および Python 3.2+ の両方と互換性がある必要があります。</p>                                                                   |
| 修復        | <p>モジュールが修復をサポートするかどうかを示します。サポートされている値は <code>True</code> または <code>False</code> です。</p> <p>この値がない場合、モジュールのデフォルトは <code>False</code> です。修復をサポートしないそれらのモジュールのオプション属性となります。</p>                                                                                                                        |
| コンテンツ     | 全スクリプトコード。                                                                                                                                                                                                                                                                                           |
| 制約        | 制約値を含むオブジェクトの名前。                                                                                                                                                                                                                                                                                     |

| 属性       | 説明                                                                                                                                                                                                       |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ドメイン     | <p>モジュールがどのようにグループ化または分類されているかの説明。含まれているモジュール一連は次のドメインを使用します。</p> <ul style="list-style-type: none"> <li>・ 同時接続の</li> <li>・ net</li> <li>・ os</li> <li>・ パフォーマンス</li> </ul>                              |
| class    | <p>モジュールによって実行されるタスクの種類の説明。含まれているモジュール一連は次のクラスを使用します。</p> <ul style="list-style-type: none"> <li>・ 回収(プログラムからの出力を回収します)</li> <li>・ 診断(一連の基準の達成/未達成)</li> <li>・ 収集(ファイルのコピーと特定のファイルへの書き込み)</li> </ul>     |
| distro   | <p>このモジュールがサポートする Linux ディストリビューションの一覧。含まれているモジュール一連は次のディストリビューションを使用します。</p> <ul style="list-style-type: none"> <li>・ alami (Amazon Linux)</li> <li>・ rhel</li> <li>・ Ubuntu</li> <li>・ suse</li> </ul> |
| 必須       | CLI オプションからモジュールが消費する必要な引数。                                                                                                                                                                              |
| optional | モジュールが使用できるオプションの引数。                                                                                                                                                                                     |
| ソフトウェア   | モジュールで使用される実行可能なソフトウェア。この属性は、デフォルトでインストールされないソフトウェアの特定を行います。Linux 用 EC2Rescue ロジックは、モジュールを実行する前に、このプログラムが存在し、実行可能であることを確認します。                                                                            |
| package  | 実行ファイル用のソースソフトウェアパッケージ。この属性は、ソフトウェアのパッケージにダウンロード用 URL やそのほかの詳細などの詳しい情報を提供するためのものです。                                                                                                                      |
| sudo     | <p>ルートアクセスがモジュールの実行に必要であるかどうかを示します。</p> <p>モジュールスクリプトで sudo チェックを行う必要はありません。値が true になると、Linux 用 EC2Rescue ロジックは実行しているユーザーがルートアクセスを所持している場合にのみモジュールを実行します。</p>                                           |

| 属性                | 説明                                                                                               |
|-------------------|--------------------------------------------------------------------------------------------------|
| perfimpact        | モジュールが実行している環境に重要な影響を及ぼす可能性があるかどうかを示します。値が true であり、--perfimpact=true 引数が存在しない場合、モジュールはスキップされます。 |
| parallelexclusive | 相互占有を必要とするプログラムを特定します。たとえば、「bpf」を指定するすべてのモジュールはシリアル方法で実行します。                                     |

## 環境変数を追加する

次の表には、利用できるモジュールの属性が一覧表示されます。

| 環境変数              | Description                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EC2RL_CALLPATH    | ec2rl.py へのパス。このパスを使用すると、lib ディレクトリを見つけて、ベンダーの Python モジュールを使用できます。                                                                                                 |
| EC2RL_WORKDIR     | 診断ツールの主要な tmp ディレクトリ。<br>デフォルト値: /var/tmp/ec2rl.                                                                                                                    |
| EC2RL_RUNDIR      | すべての出力が保存されているディレクトリ。<br>デフォルト値: /var/tmp/ec2rl/<br><date&timestamp>.                                                                                               |
| EC2RL_GATHEREDDIR | 収集されたモジュールデータを配置するルート ディレクトリ。<br>デフォルト値: /var/tmp/ec2rl/<br><date&timestamp>/mod_out/gathered/.                                                                     |
| EC2RL_NET_DRIVER  | 初めて使用されるドライバーが、インスタンスの非仮想ネットワークインターフェースでアルファベット順に順序付けされます。<br><br>例: <ul style="list-style-type: none"><li>• xen_netfront</li><li>• ixgbevf</li><li>• ena</li></ul> |
| EC2RL_SUDO        | Linux 用 EC2Rescue ガルートとして実行されている場合には true、そうでない場合には false。                                                                                                          |
| EC2RL_VIRT_TYPE   | インスタンスマタデータから提供される仮想化タイプ。<br><br>例: <ul style="list-style-type: none"><li>• default-hvm</li><li>• default-paravirtual</li></ul>                                     |

| 環境変数             | Description                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------------------|
| EC2RL_INTERFACES | システム上のインターフェースの列挙一覧。この値は、eth0 や eth1 などの名前が含まれる文字列です。これは functions.bash を介して生成され、これをソースとするモジュールのみで利用できます。 |

## YAML 構文を使用する

モジュール YAML ファイルを構築する際、以下に注意してください。

- 3 つのハイフン (---) は、ドキュメントの明示的な開始を示します。
- `!ec2rlcore.module.Module` タグは、データストリームからオブジェクトを作成する際にどのコンストラクタを呼び出すかを YAML パーサーに伝えます。`module.py` ファイル内コンストラクタを検索できます。
- `!!str` タグは、データの種類を決定する試行を行なわず、代わりにコンテンツを文字列リテラルとして解釈するように YAML パーサーに伝えます。
- パイプ文字 (`|`) は、値がリテラル形式のスカラーであることを YAML パーサーに伝えます。この場合、パーサーにはすべての空白が含まれます。インデントと改行文字が保持されるため、これはモジュールにとって重要です。
- YAML スタンダードインデントは 2 つのスペースとなり、次の例で示されます。スクリプトでスタンダードインデント (たとえば、Python では 4 つの空白) を維持して、モジュールファイル内で全コンテンツを 2 つのスペースでインデントすることを確認します。

## モジュールの例

例 1 (mod.d/ps.yaml):

```
--- !ec2rlcore.module.Module
# Module document. Translates directly into an almost-complete Module object
name: !!str ps
path: !!str
version: !!str 1.0
title: !!str Collect output from ps for system analysis
helptext: !!str |
    Collect output from ps for system analysis
    Requires --times= for number of times to repeat
    Requires --period= for time period between repetition
placement: !!str run
package:
  - !!str
language: !!str bash
content: !!str |
  #!/bin/bash
  error_trap()
  {
    printf "%0.s=" {1..80}
    echo -e "\nERROR: \"$BASH_COMMAND\" exited with an error on line ${BASH_LINENO[0]}"
    exit 0
  }
  trap error_trap ERR

  # read-in shared function
  source functions.bash
  echo "I will collect ps output from this $EC2RL_DISTRO box for $times times every $period
seconds."
```

```
for i in $(seq 1 $times); do
    ps auxww
    sleep $period
done
constraint:
requires_ec2: !!str False
domain: !!str performance
class: !!str collect
distro: !!str alami ubuntu rhel suse
required: !!str period times
optional: !!str
software: !!str
sudo: !!str False
perfimpact: !!str False
parallelexclusive: !!str
```

## 診断割り込みの送信 (上級ユーザーのみ)

### Warning

診断割り込みは、上級ユーザーが使用することを想定したものです。誤って使用すると、インスタンスに悪影響を及ぼす可能性があります。インスタンスに診断割り込みを送信すると、インスタンスのクラッシュと再起動がトリガーされ、それによりデータが失われる可能性があります。

到達できないまたは応答しない Linux インスタンスに診断割り込みを送信して、カーネルパニックを手動でトリガーできます。

Linux オペレーティングシステムは一般的に、カーネルパニックが発生するとクラッシュして再起動されます。ただし、オペレーティングシステムの具体的な動作は設定によって異なります。カーネルパニックは、インスタンスのオペレーティングシステムカーネルでクラッシュダンプファイルの生成などのタスクを実行するためにも使用できます。このクラッシュダンプファイル内の情報を使用すると、根本原因解析を実施してインスタンスのデバッグを行うことができます。

クラッシュダンプデータは、インスタンスの代わりにオペレーティングシステムによってローカルで生成されます。

インスタンスに診断割り込みを送信する前に、オペレーティングシステムのドキュメントを参照し、必要な設定変更を行うことをお勧めします。

### コンテンツ

- [サポートされるインスタンスタイプ \(p. 1182\)](#)
- [前提条件 \(p. 1182\)](#)
- [診断割り込みの送信 \(p. 1185\)](#)

## サポートされるインスタンスタイプ<sup>¶</sup>

A1 を除き、診断割り込みはすべての Nitro ベースインスタンスタイプでサポートされます。詳細については、「[Nitro ベースのインスタンス \(p. 187\)](#)」を参照してください。

## 前提条件

診断割り込みを使用する前に、インスタンスのオペレーティングシステムを設定する必要があります。これにより、カーネルパニックの発生時に必要なアクションをオペレーティングシステムで実行できます。

カーネルパニックの発生時にクラッシュダンプが生成されるように Amazon Linux 2 を設定するには

1. インスタンスに接続します。
2. kexec と kdump をインストールします。

```
[ec2-user ~]$ sudo yum install kexec-tools -y
```

3. セカンダリカーネル用に適切な量のメモリが予約されるようにカーネルを設定します。予約するメモリの量は、インスタンスで使用可能な合計メモリによって異なります。適切なテキストエディタを使用して /etc/default/grub ファイルを開き、GRUB\_CMDLINE\_LINUX\_DEFAULT から始まる行を見つけて、crashkernel=*memory\_to\_reserve* という形式で crashkernel パラメータを追加します。たとえば、160MB を予約するには、grub ファイルを次のように変更します。

```
GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=160M console=tty0 console=ttyS0,115200n8
net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
rd.shell=0"
GRUB_TIMEOUT=0
GRUB_DISABLE_RECOVERY="true"
```

4. 変更内容を保存し、grub ファイルを閉じます。
5. GRUB2 設定ファイルを再構築します。

```
[ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

6. Intel および AMD プロセッサをベースとしたインスタンスの場合、send-diagnostic-interrupt コマンドを実行すると 不明なマスク不可割り込み (NMI, unknown non-maskable interrupt) がインスタンスに送信されます。不明な NMI を受信した際にはクラッシュするようにカーネルを設定しておく必要があります。適切なテキストエディタを使用して /etc/sysctl.conf ファイルを開き、以下を追加します。

```
kernel.unknown_nmi_panic=1
```

7. インスタンスを再起動して再接続します。
8. 正しいcrashkernel パラメータを使用してカーネルが起動されていることを確認します。

```
$ grep crashkernel /proc/cmdline
```

次の出力例は、適切な設定を示しています。

```
BOOT_IMAGE=/boot/vmlinuz-4.14.128-112.105.amzn2.x86_64 root=UUID=a1e1011e-
e38f-408e-878b-fed395b47ad6 ro crashkernel=160M console=tty0 console=ttyS0,115200n8
net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 rd.emergency=poweroff
rd.shell=0
```

9. kdump サービスが実行中であることを確認します。

```
[ec2-user ~]$ systemctl status kdump.service
```

次の出力例は、kdump サービスが実行中である場合の結果を示しています。

```
kdump.service - Crash recovery kernel arming
   Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
             enabled)
     Active: active (exited) since Fri 2019-05-24 23:29:13 UTC; 22s ago
   Process: 2503 ExecStart=/usr/bin/kdumpctl start (code=exited, status=0/SUCCESS)
```

```
Main PID: 2503 (code=exited, status=0/SUCCESS)
```

#### Note

デフォルトでは、クラッシュダンプファイルは /var/crash/ に保存されます。保存先を変更するには、適切なテキストエディタを使用して /etc/kdump.conf ファイルを変更します。

カーネルバニックの発生時にクラッシュダンプが生成されるように Amazon Linux を設定するには

1. インスタンスに接続します。
2. kexec と kdump をインストールします。

```
[ec2-user ~]$ sudo yum install kexec-tools -y
```

3. セカンダリカーネル用に適切な量のメモリが予約されるようにカーネルを設定します。予約するメモリの量は、インスタンスで使用可能な合計メモリによって異なります。

```
$ sudo grubby --args="crashkernel=memory_to_reserve" --update-kernel=ALL
```

たとえば、クラッシュカーネル用に 160MB を予約するには、次のコマンドを使用します。

```
$ sudo grubby --args="crashkernel=160M" --update-kernel=ALL
```

4. Intel および AMD プロセッサをベースとしたインスタンスの場合、send-diagnostic-interrupt コマンドを実行すると 不明なマスク不可割り込み (NMI、unknown non-maskable interrupt) がインスタンスに送信されます。不明な NMI を受信した際にはクラッシュするようにカーネルを設定しておく必要があります。適切なテキストエディタを使用して /etc/sysctl.conf ファイルを開き、以下を追加します。

```
kernel.unknown_nmi_panic=1
```

5. インスタンスを再起動して再接続します。
6. 正しいcrashkernel パラメータを使用してカーネルが起動されていることを確認します。

```
$ grep crashkernel /proc/cmdline
```

次の出力例は、適切な設定を示しています。

```
root=LABEL=/ console=tty1 console=ttyS0 selinux=0 nvme_core.io_timeout=4294967295  
LANG=en_US.UTF-8 KEYTABLE=us crashkernel=160M
```

7. kdump サービスが実行中であることを確認します。

```
[ec2-user ~]$ sudo service kdump status
```

サービスが実行中であれば、コマンドから Kdump is operational 応答が返されます。

#### Note

デフォルトでは、クラッシュダンプファイルは /var/crash/ に保存されます。保存先を変更するには、適切なテキストエディタを使用して /etc/kdump.conf ファイルを変更します。

SUSE Linux Enterprise、Ubuntu、または Red Hat Enterprise Linux を設定するには

以下のウェブサイトを参照してください。

- [SUSE Linux Enterprise](#)
- [Ubuntu](#)
- [Red Hat Enterprise Linux \(RHEL\)](#)

## 診断割り込みの送信

必要な設定変更を完了したら、AWS CLI または Amazon EC2 API を使用して、診断割り込みをインスタンスに送信できます。

診断割り込みをインスタンス (AWS CLI) に送信するには

`send-diagnostic-interrupt` コマンドを使用し、インスタンス ID を指定します。

```
aws ec2 send-diagnostic-interrupt --instance-id i-1234567890abcdef0
```

# ドキュメント履歴

次の表は、Amazon EC2 ドキュメントへの重要な追加項目をまとめたものです。また、お客様からいただいたフィードバックに対応するために、ドキュメントを頻繁に更新しています。

[現在の API バージョン: 2016 月 11 月 15 日]

| 機能                             | API バージョン  | 説明                                                                                                                                                                                                                                            | リリース日            |
|--------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| ローカルゾーンを有効にするためのセルフサービスオプション   | 2016-11-15 | ローカルゾーンは、AWS マネジメントコンソールまたは AWS CLI を使用して有効にすることができます。詳細については、「 <a href="#">ローカルゾーンの有効化</a> 」を参照してください。                                                                                                                                       | 2020 年 3 月 5 日   |
| Amazon EBS マルチアタッチ             | 2016-11-15 | 単一のプロビジョンド IOPS SSD (io1) ボリュームを、同じアベイラビリティゾーンにある最大 16 の Nitro ベースのインスタンスにアタッチできるようになりました。詳細については、「 <a href="#">Amazon EBS マルチアタッチを使用した複数のインスタンスへのボリュームのアタッチ (p. 953)</a> 」を参照してください。                                                         | 2020 年 2 月 14 日  |
| AMI に関連付けられたプラットフォームの詳細と請求情報   | 2016-11-15 | オンデマンドインスタンスまたはスポットインスタンスを起動するか、リザーブドインスタンスを購入する前に、Amazon マシンイメージ (AMI) に関連付けられたプラットフォームの詳細と請求情報を確認できます。詳細については、「 <a href="#">請求情報の取得 (p. 161)</a> 」を参照してください。                                                                                | 2020 年 2 月 6 日   |
| スポットインスタンスの停止と起動               | 2016-11-15 | stop 中断動作に頼るのではなく、Amazon EBS によるスポットインスタンスを停止し、それを自由意志で起動できるようになりました。詳細については、「 <a href="#">スポットインスタンスの停止 (p. 342)</a> 」を参照してください。                                                                                                             | 2020 年 1 月 13 日  |
| リソースへのタグ付け                     | 2016-11-15 | Egress-only インターネットゲートウェイ、ローカルゲートウェイ、ローカルゲートウェイルートテーブル、ローカルゲートウェイ仮想インターフェイス、ローカルゲートウェイ仮想インターフェイスグループ、ローカルゲートウェイルートテーブル VPC の関連付け、およびローカルゲートウェイルートテーブル仮想インターフェイスグループの関連付けをタグ付けできます。詳細については、「 <a href="#">リソースにタグを付ける (p. 1121)</a> 」を参照してください。 | 2020 年 1 月 10 日  |
| セッションマネージャー                    | 2016-11-15 | Amazon EC2 コンソールからインスタンスで Session Manager セッションを開始できます。詳細については、「 <a href="#">Session Manager を使用した Linux インスタンスへの接続 (p. 529)</a> 」を参照してください。                                                                                                  | 2019 年 12 月 18 日 |
| EC2 フリートでのオンデマンドキャパシティー予約のサポート | 2016-11-15 | オンデマンドインスタンスの起動時に最初にオンデマンドキャパシティー予約を使用するように、EC2 フリートを設定できます。詳細については、「 <a href="#">EC2 フリート (p. 1121)</a> 」を参照してください。                                                                                                                          | 2019 年 12 月 16 日 |

| 機能                             | API バージョン  | 説明                                                                                                                                                                                                  | リリース日            |
|--------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
|                                |            | は、「 <a href="#">オンデマンドインスタンス用のキャパシティーの予約の使用 (p. 474)</a> 」を参照してください。                                                                                                                                |                  |
| インスタンスタイプの推奨事項                 | 2016-11-15 | AWS Compute Optimizer は、パフォーマンスの向上、コストの削減、またはその両方に役立つ Amazon EC2 インスタンスの推奨事項を提供します。詳細については、「 <a href="#">インスタンスタイプに関する推奨事項の取得 (p. 271)</a> 」を参照してください。                                              | 2019 年 12 月 3 日  |
| Inf1 インスタンス                    | 2016-11-15 | Inf1 インスタンスは、低コストで高いパフォーマンスを実現するように設計された機械学習推論チップである AWS Inferentia を使用します。                                                                                                                         | 2019 年 12 月 3 日  |
| AWS Outposts のサポート             | 2016-11-15 | Outpost サブネットにインスタンスを作成できます。詳細については、 <a href="#">AWS Outposts ユーザーガイド</a> を参照してください。                                                                                                                | 2019 年 12 月 3 日  |
| ローカルゾーン のサポート                  | 2016-11-15 | ローカルゾーン サブネット内にインスタンスを起動できます。詳細については、「 <a href="#">リージョン、アベイラビリティーボード、およびローカルゾーン</a> 」を参照してください。                                                                                                    | 2019 年 12 月 3 日  |
| Dedicated Hosts およびホストリソースグループ | 2016-11-15 | Dedicated Hosts がホストリソースグループで使用できるようになりました。詳細については、「 <a href="#">ホストリソースグループにインスタンスを作成する (p. 403)</a> 」を参照してください。                                                                                   | 2019 年 12 月 2 日  |
| Dedicated Host 共有              | 2016-11-15 | 複数の AWS アカウント間で Dedicated Hosts を共有できるようになりました。詳細については、「 <a href="#">共有 Dedicated Hosts の使用 (p. 416)</a> 」を参照してください。                                                                                | 2019 年 12 月 2 日  |
| アカウントレベルでのデフォルトのクレジット指定        | 2016-11-15 | AWS リージョンごとにアカウントレベルで、バーストパフォーマンスインスタンスファミリーごとにデフォルトのクレジット指定を設定できます。詳細については、「 <a href="#">アカウントのデフォルトのクレジット指定の設定 (p. 226)</a> 」を参照してください。                                                           | 2019 年 11 月 25 日 |
| Dedicated Host                 | 2016-11-15 | インスタンスファミリー内にある複数のインスタンスタイプをサポートするように Dedicated Host を設定できます。詳細については <a href="#">Dedicated Hosts の使用 (p. 398)</a> を参照してください。                                                                        | 2019 年 11 月 21 日 |
| Amazon EBS 高速スナップショット復元        | 2016-11-15 | EBS スナップショットに対して高速スナップショット復元を有効にすることができます。これにより、スナップショットから作成された EBS ボリュームは、作成時に完全に初期化された状態になり、プロビジョニングドパフォーマンスをすべて即座に提供できます。詳細については、「 <a href="#">Amazon EBS 高速スナップショット復元 (p. 1024)</a> 」を参照してください。 | 2019 年 11 月 20 日 |

| 機能                            | API バージョン  | 説明                                                                                                                                                                | リリース日            |
|-------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| インスタンスマタデータサービスバージョン 2        | 2016-11-15 | インスタンスマタデータのリクエストに、セッション志向な方法であるインスタンスマタデータサービスバージョン 2 を使用できます。詳細については <a href="#">インスタンスマタデータサービスの構成 (p. 594)</a> を参照してください。                                     | 2019 年 11 月 19 日 |
| Elastic Fabric Adapter        | 2016-11-15 | Elastic Fabric Adapters を インテル MPI 2019 Update 6 と併用できるようになりました。詳細については、「 <a href="#">EFA および MPI の開始方法 (p. 765)</a> 」を参照してください。                                  | 2019 年 11 月 15 日 |
| オンデマンド Windows インスタンスの休止のサポート | 2016-11-15 | オンデマンド Windows インスタンスを休止できます。サポート対象の Windows AMI の詳細については、「 <a href="#">休止の前提条件 (p. 534)</a> 」を参照してください。                                                          | 2019 年 10 月 14 日 |
| リザーブドインスタンスの購入予約のキュー登録        | 2016-11-15 | リザーブドインスタンス の購入予約を最大 3 年先までキューに入れることができます。詳細については、「 <a href="#">購入をキューに入れる (p. 294)</a> 」を参照してください。                                                               | 2019 年 10 月 4 日  |
| G4 インスタンス                     | 2016-11-15 | G4 インスタンスは NVIDIA Tesla GPU を使用して、汎用 GPU コンピューティング用のコスト効率とパフォーマンスに優れたプラットフォームを CUDA を通じて提供するか、グラフィックアプライケーションを備えた機械学習フレームワークを DirectX または OpenGL を通じて提供します。       | 2019 年 9 月 19 日  |
| 診断割り込み                        | 2016-11-15 | 到達できない、または応答しないインスタンスに診断割り込みを送信して、カーネルパニック (Linux)、ブルースクリーン/停止エラー (Windows インスタンス) をトリガーできます。詳細については、「 <a href="#">診断割り込みの送信 (上級ユーザーのみ) (p. 1182)</a> 」を参照してください。 | 2019 年 8 月 14 日  |
| 容量が最適化された配分戦略                 | 2016-11-15 | EC2 フリート または スポットフリート を使用して、起動するインスタンスの数に最適な容量でスポットプールからスポットインスタンス を起動できるようになりました。詳細については、「 <a href="#">容量最適化のための EC2 フリート の設定 (p. 473)</a> 」を参照してください。           | 2019 年 8 月 12 日  |
| オンデマンドキャパシティー予約               | 2016-11-15 | AWS アカウント間で キャパシティーの予約 を共有できるようになりました。詳細については、「 <a href="#">共有 キャパシティーの予約 の使用 (p. 439)</a> 」を参照してください。                                                            | 2019 年 7 月 29 日  |
| Elastic Fabric Adapter        | 2016-11-15 | EFA では、Open MPI 3.1.4 および Intel MPI 2019 Update 4 がサポートされるようになりました。詳細については、「 <a href="#">Elastic Fabric Adapter (p. 763)</a> 」を参照してください。                          | 2019 年 7 月 26 日  |
| 作成時に起動テンプレートにタグ付け             | 2016-11-15 | 作成時に起動テンプレートに対してタグ付けできます。詳細については、「 <a href="#">リソースにタグを付ける (p. 1121)</a> 」を参照してください。                                                                              | 2019 年 7 月 24 日  |

| 機能                                                | API バージョン  | 説明                                                                                                                                                                                          | リリース日           |
|---------------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 最大合計料金                                            | 2016-11-15 | すべての オンデマンドインスタンス および スポットインスタンス の最大時間料金を EC2 フリー フリートと スポットフリート の両方で指定できます。 詳細については、EC2 フリートの「 <a href="#">使用量の管 理 (p. 475)</a> 」と スポットフリートの「 <a href="#">使用量の管 理 (p. 328)</a> 」を参照してください。 | 2014 年 7 月 1 日  |
| EC2 Instance Connect                              | 2016-11-15 | EC2 Instance Connect は、Secure Shell (SSH) を 使用してインスタンスに接続するシンプルで安 全な方法です。 詳細については、「 <a href="#">EC2 Instance Connect を使用して Linux インスタンスに接続す る (p. 511)</a> 」を参照してください。                      | 2019 年 6 月 27 日 |
| ホスト復旧                                             | 2016-11-15 | Dedicated Host で予期しないハードウェア障害が 発生した場合にインスタンスを新しいホストで自 動的に再起動します。 詳細については、「 <a href="#">ホスト 復旧 (p. 420)</a> 」を参照してください。                                                                      | 2019 年 6 月 5 日  |
| Amazon EBS マルチ ボリュームスナップ ショット                     | 2016-11-15 | EC2 インスタンスにアタッチされている複数の EBS ボリューム間で、正確なポイントインタイム で、データ調整済みのクラッシュ整合性スナップ ショットを取得します。                                                                                                         | 2019 年 5 月 29 日 |
| デフォルトでの Amazon EBS 暗号化                            | 2016-11-15 | リージョンでデフォルトで暗号化を有効にすると、そのリージョンで作成するすべての新しい EBS ボリュームは EBS 暗号化のデフォルトの CMK を使用して暗号化されます。 詳細については、「 <a href="#">デフォルトでの暗号化 (p. 1017)</a> 」を参照してく ださい。                                          | 2019 年 5 月 23 日 |
| VPC エンドポイント、 エンドポイントサービ ス、およびエンドポイ ントサービス設定のタ グ付け | 2016-11-15 | VPC エンドポイント、エンドポイントサービス、 およびエンドポイントサービス設定にタグ付 けできます。 詳細については、「 <a href="#">リソースにタグを付 ける (p. 1121)</a> 」を参照してく ださい。                                                                          | 2019 年 5 月 13 日 |
| I3en インスタンス                                       | 2016-11-15 | 最大 100 Gbps のネットワーク帯域幅の新しいイ ンスタンス                                                                                                                                                           | 2019 年 5 月 8 日  |
| Elastic Fabric Adapter                            | 2016-11-15 | High Performance Computing (HPC) アプリケー ションを高速化するには、Elastic Fabric Adapter をインスタンスに接続します。 詳細については、「 <a href="#">Elastic Fabric Adapter (p. 763)</a> 」を参照してく ださい。                             | 2019 年 4 月 29 日 |
| T3a インスタンス                                        | 2016-11-15 | AMD EYPC プロセッサを搭載した新しいインスタ ンス。                                                                                                                                                              | 2019 年 4 月 24 日 |
| M5ad および R5ad イン スタンス                             | 2016-11-15 | AMD EYPC プロセッサを搭載した新しいインスタ ンス。                                                                                                                                                              | 2019 年 3 月 27 日 |
| Dedicated Host 予約の タグ付け                           | 2016-11-15 | Dedicated Host 予約にタグを付けることできま す。 詳細については、「 <a href="#">Dedicated Host の予約 の タグ付け (p. 415)</a> 」を参照してく ださい。                                                                                  | 2019 年 3 月 14 日 |

| 機能                                 | API バージョン  | 説明                                                                                                                                                                                                                                                        | リリース日            |
|------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| M5、M5d、R5、R5d、および z1d のベアメタルインスタンス | 2016-11-15 | ホストサーバーの物理リソースにアプリケーションが直接アクセスできるようにする新しいインスタンス。                                                                                                                                                                                                          | 2019 年 2 月 13 日  |
| パーティションプレイスメントグループ                 | 2016-11-15 | パーティションプレイスメントグループはインスタンスを複数の論理パーティションに分散させ、基盤となるハードウェアを 1 つのパーティション内のインスタンスが他のパーティション内のインスタンスと共有しないようにします。 詳細については、「 <a href="#">パーティションプレイスメントグループ (p. 793)</a> 」を参照してください。                                                                             | 2018 年 12 月 20 日 |
| p3dn.24xlarge インスタンス               | 2016-11-15 | 新しい p3dn.24xlarge インスタンスは 100 Gbps のネットワーク帯域幅を提供します。                                                                                                                                                                                                      | 2018 年 12 月 07 日 |
| EC2 Linux インスタンスの休止                | 2016-11-15 | 休止が有効になっており、前提条件を満たしている場合は、Linux インスタンスを休止状態にすることができます。 詳細については、「 <a href="#">Linux インスタンスの休止 (p. 532)</a> 」を参照してください。                                                                                                                                    | 2018 年 11 月 28 日 |
| Amazon Elastic Inference アクセラレーター  | 2016-11-15 | Amazon EI アクセラレーターをインスタンスにアタッチし、GPU アクセラレーションを追加することで、深層学習の推論を実行するコストを削減できます。 詳細については、「 <a href="#">Amazon Elastic Inference (p. 623)</a> 」を参照してください。                                                                                                     | 2018 年 11 月 28 日 |
| 100 Gbps のネットワーク帯域幅のインスタンス         | 2016-11-15 | 新しい C5n インスタンスは、最大 100 Gbps のネットワーク帯域幅を利用します。                                                                                                                                                                                                             | 2018 年 11 月 26 日 |
| Arm ベースプロセッサのインスタンス                | 2016-11-15 | 新しい A1 インスタンスは、コストを大幅に削減し、スケールアウトや Arm ベースのワークロードに最適です。                                                                                                                                                                                                   | 2018 年 11 月 26 日 |
| スポットコンソールはインスタンス群を推奨します            | 2016-11-15 | スポットコンソールは、アプリケーションのニーズに合わせた最小限のハードウェア仕様 (vCPU、メモリ、およびストレージ) を満たすために、スポットのベストプラクティス (インスタンスの多様化)に基づいたインスタンス群を推奨します。 詳細については、「 <a href="#">スポットフリートリクエストを作成する (p. 350)</a> 」を参照してください。                                                                     | 2018 年 11 月 20 日 |
| 新しい EC2 フリー リクエストタイプ: instant      | 2016-11-15 | EC2 フリートは、インスタンスタイプと購入モデル全体で同時にキャパシティーをプロビジョニングするために使用できる新しいリクエストタイプ、instant をサポートしています。 instant リクエストは、インスタンスを起動するかどうかおよびいつ起動するかについて制御する、API レスポンスで起動されたインスタンスを返します。 それ以上のアクションは実行しません。 詳細については、「 <a href="#">EC2 フリートのリクエストタイプ (p. 471)</a> 」を参照してください。 | 2018 年 11 月 14 日 |

| 機能                                  | API バージョン  | 説明                                                                                                                                                                                                                                                | リリース日            |
|-------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| AMD EYPC プロセッサのインスタンス               | 2016-11-15 | 新しい 汎用 (M5a) およびメモリ最適化インスタンス (R5a) は、マイクロサービス、中小規模のデータベース、仮想デスクトップ、開発およびテスト環境、ビジネスアプリケーションなどに低価格オプションを提供します。                                                                                                                                      | 2018 年 11 月 06 日 |
| スポット削減額情報                           | 2016-11-15 | 単一の スポットフリート またはすべての スポットインスタンス に対して スポットインスタンス を使用することによって得られた削減額を表示することができます。詳細については、「 <a href="#">スポットインスタンス 購入による削減額 (p. 333)</a> 」を参照してください。                                                                                                 | 2018 年 11 月 05 日 |
| CPU オプションを最適化するためのコンソールでのサポート       | 2016-11-15 | インスタンスを起動するとき、Amazon EC2 を使用して特定のワークロードやビジネスニーズ に合うように CPU オプションを最適化できます。詳細については、「 <a href="#">CPU オプションの最適化 (p. 571)</a> 」を参照してください。                                                                                                             | 2018 年 10 月 31 日 |
| インスタンスから起動テンプレートを作成するためのコンソールでのサポート | 2016-11-15 | Amazon EC2 コンソールを使用した新しい起動テンプレートのためのベースとしてインスタンスを使用して起動テンプレートを作成できます。詳細については、「 <a href="#">起動テンプレートの作成 (p. 456)</a> 」を参照してください。                                                                                                                   | 2018 年 10 月 30 日 |
| オンデマンドキャパシティー予約                     | 2016-11-15 | 特定のアベイラビリティーゾーンの Amazon EC2 インスタンスに対して任意の期間キャパシティーを予約できます。これにより、リザーブドインスタンス (RI) が提供する請求割引とは独立して、キャパシティー予約を登録および管理することができます。詳細については、「 <a href="#">オンデマンドキャパシティー予約 (p. 431)</a> 」を参照してください。                                                        | 2018 年 10 月 25 日 |
| 自分の IP アドレスを使用する (BYOIP)            | 2016-11-15 | すべての公開 IPv4 アドレスの範囲の一部またはすべてをオンプレミスのネットワークから AWS アカウントに導入できます。アドレス範囲を AWS に設定すると、そのアドレス範囲はアドレスプールとしてアカウントに表示されます。アドレスプールから Elastic IP アドレスを作成し、それを AWS リソースと共に使用することができます。詳細については、「 <a href="#">自分の IP アドレスを使用する (BYOIP) (p. 701)</a> 」を参照してください。 | 2018 年 10 月 23 日 |
| g3s.xlarge インスタンス                   | 2016-11-15 | g3s.xlarge インスタンスの導入により、Accelerated Computing G3 インスタンスマリナー の範囲を拡張します。                                                                                                                                                                            | 2018 年 10 月 11 日 |
| 作成時の Dedicated Host タグとコンソールのサポート   | 2016-11-15 | 作成時に Dedicated Hosts タグを付けることができ、Amazon EC2 コンソールを使用して Dedicated Host タグを管理できます。詳細については、「 <a href="#">Dedicated Hosts の割り当て (p. 399)</a> 」を参照してください。                                                                                              | 2018 年 10 月 08 日 |

| 機能                                   | API バージョン  | 説明                                                                                                                                                                                       | リリース日           |
|--------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| ハイメモリインスタンス                          | 2016-11-15 | これらのインスタンスは、大きなメモリ内のデータベースを実行するように設計されています。ホストハードウェアに直接アクセスできるペアメタルパフォーマンスを提供します。詳細については、「 <a href="#">メモリ最適化インスタンス (p. 236)</a> 」を参照してください。                                             | 2018 年 9 月 27 日 |
| f1.4xlarge インスタンス                    | 2016-11-15 | f1.4xlarge インスタンスの導入により、Accelerated Computing F1 インスタンスマリナーの範囲を拡張します。                                                                                                                    | 2018 年 9 月 25 日 |
| スポットフリート のスケジュールに基づくスケーリングのコンソールサポート | 2016-11-15 | 日付と時刻に基づいて、フリート現在の容量を増減させます。詳細については、「 <a href="#">スケーリングのスケジュールを使用したスポットフリート のスケール (p. 378)</a> 」を参照してください。                                                                              | 2018 年 9 月 20 日 |
| T3 インスタンス                            | 2016-11-15 | T3 インスタンスは、次世代のバースタブルな汎用インスタンスタイプで、ベースラインレベルの CPU パフォーマンスを提供しながら、いつでも必要なだけ CPU 使用をバーストできる機能を備えています。詳細については、「 <a href="#">バースト可能パフォーマンスインスタンス (p. 199)</a> 」を参照してください。                    | 2018 年 8 月 21 日 |
| EC2 フリート の配分戦略                       | 2016-11-15 | オンデマンド容量を料金 (最初に最低価格) または優先度 (最初に最も高い優先度) に従って達成するかどうかを指定できます。ターゲットスポット容量を割り当てる先のスポットプールの数を指定できます。詳細については、「 <a href="#">スポットインスタンス の配分戦略 (p. 472)</a> 」を参照してください。                        | 2018 年 7 月 26 日 |
| スポットフリート の配分戦略                       | 2016-11-15 | オンデマンド容量を料金 (最初に最低価格) または優先度 (最初に最も高い優先度) に従って達成するかどうかを指定できます。ターゲットスポット容量を割り当てる先のスポットプールの数を指定できます。詳細については、「 <a href="#">スポットインスタンス の配分戦略 (p. 327)</a> 」を参照してください。                        | 2018 年 7 月 26 日 |
| R5 および R5d インスタンス                    | 2016-11-15 | R5 および R5d インスタンスは、ハイパフォーマンスのデータベース、分散型のインメモリキャッシュ、およびインメモリ分析に最適です。R5d インスタンスは、NVMe インスタンスストアボリュームに付属しています。詳細については、「 <a href="#">メモリ最適化インスタンス (p. 236)</a> 」を参照してください。                   | 2018 年 7 月 25 日 |
| z1d インスタンス                           | 2016-11-15 | これらのインスタンスは、電子設計自動化 (EDA) やリレーショナルデータベースなど、コア単位のハイパフォーマンスと大容量のメモリを必要とする用途向けに設計されています。これらのインスタンスは、NVME インスタンスストアボリュームに付属しています。詳細については、「 <a href="#">メモリ最適化インスタンス (p. 236)</a> 」を参照してください。 | 2018 年 7 月 25 日 |

| 機能                      | API バージョン  | 説明                                                                                                                                                                          | リリース日           |
|-------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| スナップショットライフサイクルの自動化     | 2016-11-15 | Amazon Data Lifecycle Manager を使用して、EBS ボリュームをバックアップするスナップショットの作成と削除を自動化できます。詳細については、「 <a href="#">Amazon EBS スナップショットライフサイクルの自動化 (p. 992)</a> 」を参照してください。                  | 2018 年 7 月 12 日 |
| 起動テンプレートの CPU オプション     | 2016-11-15 | コマンドラインツールを使用して起動テンプレートを作成すると、特定のワークロードまたはビジネスニーズに合わせて CPU オプションを最適化できます。詳細については、「 <a href="#">起動テンプレートの作成 (p. 456)</a> 」を参照してください。                                         | 2018 年 7 月 11 日 |
| Dedicated Hosts のタグ付け   | 2016-11-15 | Dedicated Hosts にタグを付けることができます。詳細については、「 <a href="#">Dedicated Host のタグ付け (p. 410)</a> 」を参照してください。                                                                          | 2018 年 7 月 3 日  |
| i3.metal インスタンス         | 2016-11-15 | i3.metal インスタンスは、プロセッサとメモリなどのホストサーバーの物理リソースにアプリケーションが直接アクセスできるようにします。詳細については、「 <a href="#">ストレージ最適化インスタンス (p. 245)</a> 」を参照してください。                                          | 2018 年 5 月 17 日 |
| 最新のコンソール出力を取得する         | 2016-11-15 | <code>get-console-output</code> AWS CLI コマンドを使用すると、一部のインスタンスタイプの最新のコンソール出力を取得できます。                                                                                          | 2018 年 5 月 9 日  |
| CPU オプションの最適化           | 2016-11-15 | インスタンスを起動するとき、特定のワークロードやビジネスニーズに合うように CPU オプションを最適化できます。詳細については、「 <a href="#">CPU オプションの最適化 (p. 571)</a> 」を参照してください。                                                        | 2018 年 5 月 8 日  |
| EC2 フリート                | 2016-11-15 | EC2 フリートを使用すると、異なる EC2 インスタンスタイプとアベイラビリティーゾーン間、そして オンデマンドインスタンス、リザーブドインスタンス、スポットインスタンス 購入モデル間でインスタンスのグループを起動できます。詳細については、「 <a href="#">EC2 フリートの起動 (p. 468)</a> 」を参照してください。 | 2018 年 5 月 2 日  |
| スポットフリートのオンデマンドインスタンス   | 2016-11-15 | 常にインスタンスキャパシティーを確保できるように、スポットフリートリクエストにオンデマンドキャパシティーのリクエストを含むことができます。詳細については、「 <a href="#">スポットフリートの詳細 (p. 326)</a> 」を参照してください。                                             | 2018 年 5 月 2 日  |
| 作成中の EBS スナップショットをタグ付ける | 2016-11-15 | スナップショット作成時にタグを適用できます。詳細については、「 <a href="#">Amazon EBS スナップショットの作成 (p. 972)</a> 」を参照してください。                                                                                 | 2018 年 4 月 2 日  |
| プレイスメントグループの変更          | 2016-11-15 | インスタンスをプレイスメントグループ内またはプレイスメントグループ外に移動させたり、プレイスメントグループを変更したりできます。詳細については、「 <a href="#">インスタンスのプレイスメントグループの変更 (p. 799)</a> 」を参照してください。                                        | 2018 年 3 月 1 日  |

| 機能                            | API バージョン  | 説明                                                                                                                                                                          | リリース日            |
|-------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| 長いリソース ID                     | 2016-11-15 | より多くのリソースタイプに対して長い ID 形式を有効にできます。詳細については、「 <a href="#">リソース ID (p. 1111)</a> 」を参照してください。                                                                                    | 2018 年 2 月 9 日   |
| ネットワークパフォーマンスの向上              | 2016-11-15 | クラスタプレイスメントグループ外部のインスタンスにおいて、他のインスタンスまたは Amazon S3 との間でネットワークトラフィックを送信または受信する際に、より大きな帯域幅から利点を得られるようになりました。詳細については、「 <a href="#">ネットワーキング機能とストレージ機能 (p. 187)</a> 」を参照してください。 | 2018 年 1 月 24 日  |
| Elastic IP アドレスにタグを付ける        | 2016-11-15 | Elastic IP アドレスにタグを付けることができます。詳細については、「 <a href="#">Elastic IP アドレスのタグ付け (p. 708)</a> 」を参照してください。                                                                           | 2017 年 12 月 21 日 |
| Amazon Linux 2                | 2016-11-15 | Amazon Linux 2 は、Amazon Linux の新しいバージョンです。アプリケーションの高パフォーマンスで安定した安全な基盤が提供されます。詳細については、「 <a href="#">Amazon Linux (p. 165)</a> 」を参照してください。                                    | 2017 年 12 月 13 日 |
| Amazon Time Sync Service のご紹介 | 2016-11-15 | Amazon Time Sync Service を使用して、インスタンスの正確な時刻を維持できます。詳細については、 <a href="#">Linux インスタンスの時刻の設定 (p. 567)</a> を参照してください。                                                          | 2017 年 11 月 29 日 |
| T2 無制限                        | 2016-11-15 | T2 無制限インスタンスは、ベースラインを超えるレベルで必要なだけバーストさせることができます。詳細については、「 <a href="#">バースト可能パフォーマンスインスタンス (p. 199)</a> 」を参照してください。                                                          | 2017 年 11 月 29 日 |
| 起動テンプレート                      | 2016-11-15 | 起動テンプレートにインスタンストを起動させるパラメータの全体または一部を含めることで、インスタンス起動のたびに指定する必要がなくなります。詳細については、「 <a href="#">起動テンプレートからのインスタンスの起動 (p. 454)</a> 」を参照してください。                                    | 2017 年 11 月 29 日 |
| スプレッドプレイスメント                  | 2016-11-15 | スプレッドプレイスメントグループは、少数の重要なインスタンスが互いに分離して保持される必要があるアプリケーションに推奨されます。詳細については、「 <a href="#">スプレッドプレイスメントグループ (p. 793)</a> 」を参照してください。                                             | 2017 年 11 月 29 日 |
| H1 インスタンス                     | 2016-11-15 | H1 インスタンスは、高パフォーマンスビッグデータワークロードの用に設計されています。詳細については、「 <a href="#">ストレージ最適化インスタンス (p. 245)</a> 」を参照してください。                                                                    | 2017 年 11 月 28 日 |
| M5 インスタンス                     | 2016-11-15 | M5 インスタンスは次世代の汎用コンピュートインスタンスです。コンピューティング、メモリ、ストレージおよびネットワークリソースがバランスよく備わっています。                                                                                              | 2017 年 11 月 28 日 |

| 機能                                   | API バージョン  | 説明                                                                                                                                                                                 | リリース日            |
|--------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| スポットインスタンスの休止状態                      | 2016-11-15 | スポットサービスは、中断が発生した場合スポットインスタンスを休止させることができます。詳細については、「 <a href="#">中断したスポットインスタンスの休止 (p. 387)</a> 」を参照してください。                                                                        | 2017 年 11 月 28 日 |
| スポットフリートのターゲットの追跡                    | 2016-11-15 | スポットフリートのターゲット追跡スケーリングポリシーを設定できます。詳細については、「 <a href="#">ターゲット追跡ポリシーを使用したスポットフリートのスケーリング (p. 375)</a> 」を参照してください。                                                                   | 2017 年 11 月 17 日 |
| スポットフリートと Elastic Load Balancing の統合 | 2016-11-15 | スポットフリートに 1 つ以上のロードバランサーをアタッチできます。                                                                                                                                                 | 2017 年 11 月 10 日 |
| X1e インスタンス                           | 2016-11-15 | X1e インスタンスは、高パフォーマンスのデータベース、メモリ内データベース、またその他のメモリ負荷の大きいエンタープライズアプリケーションに最適です。詳細については、「 <a href="#">メモリ最適化インスタンス (p. 236)</a> 」を参照してください。                                            | 2017 年 11 月 28 日 |
| C5 インスタンス                            | 2016-11-15 | C5 インスタンスは、計算量の多いアプリケーション向けに設計されています。詳細については、「 <a href="#">コンピュート最適化インスタンス (p. 231)</a> 」を参照してください。                                                                                | 2017 年 11 月 6 日  |
| コンバータブルリザードインスタンスのマージと分割             | 2016-11-15 | 2 つ以上のコンバータブルリザードインスタンスを新しいコンバータブルリザードインスタンスに交換（マージ）することができます。変更プロセスを使って、コンバータブルリザードインスタンスをより小さな予約に分割することもできます。詳細については、「 <a href="#">コンバータブルリザードインスタンスの交換 (p. 313)</a> 」を参照してください。 | 2017 年 11 月 6 日  |
| P3 インスタンス                            | 2016-11-15 | P3 インスタンスは次世代のコンピューティング最適化 GPU インスタンスです。詳細については、「 <a href="#">Linux 高速コンピューティングインスタンス (p. 253)</a> 」を参照してください。                                                                     | 2017 年 10 月 25 日 |
| VPC のテナント属性を変更する                     | 2016-11-15 | VPC インスタンスのテナント属性を <code>dedicated</code> から <code>default</code> に変更することができます。詳細については、「 <a href="#">VPC のテナント属性の変更 (p. 430)</a> 」を参照してください。                                       | 2017 年 10 月 16 日 |
| 1 秒単位の請求                             | 2016-11-15 | Amazon EC2 は、Linux ベースの秒単位での課金（最低 1 分間分）を行います。                                                                                                                                     | 2017 年 10 月 2 日  |
| 中断時に停止                               | 2016-11-15 | 中断時に Amazon EC2 が スpot インスタンスを停止または終了するかを指定できます。詳細については、「 <a href="#">中断動作 (p. 386)</a> 」を参照してください。                                                                                | 2017 年 9 月 18 日  |
| NAT ゲートウェイのタグ付け                      | 2016-11-15 | NAT ゲートウェイにタグを付けることができます。詳細については、「 <a href="#">リソースにタグを付ける (p. 1121)</a> 」を参照してください。                                                                                               | 2017 年 9 月 7 日   |

| 機能                      | API バージョン  | 説明                                                                                                                                                                                                                                          | リリース日           |
|-------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| セキュリティグループルールの説明        | 2016-11-15 | 説明をセキュリティグループに追加できます。 詳細については、「 <a href="#">セキュリティグループのルール (p. 912)</a> 」を参照してください。                                                                                                                                                         | 2017 年 8 月 31 日 |
| Elastic IP アドレスの復元      | 2016-11-15 | VPC で使用するために Elastic IP アドレスを解放した場合、復元できる可能性があります。 詳細については、「 <a href="#">Elastic IP アドレスの復旧 (p. 711)</a> 」を参照してください。                                                                                                                        | 2017 年 8 月 11 日 |
| スポットフリート インスタンスにタグ付ける   | 2016-11-15 | 起動するインスタンスに自動的にタグを付けるように スpot フリート を設定できます。                                                                                                                                                                                                 | 2017 年 7 月 24 日 |
| G3 インスタンス               | 2016-11-15 | G3 インスタンスは DirectX または OpenGL を使用してグラフィックアプリケーション向けに費用対効果の高パフォーマンスのプラットフォームを提供します。 また、G3 インスタンスは NVIDIA GRID 仮想ワークステーション機能も提供し、最大で 4096x2160 の解像度の 4 つのモニターをサポートします。 詳細については、「 <a href="#">Linux 高速コンピューティングインスタンス (p. 253)</a> 」を参照してください。 | 2017 年 7 月 13 日 |
| F1 インスタンス               | 2016-11-15 | F1 インスタンスは次世代の高速コンピューティングインスタンスです。 詳細については、「 <a href="#">Linux 高速コンピューティングインスタンス (p. 253)</a> 」を参照してください。                                                                                                                                   | 2017 年 4 月 19 日 |
| リソース作成時のタグ付け            | 2016-11-15 | インスタンスやボリュームの作成時にタグを適用できます。 詳細については、「 <a href="#">リソースにタグを付ける (p. 1121)</a> 」を参照してください。 さらに、タグベースのリソースレベルアクセス権限を使用して、適用されているタグを制御できます。 詳細については、「 <a href="#">リソース作成時にタグ付けするアクセス許可の付与 (p. 847)</a> 」を参照してください。                               | 2017 年 3 月 28 日 |
| I3 インスタンス               | 2016-11-15 | I3 インスタンスは次世代のストレージ最適化インスタンスです。 詳細については、「 <a href="#">ストレージ最適化インスタンス (p. 245)</a> 」を参照してください。                                                                                                                                               | 2017 年 2 月 23 日 |
| アタッチされた EBS ボリュームで変更を行う | 2016-11-15 | ほとんどの EC2 インスタンスにアタッチされたほとんどの EBS ボリュームでは、ボリュームをデタッチしたりインスタンスを停止したりせずに、ボリュームのサイズ、タイプ、IOPS を変更できます。 詳細については、「 <a href="#">Amazon EBS Elastic Volumes (p. 1003)</a> 」を参照してください。                                                               | 2017 年 2 月 13 日 |
| IAM ロールをアタッチする          | 2016-11-15 | 既存のインスタンスの IAM ロールをアタッチ、デタッチ、または置換できます。 詳細については、「 <a href="#">Amazon EC2 の IAM ロール (p. 888)</a> 」を参照してください。                                                                                                                                 | 2017 年 2 月 9 日  |

| 機能                                     | API バージョン  | 説明                                                                                                                                                                                                                                                        | リリース日            |
|----------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| 専有 スポットインスタンス                          | 2016-11-15 | Virtual Private Cloud (VPC) のシングルテナント ハードウェアで、スポットインスタンス を実行できます。詳細については、「 <a href="#">スポットインスタンス のテナンシーの指定 (p. 337)</a> 」を参照してください。                                                                                                                      | 2017 年 1 月 19 日  |
| IPv6 サポート                              | 2016-11-15 | VPC とサブネットに IPv6 CIDR を関連付け、VPC のインスタンスに IPv6 アドレスを割り当てることができます。詳細については、「 <a href="#">Amazon EC2 インスタンスの IP アドレッシング (p. 685)</a> 」を参照してください。                                                                                                              | 2016 年 12 月 1 日  |
| R4 インスタンス                              | 2016-09-15 | R4 インスタンスは次世代のメモリ最適化インスタンスです。R4 インスタンスは、ビジネスインテリジェンス (BI)、データマイニングや分析、インメモリデータベース、ウェブスケールの分散インメモリキャッシング、構造化されていないビッグデータのリアルタイム処理を実行するアプリケーションのパフォーマンスなど、メモリを大量に消費し、レイテンシーの影響を受けやすいワークロードに最適です。詳細については、「 <a href="#">メモリ最適化インスタンス (p. 236)</a> 」を参照してください。 | 2016 年 11 月 30 日 |
| 新しい t2.xlarge および t2.2xlarge インスタンスタイプ | 2016-09-15 | T2 インスタンスは、適度なパフォーマンスを実現したり、ワークロードの必要に応じて非常に高いパフォーマンスまでバーストする機能を実現できるように設計されています。これらのインスタンスは、応答性、制限された期間における高いパフォーマンス、および低コストを必要とするアプリケーション向けのインスタンスです。詳細については、「 <a href="#">バースト可能パフォーマンスインスタンス (p. 199)</a> 」を参照してください。                                 | 2016 年 11 月 30 日 |
| P2 インスタンス                              | 2016-09-15 | P2 インスタンスは NVIDIA Tesla K80 GPU を使用し、CUDA または OpenCL プログラミングモデルを使用する汎用 GPU コンピューティング用に設計されています。詳細については、「 <a href="#">Linux 高速コンピューティングインスタンス (p. 253)</a> 」を参照してください。                                                                                     | 2016 年 29 月 9 日  |
| m4.16xlarge インスタンス                     | 2016-04-01 | 64 個の vCPU と 256 GiB の RAM を備えた m4.16xlarge インスタンスの導入に伴い、汎用 M4 ファミリーの範囲を拡張します。                                                                                                                                                                            | 2016 年 6 月 9 日   |
| スポットフリート の自動スケーリング                     |            | スポットフリート のスケーリングポリシーを設定できるようになりました。詳細については、「 <a href="#">スポットフリート のオートスケーリング (p. 373)</a> 」を参照してください。                                                                                                                                                     | 2016 年 9 月 1 日   |
| Elastic Network Adapter (ENA)          | 2016-04-01 | ENA を使用してネットワーキングを強化できるようになりました。詳細については、「 <a href="#">拡張ネットワーキングのタイプ (p. 738)</a> 」を参照ください。                                                                                                                                                               | 2016 年 6 月 28 日  |

| 機能                                                                   | API バージョン  | 説明                                                                                                                                                               | リリース日           |
|----------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 長い ID の表示および変更の拡張サポート                                                | 2016-04-01 | 他の IAM ユーザー、IAM ロール、ルートユーザーの長い ID 設定を表示および変更できるようになりました。詳細については、「 <a href="#">リソース ID (p. 1111)</a> 」を参照してください。                                                  | 2016 年 6 月 23 日 |
| AWS アカウント間での暗号化済み Amazon EBS スナップショットのコピー                            | 2016-04-01 | AWS アカウント間で暗号化済み EBS スナップショットをコピーできるようになりました。詳細については、「 <a href="#">Amazon EBS スナップショットのコピー (p. 977)</a> 」を参照してください。                                              | 2016 年 6 月 21 日 |
| インスタンスコンソールのスクリーンショットの取得                                             | 2015-10-01 | 到達不可のインスタンスをデバッグするときに、追加の情報を取得できるようになりました。詳細については、「 <a href="#">接続できないインスタンスのスクリーンショットの取得 (p. 1169)</a> 」を参照してください。                                              | 2016 年 24 月 5 日 |
| X1 インスタンス                                                            | 2015-10-01 | メモリ内データベース、ビッグデータ処理エンジン、ハイパフォーマンスコンピューティング (HPC) アプリケーションの実行用に設計されたメモリ最適化インスタンス。詳細については、「 <a href="#">メモリ最適化インスタンス (p. 236)</a> 」を参照してください。                      | 2016 年 18 月 5 日 |
| 新しい 2 タイプの EBS ボリューム                                                 | 2015-10-01 | スループット最適化 HDD (st1) と Cold HDD (sc1) ボリュームを作成できるようになりました。詳細については、「 <a href="#">Amazon EBS ボリュームの種類 (p. 933)</a> 」を参照してください。                                      | 2016 年 4 月 19 日 |
| Amazon EC2 用の新しい NetworkPacketsIn と NetworkPacketsOut のメトリクスを追加しました。 |            | Amazon EC2 用の新しい NetworkPacketsIn と NetworkPacketsOut のメトリクスを追加しました。詳細については、「 <a href="#">インスタンスマトリクス (p. 644)</a> 」を参照してください。                                   | 2016 年 3 月 23 日 |
| スポットフリートの CloudWatch メトリクス                                           |            | スポットフリートの CloudWatch メトリクスを取得できるようになりました。詳細については、「 <a href="#">スポットフリートの CloudWatch メトリクス (p. 371)</a> 」を参照してください。                                               | 2016 年 3 月 21 日 |
| スケジュールされたインスタンス                                                      | 2015-10-01 | スケジュールされたリザーブドインスタンス(スケジュールされたインスタンス)によって、毎日、毎週、または毎月ベースの指定された開始時間および期間で繰り返しキャパシティー予約を購入できます。詳細については、「 <a href="#">スケジュールされたりザーブドインスタンス (p. 317)</a> 」を参照してください。 | 2016 年 1 月 13 日 |
| 長いリソース ID                                                            | 2015-10-01 | 一部の Amazon EC2 および Amazon EBS リソースタイプに、段階的に長い ID を導入しています。オプトイン期間中に、サポートされるリソースタイプに対して長い ID 形式を有効にできます。詳細については、「 <a href="#">リソース ID (p. 1111)</a> 」を参照してください。   | 2016 年 1 月 13 日 |

| 機能                                         | API バージョン  | 説明                                                                                                                                                                                                                        | リリース日            |
|--------------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| ClassicLinkDNS サポート                        | 2015-10-01 | VPC の ClassicLink DNS サポートを有効にして、リンクされた EC2-Classic インスタンスと VPC のインスタンス間で対応された DNS ホスト名がプライベート IP アドレスに解決され、パブリック IP アドレスに解決されないようにします。詳細については、「 <a href="#">ClassicLink DNS サポートの有効化 (p. 819)</a> 」を参照してください。            | 2016 年 1 月 11 日  |
| 新しい t2.nano インスタンスタイプ                      | 2015-10-01 | T2 インスタンスは、適度なパフォーマンスを実現したり、ワークロードの必要に応じて非常に高いパフォーマンスまでバーストする機能を実現できるように設計されています。これらのインスタンスは、応答性、制限された期間における高いパフォーマンス、および低コストを必要とするアプリケーション向けのインスタンスです。詳細については、「 <a href="#">バースト可能パフォーマンスインスタンス (p. 199)</a> 」を参照してください。 | 2015 年 12 月 15 日 |
| Dedicated Host                             | 2015-10-01 | Amazon EC2 Dedicated Host は、インスタンス容量を利用したお客様専用の物理サーバーです。詳細については、「 <a href="#">Dedicated Hosts (p. 395)</a> 」を参照してください。                                                                                                    | 2015 年 11 月 23 日 |
| スポットインスタンスの所要時間                            | 2015-10-01 | スポットインスタンス で実行時間を指定できるようになりました。詳細については、「 <a href="#">スポットインスタンス の継続期間の定義 (p. 336)</a> 」を参照してください。                                                                                                                         | 2015 年 10 月 6 日  |
| スポットフリート の変更リクエスト                          | 2015-10-01 | スポットフリート リクエストのターゲット容量が変更できるようになりました。詳細については、「 <a href="#">スポットフリート リクエストの変更 (p. 360)</a> 」を参照してください。                                                                                                                     | 2015 年 9 月 29 日  |
| スポットフリート の分散配分戦略                           | 2015-04-15 | 1 つの スpot フリート リクエストを通して スpot インスタンス を複数の スpot プール に分散することができるようになりました。詳細については、「 <a href="#">スポットインスタンス の配分戦略 (p. 327)</a> 」を参照してください。                                                                                    | 2015 年 9 月 15 日  |
| スポットフリート インスタンスの分量指定                       | 2015-04-15 | アプリケーションのパフォーマンスに影響する各インスタンスのキャパシティーユニットを定義し、スポットプールごとにスポットインスタンスの支払料金を調整することができるようになりました。詳細については、「 <a href="#">スポットフリート インスタンスの分量指定 (p. 329)</a> 」を参照してください。                                                             | 2015 年 8 月 31 日  |
| 新しい再起動アラームアクションと、アラームアクションで使用する新しい IAM ロール |            | 再起動アラームアクションと、アラームアクションで使用する新しい IAM ロールが追加されました。詳細については、「 <a href="#">インスタンスを停止、終了、再起動、または復旧するアラームを作成する (p. 663)</a> 」を参照してください。                                                                                          | 2015 年 7 月 23 日  |

| 機能                                 | API バージョン  | 説明                                                                                                                                                                                                                        | リリース日           |
|------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 新しい t2.large インスタンスタイプ             |            | T2 インスタンスは、適度なパフォーマンスを実現したり、ワークロードの必要に応じて非常に高いパフォーマンスまでバーストする機能を実現できるように設計されています。これらのインスタンスは、応答性、制限された期間における高いパフォーマンス、および低コストを必要とするアプリケーション向けのインスタンスです。詳細については、「 <a href="#">バースト可能パフォーマンスインスタンス (p. 199)</a> 」を参照してください。 | 2015 年 6 月 16 日 |
| M4 インスタンス                          |            | 演算能力、メモリ、およびネットワークリソースを備えた次世代の汎用インスタンス。M4 インスタンスは、AVX2 付きのカスタム Intel 2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) プロセッサを使用します。                                                                                                | 2015 年 6 月 11 日 |
| スポットフリート                           | 2015-04-15 | 個別の スポットインスタンス を管理する代わりに、スポットインスタンス のコレクションまたは フリートを管理できます。詳細については、「 <a href="#">スポットフリート の詳細 (p. 326)</a> 」を参照してください。                                                                                                    | 2015 年 5 月 18 日 |
| Elastic IP アドレスを EC2-Classic に移行する | 2015-04-15 | VPC で使用する EC2-Classic での使用のために割り当てた Elastic IP アドレスを移行できます。詳細については、「 <a href="#">EC2-Classic からの Elastic IP アドレスの移行 (p. 809)</a> 」を参照してください。                                                                              | 2015 年 5 月 15 日 |
| 複数のディスクで構成された VM を AMI としてインポート    | 2015-03-01 | VM Import プロセスにより、複数のディスクで構成された VM を AMI としてインポートできるようになりました。詳細については、『VM Import/Export ユーザーガイド』の「 <a href="#">Importing a VM as an Image Using VM Import/Export</a> 」を参照してください。                                           | 2015 年 4 月 23 日 |
| 新しい g2.8xlarge インスタンスタイプ           |            | 新しい g2.8xlarge インスタンスは 4 つの高性能な NVIDIA GPU を利用し、大規模なレンダリング、トランスクーディング、機械学習、および超並列処理能力を必要とするその他のサーバー側ワークロードを含む GPU コンピューティングワークロードに最適です。                                                                                   | 2015 年 4 月 7 日  |

| 機能              | API バージョン | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | リリース日           |
|-----------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| D2 インスタンス       |           | <p>直接接続型インスタンストレージにある大量のデータへのシーケンシャルアクセスを必要とするアプリケーションに最適化された、次世代の Amazon EC2 高密度ストレージインスタンスです。D2 インスタンスは、高密度ストレージファミリーで最適な料金とパフォーマンスを提供するよう設計されています。2.4 GHz Intel® Xeon® E5 2676v3 (Haswell) プロセッサーを利用する D2 インスタンスは、追加の処理能力、より多くのメモリ、拡張ネットワーキングを提供して HS1 インスタンスでのパフォーマンスを向上させます。さらに、D2 インスタンスには 4 つのインスタンスサイズがあり、6 TB、12 TB、24 TB、48 TB のストレージオプションを利用できます。</p> <p>詳細については、「<a href="#">ストレージ最適化インスタンス (p. 245)</a>」を参照してください。</p>                                  | 2015 年 3 月 24 日 |
| EC2 インスタンスの自動復旧 |           | <p>Amazon CloudWatch インスタンスをモニタリングし、基になるハードウェア障害または AWS による修復を必要とする問題によりインスタンスが正常に機能しなくなった場合に、自動的にインスタンスを復旧する Amazon EC2 アラームを作成できます。復旧されたインスタンスは、インスタンス ID、IP アドレス、すべてのインスタンスマタデータを含め、元のインスタンスと同じです。</p> <p>詳細については、「<a href="#">インスタンスの復旧 (p. 551)</a>」を参照してください。</p>                                                                                                                                                                                              | 2015 年 1 月 12 日 |
| C4 インスタンス       |           | <p>次世代のコンピューティングが最適化されたインスタンスは、経済的な価格で、非常に高い CPU パフォーマンスを提供します。C4 インスタンスは、独自の 2.9 GHz Intel® Xeon® E5-2666 v3 (Haswell) プロセッサで動作します。さらに、Turbo Boost を使用して、C4 インスタンスのプロセッサのクロック速度を、1 個または 2 個のコアで 3.5GHz まで引き上げることができます。C3 コンピューティング最適化インスタンスの機能を拡張することによって、C4 インスタンスは、EC2 インスタンスで最高のプロセッサパフォーマンスをお客様に提供します。これらのインスタンスは、通信量の多いウェブアプリケーション、広告配信、バッチ処理、動画の暗号化、分散分析、高エネルギー物理学、ゲノム分析、計算流体力学に最適です。</p> <p>詳細については、「<a href="#">コンピュート最適化インスタンス (p. 231)</a>」を参照してください。</p> | 2015 年 1 月 11 日 |

| 機能                           | API バージョン  | 説明                                                                                                                                                                                                                                                  | リリース日            |
|------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| ClassicLink                  | 2014-10-01 | ClassicLink を使用すると、EC2-Classic インスタンスをアカウント内の VPC にリンクできます。これによって、VPC のセキュリティグループを EC2-Classic インスタンスに関連付け、プライベート IP アドレスを使用して EC2-Classic インスタンスと VPC 内のインスタンスが通信できるようになります。詳細については、「 <a href="#">ClassicLink (p. 812)</a> 」を参照してください。             | 2015 年 1 月 7 日   |
| スポットインスタンスの終了通知機能            |            | スポットインスタンス の中断から保護する最善の方法は、アプリケーションを耐障害性のある設計にすることです。さらに、スポットインスタンスの終了通知機能を活用できます。これによって、Amazon EC2 がスポットインスタンスを停止または削除する 2 分前に警告が送信されます。<br><br>詳細については、「 <a href="#">スポットインスタンス 中断の通知 (p. 389)</a> 」を参照してください。                                    | 2015 年 1 月 5 日   |
| DescribeVolumes ページ分割サポート    | 2014-09-01 | DescribeVolumes API 呼び出しで、MaxResults パラメータと NextToken パラメータによる結果のページ分割がサポートされるようになりました。詳細については、『Amazon EC2 API Reference』の「 <a href="#">DescribeVolumes</a> 」を参照してください。                                                                             | 2014 年 10 月 23 日 |
| T2 インスタンス                    | 2014-06-15 | T2 インスタンスは、適度なパフォーマンスを実現したり、ワークロードの必要に応じて非常に高いパフォーマンスまでバーストする機能を実現できるように設計されています。これらのインスタンスは、応答性、制限された期間における高いパフォーマンス、および低コストを必要とするアプリケーション向けのインスタンスです。詳細については、「 <a href="#">バースト可能パフォーマンスインスタンス (p. 199)</a> 」を参照してください。                           | 2014 年 30 月 6 日  |
| 新しい [EC2 Service Limits] ページ |            | Amazon EC2 コンソールの [EC2 Service Limits] ページを使用して、Amazon EC2 や Amazon VPC から提供されるリソースに対するリージョンごとの現在の制限を表示できます。                                                                                                                                        | 2014 年 6 月 19 日  |
| Amazon EBS 汎用 SSD ボリューム      | 2014-05-01 | 汎用 SSD ボリュームには、さまざまなワークロードに対応できるコスト効率の高いストレージが用意されています。これらのボリュームでは、レイテンシーは 1 桁台のミリ秒であり、長時間 3,000 IOPS にバーストでき、最大 3 IOPS/GiB のベースパフォーマンスを提供します。汎用 SSD ボリュームのサイズは、1 GiB~1 TiB になります。詳細については、「 <a href="#">汎用 SSD (gp2) ボリューム (p. 935)</a> 」を参照してください。 | 2014 年 6 月 16 日  |

| 機能                        | API バージョン  | 説明                                                                                                                                                                                                                                                                                                              | リリース日            |
|---------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Amazon EBS 暗号化            | 2014-05-01 | Amazon EBS 暗号化により、EBS データボリュームとスナップショットがシームレスに暗号化されるため、セキュアキー管理インフラストラクチャを構築および維持する必要がなくなります。EBS 暗号化サービスは、Amazon が管理するキーを使用してデータを暗号化することにより、保管中のデータのセキュリティを確保します。EC2 インスタンスをホストするサーバーで暗号化が行われるため、EC2 インスタンスと EBS ストレージとの間を移動するデータが暗号化されます。詳細については、「 <a href="#">Amazon EBS Encryption (p. 1014)</a> 」を参照してください。 | 2014 年 5 月 21 日  |
| R3 インスタンス                 | 2014-02-01 | RAM の GiB 当たりの価格が最適に設定された高パフォーマンスの次世代型メモリ最適化インスタンス。これらのインスタンスは、R3 インスタンスの vCPU ごとの大容量メモリ、優れた処理能力、拡張ネットワーキング機能によって恩恵を受けるリレーションナルおよび NoSQL データベース、インメモリ分析ソリューション、科学計算、およびその他のメモリ集約型アプリケーションに適しています。<br><br>各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「 <a href="#">Amazon EC2 インスタンスタイプ</a> 」を参照してください。                | 2014 年 4 月 9 日   |
| 新しい Amazon Linux AMI リリース |            | Amazon Linux AMI 2014.03 をリリースしました。                                                                                                                                                                                                                                                                             | 2014 年 3 月 27 日  |
| Amazon EC2 使用状況レポート       |            | Amazon EC2 使用状況レポートは、EC2 の使用コストと使用状況データを示す一連のレポートです。詳細については、「 <a href="#">Amazon EC2 使用状況レポート (p. 1132)</a> 」を参照してください。                                                                                                                                                                                         | 2014 年 1 月 28 日  |
| 追加された M3 インスタンス           | 2013-10-15 | M3 インスタンスのサイズとして m3.medium および m3.large がサポートされるようになりました。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「 <a href="#">Amazon EC2 インスタンスタイプ</a> 」を参照してください。                                                                                                                                                                | 2014 年 1 月 20 日  |
| I2 インスタンス                 | 2013-10-15 | このインスタンスは、非常に高い IOPS を提供し、連續 SSD 書き込みパフォーマンスを向上させるために Linux インスタンスで TRIM をサポートします。また、I2 インスタンスは、インスタンス間のレイテンシーが低いだけでなく、ネットワークのストレスが少なく、パケット毎秒 (PPS) が非常に大きい拡張ネットワーキングもサポートします。詳細については、「 <a href="#">ストレージ最適化インスタンス (p. 245)</a> 」を参照してください。                                                                     | 2013 年 12 月 19 日 |
| 更新された M3 インスタンス           | 2013-10-15 | M3 インスタンスサイズ (m3.xlarge と m3.2xlarge) は、SSD ボリュームと合わせてインスタンストアをサポートするようになりました。                                                                                                                                                                                                                                  | 2013 年 12 月 19 日 |

| 機能                                  | API バージョン  | 説明                                                                                                                                                                                                                                                                                                                             | リリース日            |
|-------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Linux 仮想マシンのインポート                   | 2013-10-15 | VM Import プロセスが、Linux インスタンスのインポートをサポートするようになりました。 詳細については、『 <a href="#">VM Import/Export ユーザーガイド</a> 』を参照してください。                                                                                                                                                                                                              | 2013 年 12 月 16 日 |
| RunInstances に関するリソースレベルの許可         | 2013-10-15 | AWS Identity and Access Management でポリシーを作成して、Amazon EC2 RunInstances API アクションについてリソースレベルでアクセス許可を制御できるようになりました。 詳細とポリシー例については、 <a href="#">Amazon EC2 の Identity and Access Management (p. 839)</a> を参照してください。                                                                                                                | 2013 年 11 月 20 日 |
| C3 インスタンス                           | 2013-10-15 | 計算能力が最適化されたインスタンスは、経済的な価格で、非常に高い CPU パフォーマンスを提供します。また、C3 インスタンスは、インスタンス間のレイテンシーが低いだけでなく、ネットワークのストレスが少なくパケット毎秒 (PPS) が非常に大きい拡張ネットワーキングもサポートします。これらのインスタンスは、通信量の多いウェブアプリケーション、広告配信、バッチ処理、動画の暗号化、分散分析、高エネルギー物理学、ゲノム分析、計算流体力学に最適です。<br><br>各 Amazon EC2 インスタンスタイプのハードウェア仕様については、『 <a href="#">Amazon EC2 インスタンスタイプ</a> 』を参照してください。 | 2013 年 11 月 14 日 |
| AWS Marketplace からのインスタンスの起動        |            | Amazon EC2 起動ウィザードを使用して AWS Marketplace からインスタンスを起動できるようになりました。 詳細については、『 <a href="#">AWS Marketplace インスタンスの起動 (p. 467)</a> 』を参照してください。                                                                                                                                                                                       | 2013 年 11 月 11 日 |
| G2 インスタンス                           | 2013-10-01 | これらのインスタンスはビデオ作成サービスや 3D 表示、グラフィックスを集中的に使用するアプリケーション、大規模なパラレル処理を必要とするサーバー側のワークロードに最適です。 詳細については、『 <a href="#">Linux 高速コンピューティングインスタンス (p. 253)</a> 』を参照してください。                                                                                                                                                                 | 2013 年 11 月 4 日  |
| 新しい起動ウィザード                          |            | 設計し直された新しい EC2 起動ウィザードが付属します。 詳細については、『 <a href="#">インスタンス起動ウィザードを使用してインスタンスを起動する (p. 449)</a> 』を参照してください。                                                                                                                                                                                                                     | 2013 年 10 月 10 日 |
| Amazon EC2 リザーブドインスタンスのインスタンスタイプの変更 | 2013-10-01 | 同一ファミリー内 (たとえば、M1、M2、M3、C1) であれば Linux リザーブドインスタンスのインスタンスタイプを変更できるようになりました。 詳細については、『 <a href="#">リザーブドインスタンス の変更 (p. 306)</a> 』を参照してください。                                                                                                                                                                                     | 2013 年 10 月 09 日 |
| 新しい Amazon Linux AMI リリース           |            | Amazon Linux AMI 2013.09 をリリースしました。                                                                                                                                                                                                                                                                                            | 2013 年 9 月 30 日  |

| 機能                        | API バージョン  | 説明                                                                                                                                                                                                                                                  | リリース日           |
|---------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Amazon EC2 リザーブドインスタンスの変更 | 2013-08-15 | リージョンのリザーブドインスタンスを変更できるようになりました。詳細については、「 <a href="#">リザーブドインスタンスの変更 (p. 306)</a> 」を参照してください。                                                                                                                                                       | 2013 年 9 月 11 日 |
| パブリック IP アドレスの割り当て        | 2013-07-15 | VPC でインスタンスを起動するときに、パブリック IP アドレスを割り当てることができるようになりました。詳細については、「 <a href="#">インスタンス起動時のパブリック IPv4 アドレスの割り当て (p. 691)</a> 」を参照してください。                                                                                                                 | 2013 年 8 月 20 日 |
| リソースレベルのアクセス許可の付与         | 2013-06-15 | Amazon EC2 は、新しい Amazon Resource Name (ARN) および条件キーをサポートします。詳細については、「 <a href="#">Amazon EC2 の IAM ポリシー (p. 842)</a> 」を参照してください。                                                                                                                    | 2013 年 7 月 8 日  |
| インクリメンタルスナップショットコピー       | 2013-02-01 | インクリメンタルスナップショットコピーを実行できるようになりました。詳細については、「 <a href="#">Amazon EBS スナップショットのコピー (p. 977)</a> 」を参照してください。                                                                                                                                            | 2013 年 6 月 11 日 |
| 新しい [Tags] ページ            |            | Amazon EC2 コンソールに新しい [Tags] ページを追加しました。詳細については、「 <a href="#">Amazon EC2 リソースにタグを付ける (p. 1120)</a> 」を参照してください。                                                                                                                                       | 2013 年 04 月 4 日 |
| 新しい Amazon Linux AMI リリース |            | Amazon Linux AMI 2013.03 をリリースしました。                                                                                                                                                                                                                 | 2013 年 3 月 27 日 |
| 追加された EBS 最適化インスタンスタイプ    | 2013-02-01 | 次のインスタンスタイプが、EBS 最適化インスタンスとして起動できるようになりました:<br>c1.xlarge、m2.2xlarge、m3.xlarge、m3.2xlarge。<br><br>詳細については、「 <a href="#">Amazon EBS – 最適化インスタンス (p. 1031)</a> 」を参照してください。                                                                             | 2013 年 3 月 19 日 |
| リージョン間での AMI のコピー         | 2013-02-01 | AMI をリージョン間でコピーして、整合性のあるインスタンスを複数の AWS リージョンですばやく簡単に起動できます。<br><br>詳細については、「 <a href="#">AMI のコピー (p. 155)</a> 」を参照してください。                                                                                                                          | 2013 年 3 月 11 日 |
| デフォルトの VPC へのインスタンスの起動    | 2013-02-01 | お客様の AWS アカウントは、インスタンスを EC2-Classic または VPC で、または VPC でだけで起動できる場合があり、どちらになるかはリージョンごとに異なります。インスタンスを起動できるのが VPC だけである場合は、デフォルトの VPC が自動的に作成されます。お客様がインスタンスを起動するときは、デフォルトの VPC で起動されるようになります。ただし、お客様が非デフォルト VPC を作成してその VPC でインスタンスを起動するよう指定した場合を除きます。 | 2013 年 3 月 11 日 |

| 機能                                                | API バージョン  | 説明                                                                                                                                                                                                                                                                | リリース日            |
|---------------------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| ハイメモリクラスター (cr1.8xlarge) インスタンスタイプ                | 2012-12-01 | 大容量のメモリと、高パフォーマンスの CPU およびネットワークの組み合わせです。このインスタンスは、インメモリアナリティクス、グラフ分析、科学計算アプリケーションに適しています。                                                                                                                                                                        | 2013 年 1 月 21 日  |
| ハイストレージ (hs1.8xlarge) インスタンスタイプ                   | 2012-12-01 | ハイストレージインスタンスは、ストレージ密度がきわめて高く、インスタンスあたりの順次読み込み/書き込みのパフォーマンスの高さが特徴です。データウェアハウス、Hadoop/MapReduce、並列ファイルシステムに適しています。                                                                                                                                                 | 2012 年 12 月 20 日 |
| EBS スナップショットのコピー                                  | 2012-12-01 | スナップショットのコピーを使用して、データのバックアップを作成したり、新しい Amazon EBS ボリュームを作成したり、Amazon マシンイメージ (AMI) を作成したりすることができます。詳細については、「 <a href="#">Amazon EBS スナップショットのコピー (p. 977)</a> 」を参照してください。                                                                                         | 2012 年 12 月 17 日 |
| プロビジョンド IOPS SSD ボリューム用の EBS メトリクスおよびステータスチェックの更新 | 2012-10-01 | プロビジョンド IOPS SSD ボリュームの 2 つの新しいメトリクスを含むように EBS メトリクスが更新されました。詳細については、「 <a href="#">Amazon EBS の Amazon CloudWatch メトリクス (p. 1060)</a> 」を参照してください。また、プロビジョンド IOPS SSD ボリューム用の新しいステータスチェックを追加しました。詳細については、「 <a href="#">EBS ボリュームステータスチェック (p. 960)</a> 」を参照してください。 | 2012 年 11 月 20 日 |
| Linux カーネル                                        |            | AKI ID を更新し、ディストリビューションカーネルを再編成し、PVOps セクションを更新しました。                                                                                                                                                                                                              | 2012 年 11 月 13 日 |
| M3 インスタンス                                         | 2012-10-01 | 新しいインスタンスタイプとして M3 エクストラージおよび M3 ダブルエクストララージを追加しました。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「 <a href="#">Amazon EC2 インスタンスタイプ</a> 」を参照してください。                                                                                                                       | 2012 年 11 月 10 日 |
| スポットインスタンスのリクエストステータス                             | 2012-10-01 | スポットインスタンス リクエストステータスにより、スポットリクエストの状態を簡単に判断できます。                                                                                                                                                                                                                  | 2012 年 10 月 14 日 |
| 新しい Amazon Linux AMI リリース                         |            | Amazon Linux AMI 2012.09 をリリースしました。                                                                                                                                                                                                                               | 2012 年 10 月 11 日 |
| Amazon EC2 リザードインスタンスマーケットプレイス                    | 2012-08-15 | リザードインスタンスマーケットプレイスでは、不要になった Amazon EC2 リザードインスタンスを持つ販売者と、追加の容量を求める購入者とが、互いに相手を見つけることができます。リザードインスタンスマーケットプレイスを通じて売買されたリザードインスタンスは、他のリザードインスタンス同様に動作します。異なるのは、残りの契約期間が短いことと、異なる価格で販売できることです。                                                                      | 2012 年 9 月 11 日  |

| 機能                                                                 | API バージョン  | 説明                                                                                                                                                                                                                                                                                                                                                                                             | リリース日           |
|--------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Amazon EBS 向けのプロビジョンド IOPS SSD                                     | 2012-07-20 | プロビジョンド IOPS SSD ボリュームは、整合性と応答時間の短さが重要な、データベースアプリケーションなど入出力を多用する作業負荷に対して、予測可能なハイパフォーマンスを提供します。詳細については、「 <a href="#">Amazon EBS ボリュームの種類 (p. 933)</a> 」を参照してください。                                                                                                                                                                                                                               | 2012 年 7 月 31 日 |
| Amazon EC2 向けのハイ I/O インスタンス                                        | 2012-06-15 | ハイ I/O インスタンスは、SSD ベースのローカルインスタンスストレージを使用して低レイテンシーのハイ I/O パフォーマンスを提供します。                                                                                                                                                                                                                                                                                                                       | 2012 年 7 月 18 日 |
| Amazon EC2 インスタンスの IAM ロール                                         | 2012-06-01 | <p>Amazon EC2 向けの IAM ロールは次の機能を提供します。</p> <ul style="list-style-type: none"> <li>Amazon EC2 インスタンスで実行中のアプリケーションの AWS アクセスキー。</li> <li>Amazon EC2 インスタンスの AWS アクセスキーを自動更新します。</li> <li>Amazon EC2 インスタンスで実行中の、AWS サービスにリクエストを送信するアプリケーションに対して、許可をより細かく設定できます。</li> </ul>                                                                                                                        | 2012 年 6 月 11 日 |
| 開始しやすく、中断の可能性に対応しやすくなる スポットインスタンス 機能。                              |            | <p>次のように、スポットインスタンスを管理できるようになりました。</p> <ul style="list-style-type: none"> <li>Auto Scaling 起動設定を使用して スポットインスタンスに支払う金額を指定し、スポットインスタンスに支払う金額を指定するスケジュールを設定します。詳細については、『Amazon EC2 Auto Scaling ユーザーガイド』の「<a href="#">Auto Scaling グループのスポットインスタンスの起動</a>」を参照してください。</li> <li>インスタンスの起動または終了の通知を受け取ることができます。</li> <li>AWS CloudFormation テンプレートを使用して、AWS リソースのスタック内の スポットインスタンスを起動します。</li> </ul> | 2012 年 6 月 7 日  |
| Amazon EC2 のステータスチェックのための EC2 インスタンスのエクスポートとタイムスタンプ                | 2012-05-01 | ステータスチェックが失敗した日時を示す、インスタンスステータスとシステムステータスのタイムスタンプのサポートを追加しました。                                                                                                                                                                                                                                                                                                                                 | 2012 年 5 月 25 日 |
| EC2 インスタンスのエクスポート、および Amazon VPC に対するインスタンスとシステムのステータスチェックのタイムスタンプ | 2012-05-01 | <p>Citrix Xen、Microsoft Hyper-V、および VMware vSphere 向けの EC2 インスタンスのエクスポートのサポートを追加しました。</p> <p>インスタンスとシステムのステータスチェックのタイムスタンプのサポートを追加しました。</p>                                                                                                                                                                                                                                                    | 2012 年 5 月 25 日 |

| 機能                                                                              | API バージョン  | 説明                                                                                                                                                                                                                                       | リリース日            |
|---------------------------------------------------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| クラスター・コンピュート・エイト・エクストラ・ラージ・インスタンス                                               | 2012-04-01 | VPC に cc2.8xlarge インスタンスのサポートを追加しました。                                                                                                                                                                                                    | 2012 年 4 月 26 日  |
| AWS Marketplace AMI                                                             | 2012-04-01 | AWS Marketplace AMI のサポートを追加しました。                                                                                                                                                                                                        | 2012 年 4 月 19 日  |
| 新しい Linux AMI リリース                                                              |            | Amazon Linux AMI 2012.03 をリリースしました。                                                                                                                                                                                                      | 2012 年 3 月 28 日  |
| 新しい AKI バージョン                                                                   |            | AKI バージョン 1.03 および AWS GovCloud (US) リージョン用の AKI をリリースしました。                                                                                                                                                                              | 2012 年 3 月 28 日  |
| ミディアムインスタンス、すべての AMI での 64 ビットのサポート、および Java ベースの SSH クライアント                    | 2011-12-15 | 新しいインスタンスタイプと 64 ビット情報のサポートを追加しました。Java ベースの SSH クライアントを使用して Linux インスタンスに接続する手順を追加しました。                                                                                                                                                 | 2012 年 3 月 7 日   |
| リザーブドインスタンス料金範囲                                                                 | 2011-12-15 | リザーブドインスタンス料金範囲に組み込まれた割引料金の利用方法に関する新しいセクションを追加しました。                                                                                                                                                                                      | 2012 年 3 月 5 日   |
| Amazon Virtual Private Cloud での EC2 インスタンスのための Elastic Network Interfaces (ENI) | 2011-12-01 | VPC での EC2 インスタンスのための Elastic Network Interfaces (ENI) についての新しいセクションを追加しました。詳細については、「 <a href="#">Elastic Network Interface (p. 713)</a> 」を参照してください。                                                                                     | 2011 年 12 月 21 日 |
| 新しい GRU リージョンと AKI                                                              |            | SA-East-1 リージョン用の新しい AKI リリースについての情報を追加しました。このリリースで、AKI バージョン 1.01 は廃止されました。AKI バージョン 1.02 には下位互換性があります。                                                                                                                                 | 2011 年 12 月 14 日 |
| 新しい Amazon EC2 リザーブドインスタンスの提供タイプ                                                | 2011-11-01 | インスタンスの使用目的に応じて、さまざまリザーブドインスタンス提供タイプを選択できるようになりました。                                                                                                                                                                                      | 2011 年 12 月 1 日  |
| Amazon EC2 インスタンスのステータス                                                         | 2011-11-01 | AWS で予定されている、インスタンスに影響を及ぼす可能性のあるイベントなど、インスタンスのステータスについて追加の詳細情報を表示できます。これら運用上の作業には、ソフトウェアアップデートやセキュリティパッチを適用するために必要なインスタンスの再起動や、ハードウェアに問題が生じた場合に必要となるインスタンスの廃棄などが含まれます。詳細については、「 <a href="#">インスタンスのステータスのモニタリング (p. 628)</a> 」を参照してください。 | 2011 年 11 月 16 日 |
| Amazon EC2 クラスター・コンピュート・インスタンスタイプ                                               |            | クラスター・コンピュート・エイト・エクストラ・ラージ (cc2.8xlarge) のサポートを Amazon EC2 に追加しました。                                                                                                                                                                      | 2011 年 11 月 14 日 |
| 新しい PDX リージョンと AKI                                                              |            | 新しい US-West 2 リージョン用の新しい AKI のリリースについての情報を追加しました。                                                                                                                                                                                        | 2011 年 11 月 8 日  |

| 機能                                                       | API バージョン  | 説明                                                                                                                                                                                                                                                       | リリース日           |
|----------------------------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Amazon VPC の スポットインスタンス                                  | 2011-07-15 | Amazon VPC での スポットインスタンス のサポートについての情報を追加しました。この更新により、ユーザーは Virtual Private Cloud (VPC) で スポットインスタンス を起動できます。スポットインスタンス のユーザーは スポットインスタンス を起動することで、Amazon VPC の利点を得ることができます。                                                                              | 2011年10月11日     |
| 新しい Linux AMI リリース                                       |            | Amazon Linux AMI 2011.09 のリリースについての情報を追加しました。このアップデートにより、Amazon Linux AMI からベータのタグが削除され、リポジトリを特定のバージョンにロックする機能がサポートされ、インストールされているパッケージについてセキュリティアップデートを含むアップデートが利用可能になったときに通知が表示されます。                                                                    | 2011年9月26日      |
| CLI ツールのユーザーのための VM Import 処理の簡素化                        | 2011-07-15 | VM Import 処理を簡素化し、ImportInstance と ImportVolume の機能を拡張しました。これにより、インポートタスクの作成後に、イメージが Amazon EC2 にアップロードされます。さらに、ResumeImport の導入により、アップロードが途中で止まった場合にその個所から再開できるようになりました。                                                                                | 2011年9月15日      |
| VHD ファイル形式でのインポートのサポート                                   |            | VM Import で、仮想マシンイメージファイルを VHD 形式でインポートできるようになりました。VHD ファイル形式は、Citrix Xen および Microsoft Hyper-V 仮想化プラットフォームと互換性があります。VM Import はこのリリースで、RAW、VHD、および VMDK (VMware ESX 互換) イメージ形式をサポートしています。詳細については、『 <a href="#">VM Import/Export ユーザーガイド</a> 』を参照してください。 | 2011 年 8 月 24 日 |
| VMware vCenter 用の Amazon EC2 VM Import Connector のアップデート |            | Amazon EC2 VM Import Connector for VMware vCenter 仮想アプライアンス (Connector) の1.1 バージョンについての情報を追加しました。このアップデートには、インターネットアクセスのためのプロキシサポート、エラー処理の向上、タスクプロgresバーの精度向上、および数件のバグ修正が含まれます。                                                                          | 2011年6月27日      |
| ユーザー提供のカーネルを実行するための Linux AMI の有効化                       |            | AKI のバージョン 1.01 から 1.02 への変更についての情報を追加しました。このバージョンでは、t1.micro Linux インスタンスに関連する起動エラーに対処するため、PVGRUB が更新されています。詳細については、「 <a href="#">独自の Linux カーネルを有効にする (p. 176)</a> 」を参照してください。                                                                         | 2011年6月20日      |

| 機能                                 | API バージョン  | 説明                                                                                                                                                                                                                                                                                                     | リリース日           |
|------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| スポットインスタンスのアベイラビリティーゾーンの料金変更       | 2011-05-15 | スポットインスタンス のアベイラビリティーゾーンの料金機能についての情報を追加しました。このリリースでは、スポットインスタンス リクエストおよびスポット料金履歴のクエリを行ったときに返される情報の一部として、新しいアベイラビリティーゾーンの料金オプションが追加されました。これにより、特定のアベイラビリティーゾーンでの スポットインスタンス の起動にかかる料金を調べやすくなりました。                                                                                                       | 2011年5月26日      |
| AWS Identity and Access Management |            | AWS Identity and Access Management (IAM)についての情報を追加しました。IAM を使用すると、Amazon EC2 リソースで通常ユーザーが使用できる Amazon EC2 アクションをユーザーが指定できます。詳細については、「 <a href="#">Amazon EC2 の Identity and Access Management (p. 839)</a> 」を参照してください。                                                                                 | 2011 年 4 月 26 日 |
| ユーザー提供のカーネルを実行するための Linux AMI の有効化 |            | ユーザー提供のカーネルの実行用に PVGRUB Amazon Kernel Image (AKI) を使用するために Linux AMI を有効化する方法についての情報を追加しました。詳細については、「 <a href="#">独自の Linux カーネルを有効にする (p. 176)</a> 」を参照してください。                                                                                                                                        | 2011 年 4 月 26 日 |
| 専用インスタンス                           |            | 専用インスタンスは、ホストのハードウェアレベルで物理的に隔離されているインスタンスであり、Amazon Virtual Private Cloud (Amazon VPC) 内で起動します。専用インスタンスを使うと、Amazon EC2 コンピュートインスタンスをハードウェアレベルで隔離しながら、オンデマンドの弾力性のあるプロビジョニングと、使用した分だけ支払うという料金システムを活用でき、Amazon VPC と AWS クラウドの利点を生かすことができます。詳細については、「 <a href="#">ハードウェア専有インスタンス (p. 425)</a> 」を参照してください。 | 2011 年 3 月 27 日 |
| リザーブドインスタンスに関する AWS マネジメントコンソールの更新 |            | AWS マネジメントコンソールが更新され、より簡単に、リザーブドインスタンスを表示したり、ハードウェア専有リザーブドインスタンスを含む追加のリザーブドインスタンスを購入したりできるようになりました。詳細については、「 <a href="#">リザーブドインスタンス (p. 279)</a> 」を参照してください。                                                                                                                                          | 2011 年 3 月 27 日 |
| 新しい Amazon Linux 参照 AMI            |            | 新しい Amazon Linux 参照 AMI は CentOS 参照 AMI を置き換えます。Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI セクションなど、CentOS 参照 AMI に関する情報を削除しました。                                                                                                                                                    | 2011年3月15日      |
| メタデータ情報                            | 2011-01-01 | 2011-01-01 リリースでの変更を反映するため、メタデータについての情報を追加しました。詳細については、「 <a href="#">インスタンスマタデータとユーザーデータ (p. 593)</a> 」および「 <a href="#">インスタンスマタデータのカテゴリ (p. 612)</a> 」を参照してください。                                                                                                                                      | 2011 年 3 月 11 日 |

| 機能                                                | API バージョン  | 説明                                                                                                                                                                                                              | リリース日            |
|---------------------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Amazon EC2 VM Import Connector for VMware vCenter |            | Amazon EC2 VM Import Connector for VMware vCenter 仮想アプライアンス (Connector) についての情報を追加しました。このコネクタは、VMware vSphere Client と統合するための、VMware vCenter のプラグインです。コネクタの GUI を使用して、VMware 仮想マシンを Amazon EC2 にインポートすることができます。 | 2011年3月3日        |
| ボリュームの強制デタッチ                                      |            | AWS Management Console を使用して、Amazon EBS ボリュームをインスタンスから強制的にデタッチできるようになりました。詳細については、「 <a href="#">インスタンスからの Amazon EBS ボリュームのデタッチ (p. 967)</a> 」を参照してください。                                                        | 2011 年 2 月 23 日  |
| インスタンス終了の防止                                       |            | AWS Management Console を使用してインスタンスの削除を防止できるようになりました。詳細については、「 <a href="#">インスタンスの削除保護の有効化 (p. 547)</a> 」を参照してください。                                                                                              | 2011 年 2 月 23 日  |
| CentOS 5.4 AMI のクラスタインスタンスのクロック同期ずれの修正            |            | Amazon の CentOS 5.4 AMI 上で実行されているクラスタインスタンスのクロック同期ずれを修正する方法についての情報を追加しました。                                                                                                                                      | 2011 年 1 月 25 日  |
| VM Import                                         | 2010-11-15 | 仮想マシンまたはボリュームを Amazon EC2 にインポートする VM Import についての情報を追加しました。詳細については、『 <a href="#">VM Import/Export ユーザーガイド</a> 』を参照してください。                                                                                      | 2010 年 12 月 15 日 |
| インスタンスの基本モニタリング                                   | 2010-08-31 | EC2 インスタンスの基本モニタリングについての情報を追加しました。                                                                                                                                                                              | 2010 年 12 月 12 日 |
| フィルタとタグ                                           | 2010-08-31 | リソースの一覧表示、フィルタリング、およびタグ付けについての情報を追加しました。詳細については、「 <a href="#">リソースのリスト表示とフィルタリング (p. 1116)</a> 」および「 <a href="#">Amazon EC2 リソースにタグを付ける (p. 1120)</a> 」を参照してください。                                               | 2010年9月19日       |
| インスタンス起動時の多重実行禁止                                  | 2010-08-31 | インスタンスを実行する際に多重実行を禁止する方法についての情報を追加しました。詳細については、『 <a href="#">Amazon EC2 API Reference</a> 』の「 <a href="#">べき等性の確保</a> 」を参照してください。                                                                               | 2010年9月19日       |
| マイクロインスタンス                                        | 2010-06-15 | t1.microAmazon EC2 は、特定のタイプのアプリケーションに対して、インスタンスタイプを提供します。詳細については、「 <a href="#">バースト可能パフォーマンスインスタンス (p. 199)</a> 」を参照してください。                                                                                     | 2010年9月8日        |
| Amazon EC2 の AWS Identity and Access Management   |            | Amazon EC2 が AWS Identity and Access Management (IAM) と統合されました。詳細については、「 <a href="#">Amazon EC2 の Identity and Access Management (p. 839)</a> 」を参照してください。                                                       | 2010 年 9 月 2 日   |

| 機能                                            | API バージョン  | 説明                                                                                                                                                       | リリース日           |
|-----------------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| クラスターインスタンス                                   | 2010-06-15 | Amazon EC2 は、高性能コンピューティング (HPC) アプリケーション用にクラスタコンピュートインスタンスを提供します。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「 <a href="#">Amazon EC2 インスタンスタイプ</a> 」を参照してください。 | 2010年7月12日      |
| Amazon VPC IP アドレス指定                          | 2010-06-15 | Amazon VPC ユーザーは、VPC 内で起動されたインスタンスに割り当てる IP アドレスを指定できるようになりました。                                                                                          | 2010年7月12日      |
| Amazon CloudWatch による Amazon EBS ボリュームのモニタリング |            | Amazon CloudWatch による Amazon EBS ボリュームのモニタリングが自動的に行われるようになりました。詳細については、「 <a href="#">Amazon EBS の Amazon CloudWatch メトリクス (p. 1060)</a> 」を参照してください。      | 2010 年 6 月 14 日 |
| ハイメモリエクストラーボリュームインスタンス                        | 2009-11-30 | Amazon EC2 は、ハイメモリエクストララージ (m2.xlarge) インスタンスタイプをサポートするようになりました。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「 <a href="#">Amazon EC2 インスタンスタイプ</a> 」を参照してください。  | 2010 年 2 月 22 日 |