

---

# Amazon Elastic Compute Cloud

ユーザーガイド

API Version 2014-02-01



# アマゾン ウェブ サービス

## Amazon Elastic Compute Cloud: ユーザーガイド

アマゾン ウェブ サービス

Copyright © 2014 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks of Amazon Web Services, Inc.: Amazon, Amazon Web Services Design, AWS, Amazon CloudFront, Cloudfront, Amazon DevPay, DynamoDB, ElastiCache, Amazon EC2, Amazon Elastic Compute Cloud, Amazon Glacier, Kindle, Kindle Fire, AWS Marketplace Design, Mechanical Turk, Amazon Redshift, Amazon Route 53, Amazon S3, Amazon VPC. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

Amazon EC2 とは	1
インスタンスと AMI	4
リージョンとアベイラビリティゾーン	8
ルートデバイスボリューム	15
セットアップ	21
使用開始	27
ステップ 1: インスタンスを起動する	28
ステップ 2: インスタンスに接続する	29
ステップ 3: ボリュームを追加する	32
ステップ 4: クリーンアップ	36
ベストプラクティス	37
チュートリアル: LAMP ウェブサーバーのインストール	39
チュートリアル: WordPress ブログのホスティング	46
Amazon マシンイメージ	55
AMI タイプ	57
適切な AMI の検索	60
共有 AMI	62
共有 AMI を見つける	62
AMI を一般公開する	65
特定の AWS アカウントと AMI を共有する	67
ブックマークの使用	68
共有 Linux AMI のガイドライン	69
有料 AMI	73
Amazon EBS-Backed Linux AMI の作成	77
Instance Store-Backed Linux AMI の作成	81
AMI のコピー	90
AMI の登録解除	92
Amazon Linux	94
独自の Linux カーネルの使用	102
インスタンス	108
インスタンスタイプ	109
マイクロインスタンス	110
I2 インスタンス	117
H1 インスタンス	119
HS1 インスタンス	121
R3 インスタンス	122
GPU インスタンス	124
Amazon EBS 最適化インスタンス	127
プレースメントグループ	128
インスタンスのサイズ変更	131
スポットインスタンス	134
スポットインスタンスの使用を開始する	135
スポットインスタンス価格履歴の表示	138
スポットインスタンスリクエストの作成	140
実行中のスポットインスタンスの検索	144
スポットインスタンスリクエストのキャンセル	146
スポットインスタンスの基本	148
スポットリクエストの発行	149
スポットインスタンスの制限	150
スポットリクエストのカスタマイズ	151
入札ステータスコードによるスポットリクエストの追跡	153
スポットインスタンスリクエストのタグ付け	160
スポットインスタンスのプロビジョニング、価格設定、中断を理解する	161
スポットインスタンスデータを中断から保護する	164
中断の計画	164
永続ルート EBS パーティション	165
ウォークスルー: AWS サービスでのスポットインスタンスの使用	165
Auto Scaling によるスポットインスタンスの管理	166

Auto Scaling をスポットインスタンスと一緒に管理するためのツール	167
Auto Scaling でスポットインスタンスを起動する	168
Auto Scaling により起動されたインスタンスの情報を取得する	171
スポットインスタンスの入札価格の更新	176
スポット入札リクエストの予定設定	178
Auto Scaling を使用してスポットインスタンスに関する通知を取得する	179
CloudFormation テンプレートを使用してスポットインスタンスを起動する	182
スポットインスタンスで Amazon Elastic MapReduce のジョブフローを起動	183
Amazon Virtual Private Cloud でのスポットインスタンスの起動	183
高度なタスク	186
スポットインスタンスのデータフィードの購読	186
AWS Java SDK を使用したスポットインスタンスのプログラミング	190
チュートリアル: Amazon EC2 スポットインスタンス	191
チュートリアル: Amazon EC2 スポットリクエストの高度な管理	200
スポットインスタンスでのクラスターの開始	216
リザーブドインスタンス	218
リザーブドインスタンスの使用を開始する	219
リザーブドインスタンスを使用するためのツール	222
リザーブドインスタンスの基礎	224
使用計画に基づいてリザーブドインスタンスを選択する	225
リザーブドインスタンス料金範囲を理解する	226
リザーブドインスタンスの料金メリットについて	231
リザーブドインスタンスと一括請求 ( コンソリデーティッドビルディング )	232
リザーブドインスタンスマーケットプレイス	233
リザーブドインスタンスを購入する	236
購入者になる	237
リザーブドインスタンスを購入する	238
明細書 ( 請求書 ) を読む	245
リザーブドインスタンスに関する情報を取得する	246
リザーブドインスタンスの変更	251
予約のインスタンスタイプの変更	255
変更リクエストの送信	257
リザーブドインスタンスマーケットプレイス での販売	261
販売者として登録する	262
リザーブドインスタンスを販売する	266
リザーブドインスタンスが売却された後	286
リザーブドインスタンスの要件チェックリスト	288
インスタンスメタデータとユーザーデータ	289
インスタンスのインポートとエクスポート	301
前提条件	302
Amazon EC2 に VM をインポートする	305
ステップ 1: Amazon EC2 CLI をインストールする	305
ステップ 2: VM を準備する	306
ステップ 3: VM をその仮想環境からエクスポートする	307
ステップ 4: Amazon EC2 に VM をインポートする	307
インポートタスクのステータスをチェックする	309
Amazon EBS にボリュームをインポートする	311
アップロードを再開する	312
アップロードをキャンセルする	313
アップロード後にクリーンアップする	314
ステップ 5: Amazon EC2 でインスタンスを起動する	314
Amazon EC2 インスタンスをエクスポートする	314
トラブルシューティング	316
インスタンスのライフサイクル	321
起動する	324
インスタンスの起動	325
バックアップからインスタンスを起動する	331
AWS Marketplace インスタンスの起動	332

接続 .....	334
SSH を使用した接続 .....	334
PuTTY を使用した接続 .....	338
Mindterm を使った接続 .....	343
RDP を使用して接続する .....	344
停止と起動 .....	346
再起動 .....	349
リタイア .....	350
Terminate .....	352
インスタンスの設定 .....	359
ソフトウェアの管理 .....	360
インスタンスソフトウェアの更新 .....	361
リポジトリの追加 .....	364
ソフトウェアパッケージを見つける .....	365
ソフトウェアパッケージのインストール .....	367
ソフトウェアのコンパイル準備 .....	367
ユーザーの管理 .....	369
インスタンスの時間を設定します .....	370
ホスト名の変更 .....	374
動的 DNS を使用する .....	377
ユーザーデータでインスタンスを起動する .....	378
モニタリング .....	383
自動モニタリングと手動モニタリング .....	384
モニタリングのベストプラクティス .....	386
インスタンスのステータスのモニタリング .....	386
ステータスチェックでインスタンスをモニタリングする .....	387
インスタンスのイベントのモニタリング .....	392
CloudWatch によるインスタンスのモニタリング .....	395
Amazon EC2 インスタンスの詳細モニタリングの有効化または無効化 .....	396
Amazon EC2 メトリックスを表示する .....	399
メトリックスの統計情報を取得する .....	406
特定の EC2 インスタンスの統計情報を取得する .....	406
複数のインスタンスにわたって統計情報を集計する .....	410
Auto Scaling グループごとに集計された統計情報を取得する .....	415
イメージ (AMI) ID ごとに集計された統計情報を取得する .....	418
メトリックスのグラフ化 .....	423
メトリックスをグラフ化する .....	423
複数のリソースにおける単一のメトリックスをグラフ化する .....	424
CloudWatch アラームを作成する .....	427
CPU 使用率のアラームに基づいて E メールを送信する .....	427
ロードバランサーアラームに基づいて E メールを送信する .....	429
ストレージスループットアラームに基づいて E メールを送信する .....	432
インスタンスを停止または終了するアラームを作成する .....	435
Amazon EC2 インスタンスのモニタリングスクリプト .....	451
Amazon CloudWatch Monitoring Scripts for Linux .....	452
Amazon CloudWatch Monitoring Scripts for Windows .....	458
ネットワークとセキュリティ .....	469
キーペア .....	470
セキュリティグループ .....	477
アクセスの制御 .....	485
IAM ポリシー .....	488
サポートされるリソースと条件 .....	494
ポリシーの例 .....	501
IAM ロール .....	516
ネットワークアクセス .....	522
Amazon VPC .....	524
サポートされているプラットフォーム .....	527
インスタンスの IP アドレッシング .....	529

複数のプライベート IP アドレス .....	534
Elastic IP アドレス .....	540
Elastic Network Interface .....	545
拡張ネットワーキング .....	559
ストレージ .....	564
Amazon EBS .....	566
EBS ボリューム .....	567
EBS ボリュームの種類 .....	569
ボリュームの作成 .....	570
スナップショットからの復元 .....	571
インスタンスへのボリュームのアタッチ .....	573
Amazon EBS ボリュームを使用できるようにする .....	576
ボリューム情報 .....	580
ボリュームのステータスのモニタリング .....	580
インスタンスからの Amazon EBS ボリュームのデタッチ .....	591
ボリュームの削除 .....	592
ボリュームの拡張 .....	593
Linux パーティションを拡張する .....	600
EBS スナップショット .....	605
スナップショットの作成 .....	606
スナップショットの削除 .....	607
スナップショットのコピー .....	608
スナップショットに関する情報 .....	609
スナップショットの共有 .....	610
EBS パフォーマンス .....	611
EC2 の構成 .....	612
I/O 特性 .....	614
ワークロードのデマンド .....	615
ボリュームの事前ウォーミング .....	615
RAID 構成 .....	620
ボリュームの評価 .....	626
API とコマンドの概要 .....	629
インスタンスストア .....	631
Amazon S3 .....	641
ブロックデバイスマッピング .....	643
パブリックデータセットの使用 .....	655
リソースとタグ .....	658
リソースの場所 .....	658
リソースのリスト表示とフィルタリング .....	660
リソースにタグを付ける .....	662
使用状況レポート .....	671
インスタンスの使用状況 .....	674
リザーブドインスタンスの使用率 .....	677
トラブルシューティング .....	685
インスタンスの起動 .....	685
インスタンスへの接続 .....	686
インスタンスの停止 .....	691
インスタンスの終了 .....	693
失敗したステータスチェック .....	693
インスタンスの容量 .....	718
全般 .....	718
API リクエストを行う .....	721
クエリリクエスト .....	723
API リクエストエラーのトラブルシューティング .....	725
べき等性の確保 .....	728
SOAP リクエスト .....	730
ドキュメント履歴 .....	731

# Amazon EC2 とは

---

## Abstract

ハードウェアの制約を受けずにアプリケーションを開発およびデプロイできるように、Amazon EC2 を使用して、AWS クラウドでサイズ変更可能なコンピューティングキャパシティを実現します。

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

## Amazon EC2 の機能

Amazon EC2 には次の機能があります。

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

- 残りの AWS クラウドから論理的に分離され、ユーザー独自のネットワークにオプションで接続できる、[仮想プライベートクラウド \( VPC \)](#) と呼ばれる仮想ネットワーク

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

## Amazon EC2 の使用を開始する方法

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

### 使用開始

- [Amazon EC2 でのセットアップ \(p. 21\)](#)
- [Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#)
- [Amazon EC2 Windows インスタンスの使用開始](#)

### 基礎

- [インスタンスと AMI \(p. 4\)](#)
- [インスタンスタイプ \(p. 109\)](#)
- [リージョンとアベイラビリティゾーン \(p. 8\)](#)
- [タグ \(p. 662\)](#)

### ネットワークとセキュリティ

- [Amazon EC2 のキーペア \(p. 470\)](#)
- [セキュリティグループ \(p. 477\)](#)
- [Elastic IP アドレス \( EIP \) \(p. 540\)](#)
- [Amazon EC2 と Amazon VPC \(p. 524\)](#)

### ストレージ

- [Amazon EBS \(p. 566\)](#)
- [インスタンスストア \(p. 631\)](#)

AWS がお客様に合っているかどうか検討中の場合は、[AWS セールスマン](#)までお問い合わせください。Amazon EC2 について質問がある場合は、[Amazon EC2 forum](#) をご利用ください。

## 関連サービス

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

- [Auto Scaling 開発者ガイド](#)

- [AWS CloudFormation ユーザーガイド](#)
- [AWS Elastic Beanstalk Developer Guide](#)
- [AWS OpsWorks ユーザーガイド](#)

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

インスタンスと Amazon EBS ボリュームの基本的な統計情報をモニタリングするには、Amazon CloudWatch を使用します。詳細については、「[CloudWatch によるインスタンスのモニタリング \(p. 395\)](#)」を参照してください。

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

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

## Amazon EC2 へのアクセス

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

コマンドラインインターフェイスを使用する場合は、以下のいくつかのオプションがあります。

### AWS コマンドラインインターフェイス (CLI)

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

### Amazon EC2 コマンドラインインターフェイス (CLI) ツール

Amazon EC2、Amazon EBS、および Amazon VPC 用のコマンドを提供し、Windows、Mac、および Linux/UNIX でサポートされています。開始するには、*Amazon Elastic Compute Cloud* コマンドラインリファレンスの [Setting Up the Amazon EC2 Command Line Interface Tools on Linux](#) および [Commands \(CLI Tools\)](#) を参照してください。

### AWS Tools for Windows PowerShell

PowerShell 環境でスクリプティングを行うユーザーに対して、一連のさまざまな AWS 製品用のコマンドを提供します。開始するには、[AWS Tools for Windows PowerShell ユーザーガイド](#) を参照してください。

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

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

ては、[アマゾン ウェブ サービスのツール](#)を参照してください。Amazon EC2 の使用可能な API のリストについては、[Amazon EC2 に対して使用可能な API \(p. 722\)](#) を参照してください。

## Amazon EC2 の料金表

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

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

### オンデマンドインスタンス

時間単位で使用するインスタンスに対して支払いを行い、長期的な確約や予約金支払いはありません。

### リザーブドインスタンス

インスタンスに対して低額の予約金支払いを行い、1年または3年間それを予約することで、そのインスタンスを実行する時間料金を大幅に削減できます。

### スポットインスタンス

特定のインスタンスタイプを実行するためにお客様が支払いたい最大時間料金を指定することができます。スポット料金は需要と供給に基づいて変動しますが、お客様が指定した上限価格を超えた支払いが発生することはありません。スポット価格が上限価格を超えた場合、スポットインスタンスはシャットダウンされます。

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

プロビジョニングされたサンプル環境のコストを計算するには、[AWS エコノミクスセンター](#)を参照してください。

請求書を確認するには、[AWS アカウントアクティビティページ](#)を参照してください。請求書には、料金の明細が記載された使用状況レポートへのリンクが記載されています。AWS アカウント請求の詳細については、[AWS Account Billing](#) を参照してください。

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

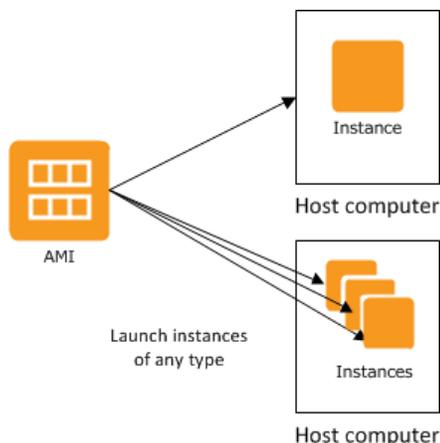
AWS 環境のコスト、セキュリティ、およびパフォーマンスを最適化できるサービスである Trusted Advisor の概要については、[AWS Trusted Advisor](#) を参照してください。

## インスタンスと AMI

### Abstract

Amazon EC2 の核となる概念であるインスタンスと AMI について学習します。

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



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

## インスタンス

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

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

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

実行中インスタンス数が制限され、保有できるインスタンスの合計数も制限されます（実行中、停止済み、または終了済みを除くその他の状態を含む）。この全体的なインスタンス制限は、実行中インスタンス制限の 2 倍です。

## インスタンスのストレージ

インスタンスのルートデバイスには、インスタンスの起動に使用されるイメージが含まれています。詳細については、「[Amazon EC2 ルートデバイスボリューム \(p. 15\)](#)」を参照してください。

インスタンスには、インスタンスストアボリュームと呼ばれるローカルストレージボリュームを含めることができます。これはブロックデバイスマッピングによって起動時に設定できます。詳細については、「[ブロックデバイスマッピング \(p. 643\)](#)」を参照してください。これらのボリュームがインスタンスに追加およびマッピングされたら、マウントして使用することができます。インスタンスが失敗、停止、または終了した場合、それらのボリュームのデータは失われます。したがって、これらのボリュームは一時データとして使用するのが最適です。重要なデータの場合、データの安全を維持するには、複数のインスタンスにわたるレプリケーション方法を使用する必要があります。あるいは、永続的なデータを Amazon S3 または Amazon EBS ボリュームに格納してください。詳細については、「[ストレージ \(p. 564\)](#)」を参照してください。

## セキュリティのベストプラクティス

- AWS Identity and Access Management ( IAM ) を使用して、ご利用のインスタンスを含め、AWS リソースへのアクセスを制御します。AWS アカウントで IAM ユーザーとグループを作成し、それぞれにセキュリティ認証情報を割り当て、それぞれが AWS のリソースとサービスに対して持つアクセスを制御できます。詳細については、「[Amazon EC2 のリソースに対するアクセスの制御 \(p. 485\)](#)」を参照してください。
- 信頼されたホストまたはネットワークのみがインスタンスのポートにアクセスできるように制限します。例えば、ポート 22 の受信トラフィックを制限することで SSH アクセスを制限できます。詳細については、「[Amazon EC2 セキュリティグループ \(p. 477\)](#)」を参照してください。
- セキュリティグループのルールを定期的に確認し、最小権限の原則を適用してください。つまり、必要なアクセス許可のみを開きます。また、さまざまなセキュリティグループを作成して、異なるセキュリティ要件を持つ各インスタンスに対応することもできます。外部ログインが許可された基本となるセキュリティグループの作成を検討し、外部ログインが許可されていないグループで残りのインスタンスを管理してください。
- AMI から起動されるインスタンスについてはパスワードベースのログインを無効にしてください。パスワードは検知または解読される恐れがあり、セキュリティ上のリスクです。詳細については、「[ルートのパスワードベースログインの無効化 \(p. 70\)](#)」を参照してください。AMI の安全な共有の詳細については、「[共有 AMI \(p. 62\)](#)」を参照してください。

## インスタンスの停止、開始、および終了

### インスタンスの停止

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

インスタンスが停止状態にあるとき、追加のインスタンス時間分が課金されることはありません。インスタンスを停止状態から実行状態に移行するたびに 1 時間分のインスタンス時間が課金されます。このような移行が 1 時間の間に何度も行われる場合であっても、その回数ごとに 1 時間分が課金されます。インスタンスが停止状態にあるときにインスタンスタイプを変更した場合、インスタンスを起動すると同時に新しいインスタンスタイプの料金が課金されます。ルートデバイスの使用を含め、インスタンスに関連する Amazon EBS の使用はすべて、Amazon EBS 料金で課金されます。

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

### インスタンスを終了した

インスタンスを終了すると、そのインスタンスは標準のシャットダウンを実行します。その後、ボリュームの `deleteOnTermination` 属性が `false` に設定されない限り、アタッチされている Amazon EBS ボリュームが削除されます。インスタンスそのものも削除され、後でインスタンスを再度起動することはできません。

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

詳細については、[インスタンスのライフサイクル \(p. 321\)](#) を参照してください。

## AMI

アマゾン ウェブ サービス (AWS) は、一般的な用途のための共通のソフトウェア設定を含む多くの Amazon マシンイメージ (AMI) を公開しています。加えて、AWS 開発者コミュニティのメンバーによって作成された、独自のカスタム AMI もあります。お客様自身でカスタム AMI を作成することもできます。必要なものがすべて含まれた新しいインスタンスを、すばやく簡単に起動できるようになります。例えば、ウェブサイトまたはウェブサービスに使用する場合は、AMIに含まれるものとして、ウェブサーバー、関連する静的コンテンツ、動的ページ用のコードが考えられます。この AMI からインスタンスを起動すると、ウェブサーバーが起動し、アプリケーションはリクエストを受け付け可能な状態になります。

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

AMI の説明には、ルートデバイスの種類 (*ebs* か *instance store*か) も明記されています。この違いが重要なのは、AMI のタイプによって、実行できる機能が大きく異なるからです。違いについての詳細は [ルートデバイスのストレージ \(p. 57\)](#) を参照してください。

# リージョンとアベイラビリティゾーン

## Abstract

インスタンスをホストできる世界各地のリージョンとアベイラビリティゾーンを説明します。

Amazon EC2 は、世界各地のロケーションでホスティングされています。これらのロケーションは、リージョンとアベイラビリティゾーンから構成されています。リージョンはそれぞれ、地理的に離れた領域です。1つのリージョンに複数のそれぞれ独立したロケーションがあり、このロケーションを「アベイラビリティゾーン」といいます。Amazon EC2 では、お客様がインスタンスなどのリソースとデータを複数のロケーションに配置できます。複数のリージョンにまたがってリソースのレプリケーションを行うには、お客様がそのように指定する必要があります。

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



### Note

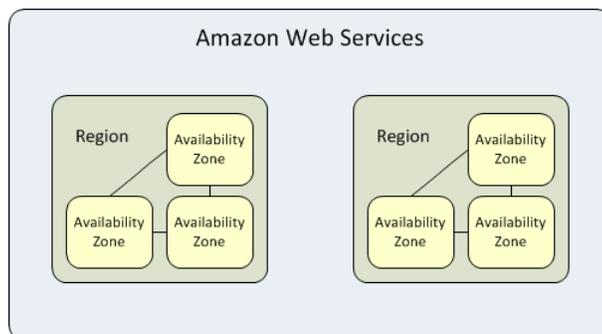
一部の AWS リソースは、リージョンおよびアベイラビリティゾーンによっては利用できない場合があります。アプリケーションをデプロイする前に、目的のリージョンまたはアベイラビリティゾーンで必要なリソースを作成できることを確認してください。

## Topics

- [リージョンとアベイラビリティゾーンに関する概念](#) (p. 8)
- [リージョンとアベイラビリティゾーンの記述](#) (p. 10)
- [リソースのリージョンの指定](#) (p. 12)
- [アベイラビリティゾーンでのインスタンスの起動](#) (p. 13)
- [別のアベイラビリティゾーンへのインスタンスの移行](#) (p. 14)

## リージョンとアベイラビリティゾーンに関する概念

各リージョンは完全に独立しています。各アベイラビリティゾーンは独立していますが、同じリージョン内のアベイラビリティゾーン同士は低レイテンシーのリンクで接続されています。次の図に、リージョンとアベイラビリティゾーンの関係を示します。



Amazon EC2 リソースには、グローバルなもの、リージョンに結び付けられているもの、アベイラビリティゾーンに結び付けられているものがあります。詳細については、「[リソースの場所 \(p. 658\)](#)」を参照してください。

## リージョン

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

Amazon EC2 では複数のリージョンが提供されるため、それぞれの要件に合った場所で Amazon EC2 インスタンスを起動できます。例えば、ヨーロッパの顧客に近づけるため、または法的要件を満たすために、ヨーロッパでインスタンスを起動することができます。次の表は、Amazon EC2 をサポートするリージョンのリストです。

コード	名前
ap-northeast-1	アジアパシフィック (東京) リージョン
ap-southeast-1	アジアパシフィック (シンガポール) リージョン
ap-southeast-2	アジアパシフィック (シドニー) リージョン
eu-west-1	欧州 (アイルランド) リージョン
sa-east-1	南米 (サンパウロ) リージョン
us-east-1	米国東部 (バージニア北部) リージョン
us-west-1	米国西部 (北カリフォルニア) リージョン
us-west-2	米国西部 (オレゴン) リージョン

リソースを表示すると、指定したリージョンに結び付けられているリソースだけが表示されます。これは、リソースは相互に分離されており、リージョン間のリソースのレプリケーションは自動的に行われないためです。

コマンドラインインターフェイスまたは API アクションを使用してインスタンスを操作するときは、そのリージョンエンドポイントを指定する必要があります。Amazon EC2 のリージョンおよびエンドポイントについては、『*Amazon Web Services General Reference*』の「[Regions and Endpoints](#)」を参照してください。

インスタンスを起動するときは、同じリージョン内にある AMI を選択する必要があります。AMI が別のリージョンにある場合は、使用しているリージョンに AMI をコピーできます。詳細については、「[AMI のコピー \(p. 90\)](#)」を参照してください。

リージョン間のすべての通信は、パブリックインターネットを通して行われます。したがって、適切な暗号方式を使用してデータを保護する必要があります。リージョン間でデータを転送すると、送信側と受信側の両方のインスタンスで、インターネットデータ転送速度に応じて料金が発生します。詳細については、「[Amazon EC2 Pricing – Data Transfer](#)」を参照してください。

## アベイラビリティゾーン

自分のアカウントで使用できるアベイラビリティゾーンをリストできます。詳細については、「[リージョンとアベイラビリティゾーンの記述 \(p. 10\)](#)」を参照してください。

インスタンスを起動するときに、アベイラビリティゾーンを自分で選択するか、自動的に選択されるようにできます。インスタンスを複数のアベイラビリティゾーンに配布する場合は、1つのインスタ

ンスで障害が発生したら別のアベイラビリティゾーンのインスタンスが要求を処理するように、アプリケーションを設計できます。

また、伸縮自在な IP アドレスを使用すると、あるアベイラビリティゾーンのインスタンスの障害を、別のアベイラビリティゾーンのインスタンスにアドレスをすばやく再マッピングすることによってマスクできます。詳細については、「[Elastic IP アドレス \( EIP \) \( p. 540 \)](#)」を参照してください。

リソースがリージョンの複数のアベイラビリティゾーンに分散されるようにするため、アベイラビリティゾーンは各アカウントの識別子に個別にマップされます。例えば、あるアカウントのアベイラビリティゾーンは別のアカウントの `us-east-1a` アベイラビリティゾーンと同じ場所にはない可能性があります `us-east-1a`。ユーザーがアカウント間でアベイラビリティゾーンを調整する手段はありません。

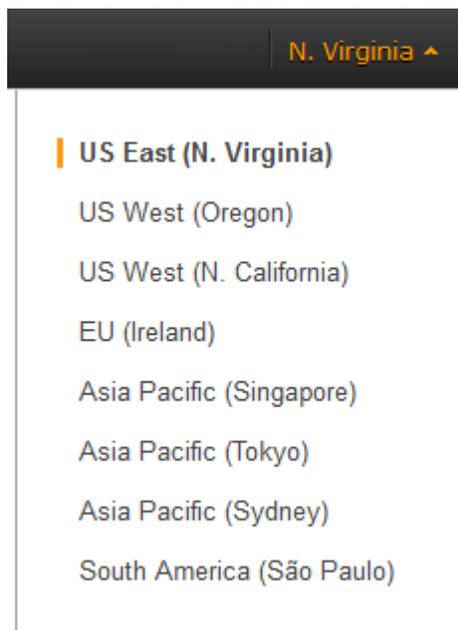
アベイラビリティゾーンが拡大すると、アベイラビリティゾーンを拡張しにくくなる場合があります。その場合、ユーザーがアベイラビリティゾーンに既にインスタンスを持っているのではない場合は、制約のあるアベイラビリティゾーンでのインスタンスの起動を制限する場合があります。最終的に、制約のあるアベイラビリティゾーンを新しいお客様に対するアベイラビリティゾーンのリストから削除することもあります。したがって、アカウントによってリージョン内で使用できるアベイラビリティゾーンの数が異なる場合があります。

## リージョンとアベイラビリティゾーンの記述

AWS Management Console またはコマンドラインインターフェイスを使用して、使用できるリージョンおよびアベイラビリティゾーンを特定できます。これらのコマンドラインインターフェイスについての詳細は、[Amazon EC2 へのアクセス \( p. 3 \)](#) を参照してください。

AWS Management Console を使用してリージョンとアベイラビリティゾーンを検索するには

1. AWS Management Console を開きます。
2. ナビゲーションバーから、リージョンセレクターのオプションを表示します。



3. Amazon EC2 コンソールを開きます。アベイラビリティゾーンは、ダッシュボードの [Service Health] の [Availability Zone Status] に一覧表示されます。

## Service Health

### Service Status:

- ✔ US East (N. Virginia):  
This service is operating normally

### Availability Zone Status:

- ✔ us-east-1b:  
Availability zone is operating normally
- ✔ us-east-1c:  
Availability zone is operating normally
- ✔ us-east-1d:  
Availability zone is operating normally

AWS CLI を使用してリージョンとアベイラビリティゾーンを見つけるには

1. 次のように `describe-regions` コマンドを使用してリージョンを記述します。

```
PROMPT> aws ec2 describe-regions
{
  "Regions": [
    {
      "Endpoint": "ec2.us-east-1.amazonaws.com",
      "RegionName": "us-east-1"
    },
    {
      "Endpoint": "ec2.ap-southeast-1.amazonaws.com",
      "RegionName": "ap-southeast-1"
    },
    {
      "Endpoint": "ec2.ap-southeast-2.amazonaws.com",
      "RegionName": "ap-southeast-2"
    },
    ...
  ]
}
```

2. 次のように `describe-availability-zones` コマンドを使用してリージョン内のアベイラビリティゾーンを記述します。us-east-1

```
PROMPT> aws ec2 describe-availability-zones --region us-east-1
{
  "AvailabilityZones": [
    {
      "State": "available",
      "RegionName": "us-east-1",
      "Messages": [],
      "ZoneName": "us-east-1b"
    },
  ],
}
```

```
{
  "State": "available",
  "RegionName": "us-east-1",
  "Messages": [],
  "ZoneName": "us-east-1c"
},
{
  "State": "available",
  "RegionName": "us-east-1",
  "Messages": [],
  "ZoneName": "us-east-1d"
}
]
```

Amazon EC2 CLI を使用してリージョンとアベイラビリティゾーンを見つけるには

1. 次のように `ec2-describe-regions` コマンドを使用してリージョンを記述します。

```
PROMPT> ec2-describe-regions
REGION  ap-northeast-1      ec2.ap-northeast-1.amazonaws.com
REGION  ap-southeast-1        ec2.ap-southeast-1.amazonaws.com
..
```

2. 次のように `ec2-describe-availability-zones` コマンドを使用してリージョン内のアベイラビリティゾーンを記述します。us-east-1

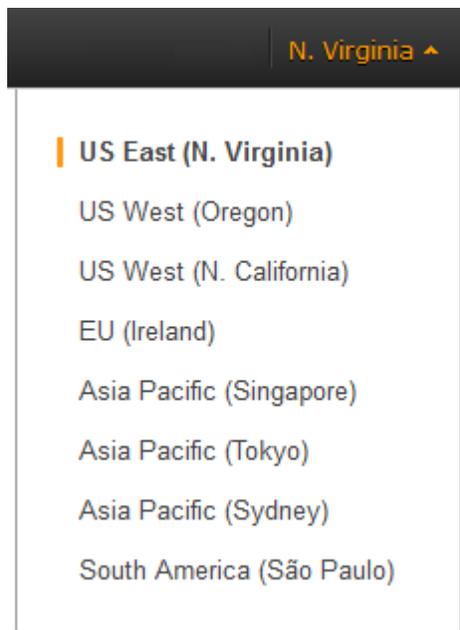
```
PROMPT> ec2-describe-availability-zones --region us-east-1
AVAILABILITYZONE us-east-1a available us-east-1
AVAILABILITYZONE us-east-1b available us-east-1
AVAILABILITYZONE us-east-1c available us-east-1
AVAILABILITYZONE us-east-1d available us-east-1
```

## リソースのリージョンの指定

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

コンソールを使用してリソースのリージョンを指定するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーのリージョンセレクターを使用します。



コマンドラインを使用してデフォルトのリージョンを指定するには

環境変数の値を、目的のリージョンエンドポイント (例: `https://ec2.us-west-1.amazonaws.com`) に設定できます。

- `AWS_DEFAULT_REGION` (AWS CLI)
- `EC2_URL` (Amazon EC2 CLI)

または、各コマンドで `--region` コマンドラインオプションを使用することもできます。例えば、`--region us-west-1` と指定します。

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

## アベイラビリティーゾーンでのインスタンスの起動

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

インスタンスを起動するときは、必要に応じて、使用するリージョン内のアベイラビリティーゾーンを指定できます。アベイラビリティーゾーンを指定しないと、自動的に選択されます。初期インスタンスを起動するときには、デフォルトのアベイラビリティーゾーンを受け入れることをお勧めします。これにより Amazon EC2 が、システムの状態と利用可能な機能に基づいて、最適なアベイラビリティーゾーンを選択できます。追加インスタンスを起動する場合、新しいインスタンスが実行中のインスタンスに近いが、分かれている必要がある場合は、アベイラビリティーゾーンだけを指定します。

コンソールを使用してインスタンスのアベイラビリティーゾーンを指定するには

1. Amazon EC2 コンソールを開きます。
2. ダッシュボードで、[Launch Instance] をクリックします。

3. ウィザードの指示にしたがって操作します。[Configure Instance Details] ページで以下の操作を行います。

- [EC2-Classic] リストからアベイラビリティゾーンのオプションを選択するか、[No Preference] を選択して最善のアベイラビリティゾーンを自動的に選択します。

**Availability Zone** ⓘ

- [EC2-VPC] リストからサブネットオプションを選択するか、[No preference (default subnet in any Availability Zone)] を選択して最善のサブネットを自動的に選択します。

**Subnet** ⓘ

[Create new subnet](#)

AWS CLI を使用してインスタンスのアベイラビリティゾーンを指定するには

`run-instances` コマンドと次のいずれかのオプションを使用できます。

- [EC2-Classic] `--placement`
- [EC2-VPC] `--subnet-id`

Amazon EC2 CLI を使用してインスタンスのアベイラビリティゾーンを指定するには

`ec2-run-instances` コマンドと次のいずれかのオプションを使用できます。

- [EC2-Classic] `--availability-zone`
- [EC2-VPC] `--subnet`

## 別のアベイラビリティゾーンへのインスタンスの移行

必要な場合は、1つのアベイラビリティゾーンから別のアベイラビリティゾーンにインスタンスを移行できます。たとえば、インスタンスのインスタンスタイプを変更しようとしたときに、現在のアベイラビリティゾーンでは新しいタイプのインスタンスを起動できない場合は、そのタイプのインスタンスを起動できるアベイラビリティゾーンにインスタンスを移行することもできます。

移行プロセスでは、次の手順に示すように、元のインスタンスから AMI を作成し、新しいアベイラビリティゾーンでインスタンスを起動して、新しいインスタンスの設定を更新します。

別のアベイラビリティゾーンにインスタンスを移行するには

1. インスタンスから AMI を作成します。手順は、オペレーティングシステムとインスタンスのルートデバイスボリュームの種類によって異なります。詳細については、使用しているオペレーティングシステムとルートデバイスボリュームに対応するドキュメントを参照してください。
  - [Amazon EBS-Backed Linux AMI の作成 \(p. 77\)](#)
  - [Instance Store-Backed Linux AMI の作成 \(p. 81\)](#)
  - [Amazon EBS-backed Windows AMI の作成](#)
  - [Instance Store-Backed Windows AMI の作成](#)
2. [EC2-VPC] インスタンスのプライベート IP アドレスを維持する必要がある場合は、現在のアベイラビリティゾーンのサブネットを削除してから、新しいアベイラビリティゾーンに元のサブネット

と同じ IP アドレス範囲のサブネットを作成する必要があります。サブネットを削除する前に、その中のすべてのインスタンスを終了する必要があります。したがって、現在のサブネット内のすべてのインスタンスを新しいサブネットに移動しなければなりません。

3. 新しいアベイラビリティゾーンまたはサブネットを指定して、作成した AMI からインスタンスを起動します。インスタンスタイプは、元のインスタンスと同じにすることも、新しいインスタンスタイプを選択することもできます。詳細については、「[アベイラビリティゾーンでのインスタンスの起動 \(p. 13\)](#)」を参照してください。
4. 元のインスタンスに Elastic IP アドレスが関連付けられていた場合は、それを新しいインスタンスに関連付けます。詳細については、「[Elastic IP アドレスを別の実行中のインスタンスに関連付ける \(p. 543\)](#)」を参照してください。
5. 元のインスタンスがリザーブインスタンスの場合、予約のアベイラビリティゾーンを変更します。(また、インスタンスタイプも変更する場合は、予約のインスタンスタイプも変更できます)。詳細については、「[変更リクエストの送信 \(p. 257\)](#)」を参照してください。
6. (オプション) 元のインスタンスを終了します。詳細については、「[インスタンスを終了する \(p. 353\)](#)」を参照してください。

## Amazon EC2 ルートデバイスボリューム

### Abstract

インスタンスのブートボリュームのタイプを選択します。

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

### Topics

- [ルートデバイスストレージの概念 \(p. 15\)](#)
- [ルートデバイスタイプによる AMI の選択 \(p. 17\)](#)
- [インスタンスのルートデバイスタイプの判別 \(p. 18\)](#)
- [永続的ルートデバイスボリュームへの変更 \(p. 18\)](#)

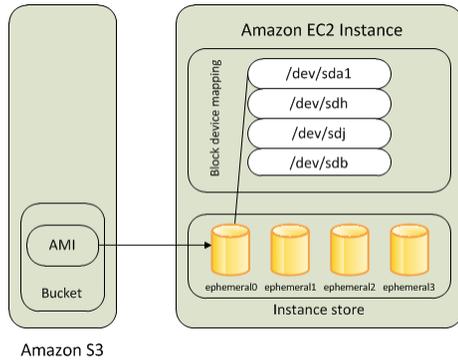
## ルートデバイスストレージの概念

Instance store-Backed AMI と Amazon EBS-Backed AMI という 2 種類の AMI のいずれかからインスタンスを起動できます。AMI の説明にはそのタイプが含まれており、場所によってルートデバイスが `ebs` (Amazon EBS-Backed の場合) または `instance store` (Instance store-Backed の場合) と表示されます。各タイプの AMI を使用して実行できることには大きな違いがあるため、タイプを区別できることは重要です。違いについての詳細は [ルートデバイスのストレージ \(p. 57\)](#) を参照してください。

### instance store-backed のインスタンス

インスタンスストアをルートデバイスに使用するインスタンスでは自動的に、インスタンスストアボリュームを利用できるようになり、その 1 つがルートデバイスボリュームとなります。インスタンスを起動すると、インスタンスのブートに使用されるイメージがルートボリューム (通常は `sda1`) にコピーされます。インスタンスストアボリュームのデータはインスタンスが実行している間は維持されますが、インスタンスが終了すると (Instance store-Backed インスタンスは [Stop] アクションをサポート

していません)、またはインスタンスが失敗すると(基盤となるドライブに問題がある場合など)、削除されます。

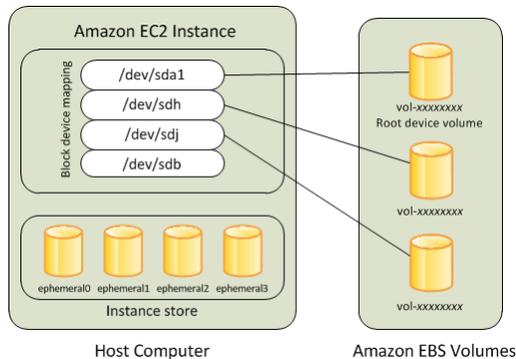


障害が発生したり終了されたりした instance store-backed インスタンスは復元できません。Amazon EC2 instance store-backed インスタンスの使用を予定している場合は、インスタンスストアのデータを複数のアベイラビリティゾーンにまたがって分散させることを強くお勧めします。また、インスタンスストアボリュームのデータを永続的ストレージに定期的にバックアップする必要があります。

詳細については、「[Amazon EC2 インスタンスストア \(p. 631\)](#)」を参照してください。

#### Amazon EBS-backed インスタンス

Amazon EBS をルートデバイスに使用するインスタンスには自動的に、Amazon EBS ボリュームがアタッチされます。Amazon EBS Backed インスタンスを起動するとき、AMI で参照されている Amazon EBS スナップショットごとに 1 つの Amazon EBS ボリュームが作成されます。他の Amazon EBS ボリュームやインスタンスストアボリュームを使用することもできます。



Amazon EBS-backed インスタンスは、停止後に再起動できます。アタッチされているボリュームに格納されているデータに影響を及ぼすこともありません。Amazon EBS-backed インスタンスが停止状態にあるときは、インスタンス関連およびボリューム関連の様々なタスクを実行できます。例えば、インスタンスのプロパティの変更、インスタンスのサイズの変更、使用しているカーネルの更新が可能です。また、デバッグなどの目的で別の実行中インスタンスにルートボリュームをアタッチすることもできます。

デフォルトでは、Amazon EBS-backed インスタンスを起動するとアタッチされるルートデバイスボリュームおよびその他の Amazon EBS ボリュームは、インスタンスが終了すると自動的に削除されます。インスタンス起動時のこの動作を変更する方法については、「[永続的ルートデバイスボリュームへの変更 \(p. 18\)](#)」を参照してください。

デフォルトでは、実行中のインスタンスにお客様が接続したすべての Amazon EBS ボリュームは、インスタンスが終了すると、データはそのままの状態です。デタッチされたボリュームを、実行中の任意のインスタンスにアタッチできます。

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

- いったん停止して再起動します。
- 関連するすべてのボリュームのスナップショットを自動的に作成し、新しい AMI を作成します。詳細については、「[Amazon EBS-Backed Linux AMI の作成 \(p. 77\)](#)」を参照してください。
- 以下の手順に従って、ボリュームを新しいインスタンスにアタッチします。
  1. ルートボリュームのスナップショットを作成します。
  2. 作成したスナップショットを使用して新しい AMI を登録します。
  3. 新しい AMI から新しいインスタンスを起動します。
  4. 残りの Amazon EBS ボリュームを古いインスタンスからデタッチします。
  5. Amazon EBS ボリュームを新しいインスタンスに再アタッチします。

通常のボリュームサイズを持つインスタンスに障害が発生した場合は 1 つ目か 2 つ目の方法、大容量のボリュームを持つインスタンスに障害が発生した場合は 3 つ目の方法を使用することをお勧めします。

## ルートデバイスタイプによる AMI の選択

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

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

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [AMIs] をクリックします。
3. フィルタリストから、イメージタイプ ( [Public images] など )、オペレーティングシステム ( [Amazon Linux] など )、および [EBS images] を選択します。
4. ( オプション ) 選択の参考になる追加情報を表示するには、[Show/Hide Columns] アイコンをクリックし、表示する列を更新して、[Close] をクリックします。
5. AMI を選択し、その AMI ID を記録します。

Instance store-Backed AMI を選択するには コンソールを使用する

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [AMIs] をクリックします。
3. フィルタリストから、イメージタイプ ( [Public images] など )、オペレーティングシステム ( [Amazon Linux] など )、および [Instance store images] を選択します。
4. ( オプション ) 選択の参考になる追加情報を表示するには、[Show/Hide Columns] アイコンをクリックし、表示する列を更新して、[Close] をクリックします。
5. AMI を選択し、その AMI ID を記録します。

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

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

- `describe-images` ( AWS CLI )
- `ec2-describe-images` ( Amazon EC2 CLI )
- `Get-EC2Image` ( AWS Tools for Windows PowerShell )

## インスタンスのルートデバイスタイプの判別

コンソールを使用してインスタンスのルートデバイスタイプを判別するには

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

コマンドラインを使用してインスタンスのルートデバイスタイプを判別するには

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

- `describe-instances` ( AWS CLI )
- `ec2-describe-instances` ( Amazon EC2 CLI )
- `Get-EC2Instance` ( AWS Tools for Windows PowerShell )

## 永続的ルートデバイスボリュームへの変更

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

### コンソールを使用したルートボリュームの永続的な変更

コンソールを使用して、インスタンスの起動時に `DeleteOnTermination` 属性を変更できます。実行中のインスタンスのこの属性を変更するには、コマンドラインを使用する必要があります。

コンソールを使用して、起動時にインスタンスのルートデバイスボリュームが存続するように変更するには

1. Amazon EC2 コンソールを開きます。
2. Amazon EC2 コンソールダッシュボードで、[Launch Instance] をクリックします。
3. [Choose an Amazon Machine Image (AMI)] ページで、使用する AMI を選択し、[Select] をクリックします。
4. ウィザードにしたがって [Choose an Instance Type] ページと [Configure Instance Details] ページを設定します。
5. [Add Storage] ページで、ルートボリュームの [Delete On Termination] チェックボックスの選択を解除します。
6. ウィザードの残りのページを完了した後、[Launch] をクリックします。

インスタンスの詳細ペインでルートデバイスボリュームの詳細を表示することにより、設定を確認できます。[Block devices] の隣にあるルートデバイスボリュームのエントリをクリックします。デフォルトでは、[Delete on termination] は [True] です。デフォルトの動作を変更した場合は、[Delete on termination] が [False] になっています。

## AWS CLI を使用して、インスタンスのルートボリュームが存続するように変更するには

AWS CLI を使用して、インスタンスの起動時またはインスタンスの実行中に `DeleteOnTermination` 属性を変更することができます。ルートデバイスは通常、`/dev/sda1` (Linux) または `xvda` (Windows) です。

### Example 起動時

`run-instances` コマンドを使用して、`DeleteOnTermination` 属性を `false` に設定するブロックデバイスマッピングを含めることにより、ルートボリュームを保持します。

```
aws ec2 run-instances --image-id ami-1a2b3c4d --block-device-mappings
"[{"DeviceName\":" /dev/sda1\", \"Ebs\":" {\"DeleteOnTermination\":false}}]"
other parameters...
```

`describe-instances` コマンドを使用して、次に示すようにコマンド出力の `/dev/sda1` で `BlockDeviceMappings` エントリを探すことで、`DeleteOnTermination` が `false` であることを確認します。

```
...
"BlockDeviceMappings": [
  {
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "Status": "attached",
      "DeleteOnTermination": false,
      "VolumeId": "vol-877166c8",
      "AttachTime": "2013-07-19T02:42:39.000Z"
    }
  }
]
...
```

### Example インスタンスの実行中

`modify-instance-attribute` コマンドを使用して、`DeleteOnTermination` 属性を `false` に設定するブロックデバイスマッピングを含めることにより、ルートボリュームを保持します。

```
aws ec2 modify-instance-attribute --instance-id i-5203422c --block-device-map
pings [{"DeviceName\":" /dev/sda1\", \"Ebs\":" {\"DeleteOnTermina
tion\":false}}]"
```

## Amazon EC2 CLI を使用して、インスタンスのルートデバイスボリュームが存続するように変更するには

Amazon EC2 CLI を使用して、インスタンスの起動時またはインスタンスの実行中に `DeleteOnTermination` 属性を変更することができます。ルートデバイスは通常、`/dev/sda1` (Linux) または `xvda` (Windows) です。

### Example 起動時

`ec2-run-instances` コマンドを使用して、ルートデバイスの `DeleteOnTermination` フラグを `false` に設定するブロックデバイスマッピングを含めます。コマンドを詳細モードで実行するための `-v` オプションを追加します。

```
ec2-run-instances ami-1a2b3c4d -b /dev/sda1::false other parameters... -v
```

Windows システム上でコマンドラインツールを使用している場合は、ブロックデバイスマッピングの値を引用符で囲む必要があります。

```
ec2-run-instances ami-1a2b3c4d -b "xvda::false" other parameters... -v
```

詳細モードでコマンドを実行すると、基になるリクエストとレスポンスを見ることができ、ここで示すように、DeleteOnTermination が false であることを確認できます。

```
...  
<blockDeviceMapping>  
  <item>  
    <deviceName>/dev/sda1</deviceName>  
    <ebs>  
      <deleteOnTermination>>false</deleteOnTermination>  
    </ebs>  
  </item>  
</blockDeviceMapping>  
...
```

#### Example インスタンスの実行中

[ec2-modify-instance-attribute](#) コマンドを使用して、DeleteOnTermination 属性を false に設定することにより、ルートボリュームを保持します。

```
ec2-modify-instance-attribute i-5203422c -b /dev/sda1::false
```

Windows システム上でコマンドラインツールを使用している場合は、ブロックデバイスマッピングの値を引用符で囲む必要があります。

```
ec2-modify-instance-attribute i-5203422c -b "xvda::false"
```

# Amazon EC2 でのセットアップ

---

## Abstract

Amazon EC2 を初めて使用する準備を整えるには、以下の手順を実行します。

Amazon EC2 を初めて使用する場合は、事前に以下のタスクをすべて実行してください。

1. [今すぐ無料アカウント作成](#) (p. 21)
2. [IAM ユーザーを作成する](#) (p. 22)
3. [キーペアを作成する](#) (p. 23)
4. [セキュリティグループの作成](#) (p. 25)

## 今すぐ無料アカウント作成

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

Amazon EC2 については、お客様が利用された分のみのお支払いとなります。AWS の新規の顧客である場合は、Amazon EC2 を無料で使い始めることができます。詳細については、「[AWS Free Usage Tier](#)」を参照してください。

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

AWS アカウントを作成するには

1. <http://aws.amazon.com> にアクセスし、[サインアップ] をクリックします。
2. 画面上の指示に従ってください。

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

次のタスクで AWS アカウント番号が必要となるので、メモしておいてください。

## IAM ユーザーを作成する

AWS のサービス ( Amazon EC2 など ) の場合は、サービスにアクセスする際に認証情報を提供する必要があります。このため、サービスのリソースにアクセスする権限があるかどうかはサービスによって判定されます。コンソールを使用するにはパスワードが必要です。AWS アカウントのアクセスキーを作成して、コマンドラインインターフェイスまたは API にアクセスすることができます。ただし、AWS アカウントの認証情報を使って AWS にアクセスすることはお勧めしません。代わりに AWS Identity and Access Management ( IAM ) を使用することをお勧めします。IAM ユーザーを作成して、管理権限を使ってこのユーザーを IAM グループに追加するか、管理権限を付与します。これで、特殊な URL と IAM ユーザーの認証情報を使って、AWS にアクセスできます。

AWS にサインアップしても、ご自分の IAM ユーザーをまだ作成していない場合は、IAM コンソールを使用して作成できます。

IAM ユーザーの管理グループを作成するには

1. IAM コンソール ( <https://console.aws.amazon.com/iam/> ) を開きます。

AWS へのサインアップ時に使用したメールアドレスとパスワードを入力します。
2. ダッシュボードから [Create a New Group of Users] をクリックします。
3. [Group Name] ボックスに `Administrators` を入力します。
4. [Administrator Access] ポリシーテンプレートを選択します。これにより、AWS リソースの任意のアクションを使用する権限がグループ内のユーザーに付与されます。選択したら [Continue] をクリックします。
5. [Create New Users] タブのボックス [1] に自分の IAM ユーザー名を入力し、[Continue] をクリックします。
6. メッセージが表示されたら、[Continue] をクリックします。
7. Amazon EC2 の CLI または API を使用する予定がある場合は、[Download Credentials] または [Show User Security Credentials] をクリックして、アクセスキーを安全な場所に保存します。完了したら、[Close Window] をクリックします。このウィンドウを閉じると、秘密のアクセスキーが取得できなくなることに注意してください。
8. AWS マネジメントコンソールを使用する予定がある場合は、ナビゲーションペインの [Users] をクリックし、以下の手順に従います。
  - a. 作成したばかりのユーザーを選択します。
  - b. 詳細ペインの [Security Credentials] タブをクリックします。
  - c. [Sign-In Credentials] で、[Manage Password] をクリックします。
  - d. [Manage Password] ダイアログボックスで、オプションを選択して [Apply] をクリックします。
  - e. 自動生成されたパスワードを割り当てた場合、[Download Credentials] または [Show User Security Credentials] をクリックし、パスワードを安全な場所に保存します。

新規の IAM ユーザーとしてサインインするには、AWS コンソールからサインアウトし、次の URL を使用します。このとき、`your_aws_account_id` はハイフンを除いた AWS アカウント番号です ( たとえば AWS アカウント番号が `1234-5678-9012` であれば、AWS アカウント ID は `123456789012` となります )。

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

作成した IAM ユーザー名とパスワードを入力します。サインインすると、ナビゲーションバーに「`your_user_name @ your_aws_account_id`」が表示されます。

サインページの URL に AWS アカウント ID を含めない場合は、アカウントのエイリアスを作成します。IAM ダッシュボードから [Create Account Alias] をクリックし、エイリアス (会社名など) を入力します。アカウントエイリアスを作成した後、サインインするには、次の URL を使用します。

```
https://your_account_alias.signin.aws.amazon.com/console/
```

アカウントの IAM ユーザーのサインインリンクを確認するには、IAM コンソールを開き、ダッシュボードの [AWS Account Alias] の下を確認します。

IAM の詳細については、[IAM と Amazon EC2 \(p. 486\)](#) を参照してください。

## キーペアを作成する

AWS ではパブリックキー暗号化を使用して、お客様のインスタンスのログイン情報を保護します。Linux インスタンスにはパスワードがありませんが、キーペアを使用することでインスタンスに安全にログインできます。インスタンスを起動するときにキーペアの名前を指定し、ログインするときに秘密キーを指定します。

キーペアをまだ作成していない場合は、Amazon EC2 コンソールを使用して作成できます。複数のリージョンでインスタンスを起動する予定がある場合は、各リージョンでキーペアを作成する必要があります。リージョンの詳細については、「[リージョンとアベイラビリティゾーン \(p. 8\)](#)」を参照してください。

キーペアを作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーで、キーペアを生成するリージョンを選択します。お客様は場所に関係なく、使用できるリージョンをどれでも選択できます。ただし、キーペアはリージョンに固有です。たとえば、米国西部 (オレゴン) リージョンでインスタンスを起動する予定がある場合、米国西部 (オレゴン) リージョンのインスタンス用にキーペアを作成する必要があります。



3. ナビゲーションペインで [Key Pairs] をクリックします。
4. [Create Key Pair] をクリックします。

5. [Create Key Pair] ダイアログボックスの [Key Pair Name] フィールドに新しいキーペアの名前を入力し、[Create] をクリックします。覚えやすい名前 ( IAM ユーザー名など ) を選び、その後に `-key-pair` を続け、さらにリージョン名を続けます。たとえば、`me-key-pair-uswest2` などです。
6. ブラウザによって秘密キーファイルが自動的にダウンロードされます。ベースファイル名はキーペアの名前として指定した名前となり、ファイル名の拡張子は `.pem` となります。プライベートキーファイルを安全な場所に保存します。



#### Important

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

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

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

詳細については、[Amazon EC2 のキーペア \(p. 470\)](#) を参照してください。

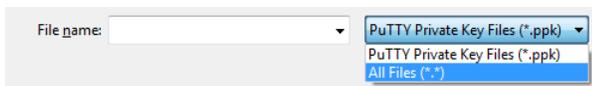
Mac または Linux を実行しているコンピュータから Linux インスタンスに接続する場合、`-i` オプションとプライベートキーへのパスを指定して、SSH クライアントに対する `.pem` ファイルを指定します。Windows を実行するコンピュータから Linux インスタンスに接続する場合は、MindTerm または PuTTY のいずれも使用できます。PuTTY を使用する予定がある場合は、それをインストールしてから、次のプロシージャを使用して `.pem` ファイルを `.ppk` ファイルに変換します。

( オプション ) PuTTY を使用して Windows から Linux インスタンスに接続するには

1. <http://www.chiark.greenend.org.uk/~sgtatham/putty/> から PuTTY をダウンロードしてインストールします。必ずスイート全体をインストールします。
2. PuTTYgen を開始します ( 例:[Start] メニューで All Programs > PuTTY > PuTTYgen をクリック ) 。
3. [Type of key to generate] の下で、[SSH-2 RSA] を選択します。



4. [Load] をクリックします。デフォルトでは、PuTTYgen には拡張子 `.ppk` を持つファイルだけが表示されます。`.pem` ファイルの場所を特定するには、すべてのタイプのファイルを表示するオプションを選択します。



5. 前の手順で作成した作成した秘密キーファイルを選択し、[Open] をクリックします。[OK] をクリックして、確認ダイアログボックスを閉じます。
6. [Save private key] をクリックします。PuTTYgen に、パスフレーズなしでキーを保存することに関する警告が表示されます。[Yes] をクリックします。
7. キーペアに使用した名前と同じ名前をキーに指定します。PuTTY は自動的にファイル拡張子 `.ppk` を加えます。

## セキュリティグループの作成

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

複数のリージョンでインスタンスを起動する予定がある場合は、各リージョンでセキュリティグループを作成する必要があります。リージョンの詳細については、「[リージョンとアベイラビリティゾーン \(p. 8\)](#)」を参照してください。

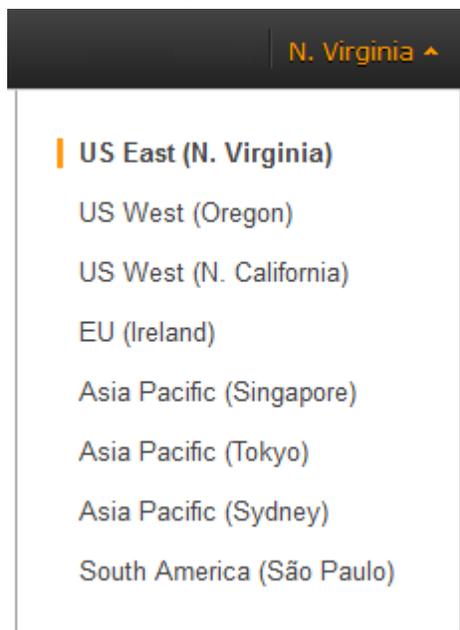


### Tip

ローカルコンピュータのパブリック IP アドレスが必要になります。このアドレスはサービスを使って取得できます。例えば、次のサービスが提供されています。  
<http://checkip.amazonaws.com/> IP アドレスを提供する別のサービスを検索するには、検索フレーズ「what is my IP address」を使用します。インターネットサービスプロバイダ (ISP) 経由で、またはファイアウォールの内側から静的な IP アドレスなしで接続する場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。

最小限の権限でセキュリティグループを作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーで、セキュリティグループのリージョンを選択します。セキュリティグループはリージョンに固有です。たとえば、米国西部 (オレゴン) リージョンでインスタンスを起動する予定がある場合、米国西部 (オレゴン) リージョンのインスタンス用にセキュリティグループを作成する必要があります。



3. ナビゲーションペインで [Security Groups] をクリックします。
4. [Create Security Group] をクリックします。
5. 新しいセキュリティグループの名前と説明を入力します。覚えやすい名前 (IAM ユーザー名など) を選び、その後、\_SG\_ を続け、さらにリージョン名を続けます。たとえば、me\_SG\_uswest2 などです。

6. [Inbound] タブで、次のルールを作成 (新しいルールごとに [Add Rule] をクリック) し、終了したら、[Create] をクリックします。
  - [Protocol] リストから [HTTP] を選択し、[Source] を [Anywhere] ( 0.0.0.0/0 ) に設定します。
  - [Type] リストから [HTTPS] を選択し、[Source] が [Anywhere] ( 0.0.0.0/0 ) に設定されていることを確認します。
  - [Type] リストから [SSH] を選択します。[Source] ボックスで、[Custom IP] が選択されていることを確認し、コンピュータまたはネットワークのパブリック IP アドレスを CIDR 表記で指定します。CIDR 表記で個々の IP アドレスを指定するには、ルーティングプレフィックスを追加します。/32 例えば、IP アドレスが 216.182.234.123 の場合は、216.182.234.123/32 を指定します。会社が特定の範囲からアドレスを割り当てている場合、範囲全体 ( 203.0.113.0/24 など ) を指定します。



#### Caution

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

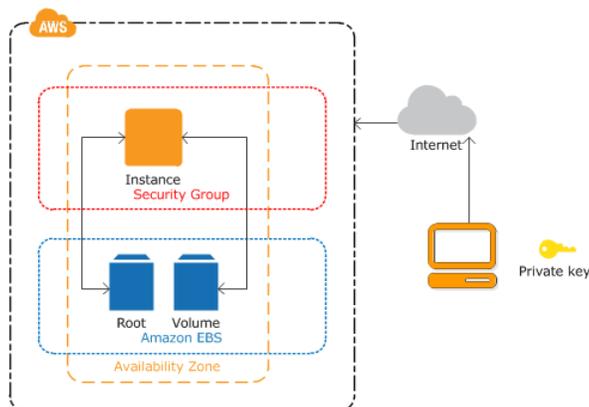
詳細については、「[Amazon EC2 セキュリティグループ \(p. 477\)](#)」を参照してください。

# Amazon EC2 Linux インスタンスの使用開始

## Abstract

Linux インスタンスを起動、接続、使用して Amazon EC2 の使用を開始しましょう。

Linux インスタンスを起動、接続、使用して Amazon Elastic Compute Cloud ( Amazon EC2 ) の使用を開始しましょう。AWS マネジメントコンソール、ポイントアンドクリックウェブベースインターフェイスを使用して、次の図に示す例のアーキテクチャを完成させます。



インスタンスは Amazon EBS バックトインスタンスです ( ルートボリュームが Amazon EBS ボリュームであることを意味します )。また、追加の Amazon EBS ボリュームを作成およびアタッチします。インスタンスが実行されるアベイラビリティゾーンを指定するか、Amazon がアベイラビリティゾーンを選択します。インスタンスを起動するときは、キーペアとセキュリティグループを指定してインスタンスをセキュリティで保護します ( この演習では、セットアップ時にキーペアとセキュリティグループを作成したことを前提としています。「[Get Set Up for Amazon EC2](#)」を参照してください )。インスタンスに接続するときは、インスタンスの起動時に指定したキーペアの秘密キーを指定する必要があります。

この演習を完了するには、次のタスクを実行します。

1. [Amazon EC2 インスタンスの起動 \(p. 28\)](#)

2. [インスタンスへの接続 \(p. 29\)](#)
3. [インスタンスへのボリュームの追加 \(p. 32\)](#)
4. [インスタンスとボリュームのクリーンアップ \(p. 36\)](#)

#### Related Topics

Windows インスタンスを起動する方法については、チュートリアル [Getting Started with Amazon Amazon EC2 Windows Instances](#) を参照してください。

AWS CLI を使用する方法については、*AWS Command Line Interface ユーザーガイド* のチュートリアル [Using Amazon EC2 through the AWS CLI](#) を参照してください。

Amazon EC2 CLI を使用する方法については、*Amazon Elastic Compute Cloud コマンドラインリファレンス* のチュートリアル [Launching an Instance Using the Amazon EC2 CLI](#) を参照してください。

他の AWS 製品や Amazon EC2 によるサービスの使用方法を説明するチュートリアルについては、[Getting Started with AWS](#) を参照してください。

## Amazon EC2 インスタンスの起動

### Abstract

AWS マネジメントコンソールを使って Amazon EC2 Linux インスタンスの起動します。

このトピックで説明しているように、AWS マネジメントコンソールを使用して Linux インスタンスを起動できます。開始する前に、[Get Set Up for Amazon EC2](#) のステップを完了していることを確認してください。インスタンスを起動した後、インスタンスに接続して使用できます。詳細については、[インスタンスへの接続 \(p. 29\)](#) を参照してください。

Windows インスタンスを起動して接続する方法をご希望の場合は、[Getting Started with Amazon EC2 Windows Instances](#) を参照してください。



#### Important

AWS にサインアップすると、[AWS 無料利用枠](#) を利用して、Amazon EC2 を無料で使い始めることができます。AWS アカウントを作成したのが過去 12 か月以内で、Amazon EC2 と Amazon EBS の無料利用枠を使い切っていない場合、無料利用枠内で利用できるオプションを選択することで、このチュートリアルでは一切費用がかかりません。それ以外の場合、インスタンスを起動したときから、インスタンスを終了するまで (このチュートリアルの最終タスク)、アイドル状態のままでも標準の Amazon EC2 使用料が発生します。無料利用枠の範囲外でこのチュートリアルを完了するための合計費用は最少です (通常は数ドルのみです)。

次の手順の目的は最初のインスタンスをすばやく起動することで、実行可能なすべての操作を説明することではありません。高度なオプションの詳細については、[Launching an Instance](#) を参照してください。

インスタンスを起動するには

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. コンソールダッシュボードで、[Launch Instance] をクリックします。
3. [Select an Amazon Machine Image (AMI)] ページには、インスタンスのテンプレートとして機能する「Amazon マシンイメージ (AMI)」と呼ばれる基本構成リストが表示されます。[64-bit Amazon Linux AMI] を選択します。この設定は「Free tier eligible」と表示されます。

4. [Select an Instance Type] ページで、インスタンスのハードウェア構成を選択できます。デフォルトでは、`t1.micro` インスタンスが選択されます。[Review and Launch] をクリックして、ウィザードが他の構成設定を完了し、すばやく起動できるようにします。
5. [Review Instance Launch] ページで、インスタンスの設定を確認できます。

[Security Group] に、ウィザードが作成および選択したセキュリティグループが表示されます。代わりに、次のステップを使ってセットアップ時に作成したセキュリティグループを選択します。

- a. [Edit security groups] をクリックします。
  - b. [Configure Security Group] ページで、[Select an existing security group] オプションが選択されていることを確認します。
  - c. 既存のセキュリティグループのリストからセキュリティグループを選択し、[Review and Launch] をクリックします。
6. [Review Instance Launch] ページで、[Launch] をクリックします。
  7. [Select an existing key pair or create a new key pair] ダイアログボックスで、[Choose an existing key pair] を選択し、設定時に作成したキーペアを選択します。

別の方法として、新しいキーペアを作成できます。[Create a new key pair] を選択し、キーペアの名前を入力して、[Download Key Pair] をクリックします。これは、プライベートキーファイルを保存できる唯一のチャンスなので、必ずダウンロードしてください。プライベートキーファイルを安全な場所に保存します。インスタンスと対応するプライベートキーの起動時には、毎回インスタンスに接続するたびに、キーペアの名前を入力する必要があります。

キーペアは、SSH 経由で Linux インスタンスに接続できるようにします。したがって、[Proceed without a Key Pair] オプションは選択しないでください。キーペアを使用せずにインスタンスを起動すると、インスタンスに接続できません。

準備ができたら、確認チェックボックスをオンにして、[Launch Instances] をクリックします。

8. インスタンスを起動することを知らせる確認ページが表示されます。[View Instances] をクリックして確認ページを閉じ、コンソールに戻ります。
9. [Instances] 画面で、インスタンスの状態を確認できます。インスタンスはすぐに起動します。インスタンスを起動すると、初期状態は `pending` になります。インスタンスを起動した後は、状態が `running` に変わり、パブリック DNS 名を受け取ります ( [Public DNS] 列が非表示の場合は、[Show/Hide] アイコンをクリックし、[Public DNS] を選択します ) 。

## インスタンスへの接続

### Abstract

Linux インスタンスに接続します。

インスタンスを起動したら、これに接続し、普通のコンピュータと同じように使用できます。

インスタンスの接続でエラーが発生した場合は、「[Troubleshooting Connecting to Your Instance](#)」を参照してください。

インスタンスに接続を試みる前に、次のタスクを完了したことを確認します。

- インスタンスのパブリック DNS 名を取得する  
Amazon EC2 コンソールを使用して、インスタンスのパブリック DNS を取得できます ( [Public DNS] 列を確認します。この列が非表示の場合は、[Show/Hide] アイコンをクリックして [Public DNS] を選択します )。もしくは、`ec2-describe-instances` コマンドを使用できます。
- プライベートキーを見つける

インスタンスの起動時に指定したキーペアの `.pem` ファイルの完全修飾パスが必要です。

- IP アドレスからインスタンスへのインバウンド SSH トラフィックを有効にする  
インスタンスに関連付けられているセキュリティグループで、IP アドレスからの受信 SSH トラフィックが許可されることを確認します。詳細については、「[Authorizing Network Access to Your Instances](#)」を参照してください。

Linux インスタンスに接続するにはいくつかの方法があります。ニーズに合った方法を選択します。

- [オプション 1: ブラウザを使用した接続 \(p. 30\)](#)
- [オプション 2: PuTTY を使用した Windows からの接続 \(p. 31\)](#)
- [オプション 3: SSH クライアントを使用した Mac または Linux からの接続 \(p. 32\)](#)

### 次のステップ

インスタンスの起動と接続が正常に完了したら、以下のいずれかの操作を実行できます。

- このチュートリアル 次の手順 [インスタンスへのボリュームの追加 \(p. 32\)](#) に進んでください。
- [Installing a LAMP Web Server](#) や [Hosting a WordPress Blog](#) など、別のチュートリアルで、このインスタンスを続けて使用します。
- このチュートリアルの最後の手順 [インスタンスとボリュームのクリーンアップ \(p. 36\)](#) にスキップして、インスタンスを終了し、料金の発生が継続しないようにします。

## オプション 1: ブラウザを使用した接続

ブラウザに Java がインストールされて有効になっている必要があります。Java を持っていない場合は、システム管理者に連絡してインストールしてもらうか、「[Install Java](#)」および「[Enable Java in your web browser](#)」の各ページに説明されているステップに従います。

ウェブブラウザを使用して Linux インスタンスに接続するには

1. Amazon EC2 コンソールから、ナビゲーションペインで [Instances] をクリックします。
2. インスタンスを選択し、[Connect] をクリックします。
3. [A Java SSH client directly from my browser ( Java required ) ] をクリックします。
4. Amazon EC2 は、インスタンスのパブリック DNS 名を自動的に検出し、[Public DNS] に自動入力します。また、インスタンスの起動時に指定したキーペアも検出します。次のように入力し、[Launch SSH Client] をクリックします。
  - a. [User name] に「`ec2-user`」と入力します。



#### Tip

Amazon Linux の場合は、デフォルトのユーザー名は `ec2-user` です。RHEL5 の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。Ubuntu の場合は、ユーザー名は `ubuntu` です。SUSE Linux の場合は、ユーザー名は `root` です。それ以外の場合は、AMI プロバイダーに確認してください。

- b. [Private key path] に、プライベートキー (`.pem`) ファイルの完全修飾パスを入力します。
- c. ブラウザキャッシュにプライベートキーの場所を格納するには、[Store in browser cache] をクリックします。これにより、Amazon EC2 はユーザーがブラウザのキャッシュをクリアするまで、以降のブラウザセッションでプライベートキーの場所を検出します。

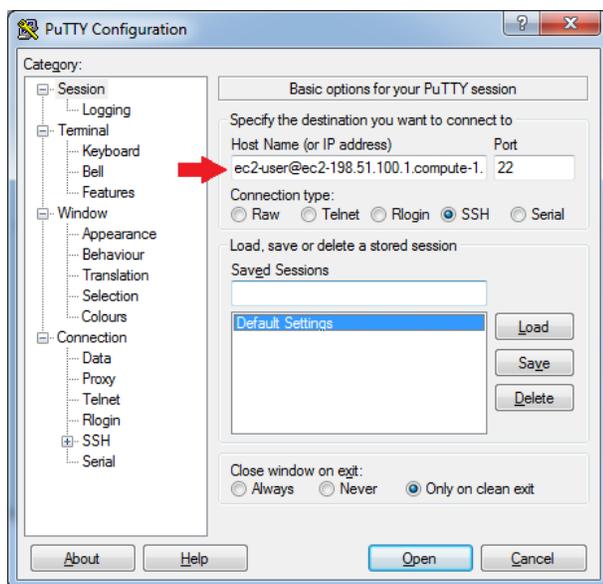
5. このホストを既知のホストに追加するかどうかの確認を求められたら、[No] をクリックします。
6. 必要に応じて、証明書を信頼するかどうかを指定する画面で [Yes] をクリックします。
7. [Run] をクリックして MindTerm クライアントを実行します。
8. ライセンス契約書に同意する場合は、[Accept] をクリックします。
9. MindTerm を実行するのが初めての場合、ホームディレクトリなどの設定を確認するための一連のダイアログボックスが表示されます。これらの設定を確認します。ウィンドウが開き、インスタンスに接続した状態になります。

## オプション 2: PuTTY を使用した Windows からの接続

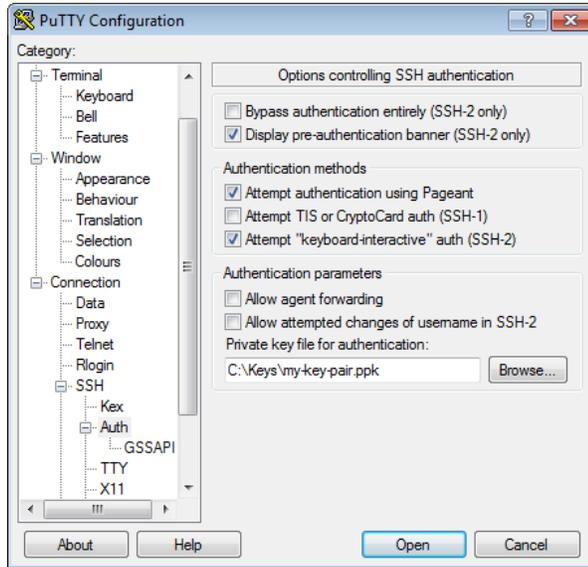
PuTTY は .pem ファイルを使用せず、.ppk ファイルを使用します。.ppk ファイルを生成していない場合は、ここで生成します。詳細については、「[To prepare to connect to a Linux instance from Windows using PuTTY](#)」を参照してください。

PuTTY を使用して Linux インスタンスに接続するには

1. PuTTY を開始します ([Start] メニューで All Programs > PuTTY > PuTTY をクリックします)。
2. [Category] ペインで [Session] を選択し、次のフィールドに入力します。
  - a. [Host Name] ボックスに「ec2-user@public\_dns\_name」と入力します。
  - b. [Connection type] で [SSH] を選択します。
  - c. [Port] が 22 であることを確認します。



3. [Category] ペインで、[Connection]、[SSH] の順に展開し、[Auth] を選択します。次のように入力します。
  - a. [Browse] をクリックします。
  - b. キーペア用に生成した .ppk ファイルを選択し、[Open] をクリックします。
  - c. [Open] をクリックして PuTTY セッションを開始します。



4. このインスタンスに接続するのが初めての場合は、接続先のホストを信頼するかどうかを尋ねるセキュリティアラートダイアログボックスが表示されます。[Yes] をクリックします。ウィンドウが開き、インスタンスに接続した状態になります。

## オプション 3: SSH クライアントを使用した Mac または Linux からの接続

ほとんどの場合、Mac または Linux コンピュータにはデフォルトで SSH クライアントがあります。SSH クライアントがあるかどうかを確認するには、コマンドラインで `ssh` と入力します。使用しているコンピュータでコマンドが認識されない場合は、OpenSSH プロジェクトから、SSH ツールの完全なスイートの無料実装が提供されています。詳細については、<http://www.openssh.org> を参照してください。

コマンドシェルを開き、次のコマンドを実行します。

```
$ ssh -i /path/key_pair.pem ec2-user@public_dns_name
```

### Tip

Amazon Linux の場合は、デフォルトのユーザー名は `ec2-user` です。RHEL5 の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。Ubuntu の場合は、ユーザー名は `ubuntu` です。SUSE Linux の場合は、ユーザー名は `root` です。それ以外の場合は、AMI プロバイダーに確認してください。

## インスタンスへのボリュームの追加

### Abstract

Amazon EBS ボリュームを Linux インスタンスに追加します。

これで Linux インスタンスを起動および接続したので、インスタンスで次のコマンドを実行して、マウントされたボリュームを表示することができます。

```
[ec2-user ~]$ df -h
```

マイクロインスタンスの場合、出力は次のようになるはずですが。

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda1	8.0G	1.1G	6.9G	14%	/
tmpfs	298M	0	298M	0%	/dev/shm

/dev/xvda1 ポリユームはルートデバイスポリユームです。インスタンスの起動に使用されるイメージが含まれています。インスタンスには追加のソフトウェアをいくつかインストールすることができます。例えば、yum コマンドを使ってパッケージをダウンロードおよびインストールできます。

データに追加のストレージが必要な場合、簡単な方法として Amazon EBS ポリユームをインスタンスに追加できます。Amazon EBS ポリユームは、インスタンスのネットワークアタッチドストレージとして機能します。起動した Linux インスタンスにポリユームを追加しましょう。最初に、EC2 コンソールを使ってポリユームを作成し、インスタンスにアタッチしてから、ポリユームをマウントして使用できるようにします。

Amazon EBS ポリユームを作成してアタッチするには

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. ナビゲーションバーで、インスタンスを作成したリージョンを選択し、ナビゲーションペインで [Instances] をクリックします。  
選択したリージョンの現在のインスタンス一覧がコンソールに表示されます。Linux インスタンスを選択します。下部のペインの [Description] タブで、インスタンスの [Availability Zone] を書き留めます。
3. ナビゲーションペインの [Elastic Block Store] で [Snapshots] をクリックします。フィルタリストから [Public Snapshots] を選択します。リストからスナップショットを選択し、そのスナップショット ID を書きとめておきます。Amazon Elastic Block Storage の無料利用枠は最大 30 GB までなので、このチュートリアルでは料金が発生しないように、30 GB 以下のスナップショットを選択します。  
このチュートリアルでは、この手順で説明したスナップショットを使って、ポリユームを作成することを前提としています。代わりに空のポリユームを作成した場合、次の追加の手順を実行してください。
4. [Create Volume] をクリックします。
5. [Create Volume] ダイアログボックスは、選択したスナップショットのスナップショット ID とポリユームサイズであらかじめ設定されています。次のように設定し、[Create] をクリックします。
  - [Standard] ポリユームタイプを選択して、標準的な EBS ポリユームを作成します。
  - インスタンスの作成時に使用した [Availability Zone] を選択します。別のゾーンを選択すると、インスタンスにポリユームをアタッチできません。
6. ナビゲーションペインの [Elastic Block Store] で、[Volumes] をクリックします。新しく作成したポリユームがそこに表示され、ポリユームの状態は `available` なので、インスタンスにアタッチする準備ができています。
7. 新しく作成したポリユームを右クリックし、[Attach Volume] を選択します。
8. [Attach Volume] ダイアログボックスで、以下のとおり設定を行ってから、[Attach] をクリックします。
  - インスタンスの名前または ID を入力して、提案されたオプションのリストからインスタンスを選択します。

- そのインスタンスの未使用のデバイス名を指定します。このチュートリアルでは、`/dev/sdf` を使用します。別のデバイス名を選択した場合、必ずそのデバイス名を書き留めてください。この情報は次の手順で必要になります。

ポリリュームの [Details] ペインで、ポリリュームの状態が `in-use` になっているので、ポリリュームがインスタンスにデバイス名 `/dev/sdf` としてアタッチされているのがわかります。ただし、インスタンスに戻って `df -h` コマンドをもう一度実行しても、ポリリュームはまだ表示されません。これは、`df -h` 用にポリリュームをマウントして、表示する必要があるためです。ただし、`lsblk` コマンドを実行すると、インスタンスに接続されているすべてのブロックデバイスが表示されます。



#### Note

一部の Linux のディストリビューションは、`lsblk` コマンドをデフォルトで提供していません。`lsblk` コマンドが機能しない場合は、代わりに `sudo fdisk -l | grep Disk` コマンドを使用できません。

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf        202:80   0  100G  0 disk
  xvdf1     202:81   0   10G  0 part
xvda1       202:1    0    8G  0 disk /
```

上の例では、`lsblk` によって、インスタンスに 2 つのブロックデバイスがアタッチされていることが報告されます。`xvda1` はルートファイルシステムのためにマウントされ ( / の `MOUNTPOINT` 値を書き留めます )、`xvdf` ( ディスクパーティション `xvdf1` が含まれます ) はマウントされません。

ポリリュームを使用できるようにするには

1. マウントするデバイスを特定します。前の手順で、新しいポリリュームが `/dev/sdf` にアタッチされました。インスタンスのオペレーティングシステムのブロックデバイスドライバに応じて、デバイスはコンソールで指定した場所 ( 前の例では `/dev/sdf` ) とは異なる場所 ( `/dev/xvdf` など ) に表示されることがあります。一部のケースでは、末尾の文字が変更されることもあります ( `/dev/xvdj` など )。常に Amazon Linux インスタンスは、コンソールで指定したデバイスパスから新しいデバイスパスへのリンクを作成しますが、他の配布版 ( Ubuntu や Red Hat など ) は予想できません。

使用できるデバイスのリストを表示するには、`lsblk` コマンドを使用します。



#### Note

一部の Linux のディストリビューションは、`lsblk` コマンドをデフォルトで提供していません。`lsblk` コマンドが機能しない場合は、代わりに `sudo fdisk -l | grep Disk` コマンドを使用できます。

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf        202:80   0  100G  0 disk
  xvdf1     202:81   0   10G  0 part
xvda1       202:1    0    8G  0 disk /
```

`xvdf` デバイスはマウントされず、`xvdf1` に 10 GB のパーティションが存在します。スナップショットからポリリュームを作成した場合、このようなポリリュームのデータには、ポリリュームのルートの代わりに、パーティションが含まれることがあります。この場合、`/dev/xvdf1` パーティシヨ

ンをマウントします (lsblk コマンドの出力では、ファイルパスの /dev/ 部分が省略されます)。xvdf にパーティションが存在しない場合、/dev/xvdf がマウントされます。

2. (オプション) 前の手順でスナップショットからポリリュームを作成する代わりに空のポリリュームを作成した場合、そのポリリュームをマウントする前に、mkfs を使ってポリリュームをフォーマットする必要があります。次のコマンドを使って、ポリリュームに ext4 ファイルシステムを作成します。`device_name` にデバイス名 (/dev/xvdf など) を置換します。



#### Caution

この手順では、空のポリリュームをマウントすることを前提としています。既にデータが含まれるポリリューム (スナップショットから復元したポリリュームなど) をマウントする場合は、ポリリュームのマウント前に mkfs を使用しないでください (代わりに、次の手順に移動します)。ステップ 1 を実行した場合、ポリリュームがフォーマットされ、既存のデータが削除されます。詳細については、[Making the Volume Available on Linux](#) を参照してください。



#### Note

SUSE Linux Enterprise Server 11 は、ext4 ファイルシステムを完全にはサポートしていません。インスタンスに SLES 11 AMI を選択する場合は、次のコマンドで ext3 を代わりに使用します。

```
[ec2-user ~]$ sudo mkfs -t ext4 device_name
```

3. デバイスを /mnt/my-data としてマウントするために、次のコマンドを実行します。

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

[Step 1 \(p. 34\)](#) で特定したデバイス名を必ず指定してください。指定しない場合、このマウントコマンドを実行すると、次のエラーが表示される可能性があります。「mount: you must specify the filesystem type」。このエラーが表示された場合、[Step 1 \(p. 34\)](#) を繰り返して、正しいデバイスパスを使用します (lsblk コマンドで取得したデバイス名に /dev/ を追加することを忘れないでください)。

4. df -h コマンドを実行すると、次のような出力が表示されます。

```
[ec2-user ~]$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/xvda1      7.9G  1.1G  6.8G  14% /  
tmpfs           298M    0  298M   0% /dev/shm  
/dev/xvdf1      10G   76M   10G   1% /mnt/my-data
```

5. 新しいポリリュームの内容を表示するには、次のコマンドを実行します。

```
[ec2-user ~]$ ls /mnt/my-data
```

この時点で、このチュートリアルの例のアーキテクチャを作成しました。必要に応じて、インスタンスを引き続きカスタマイズして使用できます。



### Important

インスタンスを無料利用枠内で起動する限り、料金はかかりません。それ以外の場合、インスタンスが起動するとすぐに、インスタンスの実行時間に応じて、インスタンスがアイドル状態の場合でも課金されます。通常のインスタンスのステータスが `shutting down` または `terminated` になると、インスタンスの使用料は発生しなくなります。

インスタンスの使用を終えるときは、必ず使用したリソースをクリーンアップします。手順は次のステップ [インスタンスとボリュームのクリーンアップ](#) (p. 36) で説明します。

## インスタンスとボリュームのクリーンアップ

### Abstract

Linux インスタンスを終了し、作業を行ったボリュームを削除すると、インスタンスがクリーンアップされます。

このチュートリアル用に作成したインスタンスと Amazon EBS ボリュームを終了したら、クリーンアップする必要があります。最初に、インスタンスを終了します。これにより、ボリュームがインスタンスからデタッチされ、ボリュームが削除されます。

インスタンスを終了するという事は、実質的には、そのインスタンスを削除するという事です。いったん終了したインスタンスに再接続することはできません。これは、インスタンスを停止するのとは異なります。インスタンスを停止すると、そのインスタンスはシャットダウンし、再開するまで時間使用料やデータ転送料はかかりません (アタッチされている Amazon EBS ボリュームストレージの使用料はかかります)。また、停止したインスタンスはいつでも再開できます。インスタンスの停止と終了の違いに関する詳細は、[インスタンスの停止](#)を参照してください。

インスタンスを終了するには

1. [Instances] ページのインスタンス一覧からインスタンスを見つけます。インスタンスが見つからない場合は、正しいリージョンを選択したことを確認します。
2. インスタンスを右クリックしてから、[Terminate] をクリックします。
3. 確認を求められたら、[Yes, Terminate] をクリックします。

EBS ボリュームは、インスタンスの終了後も持続可能です。前のステップで作成しアタッチしたのが EBS ボリュームなら、インスタンスの終了時にボリュームはデタッチされています。ただし、このボリュームを削除しない限り、ストレージ容量が無料利用枠の上限を超えた時点でボリュームストレージの使用料が発生します。ボリュームを削除すると、中のデータは消去され、そのボリュームはどのインスタンスにもアタッチできなくなります。

ボリュームを削除するには

1. [Volumes] ページのボリューム一覧から、自分が作成したボリュームを見つけます。ボリュームが見つからない場合は、正しいリージョンを選択したことを確認します。
2. ボリュームを右クリックしてから、[Delete Volume] をクリックします。
3. 確認プロンプトが表示されたら、[Yes, Delete] をクリックします。  
Amazon EC2 がボリュームの削除を開始します。

# Amazon EC2 のベストプラクティス

## Abstract

このチェックリストを使用して、Amazon EC2 の使用を評価します。

このチェックリストは、Amazon EC2 のメリットと満足感を最大限に引き出せるようにするためのものです。

- ID フェデレーション、IAM ユーザー、IAM ロールを使用して、AWS リソースおよび API へのアクセスを管理します。AWS アクセス認証情報の作成、配布、ローテーション、および取り消しを行うための認証情報管理のポリシーおよび手順を確立します。詳細については、『IAM を使用する』ガイドの「IAM ベスト プラクティス」を参照してください。
- EC2-Classic ではなく VPC でインスタンスを起動します。この動作を行う利点については、「Amazon EC2 と Amazon Virtual Private Cloud ( VPC ) ( p. 524)」を参照してください。
- セキュリティグループに対して、最も制限のないルールを適用します。詳細については、「セキュリティグループのルール ( p. 479)」を参照してください。
- データの永続性、バックアップ、および復元に対するルートデバイスタイプの影響について理解します。詳細については、「ルートデバイスのストレージ ( p. 57)」を参照してください。
- オペレーティングシステム用およびデータ用として個別に Amazon EBS ボリュームを使用します。
- インスタンスが再開したときに、動的な IP アドレスを処理するアプリケーションを設計します。詳細については、「Amazon EC2 インスタンスの IP アドレッシング ( p. 529)」を参照してください。
- 定期的にインスタンスのオペレーティングシステムやアプリケーションに対してパッチ処理、更新、および保護を行います。Amazon Linux の更新の詳細については、「ソフトウェアの管理 ( p. 360)」を参照してください。Windows サーバーの更新の詳細については、Microsoft ウェブサイトの [Windows Server Update Services] にアクセスしてください。
- Amazon EBS スナップショット ( p. 605) またはバックアップツールを使用して、定期的にインスタンスをバックアップします。
- 複数のアベイラビリティゾーンにアプリケーションの重要なコンポーネントをデプロイし、データを適切にレプリケートします。
- イベントを管理し、対応します。詳細については、「Amazon EC2 のモニタリング ( p. 383)」を参照してください。
- フェイルオーバーを処理する準備が整っていることを確認します。基本的な解決策として、手動でネットワークインターフェイスをアタッチすることも、代替インスタンスに Elastic IP アドレスを関連付けることもできます。詳細については、「Elastic Network Interface ( ENI ) ( p. 545)」を参照してください。自動化されたソリューションとして Auto Scaling を使用できます。詳細については、Auto Scaling 開発者ガイド を参照してください。

- 障害が発生した場合にインスタンスと Amazon EBS ボリュームを復元するプロセスを定期的にテストします。
- AWS リソースを追跡および識別するために、インスタンスメタデータおよびリソースのカスタムタグを使用します。詳細については、[インスタンスメタデータとユーザーデータ \(p. 289\)](#) および [Amazon EC2 リソースにタグを付ける \(p. 662\)](#) を参照してください。

# チュートリアル: LAMP ウェブサーバーのインストール

## Abstract

チュートリアル – Apache ウェブサーバーを PHP と MySQL のサポートとともに Amazon EC2 インスタンスにインストールします。

次の手順では、Apache ウェブサーバーを PHP と MySQL のサポートとともに Amazon EC2 インスタンスにインストールします (LAMP ウェブサーバーまたは LAMP スタックとも呼ばれます)。このサーバーを使用して静的ウェブサイトホストしたり、データベースとの情報の読み取りと書き込みを行う動的な PHP アプリケーションをデプロイしたりできます。

### 前提条件

このチュートリアルでは、インターネットからアクセス可能なパブリック DNS 名を持つインスタンスを既に起動したことを前提にしています。詳細については、「[Amazon EC2 インスタンスの起動 \(p. 28\)](#)」を参照してください。また、セキュリティグループを設定して、SSH (ポート 22)、HTTP (ポート 80)、HTTPS (ポート 443) 接続を有効にしている必要もあります。前提条件の詳細については、[Amazon EC2 でのセットアップ \(p. 21\)](#) を参照してください。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。Ubuntu インスタンスでの LAMP ウェブサーバーのセットアップは、このチュートリアルの範囲外です。Ubuntu の LAMP ウェブサーバーについては、Ubuntu コミュニティのドキュメントの [ApacheMySQLPHP](#) トピックを参照してください。

LAMP ウェブサーバーをインストールおよび起動するには

1. [インスタンスに接続します \(p. 29\)](#)。
2. すべてのソフトウェアパッケージが最新の状態であることを確認するため、インスタンスでソフトウェアの更新を実行します。この処理には数分かかりますが、最新の更新とバグ修正を確実に適用することが重要です。

**Note**

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

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

- これでインスタンスが最新状態になったので、Apache ウェブサーバー、MySQL、PHP ソフトウェアパッケージをインストールできます。yum groupinstall コマンドを使用すると、複数のソフトウェアパッケージと関連するすべての依存関係を同時にインストールできます。

```
[ec2-user ~]$ sudo yum groupinstall -y "Web Server" "MySQL Database" "PHP Support"
```

**Note**

非 Amazon Linux インスタンスは、グループ名にわずかな違いがある場合があります。無効なグループ名が原因で上記のコマンドが失敗する場合は、yum grouplist コマンドを使用し、「MySQL Database」ではなく「MySQL Database server」などの類似したグループを出力でスキャンし、ディストリビューション用の適切なグループ名を使用します。

- php-mysql パッケージをインストールします。

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

- Apache ウェブサーバーを起動します。

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

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

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

**Tip**

chkconfig コマンドでは、サービスを正常に有効にしたときに確認メッセージは一切表示されません。httpd が有効であることは、次のコマンドを実行して確認できます。

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

- ここで、httpd は実行レベル 2、3、4、および 5 で です (これらの値である必要があります)。
- ウェブサーバーをテストします。ウェブブラウザで、インスタンスのパブリック DNS アドレス (またはパブリック IP アドレス) を入力すると、Apache テストページが表示されます。Amazon EC2 コンソールを使用して、インスタンスのパブリック DNS を取得できます ([Public DNS] 列をチェックします。列が非表示の場合は、[Show/Hide] アイコンをクリックして、[Public DNS] を選択します)。

**Tip**

Apache テストページが表示されない場合、使用しているセキュリティグループに、HTTP (ポート 80) トラフィックを許可するルールが含まれていることを確認します。HTTP ルールをセキュリティグループに追加する方法については、[セキュリティグループへのルールの追加 \(p. 482\)](#) を参照してください。

**Important**

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

**Amazon Linux AMI Test Page**

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

<p><b>If you are a member of the general public:</b></p> <p>The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.</p> <p>If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.</p> <p>For example, if you experienced problems while visiting <code>www.example.com</code>, you should send e-mail to "<code>webmaster@example.com</code>".</p>	<p><b>If you are the website administrator:</b></p> <p>You may now add content to the directory <code>/var/www/html/</code>. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file <code>/etc/httpd/conf.d/welcome.conf</code>.</p> <p>You are free to use the image below on web sites powered by the Apache HTTP Server:</p> <div style="text-align: center;">  </div>
--	---

**Note**

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

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

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

ec2-user がこのディレクトリで複数のファイルを操作できるようにするには、ディレクトリの所有権と許可を変更する必要があります。このタスクを完了するには多くの方法があります。このチュートリアルでは、`www` グループをインスタンスに追加し、そのグループに `/var/www` ディレクトリの所有権を与え、グループの書き込み許可を追加します。そのグループの任意のメンバーは、ウェブサーバーに対してファイルの追加、削除、および変更ができるようになります。

ファイルの許可を設定するには

1. `www` グループをインスタンスに追加します。

```
[ec2-user ~]$ sudo groupadd www
```

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

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



#### Important

新しいグループを取得するには、ログアウトしてから再度ログインする必要があります。`exit` コマンドを使用するか、ターミナルウィンドウを閉じることができます。

3. ログアウトし、再度ログインして `www` グループでメンバーシップを確認します。
  - a. ログアウトします。

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

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

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

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

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

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

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

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

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

ここで、`ec2_user` ( および `www` グループの将来のメンバー ) は、Apache ドキュメントルートでファイルを追加、削除、編集できるようになります。これで、静的なウェブサイトや PHP アプリケーションなどのコンテンツを追加できます。

### LAMP ウェブサーバーをテストするには

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

1. Apache ドキュメントルートでシンプルな PHP ファイルを作成します。

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



#### Tip

このコマンドを実行しようとしたときに「`Permission denied`」というエラーが表示された場合は、ログアウトし、再度ログインして、[ファイルの許可を設定するには \(p. 42\)](#) で設定した正しいグループ許可を取得します。

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

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

PHP 情報ページが表示されます。

The screenshot shows the output of the `phpinfo()` function. At the top, it displays "PHP Version 5.3.27" next to the PHP logo. Below this, a table-like structure provides system information:

<b>System</b>	Linux ip-172-31-3-156 3.4.43-43.43.amzn1.x86_64 #1 SMP Mon May 6 18:04:41 UTC 2013 x86_64
<b>Build Date</b>	Jul 12 2013 22:04:56
<b>Configure Command</b>	"/configure" --build=x86_64-redhat-linux-gnu --host=x86_64-redhat-linux-gnu --target=x86_64-amazon-linux-gnu --program-prefix="" --prefix=/usr --exec-prefix=/usr --bindir=/usr/bin --sbindir=/usr/sbin --sysconfdir=/etc --datadir=/usr/share --includedir=/usr/include --libdir=/usr/lib64 --libexecdir=/usr/libexec --localstatedir=/var --sharedstatedir=/var/lib --mandir=/usr/share/man --infodir=/usr/share/info --cache-file=/config.cache --with-libdir=lib64 --with-config-file-path=/etc --with-config-file-scandir=/etc/php.d --disable-debug --with-pic --disable-rpath --without-pear --with-bz2 --with-exec-dir=/usr/bin --with-freetype-dir=/usr --with-png-dir=/usr --with-xpm-dir=/usr --enable-gd-native-ttf --with-t1lib=/usr --without-gdbm --with-gettext --with-gmp --with-iconv --with-jpeg-dir=/usr --with-openssl --with-pcre-regex=/usr --with-zlib --with-layout=GNU --enable-exif --enable-ftp --enable-magic-quotes --enable-sockets --with-kerberos --enable-ucd-snmp-hack --enable-shmop --enable-calendar --without-sqlite --with-libxml-dir=/usr --enable-xml --with-system-ldap --with-mhash --with-apxs2=/usr

3. `phpinfo.php` ファイルを削除します。これは有用な情報である可能性がありますが、セキュリティ上の理由から、インターネットにブロードキャストしないようにしてください。

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

### MySQL サーバーをセキュリティで保護するには

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

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

1. MySQL サーバーを起動し、`mysql_secure_installation` を実行できるようにします。

```
[ec2-user ~]$ sudo service mysqld start
Initializing MySQL database: Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
...

Starting mysqld: [ OK ]
```

2. `mysql_secure_installation` を実行します。

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

- a. プロンプトが表示されたら、アカウントのパスワードを入力します。
    - i. 現在のパスワードを入力します。デフォルトでは、アカウントにはパスワードが設定されていないので、Enter を押します。
    - ii. 「Y」と入力してパスワードを設定し、安全なパスワードを2回入力します。安全なパスワードの作成の詳細については、<http://www.pctools.com/guides/password/> を参照してください。このパスワードは必ず安全な場所に保管します。
  - b. 「Y」と入力して匿名ユーザーアカウントを削除します。
  - c. 「Y」と入力してログインを無効にします。
  - d. 「Y」と入力してテストデータベースを削除します。
  - e. 「Y」と入力して権限テーブルを再ロードし、変更を保存します。
3. (オプション) すぐに使用する予定がない場合は、MySQL サーバーを停止します。再び必要になったときに、サーバーを再起動できます。

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

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

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

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

## 関連トピック

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

- [WinSCP を使った、インスタンスへのファイルの転送 \(p. 342\)](#)
- [SCP による Linux/Unix から Linux/Unix インスタンスへのファイルの転送 \(p. 336\)](#)
- [チュートリアル: Amazon EC2 を使った WordPress ブログのホスティング \(p. 46\)](#)

Apache ウェブサーバーの詳細については、<http://httpd.apache.org/> を参照してください。MySQL データベースサーバーの詳細については、<http://www.mysql.com/> を参照してください。PHP プログラミング言語の詳細については、<http://php.net/> を参照してください。

# チュートリアル: Amazon EC2 を使った WordPress ブログのホスティング

---

## Abstract

チュートリアル – Amazon EC2 インスタンスでの WordPress ブログのインストール、構成、およびセキュリティの確保。

次の手順では、お客様の Amazon EC2 インスタンスで、WordPress ブログのインストール、構成を実行し、安全性を確立します。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

このチュートリアルは、WordPress ブログをホストするウェブサーバーを完全に制御する（これは従来のホスティングサービスでは一般的なことではありません）という点で、Amazon EC2 を使用するための優れた手引きになります。もちろんこれは、サーバーに対するソフトウェアパッケージの更新と、セキュリティパッチの維持は、お客様の責任であることも意味します。ウェブサーバー構成との直接的な対話操作を必要としない、より自動化された WordPress をインストールする場合、AWS CloudFormation サービスは、迅速に始められる WordPress テンプレートを提供します。詳細については、[AWS CloudFormation ユーザーガイド](#) の [Get Started](#) を参照してください。Windows インスタンスで WordPress ブログをホスティングする場合は、[Amazon Elastic Compute Cloud Microsoft Windows Guide](#) の [Deploying a WordPress Blog on Your Amazon EC2 Windows Instance](#) を参照してください。

### 前提条件

このチュートリアルでは、PHP と MySQL をサポートする機能的なウェブサーバーを使って、インスタンスを起動したことを前提としています。また、Amazon EC2 セキュリティグループが、HTTP と HTTPS トラフィックを許可していることも必要です。機能的なウェブサーバーがまだない場合、[チュートリアル: LAMP ウェブサーバーのインストール \(p. 39\)](#) を参照してウェブサーバーを作成してから、このチュートリアルに戻って、WordPress をインストールしてください。セキュリティグループへのルール追加の詳細については、[セキュリティグループへのルールの追加 \(p. 482\)](#) を参照してください。

## WordPress インストールパッケージをダウンロードして解凍するには

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

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
--2013-08-09 17:19:01-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 66.155.40.249, 66.155.40.250
Connecting to wordpress.org (wordpress.org)|66.155.40.249|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4028740 (3.8M) [application/x-gzip]
Saving to: latest.tar.gz

100%[=====>] 4,028,740  20.1MB/s  in 0.2s

2013-08-09 17:19:02 (20.1 MB/s) - latest.tar.gz saved [4028740/4028740]
```

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

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

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

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

1. MySQL サーバーを起動します。

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

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

**Important**

MySQL サーバーのセキュリティを確保していない場合、セキュリティ確保を行うことは非常に重要です。詳細については、[MySQL サーバーをセキュリティで保護するには \(p. 43\)](#) を参照してください。

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

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

```
mysql> CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY
'your_strong_password';
Query OK, 0 rows affected (0.00 sec)
```

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

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



#### Note

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

```
mysql> CREATE DATABASE `wordpress-db`;
Query OK, 1 row affected (0.01 sec)
```

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

```
mysql> GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"local
host";
Query OK, 0 rows affected (0.00 sec)
```

6. すべての変更を有効にするため、MySQL 権限をフラッシュします。

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

7. mysql クライアントを終了します。

```
mysql> exit
Bye
```

### wp-config.php ファイルの作成と編集を行うには

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

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

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

2. お好みのテキストエディタ ( nano、vim など ) を使って wp-config.php ファイルを編集し、インストール用の値を入力します。お好みのテキストエディタがない場合は、nano が非常に使いやすくおすすめです。

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

- a. DB\_NAME を定義する行を探して、database\_name\_here を [WordPress インストール用に MySQL ユーザーとデータベースを作成するには \(p. 47\) の Step 4 \(p. 48\)](#) で作成したデータベース名に変更します。

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

- b. DB\_USER を定義する行を探して、username\_here を [WordPress インストール用に MySQL ユーザーとデータベースを作成するには \(p. 47\) の Step 3 \(p. 47\)](#) で作成したデータベースユーザーに変更します。

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

- c. DB\_PASSWORD を定義する行を探して、password\_here を [WordPress インストール用に MySQL ユーザーとデータベースを作成するには \(p. 47\) の Step 3 \(p. 47\)](#) で作成した強力なパスワードに変更します。

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

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

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



#### Note

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

```
define('AUTH_KEY', ' #U$#[RXN8:b^-L 0(WU_+ c+WFkI~c]o)-bHw+)/Aj[wTwSiZ<Qb[mghEXcRh-');
define('SECURE_AUTH_KEY', 'Zsz._P=1/|y.Lq)Xj1kwS1y5NJ76E6EJ.AV0pCKZzB,*~*r ?6OP$eJT@;+(ndLg)');
define('LOGGED_IN_KEY', 'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2Jh^,8x>)Y|;(^[Iw]Pi+LG#A4R?7N`YB3');
define('NONCE_KEY', 'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|: ?0N]VJM%?;v2v]v+;+^9eXUahg@::cJ)');
define('AUTH_SALT', 'C$DpB4Hj[JK: ?{q1`SRVa: { :7yShy(9A@5wg+`JJVb1fk%_-Bx*M4(qc[Qg%JT!h)');
define('SECURE_AUTH_SALT', 'd!uRu#}+q#{f$Z?Z9uFPG. ${+S{n~1M&%@~gL>U>NV<zpD~@2-Es7Q10~bp28EKv}');
```

```
define('LOGGED_IN_SALT', ' ;j{00P*owZf)kVD+FVLn~>
.>./Y%Ug4#I^*LVd9QeZ^&XmK|e(76miC+&W&+^0P/');
define('NONCE_SALT',
'-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|_e1tS)8_B/, .6[=UK<J_y9?JWG');
```

- e. ファイルを保存し、テキストエディタを終了します。

### WordPress インストールを Apache ドキュメントルートに移動するには

インストールフォルダの解凍、MySQL データベースとユーザーの作成、WordPress 構成ファイルのカスタマイズが終了したため、インストールファイルをウェブサーバーのドキュメントルートに移動し、インストールスクリプトを実行して、インストールを終了する準備ができました。これらのファイルの場所は、ウェブサーバーのルートで WordPress ブログを使用できるようにするかどうか ( `my.public.dns.amazonaws.com` など)、またはサブディレクトリやフォルダに格納するか ( `my.public.dns.amazonaws.com/blog` など) によって異なります。



#### Important

ブログが使用できる場所を選択し、その場所に関する `mv` だけを実行します。以下の両方のコマンドセットを実行する場合、2 番目の `mv` コマンドでエラーメッセージが表示されます。これは、移動しようとしたファイルが存在していないことを意味します。

1. `my.public.dns.amazonaws.com` でブログを使用できるようにするには、`wordpress` フォルダ内のファイル (フォルダ自体ではありません) を Apache ドキュメントルート (Amazon Linux インスタンスの `/var/www/html`) に移動します。

```
[ec2-user wordpress]$ mv * /var/www/html/
```

2. または、その代わりにブログを `my.public.dns.amazonaws.com/blog` で使用できるようにするには、Apache ドキュメントルート内に `blog` という名前の新規フォルダを作成し、`wordpress` フォルダ内のファイル (フォルダ自体ではありません) を新しい `blog` フォルダに移動します。

```
[ec2-user wordpress]$ mkdir /var/www/html/blog
[ec2-user wordpress]$ mv * /var/www/html/blog
```



#### Important

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

### Apache ウェブサーバーのファイル許可を修正するには

WordPress で使用できる機能の中には、Apache ドキュメントルートへの書き込み権限が必要なものがあります (管理画面を使った、メディアのアップロードなど)。ウェブサーバーは `apache` ユーザーと

して実行されるため、そのユーザーを [LAMP ウェブサーバーチュートリアル \(p. 39\)](#) で作成した `www` グループに追加する必要があります。

1. `apache` ユーザーを `www` グループに追加します。

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

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

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

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

```
[ec2-user wordpress]$ sudo chmod 2775 /var/www
[ec2-user wordpress]$ find /var/www -type d -exec sudo chmod 2775 {} +
```

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

```
[ec2-user wordpress]$ find /var/www -type f -exec sudo chmod 0664 {} +
```

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

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

WordPress インストールスクリプトを実行するには

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

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

2. MySQL サーバー (`mysqld`) が実行中であることを確認します。

```
[ec2-user wordpress]$ sudo service mysqld status
mysqld (pid 4746) is running...
```

`mysqld` サービスが実行されていない場合は、起動します。

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

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

```
[ec2-user wordpress]$ sudo service httpd status
httpd (pid 502) is running...
```

httpd サービスが実行されていない場合は、起動します。

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

4. php パッケージと php-mysql パッケージがインストールされていることを確認します。出力の表示は少し異なっていることはありますが、[Installed Packages] セクションを探してください。

```
[ec2-user wordpress]$ yum list installed php php-mysql
Loaded plugins: priorities, security, update-motd, upgrade-helper
amzn-main | 2.1 kB 00:00
amzn-updates | 2.3 kB 00:00
Installed Packages
php.x86_64 5.3.27-1.0.amzn1 @amzn-
updates
php-mysql.x86_64 5.3.27-1.0.amzn1 @amzn-
updates
```



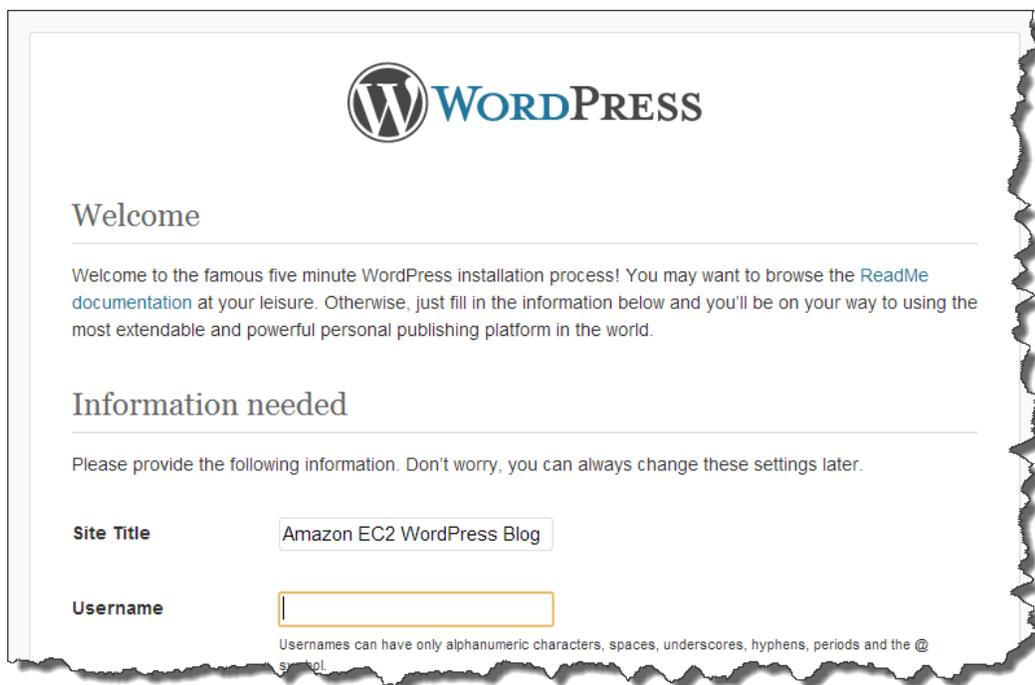
#### Note

これらのパッケージのどちらかがインストール済みとして表示されない場合、次のコマンドを使ってパッケージをインストールし、httpd サービスを再起動します。

```
[ec2-user wordpress]$ sudo yum install -y php php-mysql
[ec2-user wordpress]$ sudo service httpd restart
```

5. ウェブブラウザで WordPress ブログの URL を入力します ( インスタンスのパブリック DNS アドレス、または blog フォルダに続くアドレス )。WordPress インストール画面が表示されるはずで

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



6. 残りのインストール情報を WordPress インストールウィザードに入力します。

フィールド	値
[Site Title]	WordPress サイトの名前を入力します。
[Username]	WordPress 管理者の名前を入力します。セキュリティ上の理由から、このユーザーには一意の名前を選択します。これにより、デフォルトのユーザー名である <code>admin</code> よりも悪用される可能性が低くなるためです。
[Password]	強力なパスワードを入力し、確認のために再入力します。既存のパスワードを再利用しないでください。また、このパスワードは必ず安全な場所に保管してください。
[Your E-mail]	通知に使用する E メールアドレスを入力します。

7. [Install WordPress] をクリックして、インストールを完了します。

これで、WordPress ブログにログインして、エントリの投稿を開始することができます。

お客様の WordPress ブログの人气が高まり、コンピューティング能力の拡張が必要になった場合、より大きなインスタンスタイプへの移行を検討する必要があります。詳細は、[インスタンスのサイズ変更 \(p. 131\)](#) を参照してください。最初に割り当てたストレージ容量より大きな容量がブログで必要になった場合、インスタンスのストレージ容量を拡張することができます。詳細については、[ボリュームのストレージ領域を拡張する \(p. 593\)](#) を参照してください。MySQL データベースの拡張が必要になった場合、データベースを [Amazon RDS](#) に移動して、サービスの自動拡張機能を活用することをご検討ください。

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

# Amazon マシンイメージ ( AMI )

## Abstract

AMI はインスタンスのソフトウェア設定を提供します。

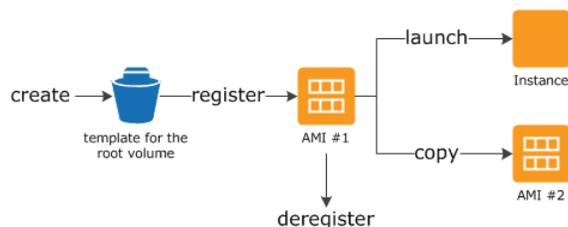
Amazon マシンイメージ ( AMI ) は、クラウドの仮想サーバーであるインスタンスの起動に必要な情報を提供します。インスタンスを起動するときに AMI を指定します。AMI からは、必要な数のインスタンスを起動できます。

AMI には次が含まれています。

- インスタンスのルートボリュームのテンプレート ( オペレーティングシステム、アプリケーションサーバー、アプリケーションなど )
- 起動許可 ( AMI を使用してインスタンスを起動する権限を特定の AWS アカウントに与える )
- インスタンスの起動時にインスタンスにアタッチするボリュームを指定するブロックデバイスマッピング

## AMI の使用

次の図は AMI のライフサイクルをまとめたものです。AMI を作成し、登録したら、それを使用して新しいインスタンスを起動できます ( AMI 所有者から起動許可を与えられた場合、AMI からインスタンスを起動することもできます )。AMI は同じリージョンにコピーすることも、異なるリージョンにコピーすることもできます。AMI からインスタンスを起動したら、AMI の登録を解除できます。



ご自分のインスタンスの基準に一致する AMI を検索できます。AWS が提供する AMI またはコミュニティが提供する AMI を検索できます。詳細については、[AMI タイプ \(p. 57\)](#) および [適切な AMI の検索 \(p. 60\)](#) を参照してください。

インスタンスに接続したら、そのインスタンスを他のサーバーとまったく同じように使用できます。インスタンスの起動、接続、使用に関する詳細については、[Amazon EC2 インスタンス \(p. 108\)](#) を参照してください。

## 独自の AMI の作成

パブリック AMI から起動するインスタンスをカスタマイズし、その設定をお客様が使用するためのカスタム AMI として保存できます。ご自分の AMI から起動するインスタンスでは、お客様が行ったすべてのカスタマイズを使用します。

AMI の作成プロセスは、インスタンスのルートストレージデバイスにより決まります。インスタンスのルートボリュームは、Amazon EBS ボリュームかインスタンスストアボリュームのいずれかになります。詳細については、[Amazon EC2 ルートデバイスボリューム \(p. 15\)](#) を参照してください。

Amazon EBS-Backed AMI の作成については、[Amazon EBS-Backed Linux AMI の作成 \(p. 77\)](#) を参照してください。Instance Store-Backed AMI の作成については、[Instance Store-Backed Linux AMI の作成 \(p. 81\)](#) を参照してください。

AMI には分類や管理のために任意のタグを付けられます。詳細については、[Amazon EC2 リソースにタグを付ける \(p. 662\)](#) を参照してください。

## AMI の購入、共有、販売

AMI を作成したら、自分だけがそれを使用できるようにプライベートとして保存したり、あるいは、AWS アカウントの指定リストと共有したりできます。コミュニティで利用できるように、カスタム AMI を公開することもできます。安全で信頼性が高く、便利な AMI を作成して、一般公開する手順はきわめて単純で、いくつかのシンプルなガイドラインにしたがうだけです。共有 AMI の作成および使用方法の詳細については、[共有 AMI \(p. 62\)](#) を参照してください。

Red Hat のような組織のサービス契約に付属する AMI など、サードパーティから AMI を購入できます。また、AMI を作成し、他の Amazon EC2 ユーザーに販売することもできます。AMI の購入と販売に関する詳細については、[有料 AMI \(p. 73\)](#) を参照してください。

## AMI の登録解除

AMI の利用が終わったら、その登録を解除できます。AMI の登録を解除すると、それを使用して新しいインスタンスを起動できなくなります。詳細については、[AMI の登録解除 \(p. 92\)](#) を参照してください。

## Amazon Linux

Amazon Linux AMI は、AWS がサポートおよび保守管理している Linux イメージの 1 つです。次に示すのは Amazon Linux の特徴の一部です。

- Amazon EC2 で実行されるアプリケーションのため、安定性があり、安全で高性能な実行環境。
- Amazon EC2 ユーザーには追加料金なしで提供。
- Linux 3.4、AWS ツールを含む Amazon EBS バックツの PV-GRUB AM。また、MySQL、PostgreSQL、Python、Ruby、Tomcat の複数のバージョンへのリポジトリアクセスが可能。
- 定期的な更新で最新のコンポーネントが追加される。更新は、インスタンスを実行するインストーラの yum リポジトリでも利用可能。

- Amazon EC2 API および AMI ツール、Python 用の Boto ライブラリ、Elastic Load Balancing ツールなど、AWS サービスとの統合を簡単にするパッケージが含まれる。

詳細については、[Amazon Linux \(p. 94\)](#) を参照してください。

## AMI タイプ

### Abstract

リージョン、オペレーティングシステム、アーキテクチャ、ルートボリュームタイプに基づいて AMI を選択します。

次の特性に基づき、使用する AMI を選択できます。

- リージョン ( [リージョンとアベイラビリティゾーン \(p. 8\)](#) を参照 )
- オペレーティングシステム
- アーキテクチャ ( 32 ビットまたは 64 ビット )
- [起動許可 \(p. 57\)](#)
- [ルートデバイスのストレージ \(p. 57\)](#)

## 起動許可

AMI の所有者は、起動許可を指定することで可用性を決定します。起動許可は次のように分類されます。

起動許可	説明
パブリック	所有者はすべての AWS アカウントに起動許可を与えます。
明示的	所有者は特定の AWS アカウントに起動許可を与えます。
暗示的	所有者には AMI の暗示的起動許可があります。

Amazon と Amazon EC2 コミュニティはさまざまなパブリック AMI を提供しています。詳細については、[共有 AMI \(p. 62\)](#) を参照してください。開発者は自分の AMI に料金を請求できます。詳細については、「[有料 AMI \(p. 73\)](#)」を参照してください。

## ルートデバイスのストレージ

すべての AMI が *Amazon EBS-Backed* と *Instance Store-Backed* のいずれかに分類されます。前者は、AMI から起動されるインスタンスのルートデバイスが、Amazon EBS スナップショットから作成される Amazon EBS ボリュームであるということです。後者は、AMI から起動されるインスタンスのルートデバイスが、Amazon S3 に格納されたテンプレートから作成されるインスタンスストアボリュームであるということです。詳細については、[Amazon EC2 ルートデバイスボリューム \(p. 15\)](#) を参照してください。

このセクションでは、2 種類の AMI の重要な相違点をまとめています。以下の表は、これらの相違点についてまとめたものです。

特徴	Amazon EBS-Backed	Amazon Instance Store-Backed
起動時間	通常 1 分以内	通常 5 分以内
サイズ制限	1 TiB	10 GiB
ルートデバイスボリューム	Amazon EBS ボリューム	インスタンスストアボリューム
データの永続性	Amazon EBS ボリュームのデータは、インスタンス終了後も保持されます。* インスタンス終了後、保持されないインスタンスストアボリュームをアタッチすることもできます。	インスタンスストアボリュームのデータは、そのインスタンスの存続中のみ保持されます。インスタンス終了後、保持される Amazon EBS ボリュームをアタッチすることもできます。
アップグレード	インスタンスの停止中に、インスタンスタイプ、カーネル、RAM ディスク、およびユーザーデータが変更可能	インスタンスの属性は、インスタンスを終了するまで固定。
料金	インスタンスの使用量、Amazon EBS ボリューム、また、Amazon EBS スナップショットとして保存した AMI に対して料金が発生します。	インスタンスの使用量や Amazon S3 に保存した AMI に対して料金が発生します。
AMI の作成/バンドル	単一のコマンドまたは呼び出しを使用	AMI ツールをインストールして使用する必要があります
停止状態	インスタンスが実行されていない停止状態にできます。ただし、ルートボリュームは Amazon EBS に保持されます。	インスタンスが実行中も終了後も、停止状態にすることはできません

\*デフォルトでは、Amazon EBS-Backed インスタンスのルートボリュームの `DeleteOnTermination` フラグは、`true` に設定されます。そのため、インスタンスの終了時にボリュームが削除されます。この設定を変更し、終了後もボリュームが保持する方法に関する詳細については、[永続的ルートデバイスボリュームへの変更 \(p. 18\)](#) を参照してください。

## AMI のルートデバイスタイプの判別

コンソールを使用して AMI のルートデバイスタイプを判別するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[AMI] をクリックし、AMI を選択します。
3. 次のように、[Details] タブで [Root Device Type] の値を確認します。
  - 値が `ebs` の場合、これは Amazon EBS-Backed AMI です。
  - 値が `instance store` の場合、これは Instance Store-Backed AMI です。

コマンドラインを使用して AMI のルートデバイスタイプを判別するには

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

- [describe-images](#) ( AWS CLI )
- [ec2-describe-images](#) ( Amazon EC2 CLI )
- [Get-EC2Image](#) ( AWS Tools for Windows PowerShell )

## サイズ制限

Amazon EC2 Instance Store-Backed AMI は、ルートデバイスで 10 GiB のストレージに制限され、Amazon EBS-Backed AMI は 1 TiB に制限されます。多くの Windows AMI は 10 GiB 制限に近づくため、Windows AMI は EBS ボリュームによりサポートされるケースがよくあります。



### Note

すべての Windows Server 2008、Windows Server 2008 R2、および Windows Server 2012 AMI はサイズが大きいため、デフォルトで Amazon EBS ボリュームによってサポートされます。

## 停止状態

Amazon EBS-Backed インスタンスは停止できますが、Amazon EC2 Instance store-Backed インスタンスは停止できません。停止すると、インスタンスの実行が停止します (ステータスが `running` から `stopping` を経て `stopped` に進む)。stopped インスタンスは Amazon EBS に存続するため、再起動できます。stopping (停止) は terminating (終了) と異なります。terminated インスタンスは再起動できません。Amazon EC2 Instance store-Backed AMI は停止できないため、実行中か終了のいずれかになります。インスタンスが停止している場合に何が行われ、何を実行できるかの詳細については、「[インスタンスの停止と起動 \(p. 346\)](#)」を参照してください。

## デフォルトのデータストレージと永続性

ルートデバイスにインスタンスストアボリュームを使用するインスタンスでは、自動的にインスタンスストアが利用できます (ルートボリュームにルートパーティションが含まれ、追加のデータを保存できます)。インスタンスが失敗するか終了すると、インスタンスストアボリューム上のすべてのデータが削除されます (ルートデバイス上のデータを除く)。1 つまたは複数の Amazon EBS ボリュームをアタッチすることで、永続的ストレージをインスタンスに追加できます。

Amazon EBS をルートデバイスに使用するインスタンスには自動的に、Amazon EBS ボリュームがアタッチされます。ボリュームは、他のボリュームと同様に、ボリュームのリストに表示されます。デフォルトでは、インスタンスは利用可能なインスタンスストアボリュームを一切使用しません。ブロックデバイスマッピングを使用して、インスタンスストレージまたは追加の Amazon EBS ボリュームを追加できます。詳細については、「[ブロックデバイスマッピング \(p. 643\)](#)」を参照してください。インスタンスを停止すると、インスタンスストアボリュームに何が発生するかについては、「[インスタンスの停止と起動 \(p. 346\)](#)」を参照してください。

## 起動時間

Amazon EBS-Backed AMI は、Amazon EC2 Instance store-Backed AMI よりも速く起動します。Amazon EC2 Instance store-Backed AMI を起動するときには、Amazon S3 からすべての部分を取得しないとインスタンスを利用できません。Amazon EBS-Backed AMI の場合、インスタンスの起動に必要な部分だけをスナップショットから取得するとインスタンスを利用できます。ただし、ルートデバイスに Amazon EBS ボリュームを使用するインスタンスのパフォーマンスは、残りの部分がスナップショットから取得され、ボリュームにロードされる少しの時間、遅くなります。インスタンスを停止し、再起動する場合は、状態が Amazon EBS ボリュームに保存されているため早く起動します。

## AMI の作成

Instance Store-Backed の Linux/Unix AMI を作成するには、インスタンス自体にインスタンスの AMI を作成する必要があります。ただし、それを支援する API アクションはありません。Instance Store-Backed の Windows AMI を作成する場合、AMI を作成する API アクションと、作成した AMI を登録する別の API アクションがあります。

AMI の作成は、EBS Backed の AMI の方がはるかに簡単です。CreateImage API アクションは、Linux/Unix と Windows の両方で AMI を作成します。この API アクションは、Amazon EBS-Backed AMI を作成し、登録します。AWS Management Console にも、実行中のインスタンスから AMI を作成できるボタンがあります。詳細については、「[Amazon EBS-Backed Linux AMI の作成 \(p. 77\)](#)」を参照してください。

## 課金方法

Instance Store-Backed の AMI の場合、AMI ストレージとインスタンスの使用量に対して課金されます。Amazon EBS-Backed の AMI の場合、AMI とインスタンスの使用料に加えて、ボリュームストレージおよび使用量に対して課金されます。

Amazon EC2 Instance Store-Backed の AMI の場合、AMI をカスタマイズしたり、新しい AMI を作成したりするたびに、各 AMI のすべての部分が Amazon S3 に保存されます。そのため、カスタマイズした各 AMI のストレージフットプリントは、AMI の完全なサイズになります。Amazon EBS-Backed の AMI の場合、AMI をカスタマイズしたり、新しい AMI を作成したりするたびに、変更のみが保存されます。そのため、最初の AMI の後にカスタマイズする後続の AMI のストレージフットプリントははるかに小さくなり、AMI ストレージ料金が少なくなります。

Amazon EBS-backed instance が停止した場合、インスタンスの使用については課金されませんが、ボリュームストレージについては引き続き課金されます。stopped ( 停止 ) 状態から running ( 実行中 ) 状態へのすべての移行に対して、1 時間内に複数回インスタンスを移行した場合でも、完全インスタンス時間が課金されます。例えば、インスタンスの時間当たりのインスタンス料金が 0.10 USD であるとします。そのインスタンスを停止せずに実行した場合、0.10 USD 課金されます。その時間中に、そのインスタンスを 2 回停止し、再起動した場合、その使用時間に対して 0.30 USD 課金されます ( 最初の 0.10 USD に再起動ごとの 2 x 0.10 USD をたす )。

## 適切な AMI の検索

### Abstract

要件に一致する AMI を検索します。

AMI を選択する前に、起動するインスタンスに関して、次の要件を検討します。

- リージョン
- オペレーティングシステム
- アーキテクチャ: 32 ビット ( i386 ) または 64 ビット ( x86\_64 )
- ルートデバイスタイプ: Amazon EBS またはインスタンスストア
- プロバイダ: Amazon Web Services、Oracle、IBM、Microsoft、またはコミュニティ

## Amazon EC2 コンソールを使用して AMI を見つける

コンソールを使用して適切な AMI を見つけるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、リージョンを選択します。お客様の居住地に関係なく、使用できるリージョンはどれでも選択できます。これは、お客様がインスタンスを起動するリージョンです。
3. ナビゲーションペインの [AMIs] をクリックします。
4. (オプション) [Filter] オプションを使用して、一覧表示された AMI を興味のある AMI に限定します。例えば、AWS が提供するすべての AMI を一覧表示するには、[Public images] を選択して、[Amazon images] を選択します。
5. (オプション) [Show/Hide Columns] アイコンをクリックして、ルートデバイスタイプなど、表示するイメージ属性を選択します。あるいは、一覧から AMI を選択し、[Details] タブにそのプロパティを表示できます。
6. AMI を選択する前に、その AMI が Instance Store-Backed と Amazon EBS-Backed のどちらであるかを確認し、その違いを認識しておくことが重要です。詳細については、「[ルートデバイスのストレージ \(p. 57\)](#)」を参照してください。
7. この AMI からインスタンスを起動するには、インスタンスを選択し、[Launch] をクリックします。詳細については、[インスタンスの起動 \(p. 324\)](#) を参照してください。まだインスタンスを起動する準備が整っていない場合は、後で使用するために AMI ID ( ami-xxxxxxx ) を書き留めておきます。

## コマンドラインを使用した AMI の検索

AMI および Amazon のパブリック AMI を検索するには、以下のいずれかのコマンドを使用できます。

- `describe-images` ( AWS CLI )

```
aws ec2 describe-images --owners self amazon
```

- `ec2-describe-images` ( Amazon EC2 CLI )

```
ec2-describe-images -o self -o amazon
```

表示される AMI の数を減らすには、フィルタを使用して、興味のある種類の AMI に限定して表示します。たとえば、次のフィルタを追加して、Amazon EBS-Backed Windows AMI のみを表示します。

- `describe-images` ( AWS CLI )

```
--filters "Name=platform,Values=Windows,Name=root-device-type,Values=ebs"
```

- `ec2-describe-images` ( Amazon EC2 CLI )

```
--filter "platform=windows" --filter "root-device-type=ebs"
```

ニーズに合った AMI を見つけたら、その ID を書き留めます。ID の形式は ami-xxxxxxx です。この AMI を使用して、インスタンスを起動できます。詳細については、『*AWS Command Line Interface ユーザーガイド*』の「[Launching an Instance Using the AWS CLI](#)」または『*Amazon Elastic Compute*

『Cloud コマンドラインリファレンス』の「[Launching an Instance Using the Amazon EC2 CLI](#)」を参照してください。

## 共有 AMI

### Abstract

パブリック AMI を検索または共有します。

共有 AMI は、開発者が作成し、他の開発者が利用できるようにした AMI です。Amazon EC2 を始める最も簡単な方法は、必要なコンポーネントが含まれている共有 AMI を使用して、カスタムコンテンツを追加することです。

共有 AMI は、ご自分の判断で使用してください。Amazon は、他の Amazon EC2 ユーザーとの間で共有される AMI の統合性や安全性を保証できません。そのため、共有 AMI を取り扱う際は、ご自分のデータセンターに外部のコードをデプロイすることを検討するのと同じように、十分な注意を払う必要があります。

信頼できる開発元の AMI を入手することをお勧めします。共有 AMI に関してご質問やお気付きになったことがあれば、[AWS フォーラム](#)をご利用ください。

Amazon のパブリックイメージにはエイリアスの所有者が設定されており、アカウントフィールドに `amazon` として表示されます。これを利用すれば、Amazon から AMI を簡単に見つけられます。他のユーザーは、AMI にエイリアスを設定できません。

### Topics

- [共有 AMI を見つける](#) (p. 62)
- [AMI を一般公開する](#) (p. 65)
- [特定の AWS アカウントと AMI を共有する](#) (p. 67)
- [ブックマークの使用](#) (p. 68)
- [共有 Linux AMI のガイドライン](#) (p. 69)

## 共有 AMI を見つける

### Abstract

共有 AMI を見つけます。

Amazon EC2 コンソールまたは Amazon EC2 CLI を使用して、共有 AMI を検索できます。

### コンソールを使用して共有 AMI を見つける

コンソールを使用して、共有しているプライベート AMI を見つけるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [AMIs] をクリックします。
3. 最初のフィルタで、[Private images] を選択します。お客様が共有を受けているすべての AMI が一覧表示されます。

コンソールを使用して、共有しているパブリック AMI を見つけるには

1. Amazon EC2 コンソールを開きます。

2. ナビゲーションペインの [AMIs] をクリックします。
3. 共有 AMI を見つけるには、[Filter] リストから [Public images] を選択します。
4. 興味のある種類の AMI のみを一覧表示するには、フィルタを使用します。例えば、[Amazon images] を選択すると、Amazon のパブリックイメージのみが表示されます。

## AWS CLI を使用して共有 AMI を見つける

コマンドラインツールを使用して共有パブリック AMI を見つけるには

AMI を一覧表示するには、`-describe-images` コマンドを使用します。次の例のように、興味のある種類の AMI に絞って一覧表示できます。

次のコマンドを使用すると、`--executable-users` オプションを使用するすべてのパブリック AMI が一覧表示されます。このリストには、お客様が所有するパブリック AMI が含まれます。

```
aws ec2 describe-images --executable-users all
```

次のコマンドを使用すると、お客様が明示的な起動許可を持つ AMI が一覧表示されます。このリストでは、お客様が所有するパブリック AMI が除外されています。

```
aws ec2 describe-images --executable-users self
```

次のコマンドを使用すると、Amazon が所有する AMI が一覧表示されます。Amazon のパブリックイメージにはエイリアスの所有者が設定されており、アカウントフィールドに `amazon` として表示されます。これを利用すれば、Amazon から AMI を簡単に見つけられます。他のユーザーは、AMI にエイリアスを設定できません。

```
aws ec2 describe-images --owners amazon
```

次のコマンドを使用すると、指定した AWS アカウントが所有する AMI が一覧表示されます。

```
aws ec2 describe-images --owners 123456789012
```

表示される AMI の数を減らすには、フィルタを使用して、興味のある種類の AMI に限定して表示します。たとえば、次のフィルタを使用すると、Windows ベースの AMI のみ表示されます。

```
--filters "Name=platform,Values=windows"
```

## Amazon EC2 CLI を使用して共有 AMI を見つける

[To find a shared public AMI using the command line tools]

AMI を一覧表示するには、`ec2-describe-images` コマンドを使用します。次の例のように、興味のある種類の AMI に絞って一覧表示できます。

次のコマンドを使用すると、`-x all` オプションを使用するすべてのパブリック AMI が一覧表示されます。このリストには、お客様が所有するパブリック AMI が含まれます。

```
ec2-describe-images -x all
```

次のコマンドを使用すると、お客様が明示的な起動許可を持つ AMI が一覧表示されます。このリストでは、お客様が所有するパブリック AMI が除外されています。

```
ec2-describe-images -x self
```

次のコマンドを使用すると、Amazon が所有する AMI が一覧表示されます。Amazon のパブリックイメージにはエイリアスの所有者が設定されており、アカウントフィールドに `amazon` として表示されます。これを利用すれば、Amazon から AMI を簡単に見つけられます。他のユーザーは、AMI にエイリアスを設定できません。

```
ec2-describe-images -o amazon
```

次のコマンドを使用すると、指定した AWS アカウントが所有する AMI が一覧表示されます。

```
ec2-describe-images -o <target_uid>
```

<target\_uid> は、検索対象の AMI を所有しているアカウント ID です。

表示される AMI の数を減らすには、フィルタを使用して、興味のある種類の AMI に限定して表示します。たとえば、次のフィルタを使用すると、Windows ベースの AMI のみ表示されます。

```
--filter "platform=windows"
```

## 共有 AMI を使用する

共有 AMI を使用する前に、次の手順を実行して、インスタンスへの好ましくないアクセスを許可する認証情報が第三者により事前にインストールされていないことと、機密データを第三者に送信する可能性があるリモートロギングが事前設定されていないことを確認します。システムセキュリティ改善についての詳細は、AMI で使用される Linux ディストリビューションの文書を確認してください。

インスタンスへのアクセスを誤って失わないように、SSH セッションを 2 つ開始して、見覚えのない認証情報を削除し、その後も SSH を使用してインスタンスにログインできることが確認されるまで、2 つ目のセッションを開いておくことをお勧めします。

1. 未許可のパブリック SSH キーを特定し、無効にします。ファイル内の唯一のキーは、AMI の起動に使用したキーである必要があります。次のコマンドを使用すると、`authorized_keys` ファイルが見つかります。

```
sudo find / -name "authorized_keys" -print -exec cat {} \;
```

2. ルートユーザーにはパスワードベースの認証を無効にします。`ssh_config` ファイルを開き、次のように `PermitRootLogin` 行を編集します。

```
PermitRootLogin without-password
```

あるいは、ルートとしてインスタンスにログインする機能を無効にできます。

```
PermitRootLogin No
```

次に、`sshd` サービスを再起動します。

3. インスタンスにログインできるユーザーアカウントが他にないか確認します。スーパーユーザー権限を持つアカウントが特に危険です。不明のアカウントがあれば、そのパスワードを削除するか、ロックします。
4. 開いていても使用していないポートと、着信接続をリスニングしている実行中のネットワークサービスをチェックします。
5. 事前設定されているリモートログインを防ぐには、既存の設定ファイルを削除し、rsyslog サービスを再起動してください。以下に例を示します。

```
sudo rm /etc/rsyslog.config
sudo service rsyslog restart
```

6. すべての cron ジョブが正当であることを確認します。

セキュリティ上のリスクとして考えられるパブリック AMI を発見した場合、AWS セキュリティチームにご連絡ください。詳細については、[AWS セキュリティセンター](#)を参照してください。

## AMI を一般公開する

### Abstract

ご自分の AMI をすべての Amazon EC2 ユーザーと共有します。

Amazon EC2 では、自分の AMI を他の AWS アカウントと共有できます。すべての AWS アカウントに AMI の起動を許可 (AMI を一般公開する) することも、特定の少数のアカウントだけに AMI の起動を許可することもできます。お客様の AMI が他の AWS アカウントによって起動されても、お客様に料金は請求されません。AMI を起動するアカウントのみに料金が請求されます。

AMI を共有する前に、[共有 Linux AMI のガイドライン \(p. 69\)](#) のセキュリティ考慮事項をお読みください。



### Note

AMI に製品コードがある場合、それを一般公開することはできません。AMI は、特定の AWS アカウントとのみ共有する必要があります。

## コンソールを使用してパブリック AMI を共有する

コンソールを使用してパブリック AMI を共有するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [AMIs] をクリックします。
3. リストで AMI を選択し、[Actions] リストから [Modify Image Permissions] を選択します。
4. [Public] ラジオ ボタンを選択し、[Save] をクリックします。

## AWS CLI を使用してパブリック AMI を共有する

各 AMI には、所有者以外でその AMI を使用してインスタンスを起動できる AWS アカウントを制御する `launchPermission` プロパティがあります。AMI の `launchPermission` プロパティを変更することで、AMI を一般公開できます。この場合、すべての AWS アカウントに起動許可を与えたり、あるいは指定した AWS アカウントとのみ AMI を共有したりできます。

AMI の起動許可を持つアカウントのリストにアカウント ID を追加したり、削除したりできます。AMI を一般公開するには、all グループを指定します。パブリック起動許可と明示的起動許可の両方を指定できます。

[To make an AMI public]

`-modify-image-attribute` コマンドを次のように使用して、指定した AMI の `launchPermission` リストに all グループを追加します。

```
aws ec2 modify-image-attribute --image-id ami-2bb65342 --launch-permission
"{\"Add\": [{\"Group\": \"all\"}]}"
```

AMI の起動許可を確認するには、次の `describe-image-attribute` コマンドを使用します。

```
aws ec2 describe-image-attribute --image-id ami-2bb65342 --attribute launchPer
mission
```

(オプション) AMI をプライベートに戻すには、その起動許可から all グループを削除します。AMI の所有者には常に起動許可が与えられるため、このコマンドの影響を受けないことにご注意ください。

```
aws ec2 modify-image-attribute --image-id ami-2bb65342 "{\"Re
move\": [{\"Group\": \"all\"}]}"
```

## Amazon EC2 CLI を使用してパブリック AMI を共有する

各 AMI には、所有者以外でその AMI を使用してインスタンスを起動できる AWS アカウントを制御する `launchPermission` プロパティがあります。AMI の `launchPermission` プロパティを変更することで、AMI を一般公開できます。この場合、すべての AWS アカウントに起動許可を与えたり、あるいは指定した AWS アカウントとのみ AMI を共有したりできます。

AMI の起動許可を持つアカウントのリストにアカウント ID を追加したり、削除したりできます。AMI を一般公開するには、all グループを指定します。パブリック起動許可と明示的起動許可の両方を指定できます。

[To make an AMI public]

`ec2-modify-image-attribute` コマンドを次のように使用して、指定した AMI の `launchPermission` リストに all グループを追加します。

```
ec2-modify-image-attribute ami-2bb65342 --launch-permission -a all
```

AMI の起動許可を検証するには、次のコマンドを使用します。

```
ec2-describe-image-attribute ami-2bb65342 -l
```

AMI をプライベートに戻すには、その起動許可から all グループを削除します。AMI の所有者には常に起動許可が与えられるため、このコマンドの影響を受けないことにご注意ください。

```
ec2-modify-image-attribute ami-2bb65342 -l -r all
```

## 特定の AWS アカウントと AMI を共有する

### Abstract

お客様の AMI を特定の Amazon EC2 ユーザーと共有します。

AMI を一般公開せず、特定の AWS アカウントとだけ共有することもできます。必要なものは AWS アカウント ID だけです。

## コンソールを使用して AMI を共有する

コンソールを使用して明示的な起動許可を与えるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [AMIs] をクリックします。
3. リストで AMI を選択し、[Actions] リストから [Modify Image Permissions] を選択します。
4. [AWS Account Number] フィールドに AMI を共有するユーザーの AWS アカウント番号を指定し、[Add Permission] をクリックします。

この AMI を複数のユーザーと共有するには、上記の手順を繰り返して、必要なすべてのユーザーを追加します。

5. スナップショットのボリューム作成アクセス許可を与えるには、[Add "create volume" permissions to the following associated snapshots when creating permissions] をオンにします。



### Note

AMI を共有するために、AMI が参照する Amazon EBS スナップショットを共有する必要はありません。共有する必要があるのは AMI 自体だけです。起動の際に、参照先の Amazon EBS スナップショットへのインスタンスアクセスが自動的に提供されます。

6. 完了したら [Save] をクリックします。

## AWS CLI を使用して AMI を共有する

AMI を共有するには、次の例のように `modify-image-attribute` コマンドを使用します。

明示的な起動許可を与えるには

次のコマンドを使用すると、指定した AWS アカウントに指定した AMI の起動許可が与えられます。

```
aws ec2 modify-image-attribute --image-id ami-2bb65342
"{\"Add\": [{\"UserId\": \"123456789012\"}]}"
```

[To remove launch permissions for an account]

次のコマンドを使用すると、指定した AWS アカウントから指定した AMI の起動許可が削除されます。

```
aws ec2 modify-image-attribute --image-id ami-2bb65342 "{\"Re
move\": [{\"UserId\": \"123456789012\"}]}"
```

[To remove all launch permissions]

次のコマンドを使用すると、指定した AWS アカウントからパブリック起動許可と明示的起動許可がすべて削除されます。AMI の所有者には常に起動許可が与えられるため、このコマンドの影響を受けないことにご注意ください。

```
aws ec2 reset-image-attribute --image-id ami-2bb65342 --attribute launchPermis  
sion
```

## Amazon EC2 CLI を使用して AMI を見つける

AMI を共有するには、次の例のように、`ec2-modify-image-attribute` コマンドを使用します。

明示的な起動許可を与えるには

次のコマンドを使用すると、指定した AWS アカウントに指定した AMI の起動許可が与えられます。

```
ec2-modify-image-attribute ami-2bb65342 -l -a 111122223333
```

[To remove launch permissions for an account]

次のコマンドを使用すると、指定した AWS アカウントから指定した AMI の起動許可が削除されます。

```
ec2-modify-image-attribute ami-2bb65342 -l -r 111122223333
```

[To remove all launch permissions]

次のコマンドを使用すると、指定した AWS アカウントからパブリック起動許可と明示的起動許可がすべて削除されます。AMI の所有者には常に起動許可が与えられるため、このコマンドの影響を受けないことにご注意ください。

```
ec2-reset-image-attribute ami-2bb65342 -l
```

## ブックマークの使用

### Abstract

他の Amazon EC2 ユーザーがお客様の共有 AMI にアクセスし、使用することを許可します。

パブリック AMI を作成した場合、あるいは AMI を別の AWS ユーザーと共有した場合は、ブックマークを作成できます。ブックマークを作成すると、ユーザーは自分のアカウントですばやく AMI にアクセスし、インスタンスを起動できます。これにより AMI リファレンスを簡単に共有できるため、時間をかけず、使用する AMI を見つけることができます。

AMI はパブリックであるか、ブックマークの送信先ユーザーと共有している必要があります。

AMI のブックマークを作成するには

1. 次の情報が含まれる URL を入力します。<region> には AMI のリージョン、<ami\_id> には AMI の ID を指定します。

```
https://console.aws.amazon.com/ec2/v2/home?region=<region>#LaunchInstanceWiz  
ard:ami=<ami_id>
```

例えば、この URL は、us-east-1 リージョンの ami-2bb65342 AMI からインスタンスを起動します。

```
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard:ami=ami-2bb65342
```

2. AMI を使用するユーザーにリンクを配信します。
3. ブックマークを使用するには、リンクをクリックするか、そのリンクをコピーしてブラウザに貼り付けます。起動ウィザードが開きます。AMI が既に選択されています。

## 共有 Linux AMI のガイドライン

### Abstract

共有 Linux AMI を作成するときは、これらのガイドラインにしたがいます。

これらのガイドラインにしたがった場合、ユーザーの使い勝手が向上し、セキュリティ上の問題にユーザーのインスタンスがさらされるリスクが減ります。

### Topics

- [起動時に AMI ツールを更新する \(p. 69\)](#)
- [ルートのパスワードベースログインの無効化 \(p. 70\)](#)
- [ルートアクセスを無効にする \(p. 70\)](#)
- [SSH ホストキーペアの削除 \(p. 70\)](#)
- [パブリックキー認証情報のインストール \(p. 71\)](#)
- [sshd DNS チェックの無効化 \( オプション \) \(p. 72\)](#)
- [自身の情報の公開 \(p. 72\)](#)
- [お客様の保護 \(p. 72\)](#)

AWS Marketplace 向けの AMI を作成する場合、[Building AMIs for AWS Marketplace](#) のガイドライン、ポリシー、ベストプラクティスを参照してください。

AMI の安全な共有についての詳細は、次の記事を参照してください。

- [パブリック AMI を安全に共有し使用する方法](#)
- [パブリック AMI の公開: セキュリティ強化とクリーンアップの要件](#)

## 起動時に AMI ツールを更新する

Instance Store-Backed の AMI の場合、起動時に、AMI で Amazon EC2 AMI 作成ツールをダウンロードして、アップグレードすることをお勧めします。これにより、共有 AMI に基づく新しい AMI に最新の AMI ツールが与えられます。

Amazon Linux の場合は、次を `/etc/rc.local` に追加します。

```
# Update the Amazon EC2 AMI tools
echo " + Updating EC2 AMI tools"
yum update -y aws-amitools-ec2
echo " + Updated EC2 AMI tools"
```

この方法では、イメージ上の他のソフトウェアが自動的に更新されます。



#### Note

自動更新するソフトウェアを決定するときは、更新によって発生する WAN トラフィックの量（お客様のユーザーはそれに対して課金されます）と、更新により AMI 上の他のソフトウェアが中断されるというリスクについて考慮してください。

他のディストリビューションの場合は、AMI ツールが最新版であることを確認してください。

## ルートのパスワードベースログインの無効化

パブリック AMI に固定のルートパスワードを使用することは、セキュリティの面で危険であり、すぐに知られるおそれがあります。初回ログイン後のパスワード変更をユーザーに任せていますが、変更されるまでの一瞬の間にパスワードが悪用される危険性があります。

この問題を解決するには、ルートユーザーのパスワードベースログインを無効にします。さらに、ルートアクセスは無効にすることをお勧めします。

ルートのパスワードベースログインを無効にするには

1. テキストエディタで `/etc/ssh/sshd_config` ファイルを開き、次の行を見つけ出します:

```
#PermitRootLogin yes
```

2. 行を次のように変更します:

```
PermitRootLogin without-password
```

この構成ファイルの場所は、ディストリビューションに応じて、または OpenSSH を実行していない場合は、異なることがあります。このような場合は、関連資料を参照してください。

## ルートアクセスを無効にする

これは、共有 AMI を操作する場合に安全な環境を確保するベストプラクティスとして知られています。安全な環境に関連する要素の 1 つとして、ルートパスワードを空のままに設定しないことが挙げられます。これを行うには、実行中のインスタンスにログインし、

```
[ec2-user ~]$ sudo passwd -l root
```

コマンドを発行して、ルートアクセスを無効にします。



#### Note

この操作が `sudo` の使用に影響を及ぼすことはありません。

## SSH ホストキーペアの削除

パブリック AMI から派生した AMI を共有する場合は、`/etc/ssh` にある既存の SSH ホストキーペアを削除します。これにより、他のユーザーがお客様の AMI を使用してインスタンスを起動したときに、SSH は、新しい固有の SSH キーペアを生成するように強制されるため、セキュリティが強化され、「中間者」攻撃の可能性を減らします。

次のリストに、削除する SSH ファイルを示します。

- ssh\_host\_dsa\_key
- ssh\_host\_dsa\_key.pub
- ssh\_host\_key
- ssh\_host\_key.pub
- ssh\_host\_rsa\_key
- ssh\_host\_rsa\_key.pub

次のコマンドを使用して、これらのファイルをすべて確実に削除できます。

```
[ec2-user ~]$ sudo shred -u /etc/ssh/*_key /etc/ssh/*_key.pub
```



#### Important

パブリック AMI から既存の SSH ホストキーペアを削除することを忘れた場合、ルーチン監査プロセスから、AMI のインスタンスを実行するすべての顧客に向けて、セキュリティ上のリスクがある可能性について通知されます。短い猶予期間の後に、AMI にプライベートのマークが付けられます。

## パブリックキー認証情報のインストール

パスワードを使用したログインを防ぐように AMI を構成したら、ユーザーが別のメカニズムを使用してログインできるようにしておく必要があります。

ユーザーは、Amazon EC2 を使用すると、インスタンスの起動時にパブリックプライベートキーペア名を指定できます。RunInstances API 呼び出し (またはコマンドライン API ツール) で有効なキーペア名を指定すると、パブリックキー (CreateKeyPair または ImportKeyPair の呼び出し後に Amazon EC2 がサーバー上に保持するキーペアの一部) を、インスタンスメタデータに対する HTTP Query を介してインスタンスで使用できるようになります。

SSH を使用してログインするには、AMI が起動時にキー値を取得し、それを /root/.ssh/authorized\_keys (または AMI 上のその他のユーザーアカウントの同等項目) に付加する必要があります。ユーザーはキーペアを使用して AMI のインスタンスを起動し、ルートパスワードを入力せずにログインできます。

Amazon Linux や Ubuntu を初めとする多くのディストリビューションでは、cloud-init パッケージを使用して、設定されたユーザーのパブリックキー認証情報を挿入します。cloud-init をサポートしていないディストリビューションの場合は、システムスタートアップスクリプト (たとえば、/etc/rc.local) に次のコードを追加して、起動時に root ユーザーに対して指定したパブリックキーを取り込みます。

```
if [ ! -d /root/.ssh ] ; then
    mkdir -p /root/.ssh
    chmod 700 /root/.ssh
fi
# Fetch public key using HTTP
curl http://169.254.169.254/latest/public-keys/0/openssh-key > /tmp/my-key
if [ $? -eq 0 ] ; then
    cat /tmp/my-key >> /root/.ssh/authorized_keys
    chmod 700 /root/.ssh/authorized_keys
    rm /tmp/my-key
fi
```

この設定は、あらゆるユーザーアカウントに適用できます。root に限定する必要はありません。



#### Note

このAMIに基づいたインスタンスを再バンドルすると、起動時に使用されたキーが組み込まれます。キーへの組み込みを阻止するには、`authorized_keys` ファイルの内容を空にする ( ファイルを削除する ) が、またはこのファイルを再バンドルから除外します。

## sshd DNS チェックの無効化 ( オプション )

sshd DNS チェックを無効にすると、sshd セキュリティが若干低下します。ただし、DNS の解決策が失敗した場合は、SSH ログインが引き続き機能します。sshd チェックを無効にしなかった場合、DNS の解決策が失敗すると、すべてのログインが阻止されます。

sshd DNS チェックを無効にするには

1. テキストエディタで `/etc/ssh/sshd_config` ファイルを開き、次の行を見つけ出します。

```
#UseDNS yes
```

2. 行を次のように変更します:

```
UseDNS no
```



#### Note

この構成ファイルの場所は、ディストリビューションに応じて、または OpenSSH を実行していない場合は、異なることがあります。このような場合は、関連資料を参照してください。

## 自身の情報の公開

現在のところ、AMI はそれぞれアカウント ID で表されるため、共有 AMI を提供したのが誰かを簡単に特定する方法はありません。

お客様の AMI の説明と AMI ID を [Amazon EC2 forum](#) に投稿することをお勧めします。これにより、新しい共有 AMI の使用に関心があるユーザーに便利な中心となる場所が提供されます。AMI を [Amazon マシンイメージ \( AMIs \)](#) ページに投稿することもできます。

## お客様の保護

前のセクションでは、共有 AMI を起動するユーザーに対して、共有 AMI をセキュアで安全にし、また、便利にする方法について説明しました。このセクションでは、お客様の AMI を利用するユーザーから、お客様ご自身を保護するためのガイドラインを説明します。

共有する AMI に、機密性のあるデータやソフトウェアは保管しないことをお勧めします。共有 AMI を起動するユーザーは、それを再バンドルしたり、自分のものとして登録したりできる可能性があります。以下のガイドラインに従って、見落としやすいセキュリティ上のリスクを回避してください。

- バンドルの前に必ずシェル履歴を削除してください。同じ AMI で複数のバンドルのアップロードを試行すると、シェル履歴にシークレットアクセスキーが含まれます。次の例は、インスタンス内からのバンドルの前に実行される最後のコマンドとなる必要があります。

```
[ec2-user ~]$ shred -u ~/.*history
```

AWS では、`ec2-bundle-vol` に `--exclude` *directory* オプションを使用して、バンドル操作に含めたくない機密情報が入っているディレクトリおよびサブディレクトリをスキップすることをお勧めしています。詳細については、『[Amazon Elastic Compute Cloud コマンドラインリファレンス](#)』の「`ec2-bundle-vol`」を参照してください。

- 実行中のインスタンスをバンドルするには、プライベートキーと X.509 証明書が必要です。これらの認証情報およびその他の認証情報を、バンドルされていない場所（インスタンスストアなど）に書き込みます。
- イメージをバンドルするときには、SSH 承認キーを除外します。Amazon パブリック AMI には、SSH 承認キーファイルとともに、インスタンスの起動に使用されるパブリックキーが保管されます。



#### Note

残念ながら、ガイドラインのこのリストに記載されている内容をすべて実践することはできません。共有 AMI を注意深く作成し、機密データが漏洩される可能性について十分考慮してください。

## 有料 AMI

### Abstract

有料 AMI と呼ばれるカスタム AMI を購入または販売します。

Amazon Kernel Image は、開発者から購入できる *有料 AMI* です。

Amazon EC2 は Amazon DevPay および AWS Marketplace と統合されており、開発者は自身が開発した AMI を他の Amazon EC2 ユーザーに有償で提供したり、インスタンスにサポートを提供したりできます。Amazon DevPay の詳細については、[Amazon DevPay](#) サイトを参照してください。

AWS Marketplace は、EC2 インスタンスの起動に使用できる AMI など、AWS で実行されるソフトウェアを購入できるオンラインストアです。AWS Marketplace AMI は、開発者ツールなど、カテゴリ別に整理されており、要件にあった製品を見つけることができます。AWS Marketplace についての詳細は、[AWS Marketplace](#) サイトを参照してください。

有料 AMI からのインスタンスの起動は、他の AMI からのインスタンスの起動と同じです。追加パラメータは必要ありません。インスタンスは、AMI の所有者が設定した料金と、Amazon EC2 でスモールインスタンスタイプを実行する場合の 1 時間あたりの料金など、関連ウェブサービスの標準使用料に基づいて課金されます。有料 AMI の所有者は、特定のインスタンスがその有料 AMI から起動されたかどうかを確認できます。



### Important

Amazon DevPay の有料 AMI はすべて Instance Store-Backed です。AWS Marketplace は Amazon EBS-Backed AMI をサポートします。

### Topics

- [ご自分の AMI を販売する \(p. 74\)](#)
- [有料 AMI を見つける \(p. 74\)](#)
- [有料 AMI の購入 \(p. 75\)](#)
- [インスタンスの製品コードを取得する \(p. 76\)](#)
- [有料サポートの利用 \(p. 76\)](#)
- [有料およびサポート対象の AMI の請求書 \(p. 77\)](#)
- [AWS Marketplace 受信登録を管理する \(p. 77\)](#)

## ご自分の AMI を販売する

AWS Marketplace と Amazon DevPay のいずれかを利用して、ご自分の AMI を販売できます。AWS Marketplace と Amazon DevPay のどちらでも、AWS で実行されるソフトウェアをお客様が購入できるようにサポートしています。ただし AWS Marketplace は、お客様に適した AMI を見つけやすくなっているため、より購入しやすくなっています。また、AWS Marketplace は、Amazon EBS-Backed AMI、リザーブドインスタンス、スポットインスタンスなど、Amazon DevPay がサポートしていない AWS 機能をサポートします。

AWS Marketplace でご自分の AMI を販売する詳細については、[Selling on AWS Marketplace](#) を参照してください。

Amazon DevPay で自分の AMI を販売する詳細については、[Using DevPay with Your Amazon EC2 AMI](#) を参照してください。

## 有料 AMI を見つける

購入できる AMI を検索する方法はいくつかあります。例えば、[AWS Marketplace](#)、Amazon EC2 コンソール、Amazon EC2 CLI を使用できます。あるいは、開発者が有料 AMI に関する情報をお客様にお知らせすることがあります。

### コンソールを使用して有料 AMI を見つける

コンソールを使用して有料 AMI を見つけるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [AMIs] をクリックします。
3. 最初の [Filter] リストから [Public images] を選択して、2 番目の [Filter] から [Marketplace images] を選択し、3 番目の [Filter] リストからオペレーティングシステムを選択します。

### AWS Marketplace を使用して有料 AMI を見つける

AWS Marketplace を使用して有料 AMI を見つけるには

1. [AWS Marketplace](#) を開きます。
2. 検索ボックスにオペレーティングシステムの名前を入力して、[Go] をクリックします。
3. 検索結果をさらに絞るには、カテゴリまたはフィルタを利用します。
4. 各製品には AMI か Software as a Service のいずれかの製品タイプのラベルが付いています。

### AWS CLI を使用して有料 Windows AMI を見つける

[To find a paid Windows AMI using the AWS CLI]

また次のように、[describe-images](#) コマンドを使用して、有料 Windows AMI を見つけることもできます。

```
ec2-describe-images --owners aws-marketplace
```

このコマンドは、有料 AMI の製品コードなど、各 AMI を説明するさまざまな詳細を返します。[describe-images](#) からの出力には、次のような製品コードのエントリがあります：

```
"ProductCodes": [  
  {  
    "ProductCodeId": "product_code",  
    "ProductCodeType": "marketplace"  
  }  
],
```

## Amazon EC2 CLI を使用して有料 Windows AMI を見つける

[To find a paid Windows AMI using the Amazon EC2 CLI]

また次のように、[ec2-describe-images](#) コマンドを使用して、有料 Windows AMI を見つけることもできます。

```
ec2-describe-images -o aws-marketplace
```

このコマンドは、有料 AMI の製品コードなど、各 AMI を説明するさまざまな詳細を返します。ec2-describe-images からの次の出力例に製品コードが含まれています。

IMAGE	ami-a5bf59cc	image_source	123456789012	available public
<i>product_code</i>	x86_64	machine		instance-store

## 有料 AMI の購入

AMI を使用してインスタンスを起動するには、有料 AMI にサインアップする ( 購入する ) 必要があります。

通常、有料 AMI の販売者は、価格や購入サイトへのリンクなど、AMI に関する情報を提供します。リンクをクリックすると、最初に AWS へのログインが求められます。ログイン後、AMI を購入できます。



### Important

Amazon DevPay から購入した有料 AMI を使用する場合、リザーブドインスタンスの割引は得られません。つまり、リザーブドインスタンスを購入した場合、有料 AMI を起動するとき、インスタンスに関連付けられている低価格が適用されません。お客様は、有料 AMI の販売者が指定した価格を常に支払います。

## コンソールを使用して有料 AMI を購入する

Amazon EC2 起動ウィザードを使用して有料 AMI を購入できます。詳細については、[AWS Marketplace インスタンスの起動 \(p. 332\)](#) を参照してください。

## AWS Marketplace を使用して製品の受信登録をする

AWS Marketplace を利用するには、AWS アカウントが必要です。AWS Marketplace 製品からインスタンスを起動するには、Amazon EC2 サービスの利用にサインアップして、インスタンスの起動元から製品の受信登録をする必要があります。AWS Marketplace の製品を受信登録するには、2 つの方法があります。

- AWS Marketplace ウェブサイト: 1-Click デプロイメント機能で、事前に設定したソフトウェアをすばやく起動できます。

- Amazon EC2 起動ウィザード: AMI を検索し、ウィザードからインスタンスを直接起動できます。詳細については、[AWS Marketplace インスタンスの起動 \(p. 332\)](#) を参照してください。

## 開発者から有料 AMI を購入する

有料 AMI の開発者は、AWS Marketplace のリストにない有料 AMI をユーザーが購入できるように手配できます。開発者は、Amazon から製品を購入できるように、お客様にリンクを送ります。お客様がご自分の Amazon.com 認証情報でサインインして、Amazon.com アカウントに保存しているクレジットカードを選択すると、AMI の購入時に使用できます。

## インスタンスの製品コードを取得する

インスタンスのメタデータを利用すれば、ご自分のインスタンスの製品コードが Amazon DevPay と AWS Marketplace のどちらであるのかが判断できます。メタデータの取得については、[インスタンスメタデータとユーザーデータ \(p. 289\)](#) を参照してください。

製品コードを取得するには、次のクエリを使用します。

```
GET http://169.254.169.254/2007-03-01/meta-data/product-codes
```

インスタンスに製品コードが含まれる場合、Amazon EC2 はそれを返します。以下に例を示します。

```
774F4FF8
```

## 有料サポートの利用

Amazon EC2 は、開発者がソフトウェア（またはそれに由来する AMI）のサポートを提供できるように手配します。開発者は、お客様がサインアップして使用できるサポート製品を提供することができます。サポート製品にサインアップすると、開発者はお客様に製品コードを渡します。お客様はそのコードをご自分の AMI に関連付ける必要があります。これにより、開発者は、お客様のインスタンスがサポート対象であることを確認できます。また、お客様が製品からインスタンスを実行すると、開発者が定めた製品の利用規約にしたがい、お客様に課金されます。



### Important

リザーブドインスタンスとともにサポート製品を使用することはできません。お客様は常に、サポート製品の販売者が指定した価格を支払います。

製品コードと自分の AMI を関連付けるには、次のコマンドの 1 つを使用します。*ami\_id* は AMI の ID で、*product\_code* は製品コードです。

- `modify-image-attribute` (AWS CLI)

```
aws ec2 modify-image-attribute --image-id ami_id --product-codes "product_code"
```

- `ec2-modify-image-attribute` (Amazon EC2 CLI)

```
ec2-modify-image-attribute ami_id --product-code product_code
```

一度設定した製品コード属性を変更したり、削除したりすることはできません。

## 有料およびサポート対象の AMI の請求書

有料またはサポートされた AMI の使用料金がお客様のクレジットカードに請求され、その金額を記載した E メールが毎月末に届きます。これは通常の Amazon EC2 使用料金とは別に請求されます。詳細については、[Paying For AWS Marketplace Products](#) を参照してください。

## AWS Marketplace 受信登録を管理する

AWS Marketplace ウェブサイトでは、ご自分の受信登録の詳細を確認したり、ベンダーの取扱説明を表示したり、ご自分の受信登録を管理したりできます。

ご自分の受信登録の詳細を確認するには

1. [AWS Marketplace](#) にログインします。
2. [Your Account] をクリックします。
3. [Manage Your Software Subscriptions] をクリックします。
4. 現在の受信登録がすべて表示されます。実行中のインスタンスに接続するためのユーザー名など、製品の使用に関する特定の取扱説明を表示するには、[Usage Instructions] をクリックします。

AWS Marketplace 受信登録をキャンセルするには

1. 受信登録から実行しているインスタンスがあれば、それをすべて終了します。
  - a. Amazon EC2 コンソールを開きます。
  - b. ナビゲーションペインの [Instances] をクリックします。
  - c. インスタンスを選択し、[Actions] メニューから [Terminate] を選択します。メッセージが表示されたら、[Yes, Terminate] をクリックします。
2. [AWS Marketplace](#) にログインして、[Your Account] をクリックし、[Manage Your Software Subscriptions] をクリックします。
3. [Cancel subscription] をクリックします。キャンセルを確認するメッセージが表示されます。



### Note

受信登録をキャンセルすると、その AMI からインスタンスを起動できなくなります。その AMI を再度使用するには、AWS Marketplace ウェブサイトか、Amazon EC2 コンソールの起動ウィザードで、再度キャンセルした AMI を受信登録する必要があります。

## Amazon EBS-Backed Linux AMI の作成

### Abstract

カスタマイズした Amazon EBS-Backed Linux インスタンスから、独自の AMI を作成します。

Amazon EBS-Backed Linux AMI を作成するには、既存の Amazon EBS-Backed Linux AMI から起動したインスタンスから始めます。ニーズに合わせてインスタンスをカスタマイズしたら、新しい AMI を作成し、登録します。新しい AMI を使用して、カスタマイズした新しいインスタンスを起動できます。

Amazon EBS-Backed Windows AMI を作成する場合、「[Amazon Elastic Compute Cloud Microsoft Windows Guide](#)」の「[Creating an Amazon EBS-Backed Windows AMI](#)」を参照してください。

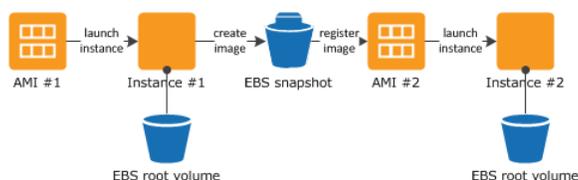
AMI の作成プロセスは、Instance Store-Backed AMI の場合とは異なります。Amazon EBS-Backed インスタンスと Instance Store-Backed インスタンスの違いの詳細と、インスタンスのルートデバイスタイプを判別する方法については、「[ルートデバイスのストレージ \(p. 57\)](#)」を参照してください。Instance Store-Backed Linux AMI を作成する必要がある場合は、[Instance Store-Backed Linux AMI の作成 \(p. 81\)](#) を参照してください。

#### Topics

- [Amazon EBS-Backed AMI の作成プロセスの概要 \(p. 78\)](#)
- [インスタンスから AMI を作成する \(p. 79\)](#)
- [スナップショットから AMI を作成する \(p. 80\)](#)

## Amazon EBS-Backed AMI の作成プロセスの概要

次の図は、Amazon EBS-Backed AMI の作成プロセスについてまとめたものです。



最初に、作成する AMI に似ている AMI からインスタンスを起動します。インスタンスに接続し、それをカスタマイズできます。インスタンスのカスタマイズが終わったら、データの整合性を維持するために、AMI を作成する前にいったんインスタンスを停止することをお勧めします。その後、イメージを作成できます。作成した Amazon EBS-Backed AMI は自動的に登録されます。

インスタンス上のすべてを停止し、作成プロセス中に一貫した状態が維持されるように、Amazon EC2 は AMI を作成する前にインスタンスの電源を落とします。インスタンスが一貫した状態にあり、適切に AMI を作成できる場合、インスタンスの電源を落として再起動しないように、Amazon EC2 に指定できます。xfs などの一部のファイルシステムでは、アクティビティのフリーズおよびフリーズ解除が可能のため、インスタンスを再起動しなくてもイメージを安全に作成することが可能です。

AMI 作成プロセスの間、Amazon EC2 はインスタンスのルートボリュームとインスタンスにアタッチされているその他の Amazon EBS ボリュームのスナップショットを作成します。ボリュームのサイズによっては、AMI 作成プロセス全体の完了に数分かかる場合があります ( 最長で 24 時間かかることもあります )。このプロセスをスピードアップするには、AMI を作成する直前にボリュームのスナップショットを作成することをお勧めします。詳細については、[Amazon EBS スナップショットの作成 \(p. 606\)](#) を参照してください。

プロセスが完了すると、新しい AMI と、インスタンスのルートボリュームから作成されたスナップショットが与えられます。ユーザーが新しい AMI を使用してインスタンスを起動すると、Amazon はスナップショットを使用して、そのルートボリュームのために新しい Amazon EBS ボリュームを作成します。AMI とスナップショットはどちらも削除されるまで、ユーザーのアカウントに料金を発生させます。詳細については、[AMI の登録解除 \(p. 92\)](#) を参照してください。

ルートデバイスボリュームに加えて、インスタンスストアボリュームまたは Amazon EBS ボリュームをインスタンスに追加した場合、新しい AMI のブロックデバイスマッピングにこれらのボリュームの情報が含まれ、新しい AMI から起動するインスタンスのブロックデバイスマッピングに自動的にこれらのボリュームの情報が含まれます。新しいインスタンスのブロックデバイスマッピングに指定されているインスタンスストアボリュームは新しく、AMI の作成に使用したインスタンスのインスタンスストアボリュームからのデータは含まれていません。Amazon EBS ボリュームのデータは永続的です。詳細については、[ブロックデバイスマッピング \(p. 643\)](#) を参照してください。

## インスタンスから AMI を作成する

AWS Management Console またはコマンドラインを利用して AMI を作成できます。

コンソールを使用してインスタンスから AMI を作成するには

1. Amazon EC2 コンソールを開きます。
2. ルートデバイスに Amazon EBS ボリュームを使用する実行中のインスタンスがない場合、インスタンスを起動する必要があります。手順については、「[インスタンスの起動 \(p. 325\)](#)」を参照してください。
3. ( オプション ) インスタンスに接続し、それをカスタマイズします。例えば、ソフトウェアとアプリケーションをインストールしたり、データをコピーしたり、追加の Amazon EBS ボリュームをアタッチしたりできます。インスタンスへの接続の詳細については、[インスタンスへの接続 \(p. 334\)](#)を参照してください。
4. ( オプション ) インスタンスにアタッチされているすべてのボリュームのスナップショットを作成します。これにより、AMI の作成時間が短縮される場合があります。スナップショット作成についての詳細は、[Amazon EBS スナップショットの作成 \(p. 606\)](#)を参照してください。
5. ナビゲーションペインの [Instances] をクリックし、インスタンスを選択します。[Actions] をクリックし、[Create Image] をクリックします。



### Tip

このオプションが無効になっている場合、そのインスタンスは Amazon EBS-Backed インスタンスではありません。

6. [Create Volume] ダイアログボックスで、以下を指定し、[Create Image] をクリックします。
  - a. イメージの一意な名前。
  - b. ( オプション ) イメージの説明 ( 最大 255 文字 ) 。
  - c. デフォルトでは、Amazon EC2 はインスタンスをシャットダウンし、アタッチされているボリュームのスナップショットを作成して、AMI を作成して登録し、インスタンスを再起動します。インスタンスをシャットダウンしない場合、[No reboot] を選択します。



### Warning

[No reboot] を選択した場合、Amazon では作成されたイメージのファイルシステムの整合性を保証できません。

- d. ( オプション ) ルートボリューム、Amazon EBS ボリューム、インスタンスストアボリュームを次のように変更できます。
  - ルートボリュームのサイズを変更するには、[Type] 列で [Root] ボリュームを見つけ、[Size] フィールドに入力します。
  - インスタンスの起動に使用された AMI のブロックデバイスマッピングで指定された Amazon EBS ボリュームを非表示にするには、リストから EBS ボリュームを見つけ、[Delete] をクリックします。
  - Amazon EBS ボリュームを追加するには、[Add New Volume] をクリックし、[Type] リストから [EBS] を選択して、フィールドに入力します。その後新しい AMI からインスタンスを起動すると、追加されたボリュームは自動的にインスタンスにアタッチされます。空のボリュームはフォーマットしてマウントする必要があります。スナップショットベースのボリュームはマウントする必要があります。

- インスタンスの起動に使用された AMI のブロックデバイスマッピングで指定されたインスタンスストアボリュームを非表示にするには、リストからボリュームを見つけ、[Delete] をクリックします。
  - インスタンスストアボリュームを追加するには、[Add New Volume] をクリックし、[Type] リストから [Instance Store] を選択して、[Device] リストからデバイス名を選択します。その後新しい AMI からインスタンスを起動すると、追加されたボリュームは自動的に初期化されてマウントされます。これらのボリュームには、AMI の作成に使用された実行中のインスタンスのインスタンスストアボリュームのデータは含まれません。
7. ナビゲーションペインの [AMI] をクリックして、AMI のステータスを表示します。新しい AMI が作成される間、そのステータスは `pending` です。通常、このプロセスが完了するには数分かかります。完了後、AMI のステータスは `available` になります。
  8. ( オプション ) ナビゲーションペインの [Snapshots] をクリックして、新しい AMI に作成されたスナップショットを表示します。ユーザーがこの AMI からインスタンスを起動すると、Amazon はこのスナップショットを使用して、ルートデバイスボリュームを作成します。

コマンドラインを使用してインスタンスから AMI を作成するには

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

- `create-image` ( AWS CLI )
- `ec2-create-image` ( Amazon EC2 CLI )
- `New-EC2Image` ( AWS Tools for Windows PowerShell )

## スナップショットから AMI を作成する

インスタンスのルートデバイスボリュームのスナップショットがある場合、AWS Management Console またはコマンドラインを使用して、そのスナップショットから AMI を作成できます。

コンソールを使用してスナップショットから AMI を作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic Block Store] の [Snapshots] をクリックします。
3. スナップショットを選択し、[Actions] リストから [Create Image] を選択します。
4. [Create Image from EBS Snapshot] ダイアログボックスで、次の操作を行います。
  - a. [Architecture] リスト ( 32 ビットの場合は `i386`、64 ビットの場合は `x86_64` ) からアーキテクチャを選択します。
  - b. [Kernel ID] リストから AKI を選択します。デフォルトの AKI を選択するか、AKI を選択しない場合、この AMI を使用してインスタンスの起動するたびに AKI を指定するように要求されます。また、デフォルトの AKI にインスタンスとの互換性がない場合、インスタンスのヘルスチェックが失敗する可能性があります。
  - c. [Create] をクリックします。

コマンドラインを使用してスナップショットから AMI を作成するには

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

- [register-image](#) ( AWS CLI )
- [ec2-register](#) ( Amazon EC2 CLI )
- [Register-EC2Image](#) ( AWS Tools for Windows PowerShell )

## Instance Store-Backed Linux AMI の作成

### Abstract

カスタマイズした Instance Store-Backed Linux インスタンスから独自の AMI を作成します。

Instance Store-Backed Linux AMI を作成するには、既存の Instance Store-Backed Linux AMI から起動したインスタンスから始めます。ニーズに合わせてインスタンスをカスタマイズしたら、ボリュームをバンドルし、新しい AMI を登録します。新しい AMI を使用して、カスタマイズした新しいインスタンスを起動できます。

instance store-backed Windows AMI を作成する場合、『*Amazon Elastic Compute Cloud Microsoft Windows Guide*』の「[Instance Store-Backed Windows AMI の作成](#)」を参照してください。

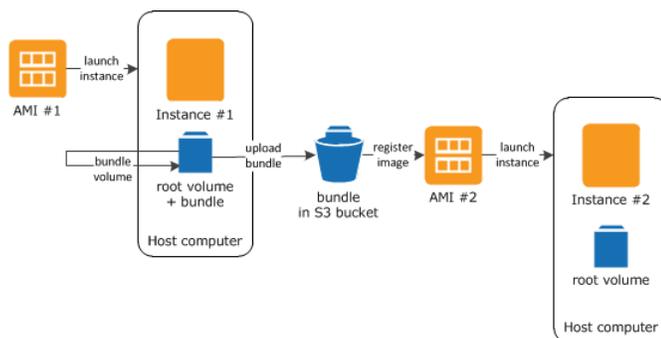
AMI の作成プロセスは、Instance Store-Backed AMI の場合とは異なります。Amazon EBS-Backed インスタンスと Instance Store-Backed インスタンスの違いの詳細と、インスタンスのルートデバイスタイプを判別する方法については、「[ルートデバイスのストレージ \(p. 57\)](#)」を参照してください。Amazon EBS-backed Linux AMI を作成する必要がある場合は、「[Amazon EBS-Backed Linux AMI の作成 \(p. 77\)](#)」を参照してください。

### Topics

- [Instance Store-Backed AMI の作成プロセスの概要 \(p. 81\)](#)
- [前提条件 \(p. 82\)](#)
- [インスタンスから AMI を作成する Store-Backed Linux インスタンス \(p. 82\)](#)
- [Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換する \(p. 87\)](#)

## Instance Store-Backed AMI の作成プロセスの概要

次の図は、Instance Store-Backed インスタンスから AMI を作成するプロセスをまとめたものです。



最初に、作成する AMI に似ている AMI からインスタンスを起動します。インスタンスに接続し、それをカスタマイズできます。インスタンスのカスタマイズが終わったら、それをバンドルできます。バンドルプロセスが完了するには数分間かかります。プロセスが完了すると、バンドルが与えられます。バンドルは、イメージマニフェスト ( `image.manifest.xml` ) と、ルートボリュームのテンプレートを含むファイル ( `image.part.` ) から構成されます。次に、バンドルを Amazon S3 バケットにアップロードし、AMI を登録します。

お客様が新しい AMI を使用してインスタンスを起動すると、Amazon はユーザーが Amazon S3 にアップロードしたバンドルを使用してインスタンスのルートボリュームを作成します。Amazon S3 のバンドルで使用されるストレージ領域については、お客様がその領域を削除するまでアカウントに料金が発生します。詳細については、「[AMI の登録解除 \(p. 92\)](#)」を参照してください。

ルートデバイスボリュームに加えて、インスタンスストアボリュームをインスタンスに追加した場合、新しい AMI のブロックデバイスマッピングにこれらのボリュームの情報が含まれ、新しい AMI から起動するインスタンスのブロックデバイスマッピングに自動的にこれらのボリュームの情報が含まれます。詳細については、[ブロックデバイスマッピング \(p. 643\)](#) を参照してください。

## 前提条件

AMI を作成するには、最初に次のタスクを完了する必要があります。

- AMI ツールをインストールします。詳細については、[Set Up the AMI Tools](#) を参照してください。
- API ツールをインストールします。詳細については、「[Setting Up the Amazon EC2 Command Line Interface Tools on Linux](#)」を参照してください。
- バンドルに Amazon S3 バケットがあることを確認します。Amazon S3 バケットを作成するには、Amazon S3 コンソールを開き、[Create Bucket] をクリックします。



### Note

AWS CLI `mb` コマンドを使用して、バケットを作成することもできます。AWS CLI の使用を開始するには、『[AWS Command Line Interface ユーザーガイド](#)』を参照してください。

- 次の認証情報があることを確認します。
  - AWS アカウント ID。アカウント ID を取得するには、[Your Security Credentials](#) に進み、[Account Identifiers] を展開します。
  - X.509 証明書とプライベートキー。X.509 証明を作成する必要がある場合、[Your Security Credentials](#) にアクセスして、[X.509 Certificates] を展開し、[Create New Certificate] をクリックします。X.509 証明書とプライベートキーは、AMI の暗号化/復号化に使用されます。
  - お客様のアクセスキー ID。アクセスキー ID を取得または作成する必要がある場合、[Your Security Credentials](#) にアクセスして、[Access Keys] を展開します。
  - お客様のシークレットアクセスキー。シークレットアクセスキーは取得できません。そのため、シークレットアクセスキーを見つけられない場合、新規作成する必要があります。シークレットアクセスキーを作成するには、[Your Security Credentials](#) にアクセスして、[Access Keys] を展開し、[Create New Access Key] をクリックします。
- インスタンスに接続し、カスタマイズします。例えば、ソフトウェアとアプリケーションをインストールしたり、データをコピーしたり、一時ファイルを削除したり、Linux 設定を変更したりできます。

## インスタンスから AMI を作成する Store-Backed Linux インスタンス

Amazon EC2 AMI ツールの使用準備を整えるには (HVM インスタンスのみ)

Amazon EC2 AMI ツールでは、GRUB のレガシーが正常に起動する必要があります。一部の AMI (特に Ubuntu) は GRUB 2 を使用するよう設定されています。インスタンスで GRUB のレガシーを使用しているかどうかを確認し、使用していない場合はインストールして設定する必要があります。

AMI ツールが正常に機能するためには、HVM インスタンスにパーティションツールがインストールされている必要もあります。

1. インスタンスで GRUB のレガシーを使用しているかどうかを確認します。
  - a. ブロックデバイスの一覧を表示し、ルートブロックデバイスを探します。

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda        202:0    0   8G  0 disk
  xvda1     202:1    0   8G  0 part /
xvddb       202:16   0  30G  0 disk /media/ephemeral0
```

この例では、ルートデバイス (/ の MOUNTPOINT で示される) は /dev/xvda1 です。ルートブロックデバイスは、その親である /dev/xvda です。

- b. ルートブロックデバイス GRUB のバージョンを確認します。

```
[ec2-user ~]$ sudo file -s /dev/xvda
/dev/xvda: x86 boot sector; GRand Unified Bootloader, stage1 version
0x3, stage2 address 0x2000, 1st sector stage2 0x800, stage2 segment
0x200, GRUB version 0.94; partition 1: ID=0xee, starthead 254,
startsector 1, 16777215 sectors, extended partition table (last)\011,
code offset 0x48
```

前述の例では、GRUB バージョンは GRUB のレガシーである 0.94 です。GRUB バージョンが 0.9x 以下の場合、[Step 3 \(p. 83\)](#) に進むことができます。この出力で GRUB バージョンを確認できない場合は、`grub-install -v` コマンドを試してください。

```
ubuntu:~$ grub-install -v
grub-install (GRUB) 1.99-21ubuntu3.10
```

この例では、GRUB バージョンが 0.9x よりも大きいいため、GRUB のレガシーをインストールする必要があります。[Step 2 \(p. 83\)](#) に進みます。

2. お使いのディストリビューションのパッケージマネージャを使用して `grub` パッケージをインストールし、GRUB のレガシーをインストールします。Ubuntu インスタンスの場合は、次のコマンドを使用します。

```
ubuntu:~$ sudo apt-get install -y grub
```

インスタンスで GRUB のレガシーを使用していることの検証は、`grub --version` コマンドを使用して行えます。

```
ubuntu:~$ grub --version
grub (GNU GRUB 0.97)
```

3. お使いのディストリビューションのパッケージマネージャを使用して、次のパーティション管理パッケージをインストールします。
  - `gdisk` (ディストリビューションによっては代わりにパッケージ `gptfdisk` が呼び出される場合があります)。
  - `kpartx`

Ubuntu インスタンスの場合は、次のコマンドを使用します。

```
ubuntu:~$ sudo apt-get install -y gdisk kpartx
```

4. インスタンスのカーネルパラメータを確認します。

```
ubuntu:~$ cat /proc/cmdline  
BOOT_IMAGE=/boot/vmlinuz-3.2.0-54-virtual root=UUID=4f392932-ed93-4f8f-ae7-  
72bc5bb6ca9d ro console=ttyS0 xen_emul_unplug=unnecessary
```

カーネルおよびルートデバイスのパラメータ `console=ttyS0` と `xen_emul_unplug=unnecessary` を書き留めます。

5. `/boot/grub/menu.lst` のカーネル エントリを確認します。

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst  
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro con  
sole=hvc0  
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single  
kernel /boot/memtest86+.bin
```

`console` パラメータが `hvc0` をポイントしている (`ttyS0` ではない) こと、および `xen_emul_unplug=unnecessary` パラメータが未指定であることに注意してください。

6. `/boot/grub/menu.lst` ファイルを任意のテキストエディタで (`vim` や `nano` など) で編集して、コンソールを変更し、先ほど確認したパラメータをブートエントリに追加します。

```
title          Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual  
root           (hd0)  
kernel         /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs  
ro console=ttyS0 xen_emul_unplug=unnecessary  
initrd         /boot/initrd.img-3.2.0-54-virtual  
  
title          Ubuntu 12.04.3 LTS, kernel 3.2.0-54-virtual (recovery mode)  
root           (hd0)  
kernel         /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs  
ro single console=ttyS0 xen_emul_unplug=unnecessary  
initrd         /boot/initrd.img-3.2.0-54-virtual  
  
title          Ubuntu 12.04.3 LTS, memtest86+  
root           (hd0)  
kernel         /boot/memtest86+.bin
```

7. カーネルエントリに適切なパラメータが含まれていることを確認します。

```
ubuntu:~$ grep ^kernel /boot/grub/menu.lst  
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro con  
sole=ttyS0 xen_emul_unplug=unnecessary  
kernel /boot/vmlinuz-3.2.0-54-virtual root=LABEL=cloudimg-rootfs ro single  
console=ttyS0 xen_emul_unplug=unnecessary  
kernel /boot/memtest86+.bin
```

Instance Store-Backed Linux インスタンスから AMI を作成するには

この手順では、「[前提条件 \(p. 82\)](#)」に記載された前提条件が満たされていることを前提としています。

1. インスタンスに認証情報をアップロードします。Amazon ではこれらの認証情報を使用して、お客様と Amazon EC2 だけがお客様の AMI にアクセスできるようにします。

- a. 次のように、認証情報のための一時ディレクトリをインスタンスに作成します。

```
[ec2-user ~]$ mkdir /tmp/cert
```

それにより、作成したイメージから認証情報を除外できます。

- b. [scp](#) (p. 336) などの安全なコピーツールを使用して、コンピュータからインスタンスの `/tmp/cert` ディレクトリに X.509 証明書とプライベートキーをコピーします。次の `scp` コマンドの `-i my-private-key.pem` オプションは、X.509 プライベートキーではなく、SSH でインスタンスに接続するために使用するプライベートキーです。以下に例を示します。

```
you@your_computer:~ $ scp -i my-private-key.pem /path/to/pk-HKZYK
TAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem /path/to/cert-HKZYKTAIG2ECMXY
IBH3HXV4ZBEXAMPLE.pem ec2-user@ec2-67-202-51-223.compute-1.amazon
aws.com:/tmp/cert/
pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 717      0.7KB/s   00:00
cert-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem 100% 685      0.7KB/s   00:00
```

2. `ec2-bundle-vol` コマンドを使用して、バンドルを Amazon S3 にアップロードする準備をします。`-e` オプションを指定して、認証情報を保存するディレクトリを除外します。デフォルトでは、バンドルプロセスで機密情報を含んでいる可能性があるファイルを除外します。そのようなファイルには、`*.sw`、`*.swo`、`*.swp`、`*.pem`、`*.priv`、`*id_rsa*`、`*id_dsa*`、`*.pgp`、`*.jks`、`*/.ssh/authorized_keys`、`*/.bash_history` があります。これらのファイルをすべて含めるには、`--no-filter` オプションを使用します。これらのファイルの一部を含めるには、`--include` オプションを使用します。



#### Important

AMI バンドルプロセスは、デフォルトで、ルートボリュームを表す `/tmp` ディレクトリに、圧縮され暗号化された一連のファイルを作成します。バンドルを格納するのに十分な空きディスク領域が `/tmp` にない場合、`-d /path/to/bundle/storage` オプションを使用して、バンドルを格納する別の場所を指定する必要があります。インスタンスによっては、エフェメラルストレージが `/mnt` または `/media/ephemeral0` にマウントされて使用可能になっている場合があります。あるいは、バンドルを格納する新しい Amazon EBS ボリュームを作成 (p. 570)、アタッチ (p. 573)、およびマウント (p. 576) することもできます。

- a. `ec2-bundle-vol` コマンドは、`root` として実行する必要があります。ほとんどのコマンドで、`sudo` を使用することでアクセス許可を昇格させることができますが、この場合は、環境変数を維持するために `sudo -E su` を実行する必要があります。

```
[ec2-user ~]$ sudo -E su
```

- b. 次の引数を指定して `ec2-bundle-vol` コマンドを実行します。バンドルを格納するのに十分な空きディスク領域が `/tmp` にない場合は、`-d /path/to/bundle/storage` オプションを使用して、利用可能な領域がある場所を指定します。HVM インスタンスの場合は、必ず `--partition` フラグを追加します。そうしないと、AMI が起動しません。このコマンドと使用可能なオプションの詳細については、『[Amazon Elastic Compute Cloud コマンドラインリファレンス](#)』の「[ec2-bundle-vol](#)」を参照してください。

```
[root ec2-user]# $EC2_AMITool_HOME/bin/ec2-bundle-vol -k /tmp/cert/pk-  
HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -c /tmp/cert/cert-HKZYKTAIG2ECMXY  
IBH3HXV4ZBEXAMPLE.pem -u your_aws_account_id -r x86_64 -e /tmp/cert
```

イメージの作成には数分かかります。このコマンドが完了すると、tmp ディレクトリにバンドル ( image.manifest.xml と複数の image.part.xx ファイル ) が含まれます。

- c. root シェルを終了します。

```
[root ec2-user]# exit
```

3. **ec2-upload-bundle** コマンドを使用して、バンドルを Amazon S3 にアップロードします。バケットにバンドルのプレフィックス ( ディレクトリ ) が存在しない場合、このコマンドを実行することによって作成されることに注意してください。



#### Note

[Step 2.b \(p. 85\)](#) で `-d /path/to/bundle/storage` オプションを使用してパスを指定した場合は、次のコマンドで `/tmp` ではなく `-m` オプションで同じパスを使用します。

```
[ec2-user ~]$ ec2-upload-bundle -b my-s3-bucket/bundle_folder/bundle_name  
-m /tmp/image.manifest.xml -a your_access_key_id -s your_secret_access_key  
--region us-west-2
```

4. ( オプション ) バンドルを Amazon S3 にアップロードしたら、次の `rm` コマンドを使用して、インスタンスの `/tmp` ディレクトリからバンドルを削除できます。



#### Note

[Step 2.b \(p. 85\)](#) で `-d /path/to/bundle/storage` オプションを使用してパスを指定した場合は、次のコマンドで `/tmp` ではなく同じパスを使用します。

```
[ec2-user ~]$ sudo rm /tmp/image.manifest.xml /tmp/image.part.* /tmp/image
```

5. **ec2-register** コマンドを使用して、AMI を登録します。 `-o` と `-w` は、環境変数の `AWS_ACCESS_KEY` と `AWS_SECRET_KEY` を設定している場合は必要ありません。



#### Important

HVM AMI の場合は、`--virtualization-type hvm` フラグを追加します。

```
[ec2-user ~]$ ec2-register my-s3-bucket/bundle_folder/bundle_name/image.manifest.xml -n AMI_name -o your_access_key_id -w your_secret_access_key --region us-west-2
```

## Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換する

Instance Store-Backed Linux AMI は、Amazon EBS-Backed Linux AMI に変換できます。



### Important

Instance Store-Backed Windows AMI から Amazon EBS-Backed Windows AMI への変換、および所有していない AMI の変換はできません。

Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換するには

1. Amazon EBS-Backed AMI から Amazon Linux インスタンスを起動します。詳細については、「[インスタンスの起動 \(p. 325\)](#)」を参照してください。Amazon Linux インスタンスには、Amazon EC2 コマンドラインと AMI ツールがプリインストールされています。
2. Instance Store-Backed AMI をバンドルするのに使用した X.509 プライベートキーをインスタンスにアップロードします。Amazon はこのキーを使用して、お客様と Amazon EC2 だけがお客様の AMI にアクセスできるようにします。

- a. 次のように、X.509 プライベートキーのインスタンスに一時ディレクトリを作成します。

```
[ec2-user ~]$ mkdir /tmp/cert
```

- b. scp などの安全なコピーツールを使用して、コンピュータから [/tmp/cert \(p. 336\)](#) ディレクトリに X.509 プライベートキーをコピーします。次のコマンドの `my-private-key` パラメータは、SSH でインスタンスに接続するために使用するプライベートキーです。以下に例を示します。

```
you@your_computer:~ $ scp -i my-private-key.pem /path/to/pk-HKZYK  
TAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem ec2-user@ec2-67-202-51-223.compute-  
1.amazonaws.com:/tmp/cert/  
pk-HKZYKTAIG2ECMXIYIBH3HXV4ZBEXAMPLE.pem 100% 717 0.7KB/s 00:00
```

3. AWS アクセスキーおよび秘密キーの環境変数を設定します。

```
[ec2-user ~]$ export AWS_ACCESS_KEY=your_access_key_id  
[ec2-user ~]$ export AWS_SECRET_KEY=your_secret_access_key
```

4. 新しい AMI の Amazon EBS ボリュームを準備します。

- a. `ec2-create-volume` コマンドを使用して、インスタンスと同じアベイラビリティーゾーンに空の Amazon EBS ボリュームを作成します。コマンド出力のボリューム ID を書き留めてください。



### Important

この Amazon EBS ボリュームは、元のインスタンスストアのルートボリュームと同じサイズ以上である必要があります。

```
[ec2-user ~]$ ec2-create-volume --size 10 --region us-west-2 --availability-zone us-west-2b  
VOLUME volume_id 10 us-west-2b creating 2014-01-24T23:11:45+0000  
standard
```

- b. `ec2-attach-volume` コマンドを使用して、Amazon EBS-Backed インスタンスにボリュームをアタッチします。

```
[ec2-user ~]$ ec2-attach-volume volume_id -i instance_id --device /dev/sdb  
--region us-west-2  
ATTACHMENT volume_id instance_id /dev/sdb attaching 2014-01-  
24T23:15:34+0000
```

5. バンドルのフォルダを作成します。

```
[ec2-user ~]$ mkdir /tmp/bundle
```

6. `ec2-download-bundle` コマンドを使用して、Instance Store-Backed AMI のバンドルを `/tmp/bundle` にダウンロードします。

```
[ec2-user ~]$ ec2-download-bundle -b my-s3-bucket/bundle_folder/bundle_name  
-m image.manifest.xml -a $AWS_ACCESS_KEY -s $AWS_SECRET_KEY --privatekey  
/path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem -d /tmp/bundle
```

7. `ec2-unbundle` コマンドを使用して、バンドルからイメージファイルを再作成します。  
a. バンドルフォルダにディレクトリを変更します。

```
[ec2-user ~]$ cd /tmp/bundle/
```

- b. `ec2-unbundle` コマンドを実行します。

```
[ec2-user bundle]$ ec2-unbundle -m image.manifest.xml --privatekey  
/path/to/pk-HKZYKTAIG2ECMXYIBH3HXV4ZBEXAMPLE.pem
```

8. バンドルを解除したイメージから新しい Amazon EBS ボリュームにファイルをコピーします。

```
[ec2-user bundle]$ sudo dd if=/tmp/bundle/image of=/dev/sdb bs=1M
```

9. 新しい Amazon EBS ボリュームのマウントポイントを作成し、ボリュームをマウントします。

```
[ec2-user bundle]$ sudo mkdir /mnt/ebs  
[ec2-user bundle]$ sudo mount /dev/sdb /mnt/ebs
```

10. EBS ボリュームの `/etc/fstab` ファイルを任意のテキストエディタ ( `vim` や `nano` など ) で開き、インスタンスストア ( エフェメラル ) ボリュームのエントリがあれば削除します。Amazon EBS

ボリュームは `/mnt/ebs` にマウントされているため、`fstab` ファイルは `/mnt/ebs/etc/fstab` に配置されています。

```
[ec2-user bundle]$ sudo nano /mnt/ebs/etc/fstab
#
LABEL=/          /              ext4      defaults,noatime 1    1
tmpfs            /dev/shm       tmpfs     defaults         0    0
devpts           /dev/pts       devpts    gid=5,mode=620  0    0
sysfs            /sys           sysfs     defaults         0    0
proc             /proc          proc      defaults         0    0
/dev/sdb         /media/ephemeral0 auto      defaults,comment=cloudconfig
0                2
```

前述の例では、最後の行を削除する必要があります。

11. ボリュームをアンマウントし、インスタンスからデタッチします。

```
[ec2-user bundle]$ sudo umount /mnt/ebs
[ec2-user bundle]$ ec2-detach-volume volume_id --region us-west-2
ATTACHMENT volume_id instance_id /dev/sdb detaching 2014-01-24T23:15:34+0000
```

12. 次のように、新しい Amazon EBS ボリュームから AMI を作成します。

- a. 新しい Amazon EBS ボリュームのスナップショットを作成します。

```
[ec2-user bundle]$ ec2-create-snapshot --region us-west-2 -d
"your_snapshot_description" -O $AWS_ACCESS_KEY -W $AWS_SECRET_KEY
volume_id
SNAPSHOT snapshot_id volume_id pending 2014-01-25T00:18:48+0000
1234567891011 10 your_snapshot_description
```

- b. スナップショットが完了していることを確認します。

```
[ec2-user bundle]$ ec2-describe-snapshots --region us-west-2 snapshot_id
SNAPSHOT snapshot_id volume_id completed 2014-01-25T00:18:48+0000 100%
1234567891011 10 your_snapshot_description
```

- c. 元の AMI で使用されたアーキテクチャとカーネルイメージ ( `aki` ) を、`ec2-describe-images` コマンドを使用して特定します。このステップでは、元の Instance Store-Backed AMI の AMI ID が必要です。

```
[ec2-user bundle]$ ec2-describe-images --region us-west-2 ami-id
IMAGE ami-8ef297be amazon/amzn-ami-pv-2013.09.2.x86_64-s3 amazon available
public x86_64 machine aki-fc8f11cc instance-store paravirtual xen
```

この例では、アーキテクチャは `x86_64` で、カーネルイメージ ID は `aki-fc8f11cc` です。次のステップでこれらの値を使用します。前述のコマンドの出力では `ari` ID もリストされるので、これも書き留めます。

- d. 新しい Amazon EBS ボリュームのスナップショット ID と前のステップで書き留めた値を使用して、新しい AMI を登録します。前述のコマンド出力に `ari` ID がリストされていた場合は、その ID を次のコマンドで `--ramdisk` `ari_id` を使用して指定します。

```
[ec2-user bundle]$ ec2-register --region us-west-2 -n your_new_ami_name  
-s snapshot_id -a x86_64 --kernel aki-fc8f11cc  
IMAGE new-ami-id
```

13. (オプション) 新しい AMI からインスタンスを起動できることをテストした後で、この手順で作成した Amazon EBS ボリュームを削除できます。

```
$ ec2-delete-volume volume_id
```

## AMI のコピー

### Abstract

独自の AMI を別の AWS リージョンにコピーします。

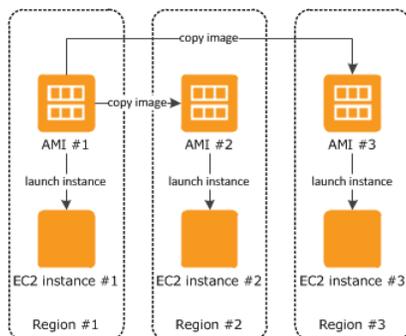
所有している Amazon マシンイメージ (AMI) を他の AWS リージョンに簡単にコピーし、アプリケーションを拡大して AWS の地理的に多様なリージョンを活用できます。

AMI をコピーすると、次の利点があります。

- 一貫性のあるグローバルなデプロイメント: 1 つのリージョンから別のリージョンに AMI をコピーし、同じ AMI を基にして一貫性のあるインスタンスを別のリージョンに起動できます。
- スケーラビリティ: ユーザーの場所にかかわらず、ユーザーのニーズに合った世界規模のアプリケーションをより簡単に設計できます。
- パフォーマンス: アプリケーションを配布したり、アプリケーションの重要なコンポーネントをユーザーの近くに配置したりすることでパフォーマンスを向上できます。また、インスタンスの種類やその他の AWS サービスなど、リージョン固有の機能を活用することもできます。
- 高可用性: アプリケーションを設計し、AWS リージョン全体にわたってデプロイして可用性を高めることができます。

## AMI コピー

Amazon EBS-Backed AMI と Instance Store-Backed AMI の両方をコピーできます。AMI をコピーできるリージョンの数に制限はありません。AMI を同じリージョンにコピーすることもできます。AMI の各コピーは新しい AMI になり、固有の AMI ID を持ちます。ユーザーが AMI からインスタンスを起動するとき、下の図のように、Amazon はユーザーが選択した AMI と同じリージョンにそのインスタンスを起動します。



AMI をコピーするとき、新しい AMI は、元の AMI から完全に独立しており、元の (ソース) AMI へのリンクはありません。元の AMI に影響を与えずに、新しい AMI を変更できます。その逆もまた真です。新しい AMI に影響を与えずに、元の AMI を変更できます。そのため、元の AMI を変更し、それらの変更を対象リージョンの AMI に反映させる場合、元の AMI を対象のリージョンに再度コピーする必要があります。

Amazon は元の AMI から新しい AMI に起動許可、ユーザー定義のタグ、Amazon S3 バケット許可をコピーしません。コピー操作が完了すると、起動許可、ユーザー定義のタグ、Amazon S3 バケット許可を新しい AMI に適用できます。

Amazon は、新しい AMI に合った AKI と ARI の検索を選択された対象リージョンで試行します。合致する AKI または ARI が見つからない場合、Amazon は AMI をコピーしません。Amazon が推奨する AKI と ARI を使用している場合は、コピー操作によって、適切な AKI と ARI とともに AMI が対象のリージョンに登録されます。「Failed to find matching AKI/ARI」というエラーメッセージが表示された場合、対象のリージョンに元の AMI で指定されたものに一致する AKI または ARI が含まれていないことを意味します。AMI で PV-GRUB AKI を使用している場合は、AMI を更新して、最新バージョンの PV-GRUB を利用できます。PV-GRUB と AKI の詳細については、[独自の Linux カーネルの使用 \(p. 102\)](#) を参照してください。

AMI のコピーには課金されません。ただし、標準のストレージ料金とデータ転送料金が適用されます。

## Amazon EC2 AMI のコピー

AMI をコピーする前に、ソース AMI のすべてのコンテンツが、異なるリージョンでの実行をサポートするように更新されていることを確認する必要があります。例えば、データベース接続文字列や同様のアプリケーション設定データが、適切なリソースを指すように更新する必要があります。それ以外の場合、対象のリージョンの新しい AMI から起動したインスタンスは元のリージョンのリソースをまだ使用している可能性があり、それによりパフォーマンスとコストに影響が及ぶことがあります。

AWS Management Console またはコマンドラインを利用して AMI をコピーできます。

コンソールを使用して AMI をコピーするには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーから、AMI を含むリージョンを選択して、コピーします。
3. ナビゲーションペインの [AMIs] をクリックします。
4. コピーする AMI を選択し、[Actions] をクリックし、[Copy AMI] を選択します。
5. [AMI Copy] ページで、次のフィールドを設定し、[Copy AMI] をクリックします。
  - [Destination region]: AMI をコピーするリージョンを選択します。
  - [Name]: 新しい AMI の名前を指定します。
  - [Description]: デフォルトでは、元の AMI と新しい AMI を見分けられるよう、元の AMI に関する情報が説明に含まれています。この説明は、必要に応じて変更できます。
6. コピー操作が開始したことを知らせる確認ページが表示され、新しい AMI の ID が提供されます。

コピー操作の進捗状況をすぐに確認するには、与えられたリンクをクリックし、対象のリージョンに切り替えます。進捗状況を後で確認するには、[Done] をクリックします。進捗状況を確認する段階に入ったら、ナビゲーションペインを使用して、対象のリージョンに切り替えます。

対象の AMI の初期ステータスは `pending` です。ステータスが `available` になると、操作は完了します。

コマンドラインを使用して AMI をコピーするには

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

- [copy-image](#) ( AWS CLI )
- [ec2-copy-image](#) ( Amazon EC2 CLI )
- [Copy-EC2Image](#) ( AWS Tools for Windows PowerShell )

## 保留中の AMI コピー操作を中止する

AWS Management Console またはコマンドラインを使用して、保留中の AMI コピーを中止できます。

コンソールを使用して AMI のコピー操作を中止するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーのリージョンセレクターから対象のリージョンを選択します。
3. ナビゲーションペインの [AMIs] をクリックします。
4. コピーを中止する AMI を選択し、[Actions] をクリックし、[Deregister] をクリックします。
5. 確認を求められたら、[Continue] をクリックします。

コマンドラインを使用して AMI コピー操作を中止するには

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

- [deregister-image](#) ( AWS CLI )
- [ec2-deregister](#) ( Amazon EC2 CLI )
- [Unregister-EC2Image](#) ( AWS Tools for Windows PowerShell )

## AMI の登録解除

### Abstract

AMI の利用が終わったら、その登録を解除します。

AMI の利用が終わったら、その登録を解除できます。AMI の登録を解除すると、それを使用して新しいインスタンスを起動できなくなります。

AMI の登録を解除しても、AMI から既に起動したインスタンスに影響を与えることはありません。そのようなインスタンスの使用に対しては引き続き課金されます。そのため、使用が終わったら、インスタンスを終了することをお勧めします。

AMI のクリーンアップに使用する手順は、Amazon EBS-Backed と Instance Store-Backed で異なります。

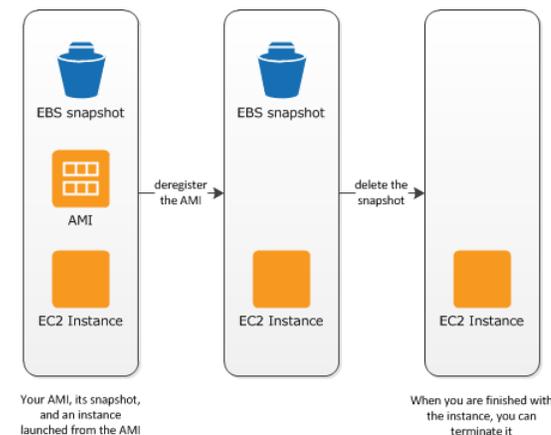
### Topics

- [Amazon EBS-Backed AMI をクリーンアップする \(p. 93\)](#)
- [Instance Store-Backed AMI をクリーンアップする \(p. 93\)](#)

## Amazon EBS-Backed AMI をクリーンアップする

Amazon EBS-Backed AMI の登録を解除しても、AMI の作成時に作成したスナップショットに影響を与えることはありません。Amazon EBS のこのスナップショットの使用に対しては引き続き課金されます。そのため、使用が終わったら、スナップショットは削除することをお勧めします。

次の図は、Amazon EBS-Backed AMI のクリーンアッププロセスについてまとめたものです。



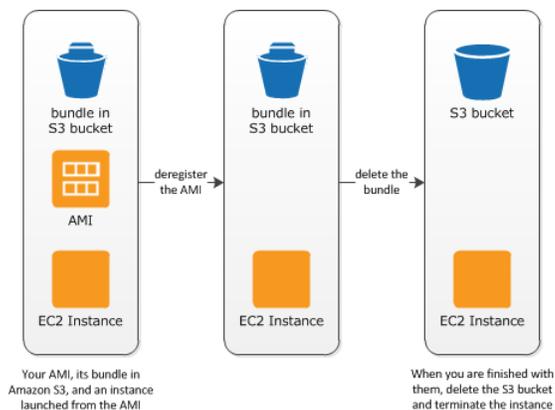
Amazon EBS-Backed AMI をクリーンアップするには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [AMIs] をクリックします。AMI を選択し、[Actions] をクリックして、[Deregister] をクリックします。確認を求められたら、[Continue] をクリックします。  
  
AMI の現在のステータスは `unavailable` です。
3. ナビゲーションペインで、[Snapshots] をクリックします。スナップショットを選択し、[Delete Snapshot] をクリックします。確認を求めるメッセージが表示されたら、[Yes, Delete] をクリックします。
4. ( オプション ) AMI から起動したインスタンスの使用が終わったら、それを終了します。ナビゲーションペインの [Instances] をクリックします。インスタンスを選択して、[Actions] をクリックし、[Terminate.] をクリックします。確認を求めるメッセージが表示されたら、[Yes, Terminate] をクリックします。

## Instance Store-Backed AMI をクリーンアップする

Instance Store-Backed AMI の登録を解除しても、AMI の作成時に Amazon S3 にアップロードしたファイルに影響を与えることはありません。Amazon S3 のそれらのファイルの使用に対しては引き続き課金されます。そのため、使用が終わったら、それらのファイルは削除することをお勧めします。

次の図は、Instance Store-Backed AMI のクリーンアッププロセスをまとめたものです。



### Instance Store-Backed AMI をクリーンアップするには

1. 次のように、`ec2-deregister` コマンドを使用して AMI の登録を解除します。

```
ec2-deregister ami_id
```

AMI の現在のステータスは `unavailable` です。

2. 次のように、`ec2-delete-bundle` コマンドを使用してバンドルを削除します。

```
ec2-delete-bundle -b myawsbucket/myami -a your_access_key_id -s  
your_secret_access_key -p image
```

3. (オプション) AMI から起動したインスタンスの使用が終わったら、次のように、`ec2-terminate-instances` コマンドを使用してそれを終了します。

```
ec2-terminate-instances instance_id
```

4. (オプション) バンドルをアップロードした Amazon S3 バケットの使用が終わったら、バケットを削除できます。Amazon S3 バケットを削除するには、Amazon S3 コンソールを開き、バケットを選択して、[Actions] をクリックし、[Delete] をクリックします。

## Amazon Linux

### Abstract

Amazon Linux は Amazon EC2 で使用するために AWS が提供する Linux イメージです。

Amazon Linux はアマゾン ウェブ サービス (AWS) が提供します。これは Amazon EC2 上で実行するアプリケーションのために安定した安全で高性能な実行環境を提供できるよう設計されています。またこれには、起動設定ツールおよび多くの AWS 人気ライブラリやツールなど、AWS の統合を容易にするいくつかのパッケージも含まれています。AWS は Amazon Linux を実行しているすべてのインスタンスに、現行のセキュリティとメンテナンスの更新を提供します。

Amazon Linux インスタンスを起動するには、LinuxAMI; を使用します。AWS は、Amazon Linux AMI を Amazon EC2 ユーザーに追加料金なしで提供しています。

### Topics

- [Amazon Linux AMI の検索 \(p. 95\)](#)

- [Amazon Linux インスタンスを起動し、それに接続する \(p. 95\)](#)
- [Amazon Linux AMI イメージの特定 \(p. 95\)](#)
- [組み込まれている AWS コマンドラインツール \(p. 96\)](#)
- [cloud-init \(p. 97\)](#)
- [リポジトリの設定 \(p. 98\)](#)
- [パッケージの追加 \(p. 99\)](#)
- [参照のためのソースパッケージへのアクセス \(p. 99\)](#)
- [アプリケーションの開発 \(p. 100\)](#)
- [インスタンスストアアクセス \(p. 100\)](#)
- [製品ライフサイクル \(p. 100\)](#)
- [セキュリティの更新 \(p. 101\)](#)
- [サポート \(p. 101\)](#)

## Amazon Linux AMI の検索

最新の Amazon Linux AMI のリストについては、「[Amazon Linux AMI](#)」を参照してください。

## Amazon Linux インスタンスを起動し、それに接続する

希望する AMI が見つかったら、AMI ID を控えておきます。AMI ID を使ってインスタンスを起動し、そのインスタンスに接続することができます。

デフォルトでは、Amazon Linux はリモートルート SSH を許可しません。また、パスワード認証は、パスワードのブルートフォース攻撃を防ぐために無効になっています。Amazon Linux インスタンスへの SSH ログインを有効にするには、起動時にキーペアをインスタンスに提供する必要があります。また、インスタンスを起動するときに使用するセキュリティグループで、SSH アクセスを許可するよう設定する必要があります。デフォルトでは、SSH を使用してリモートログインできる唯一のアカウントは `ec2-user` です。このアカウントには `sudo` 特権もあります。リモートルートログインを有効にする場合は、このログインが、キーペアおよびセカンダリユーザーを使用する場合よりも安全性が低いことに注意してください。

Amazon Linux インスタンスの起動と使用についての詳細は、[インスタンスの起動 \(p. 324\)](#) を参照してください。Amazon Linux インスタンスへの接続についての詳細は、[Linux/Unix インスタンスへの接続 \(p. 335\)](#) を参照してください。

## Amazon Linux AMI イメージの特定

各イメージには AMI を特定する一意の `/etc/image-id` が含まれています。このファイルには、イメージに関する情報が含まれています。

次は、`/etc/image-id` ファイルの例です。

```
[ec2-user ~]$ cat /etc/image-id
image_name="amzn-ami-pv"
image_version="2014.03"
image_arch="x86_64"
image_file="amzn-ami-pv-2014.03.rc-1.x86_64.ext4"
image_stamp="6270-181c"
image_date="20140320044105"
```

```
recipe_name="amzn ami"  
recipe_id="5f291ea8-31de-0e3c-05c2-8f26-942c-852a-0d5f98fa"
```

image\_name、image\_version、および image\_arch 項目は、イメージを作成するときに Amazon が使用したビルドレシピに基づいています。image\_stamp は、イメージの作成中に生成されたランダムな一意の 16 進値にすぎません。image\_date 項目の形式は YYYYMMDDhhmmss で、イメージを作成した UTC 時間です。recipe\_name と recipe\_id は、イメージを作成するときに Amazon が使用したビルドレシピの名前と ID を参照し、これにより、現在実行中の Amazon Linux バージョンが特定されます。このファイルは、yum リポジトリから更新をインストールしても変更されません。

Amazon Linux には、インストールされている現在のリリースを示す /etc/system-release ファイルが含まれています。このファイルは、yum によって更新され、system-release RPM の一部です。

次は、/etc/system-release ファイルの例です。

```
[ec2-user ~]$ cat /etc/system-release  
Amazon Linux AMI release 2014.03
```

Amazon Linux には、/etc/system-release-cpe 内の機械による読み取りが可能なバージョンの /etc/system-release ファイルも含まれます。また、MITRE の CPE 仕様に準拠しています (CPE)。

## 組み込まれている AWS コマンドラインツール

Amazon Linux またはデフォルトリポジトリには、AWS の統合および使用量によく使用される次のコマンドラインツールが含まれています。

- aws-amitools-ec2
- aws-apitools-as
- aws-apitools-cfn
- aws-apitools-common
- aws-apitools-ec2
- aws-apitools-elb
- aws-apitools-iam
- aws-apitools-mon
- aws-apitools-rds
- aws-cfn-bootstrap
- aws-cli
- aws-scripts-ses

これらのツールの設定を簡素化するために、認証情報ファイルのインストール後、AWS\_CREDENTIAL\_FILE、JAVA\_HOME、AWS\_PATH、PATH、および製品固有の環境変数を準備するためのシンプルなスクリプトが組み込まれます。

また、次に説明するように、複数のバージョンの API と AMI ツールをインストールできるように、これらのツールの希望するバージョンにアクセスするシンボリックリンクを /opt/aws に配置しました。

`/opt/aws/bin`

インストールされている各ツールディレクトリにある、/bin ディレクトリへのシンボリックリンク。

`/opt/aws/{apitools|amitools}`

インストールされた最新バージョンにアタッチされている `name-version` フォームと `name` シンボリックリンクのディレクトリに製品がインストールされます。

```
/opt/aws/{apitools|amitools}/name/environment.sh
```

/etc/profile.d/aws-apitools-common.sh により使用され、EC2\_HOME などの製品固有の環境変数を設定します。

## cloud-init

cloud-init パッケージは、Canonical によって構築されたオープンソースアプリケーションであり、Amazon EC2 などのクラウドコンピューティング環境で Linux イメージをブートストラップするときに使用されます。Amazon Linux にはカスタマイズされたバージョンの cloud-init が含まれています。これにより、起動時のインスタンスに対するアクションを指定することができます。インスタンスの起動時に、ユーザーデータフィールドを使用して必要なアクションを cloud-init に渡すことができます。つまり、さまざまなユースケースに対して共通の AMI を使用し、起動時にその AMI を動的に設定できます。また、Amazon Linux は cloud-init を使用して、ec2-user アカウントの初期設定を実行します。

cloud-init の詳細については、<https://help.ubuntu.com/community/CloudInit> を参照してください。

Amazon Linux は次の cloud-init アクションを使用します ( /etc/sysconfig/cloudinit で設定できます )。

- action: INIT ( 常に実行 )
  - デフォルトロケールの設定
  - ホスト名の設定
  - ユーザーデータの解析と処理を行う
- action: CONFIG\_SSH
  - ホスト プライベート SSH キーを生成する
  - 容易にログインおよび管理できるように、ユーザーのパブリック SSH キーを .ssh/authorized\_keys に追加する
- action: PACKAGE\_SETUP
  - yum repo を準備する
  - ユーザーデータで定義されたパッケージアクションの処理
- action: RUNCMD
  - シェルコマンドの実行
- action: RUN\_USER\_SCRIPTS
  - ユーザーデータにあるユーザースクリプトの実行
- action: CONFIG\_MOUNTS
  - エフェメラルドライブのマウント
- action: CONFIG\_LOCALE
  - ユーザーデータに基づく、ロケール設定ファイルのロケールを設定する

## サポートされているユーザーデータ形式

cloud-init パッケージでは、さまざまな形式のユーザーデータを処理できます。

- Gzip
  - ユーザーデータが gzip で圧縮されている場合、cloud-init はデータを解凍し、適切に処理します。
- MIME マルチパート

- MIME マルチパートファイルを使用して、複数のデータタイプを指定できます。例えば、ユーザーデータスクリプトとクラウド設定タイプの両方を指定できます。マルチパートファイルのパートの形式がサポートされている形式のいずれかの場合、そのパートは `cloud-init` で処理できます。
- Base64 デコード
  - ユーザーデータが base64 でエンコードされている場合、`cloud-init` は、デコードされたデータをサポートされているタイプのいずれかとして認識できるか確認します。デコードされたデータを認識できる場合、データをデコードし、適切に処理します。認識できない場合、base64 データは変更されません。
- ユーザーデータスクリプト
  - 「#!」または「Content-Type: text/x-shellscript」で始まります。
  - このスクリプトは、初回の起動サイクル時に `/etc/init.d/cloud-init-user-scripts` によって実行されます。これは起動プロセスの後半（初期設定アクションが実行された後）に実行されます。
- インクルードファイル
  - 「#include」または「Content-Type: text/x-include-url」で始まります。
  - このコンテンツはインクルードファイルです。ファイルには URL の一覧（1 行に 1 つの URL）が含まれます。各 URL が読み取られ、そのコンテンツが同じルールセットを使用して渡されます。URL から読み取られたコンテンツは gzip、MIME マルチパート、またはプレーンテキスト形式になります。
- クラウド設定データ
  - 「#cloud-config」または「Content-Type: text/cloud-config」で始まります。
  - このコンテンツはクラウド設定データです。サポートされている設定形式のコメント付きサンプルについては、例を参照してください。
- クラウドブートフック
  - 「#cloud-boothook」または「Content-Type: text/cloud-boothook」で始まります。
  - このコンテンツはブートフックデータです。このデータは `/var/lib/cloud` にあるファイルに保存され、すぐに実行されます。
  - これは最初に使用可能な「フック」です。1 回だけ実行するためのメカニズムはありません。ブートフックは自身でこの点に対処する必要があります。環境変数 `INSTANCE_ID` でインスタンス ID が指定されています。この変数を使用して、インスタンスあたり 1 つのブートフックデータのセットを提供します。

## リポジトリの設定

2011 年 9 月リリースの Amazon Linux より、Amazon Linux AMI はスナップショットとして扱われるようになり、`yum update -y` の実行時には、最新のパッケージを常に提供するリポジトリおよび更新構造を備えるようになりました。

このリポジトリ構造は、Amazon Linux のあるバージョンから次のバージョンへのローリング更新を可能にするように、連続的な更新が配信されるように設定されています。たとえば、前のバージョンの Amazon Linux AMI（2013.09 やそれより前のバージョンなど）からインスタンスを起動し、`yum update -y` を実行する場合でも、最新のパッケージが利用できます。

Amazon Linux のローリング更新を無効にするには、*lock-on-launch* 機能を有効にします。lock-on-launch 機能は、新たに起動したインスタンスが、指定したリリースの AMI からのみ更新を受け取れるようにロックします。たとえば、2014.03 AMI に移行する準備ができるまでは、2013.09 AMI を起動したときに、2014.03 AMI より前にリリースされた更新のみを受け取るよう指定できます。新しいインスタンスで lock-on-launch を有効にするには、EC2 コンソールまたは `-f` フラグが設定された `ec2-run-instances` コマンドを使用して、次のユーザーデータを `cloud-init` に渡してインスタンスを起動します。

```
#cloud-config
repo_releasever: 2013.09
```

既存のインスタンスを現在の AMI リリースバージョンにロックするには

1. `/etc/yum.conf` を編集します。
2. `releasever=latest` をコメントアウトします。
3. `yum clean all` を実行し、キャッシュを消去します。

## パッケージの追加

Amazon Linux は、各 Amazon EC2 のリージョンでホストされているオンラインパッケージリポジトリと一緒に使用するように設計されています。これらのリポジトリは、Amazon Linux AMI パッケージの継続的な更新だけでなく、何百もの一般的なオープンソースのサーバーアプリケーションへのアクセスを提供します。リポジトリはすべてのリージョンに存在し、yum 更新ツールを使用してアクセスできるほか、[Amazon Linux AMI パッケージのサイト](#)でもアクセスできます。各リージョンでリポジトリをホストしているため、データ転送料金なしで、更新を迅速にデプロイできます。パッケージをインストールするには、次の yum コマンドを発行します。

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

EnterpriseLinux ( EPEL ) リポジトリの追加パッケージへのアクセスが設定されていますが、デフォルトでは有効になっていません。EPEL は、Amazon Linux リポジトリのパッケージのほか、サードパーティ製のパッケージを提供します。AWS はサードパーティ製のパッケージをサポートしていません。

必要なアプリケーションが Amazon Linux に含まれていない場合は、そのアプリケーションを直接 Amazon Linux インスタンスにインストールできます。Amazon Linux は、RPM および yum を使用してパッケージを管理します。これが最も簡単に新しいアプリケーションをインストールする方法になるでしょう。Amazon Linux 中央リポジトリで利用可能なアプリケーションは多数あるので、アプリケーションがそのリポジトリで利用できるかどうかを最初に必ず確認する必要があります。これらのアプリケーションは、Amazon Linux インスタンスに簡単に追加できます。

実行中の Amazon Linux インスタンスにアプリケーションをアップロードするには、`scp` または `sftp` を使用し、インスタンスにログオンしてアプリケーションを設定します。組み込みの `cloud-init` パッケージから `PACKAGE_SETUP` アクションを使用して、インスタンスの起動時にアプリケーションをアップロードすることもできます。詳細については、[cloud-init \(p. 97\)](#) を参照してください。



### Important

VPC ( Virtual Private Cloud ) でインスタンスを実行している場合、yum リポジトリにアクセスするために、インターネットゲートウェイを VPC にアタッチする必要があります。詳細については、[Amazon Virtual Private Cloud ユーザーガイドのインターネットゲートウェイ](#)を参照してください。

## 参照のためのソースパッケージへのアクセス

Amazon Linux で提供されているツールを使用して、インスタンスにインストールしたパッケージソースを参照するために表示できます。ソースパッケージは、Amazon Linux およびオンラインパッケージリポジトリに含まれるすべてのパッケージで利用できます。インストールするソースパッケージのパッケージ名を確認し、`get_reference_source` コマンドを使用して実行中のインスタンス内にソースを表示します。以下に例を示します。

```
[ec2-user ~]$ get_reference_source -p bash
```

レスポンスの例を次に示します。

```
Requested package: bash
Found package from local RPM database: bash-4.1.2-15.17.amzn1.x86_64
Corresponding source RPM to found package :
bash-4.1.2-15.17.amzn1.src.rpm

Are these parameters correct? Please type 'yes' to continue: yes
Source RPM downloaded to:
/usr/src/srpm/debug/bash-4.1.2-15.17.amzn1.src.rpm
```

ソース RPM は、インスタンスの `/usr/src/srpm/debug` ディレクトリにあります。ソース RPM はこの場所から解凍されます。そして、標準の RPM ツールを使用して、参照するためにソースツリーを表示できます。デバッグが完了したら、パッケージを利用できます。



#### Important

VPC ( Virtual Private Cloud ) でインスタンスを実行している場合、yum リポジトリにアクセスするために、インターネットゲートウェイを VPC にアタッチする必要があります。詳細については、*Amazon Virtual Private Cloud ユーザーガイド* の [インターネットゲートウェイ](#) を参照してください。

## アプリケーションの開発

Amazon Linux の yum リポジトリには、Linux 開発ツールのフルセットが用意されています。Amazon Linux でアプリケーションを開発するには、yum で必要な開発ツールを選択します。あるいは、CentOS および他の同様のディストリビューションで開発された多くのアプリケーションは、Amazon Linux で実行できるはずです。

## インスタンスストアアクセス

インスタンスストアドライブ ephemeral0 は、Amazon Instance Store-Backed AMI でのみ、`/media/ephemeral0` にマウントされます。これは、インスタンスストアドライブを `/mnt` にマウントする他の多くのイメージとは異なります。

## 製品ライフサイクル

Amazon Linux AMI のセキュリティと機能強化は定期的に更新されます。Amazon Linux インスタンスでデータまたはカスタム設定を保存する必要がない場合は、最新の Amazon Linux AMI で新しいインスタンスを再起動できます。Amazon Linux インスタンスでデータまたはカスタム設定を保存する必要がある場合は、Amazon Linux yum リポジトリを介してこれらのインスタンスを維持できます。yum リポジトリには、更新されたすべてのパッケージが含まれます。実行中のインスタンスにこれらの更新を適用するよう選択できます。

新しいバージョンの AMI がリリースされても、古いバージョンの AMI と更新パッケージは引き続き利用できます。AWS サポートを介して古いバージョンの Amazon Linux; サポートをお求めの場合、Amazon では、サポートプロセスの一環として、新しいバージョンに移行するようお客様にお願いすることがあります。

## セキュリティの更新

セキュリティの更新は、Amazon Linux AMI yum リポジトリと更新された Amazon Linux AMI を介して提供されます。セキュリティアラートは、[Amazon Linux AMI セキュリティセンター](#)で公開されます。AWS セキュリティポリシーの詳細については、またはセキュリティの問題を報告するには、[AWS セキュリティセンター](#)にアクセスしてください。

Amazon Linux AMI は、起動時にセキュリティの更新をダウンロードおよびインストールするよう設定されています。これは `repo_upgrade` と呼ばれる `cloud-init` 設定を介して制御されます。次の `cloud-init` 設定のスニペットは、インスタンス初期化に渡すユーザーデータテキストで設定を変更する方法を示しています。

```
#cloud-config
repo_upgrade: security
```

`repo_upgrade` 設定に使用できる値は次のようになります。

Amazon によってセキュリティ更新としてマークされた保留中の更新を適用します。

### **bugfix**

Amazon によってバグフィックスとしてマークされた更新を適用します。バグフィックスは大きなサイズの更新セットで、セキュリティ更新および他のさまざまな小さなバグに対する修正が含まれます。

分類に関係なく、使用できる適切な更新すべてを適用します。

### **none**

起動時に更新をインスタンスに適用しません。

`repo_upgrade` のデフォルトの設定は `security` です。つまり、ユーザーデータで別の値を指定しない場合、Amazon Linux AMI は、デフォルトにより、インストールされているすべてのパッケージに対して、起動時にセキュリティを更新します。Amazon Linux AMI はまた、`/etc/motd` ファイルを利用して、ログイン時に利用できる更新の数を表示し、インストールされているパッケージすべての更新をお知らせします。これらの更新をインストールするには、インスタンスで `sudo yum upgrade` を実行する必要があります。



### **Important**

VPC ( Virtual Private Cloud ) でインスタンスを実行している場合、yum リポジトリにアクセスするために、インターネットゲートウェイを VPC にアタッチする必要があります。詳細については、[Amazon Virtual Private Cloud ユーザーガイドのインターネットゲートウェイ](#)を参照してください。

## サポート

基本的な Amazon Linux AMI のインストールと使用に関するサポートは、AWS サポートの加入者向けサービスとなっています。詳細については、[AWS サポート](#)を参照してください。

Amazon Linux についてご質問がありましたら、[Amazon EC2 forum](#) に投稿することをお勧めします。

## 独自の Linux カーネルの使用

### Abstract

AMI を作成するとき、PV-GRUB を使用するカーネルを提供します。

インスタンスでユーザー指定カーネルを有効にするために、Amazon は PV-GRUB というシステムを使用する Amazon カーネルイメージ (AKI) を公開しました。PV-GRUB は、パッチが適用されたバージョンの GNU GRUB 0.97 を実行する準仮想ブートローダです。インスタンスを開始するとき、PV-GRUB は、お客様のイメージの `menu.lst` ファイルが指定するカーネルを読み込みます。

PV-GRUB は標準の `grub.conf` または `menu.lst` コマンドを認識しますこれにより、現在サポートされているすべての Linux ディストリビューションとともに利用できます。Ubuntu 10.04 LTS、Oracle Enterprise Linux、CentOS 5.x など、古いディストリビューションでは特別な「ec2」や「xen」カーネルパッケージが必要です。新しいディストリビューションでは、デフォルトのカーネルパッケージに必要なドライバが含まれています。

最新の準仮想 AMI では、デフォルトで PV-GRUB AKI を使用します ( Amazon EC2 Launch Wizard Quick Start メニューで利用できるすべての準仮想 Linux AMI が含まれています )。そのため、使用するカーネルにディストリビューションとの互換性がある場合、インスタンスで別のカーネルを使用するために必要な追加の手順はありません。AMI のカーネルイメージが PV-GRUB AKI であることを Amazon EC2 コマンドラインツール ( チェックするカーネルイメージ ID を代用します ) で検証できます。

```
$ ec2-describe-images -a -F image-id=aki-880531cd
IMAGE    aki-880531cd    amazon/pv-grub-hd0_1.04-x86_64.gz ...
```

出力の `[name]` フィールドに `pv-grub` が入っている必要があります。

### Topics

- [PV-GRUB の制約事項 \(p. 102\)](#)
- [GRUB の設定 \(p. 103\)](#)
- [Amazon PV-GRUB カーネルイメージ ID \(p. 104\)](#)
- [PV-GRUB の更新 \(p. 106\)](#)

## PV-GRUB の制約事項

PV-GRUB には次の制約事項があります。

- PV-GRUB の 64 ビットバージョンを使用して 32 ビットカーネルを起動したり、PV-GRUB の 32 ビットバージョンを使用して 64 ビットカーネルを起動したりすることはできません。
- PV-GRUB AKI の使用時には、Amazon ラムディスクイメージ (ARI) を指定できません。
- AWS は、PV-GRUB が EXT2、EXT3、EXT4、JFS、XFS、ReiserFS のファイルシステム形式で動作することをテストし、確認しています。その他のファイルシステム形式では動作しない場合があります。
- PV-GRUB は、gzip、bzip2、lzo、xz 圧縮形式を利用して圧縮されたカーネルを起動できます。
- Cluster AMI は PV-GRUB をサポートせず、また、必要としません。完全ハードウェア仮想化 (HVM) が使用されるためです。準仮想インスタンスは PV-GRUB を使用して起動します。一方、HVM インスタンスボリュームは実際のディスクのように扱われ、その起動プロセスはパーティション分割ディスクとブートローダーを備えるペアメタルオペレーティングシステムの起動プロセスに似ています。
- PV-GRUB バージョン 1.03 以前では、GPT パーティショニングをサポートしません。MBR パーティショニングがサポートされています。
- Amazon EBS で Logical Volume Manager (LVM) を使用する場合、LVM の外側に別の起動パーティションが必要です。その場合、LVM で論理ボリュームを作成できます。

## GRUB の設定

PV-GRUB を起動するには、GRUB `menu.lst` ファイルがイメージに含まれている必要があります。このファイルの最も一般的な場所は `/boot/grub/menu.lst` です。

次の例は、PV-GRUB AKI を使用して AMI を起動する `menu.lst` 設定ファイルです。この例では、Amazon Linux 2013.09 (この AMI の元々のカーネル) と Vanilla Linux 3.11.6 (<https://www.kernel.org/> の新しいバージョンの Vanilla Linux カーネル) の 2 つのカーネルエントリを選択できます。Vanilla エントリは、この AMI の元々のエントリからコピーされました。kernel と initrd パスは新しい場所に更新されました。[default 0] パラメータは、ブートローダをそれが検出した最初のエントリ (この場合、Vanilla エントリ) にポイントします。[fallback 1] パラメータは、最初のエントリの起動に問題が発生した場合、次のエントリにブートローダをポイントします。

```
default 0
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 3.11.6
root (hd0)
kernel /boot/vmlinuz-3.11.6 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-3.11.6

title Amazon Linux 2013.09 (3.4.62-53.42.amzn1.x86_64)
root (hd0)
kernel /boot/vmlinuz-3.4.62-53.42.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-3.4.62-53.42.amzn1.x86_64.img
```

`menu.lst` ファイルにフォールバックカーネルを指定する必要はありません。ただし、新しいカーネルをテストするときは、フォールバックを設定することをお勧めします。PV-GRUB では、新しいカーネルにエラーがあった場合に別のカーネルにフォールバックできます。フォールバックカーネルを設定すると、新しいカーネルが見つからない場合でもインスタンスを起動できます。

PV-GRUB は、次の場所で `menu.lst` をチェックします。その際、それが検出した最初の場所が利用されます。

- `(hd0)/boot/grub`
- `(hd0,0)/boot/grub`
- `(hd0,0)/grub`
- `(hd0,1)/boot/grub`
- `(hd0,1)/grub`
- `(hd0,2)/boot/grub`
- `(hd0,2)/grub`
- `(hd0,3)/boot/grub`
- `(hd0,3)/grub`

PV-GRUB 1.03 以前では、このリストの最初の 2 つの場所うちの 1 つのみがチェックされることに注意してください。

## Amazon PV-GRUB カーネルイメージ ID

PV-GRUB AKI はすべての Amazon EC2 リージョンで利用できます。32 ビットと 64 ビットの両方のアーキテクチャタイプに AKI があります。最新の AMI では、デフォルトで PV-GRUB AKI が使用されます。

すべてのバージョンの PV-GRUB AKI がすべてのインスタンスタイプと互換性があるとは限らないため、常に最新バージョンの PV-GRUB AKI を使用することをお勧めします。次のコマンドを使用し、現在のリージョンの PV-GRUB AKI のリストを取得します。

```
$ ec2-describe-images -o amazon --filter "name=pv-grub-*.gz"
```

PV-GRUB は、ap-southeast-2 リージョンで利用できる唯一の AKI であることにご注意ください。このリージョンにコピーする AMI が、このリージョンで利用できる PV-GRUB のバージョンを使用していることを確認してください。

各リージョンの現在の AKI ID は次のとおりです。現在、hd0 AKI と hd00 AKI には違いがありません。ただし、下位互換性のために両方の命名体系を引き続き提供します。新しい AMI は、hd0 AKI を使用して登録する必要があります。

### ap-northeast-1, アジアパシフィック ( 東京 ) リージョン

イメージ ID	イメージ名
aki-136bf512	pv-grub-hd0_1.04-i386.gz
aki-176bf516	pv-grub-hd0_1.04-x86_64.gz
aki-196bf518	pv-grub-hd00_1.04-i386.gz
aki-1f6bf51e	pv-grub-hd00_1.04-x86_64.gz

### ap-southeast-1, アジアパシフィック ( シンガポール ) リージョン

イメージ ID	イメージ名
aki-ae3973fc	pv-grub-hd0_1.04-i386.gz
aki-503e7402	pv-grub-hd0_1.04-x86_64.gz
aki-563e7404	pv-grub-hd00_1.04-i386.gz
aki-5e3e740c	pv-grub-hd00_1.04-x86_64.gz

### ap-southeast-2, アジアパシフィック ( シドニー ) リージョン

イメージ ID	イメージ名
aki-cd62fff7	pv-grub-hd0_1.04-i386.gz
aki-c362fff9	pv-grub-hd0_1.04-x86_64.gz
aki-c162fffb	pv-grub-hd00_1.04-i386.gz
aki-3b1d8001	pv-grub-hd00_1.04-x86_64.gz

eu-west-1, 欧州 ( アイルランド ) リージョン

イメージ ID	イメージ名
aki-68a3451f	pv-grub-hd0_1.04-i386.gz
aki-52a34525	pv-grub-hd0_1.04-x86_64.gz
aki-5ea34529	pv-grub-hd00_1.04-i386.gz
aki-58a3452f	pv-grub-hd00_1.04-x86_64.gz

sa-east-1, 南米 ( サンパウロ ) リージョン

イメージ ID	イメージ名
aki-5b53f446	pv-grub-hd0_1.04-i386.gz
aki-5553f448	pv-grub-hd0_1.04-x86_64.gz
aki-5753f44a	pv-grub-hd00_1.04-i386.gz
aki-5153f44c	pv-grub-hd00_1.04-x86_64.gz

us-east-1, 米国東部 ( バージニア北部 ) リージョン

イメージ ID	イメージ名
aki-8f9dcae6	pv-grub-hd0_1.04-i386.gz
aki-919dcaf8	pv-grub-hd0_1.04-x86_64.gz
aki-659ccb0c	pv-grub-hd00_1.04-i386.gz
aki-499ccb20	pv-grub-hd00_1.04-x86_64.gz

us-gov-west-1, AWS GovCloud ( 米国 )

イメージ ID	イメージ名
aki-1fe98d3c	pv-grub-hd0_1.04-i386.gz
aki-1de98d3e	pv-grub-hd0_1.04-x86_64.gz
aki-63e98d40	pv-grub-hd00_1.04-i386.gz
aki-61e98d42	pv-grub-hd00_1.04-x86_64.gz

us-west-1, 米国西部 ( 北カリフォルニア ) リージョン

イメージ ID	イメージ名
aki-8e0531cb	pv-grub-hd0_1.04-i386.gz
aki-880531cd	pv-grub-hd0_1.04-x86_64.gz
aki-960531d3	pv-grub-hd00_1.04-i386.gz

イメージ ID	イメージ名
aki-920531d7	pv-grub-hd00_1.04-x86_64.gz

us-west-2, 米国西部 ( オレゴン ) リージョン

イメージ ID	イメージ名
aki-f08f11c0	pv-grub-hd0_1.04-i386.gz
aki-fc8f11cc	pv-grub-hd0_1.04-x86_64.gz
aki-e28f11d2	pv-grub-hd00_1.04-i386.gz
aki-e68f11d6	pv-grub-hd00_1.04-x86_64.gz

## PV-GRUB の更新

すべてのバージョンの PV-GRUB AKI がすべてのインスタンスタイプと互換性があるとは限らないため、常に最新バージョンの PV-GRUB AKI を使用することをお勧めします。また、古いバージョンの PV-GRUB はすべてのリージョンで使用できるわけではないため、旧バージョンを使用する AMI を、そのバージョンをサポートしないリージョンにコピーした場合、カーネルのイメージを更新するまで、その AMI から起動されたインスタンスを起動できなくなります。次の手順を使用してインスタンスの PV-GRUB のバージョンを確認し、必要に応じて更新します。

PV-GRUB のバージョンを確認するには

1. インスタンスのカーネル ID を見つけます。

```
$ ec2-describe-instance-attribute instance_id --kernel --region region
kernel instance_id aki-fc8f11cc
```

このインスタンスのカーネル ID は、aki-fc8f11cc です。

2. このカーネル ID のバージョン情報を表示します。

```
$ ec2-describe-images aki-fc8f11cc --region region
IMAGE aki-fc8f11cc amazon/pv-grub-hd0_1.04-x86_64.gz ...
```

このカーネルイメージは PV-GRUB 1.04 です。PV-GRUB のバージョンが最新バージョン ( [Amazon PV-GRUB カーネルイメージ ID \(p. 104\)](#) を参照 ) でない場合、次の手順を使用して更新する必要があります。

PV-GRUB のバージョンを更新するには

インスタンスが古いバージョンの PV-GRUB を使用している場合は、最新バージョンに更新する必要があります。

1. [Amazon PV-GRUB カーネルイメージ ID \(p. 104\)](#) で、使用するリージョンとプロセッサアーキテクチャーの最新の PV-GRUB AKI を特定します。
2. インスタンスを停止します。使用されるカーネルイメージを変更するには、インスタンスを停止する必要があります。

```
$ ec2-stop-instances instance_id --region region  
INSTANCE instance_id stopped stopped
```

3. インスタンスに使用するカーネルイメージを変更します。

```
$ ec2-modify-instance-attribute --kernel kernel_id --region region instance_id
```

4. インスタンスを再起動します。

```
$ ec2-start-instances --region region instance_id
```

# Amazon EC2 インスタンス

---

## Abstract

インスタンスの設定、起動、接続、および管理。

Amazon EC2 を初めて使用する場合は、次のトピックを参照して使用を開始してください。

- [Amazon EC2 とは \(p. 1\)](#)
- [Amazon EC2 でのセットアップ \(p. 21\)](#)
- [Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#)
- [Amazon EC2 Windows インスタンスの使用開始](#)
- [インスタンスのライフサイクル \(p. 321\)](#)

実稼働環境を起動する前に、以下の質問に答える必要があります。

- Q: ニーズに最も合っている購入オプションはどれですか。  
Amazon EC2 はオンデマンドインスタンス ( デフォルト )、[スポットインスタンス \(p. 134\)](#)、および [リザーブドインスタンス \(p. 218\)](#) をサポートしています。詳細については、[Amazon EC2 料金表](#) を参照してください。
- Q: ニーズに最も合っているインスタンスタイプはどれですか。  
Amazon EC2 には、アプリケーションを実行するために必要な CPU、メモリ、ストレージ、ネットワークキャパシティを選択できるようにするため、さまざまなインスタンスタイプが用意されています。詳細については、「[インスタンスタイプ \(p. 109\)](#)」を参照してください。
- Q: ニーズに合っているルートボリュームのタイプはどれですか。  
各インスタンスは Amazon EBS またはインスタンスストアによってサポートされています。必要なルートボリュームのタイプに基づいて AMI を選択します。詳細については、「[ルートデバイスのストレージ \(p. 57\)](#)」を参照してください。
- Q: 仮想プライベートクラウドを使用するとメリットがありますか。  
EC2-Classic または EC2-VPC でインスタンスを起動できる場合は、どちらのプラットフォームがニーズに合っているかを判断する必要があります。詳細については、「[サポートされているプラットフォーム \(p. 527\)](#)」および「[Amazon EC2 と Amazon Virtual Private Cloud \( VPC \) \(p. 524\)](#)」を参照してください。

# インスタンスタイプ

## Abstract

Amazon EC2 では、コンピューティング、メモリ、ストレージ、ネットワークの能力が異なるさまざまなインスタンスタイプが提供されています。

インスタンスを起動するときは、指定したインスタンスタイプによって、インスタンスに使用するホストコンピュータのハードウェアが決まります。インスタンスタイプごとに、コンピューティング、メモリ、ストレージの能力が異なります。インスタンスタイプは、インスタンス上で実行するアプリケーションやソフトウェアの要件に基づいて選択します。

Amazon EC2 は、基になっているハードウェアに関係なく、各インスタンスに、一貫した予測可能な CPU 能力を提供します。

Amazon EC2 は、CPU、メモリ、インスタンスストレージなどのホストコンピュータの一部のリソースを、特定のインスタンス専用に割り当てます。ネットワークやディスクサブシステムなどの他のホストコンピュータリソースは、インスタンス間で共有されます。ホストコンピュータの各インスタンスが、これらの共有リソースの1つを可能な限り利用しようとする場合、それぞれのインスタンスは、そのリソースの共有分を等しく受け取ります。ただし、リソースの使用率が低い場合は、1つのインスタンスがそのリソースを多く使用できます。

各インスタンスタイプは、共有リソースからより高い、またはより低い最小性能を提供します。例えば、高速の I/O パフォーマンスを実行するインスタンスタイプは、共有リソースに対してより大きな割り当てを取得します。共有リソースをより大きく配分することによって、I/O 性能のばらつきを抑えることもできます。ほとんどのアプリケーションでは、中程度の I/O 性能があれば十分です。ただし、より高い、またはより一貫した I/O パフォーマンスを必要とするアプリケーションの場合は、より I/O パフォーマンスの高いインスタンスタイプを使用することを検討してください。

インスタンスの最大トランスミッション単位 (MTU) はインスタンスタイプに依存します。CC2、C3、R3、CG1、CR1、G2、HS1、H11、I2、M3 の各インスタンスタイプは、9001 MTU (ジャンボフレーム) を提供します。他のインスタンスタイプは、1500 MTU (Ethernet v2 フレーム) を提供します。

Amazon EBS I/O 用にさらに多くの専用容量を取得するには、一部のインスタンスタイプを Amazon EBS 最適化インスタンスとして起動できます。詳細については、[Amazon EBS 最適化インスタンス \(p. 127\)](#) を参照してください。

高パフォーマンスコンピューティング (HPC) アプリケーション用にインスタンスを最適化するには、一部のインスタンスタイプをプレイズメントグループで起動できます。詳細については、[プレイズメントグループ \(p. 128\)](#) を参照してください。

インスタンスを起動すると、準仮想 (PV) とハードウェア仮想マシン (HVM) の2種類のうち1つが使用されます。仮想化タイプは、インスタンスの起動に使用される AMI によって決まります。インスタンスタイプには、PV と HVM の両方をサポートしているものと、どちらか一方のみをサポートしているものもあります。HVM 仮想化は、AWS プラットフォームによって提供されるハードウェアアシストテクノロジーを使用します。HVM 仮想化を使用すると、ゲスト VM はネイティブハードウェアプラットフォーム上で動作しているかのように動作します。ただし、パフォーマンスの向上のために PV ネットワークとストレージドライバは使用します。

## 利用可能なインスタンスタイプ

Amazon EC2 では、次の表で示すインスタンスタイプが提供されます。

インスタンスファミリー	インスタンスタイプ
汎用	m1.small   m1.medium   m1.large   m1.xlarge   m3.medium   m3.large   m3.xlarge   m3.2xlarge

インスタンスファミリー	インスタンスタイプ
コンピューティングの最適化	c1.medium   c1.xlarge   c3.large   c3.xlarge   c3.2xlarge   c3.4xlarge   c3.8xlarge   cc2.8xlarge
メモリを最適化	m2.xlarge   m2.2xlarge   m2.4xlarge   r3.large   r3.xlarge   r3.2xlarge   r3.4xlarge   r3.8xlarge   cr1.8xlarge
ストレージの最適化	hi1.4xlarge   hs1.8xlarge   i2.xlarge   i2.2xlarge   i2.4xlarge   i2.8xlarge
マイクロインスタンス	t1.micro
GPU インスタンス	cg1.4xlarge   g2.2xlarge

## ハードウェア仕様

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Instance Type Details](#)」を参照してください。

お客様のニーズに最適なインスタンスタイプを決定するには、インスタンスを起動し、独自のベンチマークアプリケーションを使用することをお勧めします。支払いはインスタンス時間単位なので、決定する前に複数のインスタンスタイプをテストすると、便利なおうえ、コストを抑えることができます。ニーズが変化した場合は、後でインスタンスのサイズを変更できます。詳細については、[インスタンスのサイズ変更 \(p. 131\)](#) を参照してください。

## マイクロインスタンス

### Abstract

マイクロインスタンスは、非常に低コストなオプションで、小規模なコンピューティングリソースを提供します。

マイクロインスタンス (t1.micro) は、少量かつ一定量の CPU リソースを提供し、追加サイクルが利用可能であるときは、CPU 処理能力を短期バーストとして増大させることができます。このタイプが適しているのは、低スループットのアプリケーションやウェブサイトが定期的に追加計算サイクルを必要とする場合です。

t1.micro インスタンスは、Amazon EBS-Backed インスタンスとしてのみ利用できます。

このドキュメントでは、t1.micro インスタンスの動作とその適用方法について説明します。ここでは、特定の動作の詳細を説明するのではなく、インスタンスの動作の様子を描いて示すことを重視します。インスタンスの動作が理解できれば、マルチテナント型共有ウェブホストシステムなどの場合に比べて、パフォーマンスをより正確に掌握することができます。

### Topics

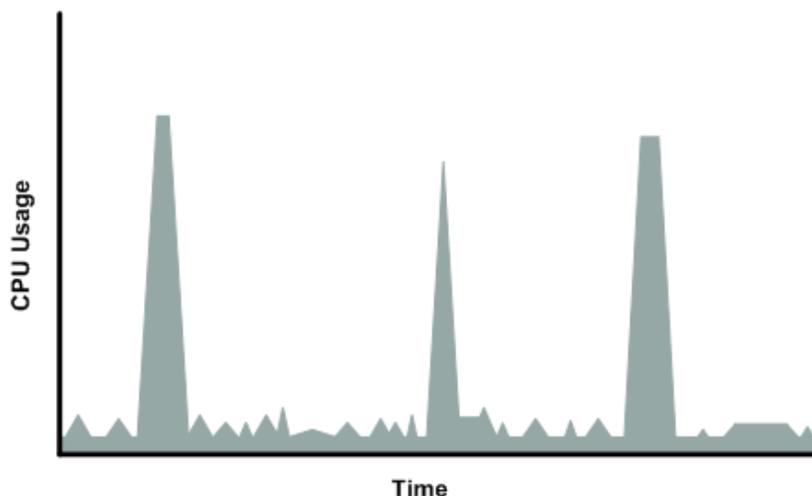
- [ハードウェア仕様 \(p. 111\)](#)
- [マイクロインスタンスの最適なアプリケーション \(p. 111\)](#)
- [スパイク中に利用可能な CPU リソース \(p. 113\)](#)
- [インスタンスが割り当てられたリソースを使用する場合 \(p. 113\)](#)
- [m1.small インスタンスタイプとの比較 \(p. 115\)](#)
- [マイクロインスタンスの AMI 最適化 \(p. 117\)](#)

## ハードウェア仕様

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Instance Type Details](#)」を参照してください。

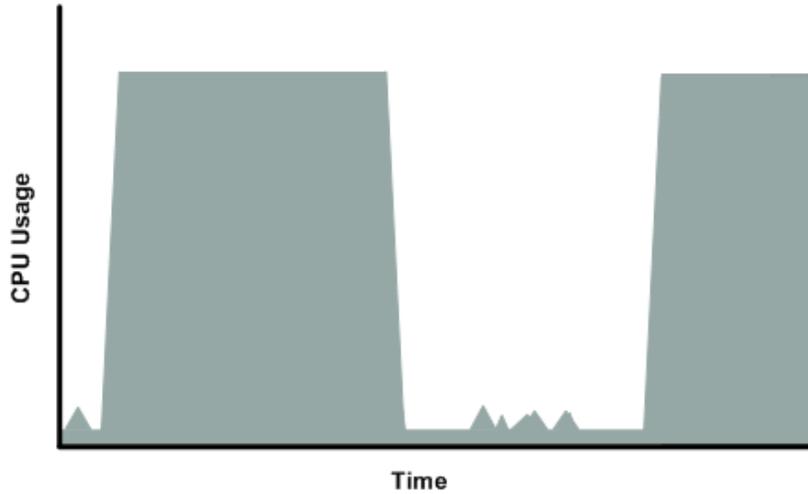
## マイクロインスタンスの最適なアプリケーション

t1.micro インスタンスは、ワークロードの CPU 使用率のスパイク期間 ( 次の図を参照 ) に対応するための CPU リソースを提供します。

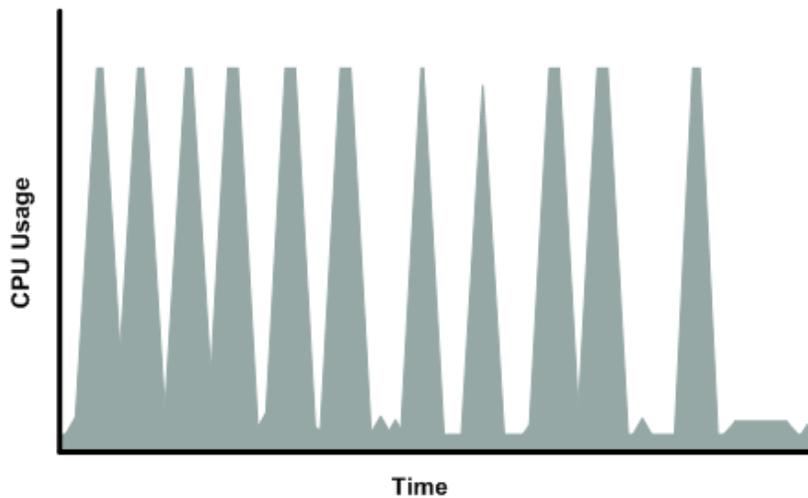


インスタンスは、2つの基本レベルの CPU 使用率のみで機能するように設計されています。1つは通常の低バックグラウンドレベルで、もう1つはバックグラウンドレベルよりはるかに高い短期スパイクレベルです。インスタンスは、最大2つの EC2 コンピューティングユニット ( ECU ) で動作することが許可されています ( 1つの ECU が 1.0~1.2 GHz 2007 Opteron、または 2007 Xeon プロセッサと同等の CPU 能力を提供 )。最大レベルとバックグラウンドレベルとの比率は大きくなるように設計されています。t1.micro インスタンスは、アプリケーションで1分あたり数十のリクエストをサポートするように設計されています。ただし実際のパフォーマンスは、アプリケーションで申請された各リクエストが必要とする CPU リソースの量に応じて、大幅に変わる可能性があります。

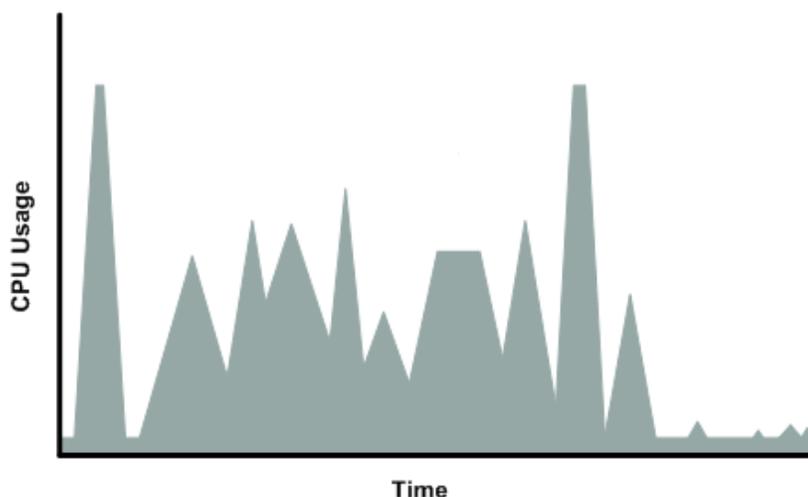
アプリケーションの CPU 使用率プロファイルは、上記の図に示すプロファイルと同じとは限りません。次の図は、t1.micro インスタンスに適していないアプリケーションのプロファイルを表示しています。アプリケーションには、リクエストごとに連続したデータ処理 CPU リソースが必要であるため、CPU 使用率が長時間にわたって高レベルを維持する結果となり、t1.micro インスタンスでは対処できません。



次の図は、`t1.micro` インスタンスに不適切なもう1つのプロファイルを示しています。ここではCPU利用率のスパイクの1つ1つは短期間ですが、マイクロインスタンスで対処できないほど頻発しています。



次の図は、`t1.micro` インスタンスに不適切なもう1つのプロファイルを示しています。ここではスパイクは頻発していませんが、スパイク間のバックグラウンドレベルが `t1.micro` インスタンスで対処できないほど高くなっています。



t1.micro インスタンスに適切ではない前記の各ワークロードのケースでは、異なるインスタンスタイプのご検討いただくことをお勧めします。インスタンスタイプの詳細については、「[インスタンスタイプ](#) (p. 109)」を参照してください。

## スパイク中に利用可能な CPU リソース

計算リソースの要求のスパイクに対応するためにインスタンスがバーストすると、ホスト上の未使用のリソースが使用されます。利用可能な量は、スパイク発生時の競合の程度に応じて異なります。ホスト上の他のインスタンスがスパイク中であるかどうかに関係なく、インスタンスには必ず CPU リソースが適用されます。

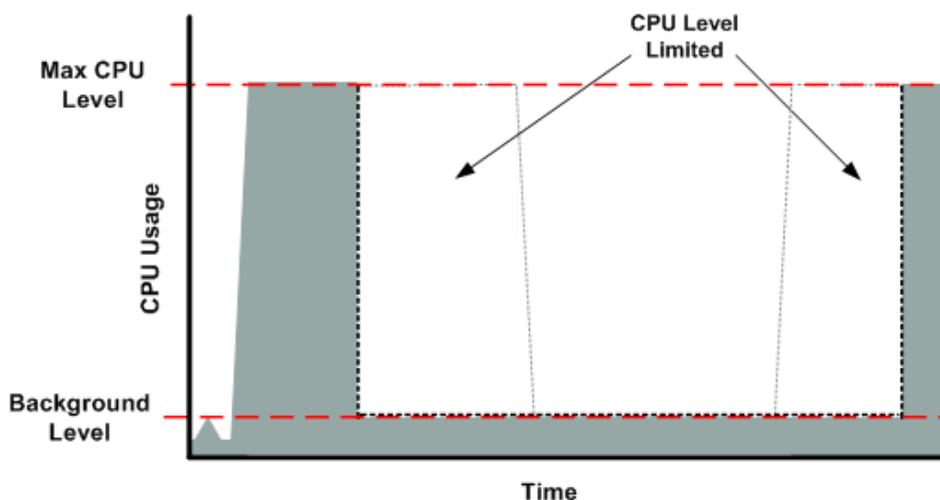
## インスタンスが割り当てられたリソースを使用する場合

ある期間内にはアプリケーションは CPU リソースの一部のみを使用するものと想定しています。アプリケーションが、インスタンスに割り当てられた CPU リソースを超える量を使用する場合は、CPU の使用レベルが低レベルになるように、一時的にインスタンスを制限します。インスタンスが割り当てられたすべてのリソースを使用し続けると、パフォーマンスが低下します。CPU の使用レベルを制限する時間を延長することで、インスタンスの再バーストが許可されるまでの時間を長くします。

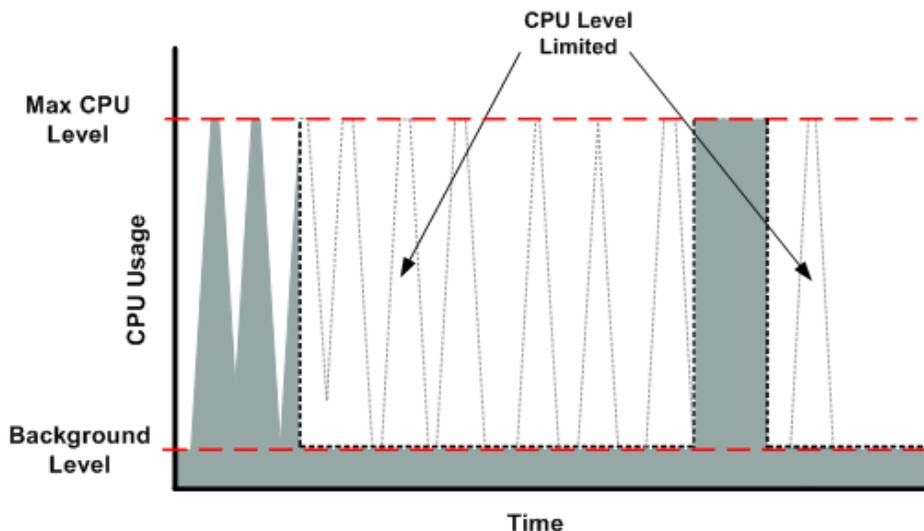
t1.micro インスタンスの CloudWatch モニタリングを有効にすると、AWS Management Console で「CPU 平均利用率」グラフを使用して、インスタンスが割り当てられているすべての CPU リソースを定期的に使用しているかどうかを判別できます。各期間の最大値を確認することをお勧めします。最大値が 100% である場合は、Auto Scaling を使用してスケールアウトする (t1.micro インスタンスを追加し、ロードバランサーを使用する) が、またはより大きいサイズのインスタンスタイプに移行することをお勧めします。詳細については、[Auto Scaling 開発者ガイド](#) を参照してください。

次の図は、前のセクションで使用した十分最適化されていない 3 つのプロファイルを示しており、インスタンスが割り当てられたリソースをすべて使用するため、その CPU レベルを制限しなければならない例です。インスタンスが割り当てられたリソースをすべて使用する場合は、そのレベルを低バックグラウンドレベルに制限します。

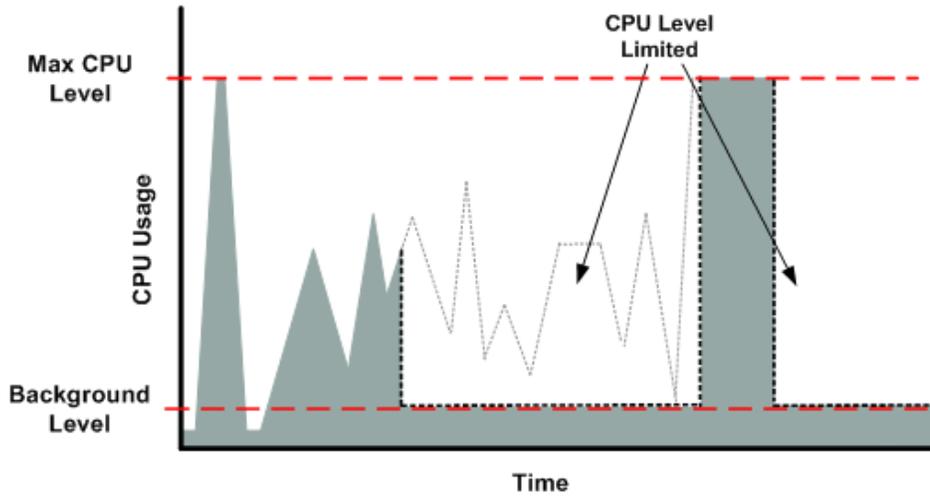
次の図は、データ処理 CPU 使用率が長時間にわたり高レベルを維持する場合を示しています。CPU は許可される最大レベルに達し、その期間にインスタンスに割り当てられているリソースがすべて使用されるまで、そのレベルを維持しています。その時点で、低バックグラウンドレベルで機能するようにインスタンスを制限します。そのレベルより高いレベルまでバーストすることを許可するまで、低レベルが維持されます。インスタンスは、割り当てられているリソースをすべて使用し、制限されるまで、そのレベルを維持します (グラフには示されていません)。



次の図では、要求が頻繁すぎる状況を示します。このインスタンスでは、ごくわずかのリクエストが出されただけで割り当てソースが使い尽くされ、CPU レベルが制限されます。制限を解除すると、インスタンスはリクエストの処理に追いつくように CPU 使用率を最大限にするため、再度インスタンスを制限します。

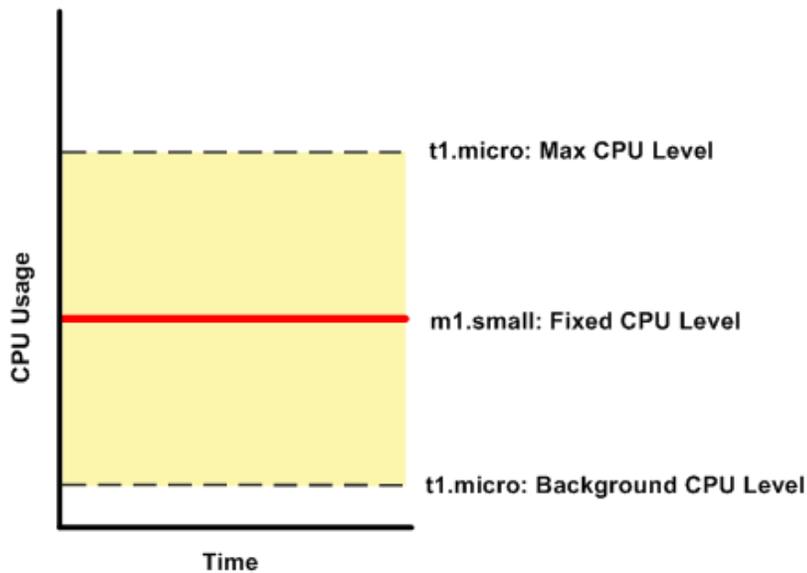


次の図では、バックグラウンドレベルが高すぎる状況を示します。インスタンスは、制限されることを目的として、最大 CPU レベルで機能し続けなければならないわけではありません。インスタンスを制限するのは、通常のバックグラウンドレベルを上回るレベルで機能し続け、割り当てられているリソースが一定期間にすべて使用される場合です。このような場合も（前の例を参照）、インスタンスの作業に遅れが出るため、インスタンスを制限します。



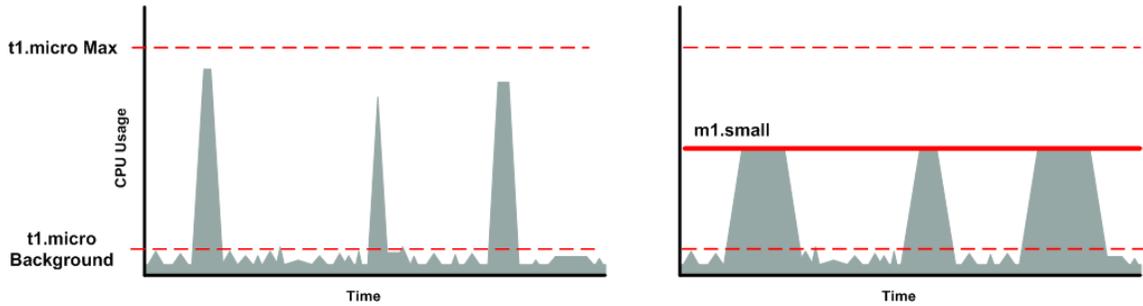
## m1.small インスタンスタイプとの比較

t1.micro インスタンスは、時間によって異なるレベルの CPU リソースを提供します (最大 2 ECU)。比較すると、m1.small インスタンスタイプは常時 1 つの ECU を提供します。次の図は、相違点を示しています。

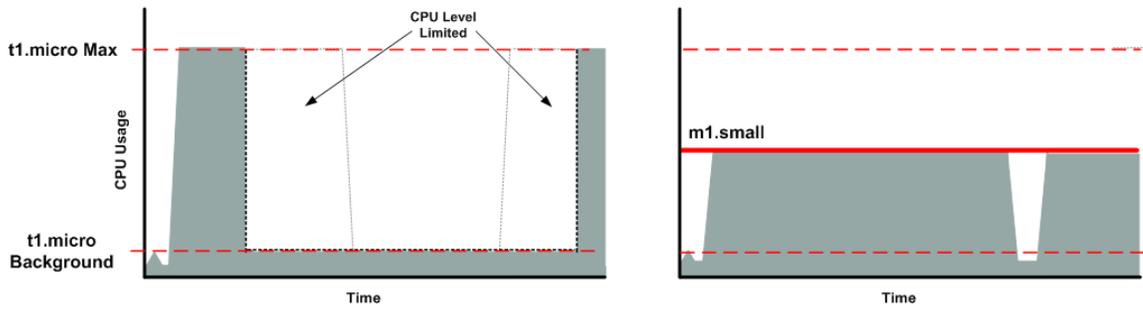


次の図では、t1.micro インスタンスの CPU 使用率を、これまでのセクションで説明したさまざまなシナリオの m1.small インスタンスと比較しています。

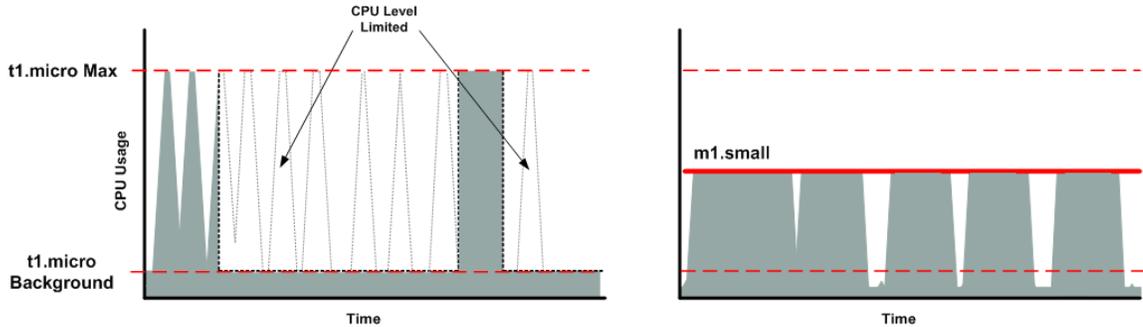
最初の図は、t1.micro インスタンスの最適なシナリオ (左側のグラフ) と、m1.small インスタンスの場合の状態 (右側のグラフ) を示しています。この場合は、t1.micro インスタンスを制限する必要はありません。m1.small インスタンスでの処理時間は、t1.micro インスタンスと比較すると、CPU 要求のスパイクごとに長くなっています。



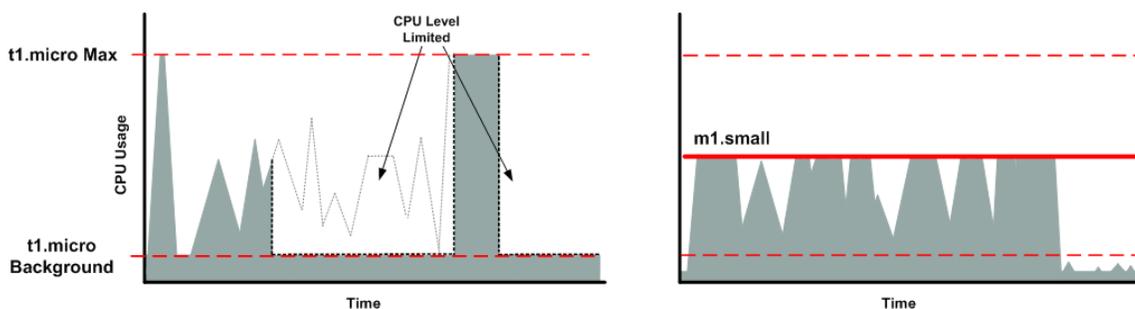
次の図は、t1.micro インスタンスに割り当てられたリソースがすべて使い切られたためにデータ処理リクエストが出されたというシナリオと、m1.small インスタンスの場合の状態を示しています。



次の図は、データ処理リクエストが頻繁に出されたため t1.micro インスタンスの割り当てリソースがすべて使い果たされたというシナリオと、m1.small インスタンスの場合の状態を示しています。



次の図は、バックグラウンドレベルで t1.micro インスタンスに割り当てられたリソースがすべて使い切られた状況と、m1.small インスタンスの場合の状態を示しています。



## マイクロインスタンスの AMI 最適化

t1.micro インスタンスタイプに合わせて AMI を最適化するときは、以下のベストプラクティスにしたがうことをお勧めします。

- 600 MB の RAM で実行されるように AMI を設計する
- CPU 時間を使用する定期プロセス ( 例: cron ジョブ、デーモン ) の数を制限する

Linux の場合は、スワップ領域と仮想メモリを使用してパフォーマンスを最適化できます ( 例えば、ルートファイルシステムから個別のパーティションでスワップ領域を設定できます )。

Windows の場合は、AMI またはインスタンスの重要な構成変更を行う ( 例: サーバーの役割を有効にする、サイズの大きいアプリケーションをインストールする ) ときに、インスタンスのパフォーマンスが制限されることがあります。これは、このような変更ではメモリの使用量が増加して、CPU リソースを長時間実行しなければならない可能性があるためです。AMI にこのような変更を加えるときには、まず、より大きいインスタンスタイプを使用し、次に、通常のオペレーション用の t1.micro インスタンスで AMI を実行することをお勧めします。

## I2 インスタンス

### Abstract

I2 インスタンスは、非常に高速な SSD-Backed インスタンスストレージを提供します。このインスタンスは、非常に高いランダム I/O パフォーマンスと IOPS あたりの安価なコストに最適化されています。

I2 インスタンスは、数万回の低レイテンシーとランダム I/O オペレーション/秒 ( IOPS ) をアプリケーションに提供するように最適化されています。このインスタンスは、以下のシナリオに最適です。

- NoSQL データベース ( Cassandra や MongoDB など )
- クラスター化データベース
- オンライントランザクション処理 ( OLTP ) システム

I2 インスタンスをプレイズメントグループにクラスター化できます。クラスタープレイズメントグループは、1つのアベイラビリティゾーン内で、インスタンス間の低レイテンシーで高帯域幅の接続を提供します。詳細については、[プレイズメントグループ \(p. 128\)](#) を参照してください。

I2 インスタンス用の拡張ネットワーク機能を実効にできます。詳細については、[VPC 内の Linux インスタンスでの拡張ネットワークの有効化 \(p. 559\)](#) または [Enabling Enhanced Networking on Windows Instances in a VPC](#) を参照してください。

### Topics

- ハードウェア仕様 (p. 118)
- I2 インスタンスの制限 (p. 118)
- SSD ストレージ (p. 118)
- SSD I/O パフォーマンス (p. 118)

## ハードウェア仕様

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Instance Type Details](#)」を参照してください。

## I2 インスタンスの制限

I2 インスタンスは、HVM AMI を使用して起動する必要があります。Amazon Linux、Ubuntu、Red Hat Enterprise Linux、および SUSE Enterprise Linux の各 Linux ディストリビューションがサポートされています。Microsoft Windows Server 2012 および Microsoft Windows Server 2008 R2 の各バージョンの Windows がサポートされています。その他のオペレーティングシステムで HVM AMI の使用をご希望の場合は、[Amazon EC2 forum](#) に投稿してお知らせください。

実行できる I2 インスタンスの数は制限されています。次の表に示すデフォルトの制限よりも多くの I2 インスタンスが必要な場合は、[Amazon EC2 インスタンス申請フォーム](#)を使用して、I2 インスタンスの増加を申請できます。

インスタンスサイズ	デフォルトのインスタンス制限
i2.xlarge	8
i2.2xlarge	8
i2.4xlarge	4
i2.8xlarge	2

## SSD ストレージ

I2 インスタンスのプライマリデータストレージは、SSD-Backed インスタンスストレージです。すべてのインスタンスストレージのように、これらのボリュームはインスタンスの運用期間中のみ永続化されます。インスタンスを終了すると、アプリケーションとそのインスタンスストアのデータは消去されます。Amazon EBS-Backed AMI を使用するときは、インスタンスを開始して停止できます。ただし、インスタンスを停止すると、Amazon EBS ボリュームに保存されたデータは永続化されますが、インスタンスストアボリュームのデータは永続化されません。インスタンスストレージに保存したデータは、定期的にバックアップするかレプリケートすることをお勧めします。

インスタンスストアボリュームの詳細については、[Amazon EC2 インスタンスストア \(p. 631\)](#) を参照してください。

## SSD I/O パフォーマンス

I2 インスタンスから最高の IOPS 性能を得るには、最新バージョンの [Amazon Linux AMI](#)、またはカーネルバージョン 3.8 以降の別の Linux AMI を使用することをお勧めします。カーネルバージョン 3.8 以降の Linux AMI を使用して、インスタンスで利用できるすべての SSD-Backed インスタンスストアボリュームを利用する場合は、少なくとも次の表に一覧表示されている最小のランダム IOPS (4,096 バイトブロックサイズ) を取得できます。それ以外の場合は、IOPS 性能は表に示されているよりも低くなります。

インスタンスサイズ	読み取り IOPS	最初の書き込み IOPS
i2.xlarge	35,000	35,000
i2.2xlarge	75,000	75,000
i2.4xlarge	175,000	155,000
i2.8xlarge	365,000	315,000

インスタンス用に SSD ベースのインスタンスストレージを使用するにつれて、アーカイブできる書き込み IOPS の数は減ります。これは、SSD コントローラーが実行する必要がある追加の作業が原因です。SSD コントローラーは、利用可能な領域を見つけ、既存のデータを再書き込みし、未使用の領域を消去して、再書き込みができるようにします。このガベージコレクションというプロセスにより、SSD への内部的な書き込み増幅が発生し、ユーザーの書き込み操作に対する SSD 書き込み操作の割合として表示されます。書き込み操作が 4,096 バイトの倍数でないか、4,096 バイトの境界に整合していない場合、パフォーマンスの低下はさらに大きくなります。少量のバイト数または整合していないバイト数で書き込む場合、SSD コントローラーは周辺のデータを読み取り、その結果を新しい場所に保存する必要があります。このパターンにより、書き込み増幅が大幅に増え、レイテンシーが増加し、I/O パフォーマンスが大きく低下します。

SSD コントローラーは、複数の方法を利用すると、書き込み増幅の影響を減らすことができます。このような方法の 1 つには、SSD インスタンスストレージに領域を予約し、コントローラーが書き込み操作に利用できる領域をより効率的に管理できるようにすることです。これを **オーバープロビジョニング** と呼びます。I2 インスタンスに提供された SSD ベースのインスタンスストアボリュームには、オーバープロビジョニングに対して予約された領域がありません。書き込み増幅を減らすには、ボリュームの 10% を未使用の状態のままにし、SSD コントローラーがこれをオーバープロビジョニングに使用できるようにします。これにより、使用できるストレージが減りますが、パフォーマンスは向上します。

また、TRIM コマンドを使用して、書き込んだデータが不要になるたびに SSD コントローラーに通知することもできます。これにより、より多くの空き領域がコントローラーに与えられ、その結果書き込み増幅が減り、パフォーマンスが向上します。TRIM コマンドの使用の詳細については、インスタンス用のオペレーティングシステムのドキュメントを参照してください。

## HI1 インスタンス

### Abstract

HI1 インスタンスは、非常に高速の SSD-Backed インスタンスストレージを提供します。このインスタンスストレージは、非常に高いランダム I/O パフォーマンスと、IOPS あたりの安価なコストに最適化されています。

HI1 I/O インスタンス ( `hi1.4xlarge` ) は、数万回の低レイテンシーとランダム I/O オペレーション/秒 ( IOPS ) をアプリケーションに提供します。このインスタンスは、以下のシナリオに最適です。

- NoSQL データベース ( Cassandra や MongoDB など )
- クラスター化データベース
- オンラインランザクション処理 ( OLTP ) システム

HI1 インスタンスは、プレイズメントグループでクラスター化できます。詳細については、[プレイズメントグループ \(p. 128\)](#) を参照してください。

デフォルトでは、最大 2 つのインスタンスを実行できます。hi1.4xlargehi1.4xlarge インスタンスが 2 つ以上必要な場合は、[Amazon EC2 Instance Request Form](#) を使用して、インスタンスの追加を申請できます。

### Topics

- [ハードウェア仕様 \(p. 120\)](#)
- [ディスク I/O パフォーマンス \(p. 120\)](#)
- [SSD ストレージ \(p. 120\)](#)

## ハードウェア仕様

hi1.4xlarge インスタンスタイプは、Solid State Drive ( SSD ) テクノロジーに基づいています。

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Instance Type Details](#)」を参照してください。

## ディスク I/O パフォーマンス

Linux 準仮想 ( PV ) AMI を使用すると、H1 I/O インスタンスは、4 KB ランダム読み取りについては 120,000 IOPS 以上、4 KB ランダム書き込みについては 10,000 ~ 85,000 IOPS ( アクティブな論理ブロックアドレス指定範囲によって異なる ) を、2 つの SSD データボリューム全体でアプリケーションに提供できます。ハードウェア仮想マシン ( HVM ) と Windows の AMI でのパフォーマンスは、4 KB ランダム読み取りについては約 90,000 IOPS、4 KB ランダム書き込みについては 9,000 ~ 75,000 IOPS となります。

最大シーケンシャルスループットは、どの AMI タイプでも ( Linux PV、Linux HVM、Windows )、読み取りについては 2 GB/秒程度、書き込みについては 1.1 GB/秒程度です。

## SSD ストレージ

このセクションでは、SSD ストレージについて知っておく必要がある重要な情報を示します。SSD ストレージ

- 主要なデータソースは、SSD ストレージを含むインスタンスストアです。
- 読み取りパフォーマンスは一貫し、書き込みパフォーマンスは多様です。
- 書き込み増幅が発生する可能性があります。
- TRIM コマンドは現在サポートされていません。

## SSD ストレージによるインスタンスストア

hi1.4xlarge インスタンスは Amazon EBS-Backed ルートデバイスを使用できます。ただし、その主要なデータストレージは、インスタンスストア内の SSD ボリュームによって提供されます。他のインスタンスストアボリュームと同様に、これらのインスタンスストアボリュームはインスタンスの運用期間中のみ永続化されます。hi1.4xlarge インスタンスのルートデバイスは Amazon EBS-Backed であるため、インスタンスを起動/停止できます。インスタンスを停止すると、アプリケーションは永続化されませんが、インスタンスストアの運用データは永続化されません。インスタンスストアボリュームの詳細については、「[Amazon EC2 インスタンスストア \(p. 631\)](#)」を参照してください。

## 多様な書き込みパフォーマンス

書き込みパフォーマンスは、アプリケーションがどのように論理ブロックアドレス指定 ( LBA ) 空間を指定するかによって異なります。アプリケーションが LBA 空間全体を使用している場合、書き込みパフォーマンスが約 90% 低下する可能性があります。アプリケーションを評価し、キューの深さ ( ボリュームに対する保留中の I/O リクエストの数 ) と I/O サイズを監視します。

## 書き込み増幅

書き込み増幅とは、フラッシュメモリと SSD に関連する望ましくない状況を指し、実際に書き込まれる物理的情報量が、意図した論理的書き込み量の倍数になることを意味します。フラッシュメモリは再

書き込みする前に消去する必要があるため、これらのオペレーションを実行するプロセスでは、ユーザーデータとメタデータを1回以上移動（または再書き込み）することになります。この増幅効果によって、SSDの運用期間中、必要な書き込みの数が増加し、信頼できる動作を実行する時間が短くなります。hi1.4xlarge インスタンスは、書き込み増幅を最小化することを目的とするプロビジョニングモデルを使用して設計されています。

ランダム書き込みは、シリアル書き込みよりも書き込み増幅に与える影響が大きくなります。書き込み増幅が懸念される場合は、ストレージの全テビバイト数よりも少ない容量をアプリケーションに割り当てます（オーバープロビジョニングとも呼ばれます）。

## TRIM コマンド

TRIM コマンドによって、オペレーティングシステムは、以前に保存されたデータのブロックが使用されなくなったと見なされることを SSD に通知できます。TRIM によって、書き込み増幅の影響が制限されます。

TRIM のサポートは HI1 インスタンスでは利用できません。TRIM のサポートには、I2 インスタンスを使用します。詳細については、[I2 インスタンス \(p. 117\)](#) を参照してください。

# HS1 インスタンス

## Abstract

HS1 インスタンスは、高密度ストレージ、低いストレージコスト、高シーケンシャル I/O パフォーマンスを提供します。

HS1 インスタンス (hs1.8xlarge) はストレージ密度がきわめて高く、インスタンスあたりの順次読み取り/書き込みのパフォーマンスの高さが特徴です。このインスタンスは、以下のシナリオに最適です。

- データウェアハウス
- Hadoop/MapReduce
- 並列ファイルシステム

HS1 インスタンスはプレイズメントグループでクラスター化できます。詳細については、[プレイズメントグループ \(p. 128\)](#) を参照してください。

デフォルトでは、最大2つのHS1インスタンスを実行できます。HS1インスタンスが2つ以上必要な場合は、[Amazon EC2 インスタンス申請フォーム](#)を使用して、インスタンスの追加を申請できます。

## Topics

- [ハードウェア仕様 \(p. 121\)](#)
- [ディスク I/O パフォーマンス \(p. 122\)](#)
- [ストレージ情報 \(p. 122\)](#)

## ハードウェア仕様

HS1 インスタンスは、Amazon Elastic Block Store (Amazon EBS) -Backed および Instance store-Backed Amazon マシンイメージ (AMI) の両方をサポートしています。HS1 インスタンスサポートは、準仮想化 (PV) およびハードウェア仮想マシン (HVM) AMI の両方をサポートしています。HS1 インスタンスは、ブロックサイズ 2 MiB を使用するとき、2.4 GB/秒の順次読み取りおよび 2.6 GB/秒の順次書き込みパフォーマンスを実現できます。

Microsoft Windows Server を使用する場合は、HS1 インスタンスがサポートされるのは、Microsoft Windows Server AMI for Cluster Instance Type のみとなります。HS1 インスタンスは現在 Amazon EBS

の最適化をサポートしていませんが、高帯域幅ネットワークを提供しており、Amazon EBS プロビジョニング IOPS ( I/O オペレーション/秒 ) ポリユームとともに使用して、一貫性とパフォーマンスの向上を図ることができます。

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Instance Type Details](#)」を参照してください。

## ディスク I/O パフォーマンス

HS1 インスタンスは、ブロックサイズ 2 MiB を使用するとき、2.6 GB/秒の順次読み取りおよび書き込みパフォーマンスを実現できます。

## ストレージ情報

このセクションでは、HS1 インスタンスと組み合わせて使用するストレージについて知っておく必要がある重要な情報を示します。

### インスタンスストアと HS1 インスタンス

HS1 インスタンスは、インスタンスストアと Amazon EBS ルートデバイスボリュームの両方をサポートしています。ただし、Amazon EBS-Backed インスタンスを使用する場合でも、主要なデータストレージは、インスタンスストア内のハードディスクドライブによって提供されます。他のインスタンスストアボリュームと同様に、これらのインスタンスストアボリュームはインスタンスの運用期間中のみ永続化されます。したがって、Amazon EBS-Backed ルートボリュームを使用するときインスタンスを停止すると、アプリケーションは維持されますが、インスタンスストア内の本番データは維持されません。インスタンスストアボリュームの詳細については、「[Amazon EC2 インスタンスストア \(p. 631\)](#)」を参照してください。

### ディスクの初期化

HS1 インスタンスを安定した状態で長期間使用する場合は、パフォーマンスの向上のために、まずハードディスクを初期化することをお勧めします。このプロセスの完了には6時間程度もかかる可能性があります。

### メモリ制限の設定

多くの場合、AMI ベースでの Linux では `CONFIG_XEN_MAX_DOMAIN_MEMORY` が 70 に設定されています。117 GiB のメモリに適合するように、この値を設定することをお勧めします。

### ユーザー制限 ( ulimit ) の設定

多くの場合、AMI ベースでの Linux はデフォルトの ulimit 1024 に設定されています。ulimit を 2048 に引き上げることをお勧めします。

## R3 インスタンス

### Abstract

優れたメモリパフォーマンスと、高持続性帯域幅を実現するには、R3 インスタンスを使用します。

優れたメモリパフォーマンスと、高持続性帯域幅を実現するには、R3 インスタンスが最適です。このインスタンスは、以下のシナリオに最適です。

- リレーショナルデータベースおよび NoSQL データベース (たとえば、MongoDB など)
- インメモリ分析
- Memcache/Redis アプリケーション (たとえば、Elasticache)

R3 インスタンスはプレイズメントグループでクラスター化できます。クラスタープレイズメントグループは、1つのアベイラビリティゾーン内で、インスタンス間の低レイテンシーで高帯域幅の接続を提供します。詳細については、[プレイズメントグループ \(p. 128\)](#) を参照してください。

R3 インスタンス用に拡張ネットワーク機能の有効にすることができます。詳細については、[VPC 内の Linux インスタンスでの拡張ネットワークの有効化 \(p. 559\)](#) または [Enabling Enhanced Networking on Windows Instances in a VPC](#) を参照してください。

#### Topics

- [ハードウェア仕様 \(p. 123\)](#)
- [オペレーティングシステムのサポート \(p. 123\)](#)
- [HVM AMI のサポート \(p. 123\)](#)
- [R3 インスタンスのデフォルトの制限 \(p. 123\)](#)
- [SSD ストレージ \(p. 124\)](#)
- [SSD I/O パフォーマンス \(p. 124\)](#)
- [TRIM サポート \(p. 124\)](#)

## ハードウェア仕様

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Instance Type Details](#)」を参照してください。

## オペレーティングシステムのサポート

次の Linux オペレーティングシステムをサポートします。Amazon Linux、Ubuntu、Red Hat Enterprise Linux、および SUSE Enterprise Linux。さらに、Microsoft Windows Server 2012 および Microsoft Server 2008 R2 AMI を使用して、R3 インスタンスを起動できます。その他のオペレーティングシステムで HVM AMI の使用をご希望の場合は、[Amazon EC2 forum](#) に投稿してお知らせください。

## HVM AMI のサポート

R3 インスタンスにはハイメモリ (最大 244 GiB の RAM) があり、その処理能力を活用するためには 64 ビットのオペレーティングシステムが必要です。HVM AMI は、ハイメモリインスタンスタイプの準仮想化 (PV) AMI よりも優れたパフォーマンスを提供します。これらの理由から、R3 インスタンスは 64 ビットの HVM AMI のみをサポートします。さらに、拡張ネットワークのメリットを活用するには、HVM AMI が必要です。

## R3 インスタンスのデフォルトの制限

同時に実行できる R3 インスタンスの数を制限しています。次の表に示すデフォルトの制限よりも多くの R3 インスタンスが必要な場合は、[Amazon EC2 インスタンス申請フォーム](#) を使用して、増加を申請できます。

インスタンスタイプ	デフォルトのインスタンス制限
r3.large	20
r3.xlarge	20
r3.2xlarge	20
r3.4xlarge	10
r3.8xlarge	5

## SSD ストレージ

R3 インスタンスのプライマリデータストレージは、SSD ベースのインスタンスストレージです。すべてのインスタンスストレージのように、これらのボリュームはインスタンスの運用期間中のみ永続化されます。インスタンスを終了すると、アプリケーションとそのインスタンスストアのデータは消去されます。Amazon EBS-Backed AMI を使用するときは、インスタンスを停止して、後で再起動することができます。ただし、インスタンスを停止すると、Amazon EBS ボリュームに保存されているデータは保持されますが、インスタンスストアボリュームのデータは保持されません。インスタンスストレージに保存したデータは、定期的にバックアップするかレプリケートすることをお勧めします。

インスタンスストアボリュームの詳細については、[Amazon EC2 インスタンスストア \(p. 631\)](#) を参照してください。

## SSD I/O パフォーマンス

最大の R3 インスタンス ( `r3.8xlarge` ) では、150,000 以上の 4 キロバイト ( KB ) ランダムリード IOPS および 130,000 以上の 4 KB ランダム初回書き込み IOPS が可能です。R3 インスタンスから最高の IOPS 性能を得るには、最新バージョンの [Amazon Linux AMI](#)、またはカーネルバージョン 3.8 以降の別の Linux AMI を使用することをお勧めします。カーネルバージョン 3.8 以降の Linux AMI を使用しないと、このインスタンスタイプで可能な最高の IOPS パフォーマンスを実現できません。

## TRIM サポート

また、TRIM コマンドを使用して、書き込んだデータが不要になるたびに SSD コントローラーに通知することもできます。これにより、より多くの空き領域がコントローラーに与えられ、その結果書き込み増幅が減り、パフォーマンスが向上します。TRIM コマンドの使用に関する詳細は、使用するインスタンスのオペレーティングシステム固有のドキュメントを参照してください。

## GPU インスタンス

### Abstract

GPU インスタンスは、グラフィックスや汎用的な GPU コンピューティングアプリケーションを対象としています。

高度な並列処理機能を必要とする場合は、GPU インスタンスを使用するのが適しています。GPU インスタンスは、最大 1,536 個の CUDA コアと 4 GB のビデオメモリを備える NVIDIA GPU にアクセスできます。GPU インスタンスを使用すると、Compute Unified Device Architecture ( CUDA ) または OpenCL パラレルコンピューティングフレームワークを活用することにより、多数のサイエンス、エンジニアリング、およびレンダリングアプリケーションを高速化することができます。また、ゲームストリーミング、3D アプリケーションストリーミング、およびその他のグラフィックスワークロードを含む、グラフィックアプリケーションにも使用できます。

GPU インスタンスは、HVM ベースのインスタンスとして実行されます。ハードウェア仮想マシン ( HVM ) 仮想化は、AWS プラットフォームによって提供されるハードウェアアシストテクノロジーを使用します。ゲスト VM は、性能の向上のために準仮想化 ( PV ) ネットワークおよびストレージドライバをまだ使用している点を除き、HVM 仮想化を使用して、ネイティブハードウェアプラットフォーム上で動作しているかのように動作します。これにより、Amazon EC2 は、各 GPU インスタンスで 1 つ以上の個別の GPU への専用アクセス権を提供できます。

GPU インスタンスをクラスタープレイスメントグループにクラスター化できます。クラスタープレイスメントグループは、1 つのアベイラビリティゾーン内で、インスタンス間の低レイテンシーで高帯域幅の接続を提供します。詳細については、[プレイスメントグループ \(p. 128\)](#) を参照してください。

### Topics

- [ハードウェア仕様 \(p. 125\)](#)

- [GPU インスタンスの制限 \(p. 125\)](#)
- [GPU インスタンスの AMI \(p. 125\)](#)
- [Linux での NVIDIA ドライバのインストール \(p. 125\)](#)
- [Windows での NVIDIA ドライバのインストール \(p. 126\)](#)

## ハードウェア仕様

各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「[Instance Type Details](#)」を参照してください。

## GPU インスタンスの制限

現在、GPU インスタンスには以下の制限があります。

- すべてのリージョンで利用できません。
- HVM AMI から起動する必要があります。
- NVIDIA ドライバがインストールされていない限り、GPU にアクセスできません。
- Amazon DevPay で使用することはできません。
- 実行できるインスタンス数は制限されています。詳細については、Amazon EC2 のよくある質問の [How many instances can I run in Amazon EC2?](#) を参照してください。これらの制限の拡大を申請するには、[Request to Increase Amazon EC2 Instance Limit](#) というフォームを使用してください。

## GPU インスタンスの AMI

NVIDIA では、すぐに使用を開始できるように、Amazon Linux および Windows 向けに GPU インスタンス用の AMI を提供しています。これらのリファレンス AMI には、NVIDIA GPU の完全な機能とパフォーマンスを有効にする NVIDIA ドライバが含まれています。NVIDIA ドライバがある AMI の一覧については、[AWS Marketplace \( NVIDIA GRID \)](#) を参照してください。

CG1 インスタンスは任意の HVM AMI を使用して起動できます。

Windows Server 2012 および Windows Server 2008 R2 AMI を使用して、G2 インスタンスを起動できます。さらに、Linux HVM AMI は、Amazon Linux、SUSE Enterprise Linux や Ubuntu の各オペレーティングシステムで起動できます。別のオペレーティングシステム用の AMI から G2 インスタンスを起動するときに次のエラーが発生した場合は、[カスタマーサービス](#)にお問い合わせいただくか、[Amazon EC2 forum](#) を通じてお問い合わせください。

```
Client.UnsupportedOperation: Instances of type 'g2.2xlarge' may not be launched from AMI <ami-id>.
```

G2 インスタンスを起動した後で、インスタンスから独自の AMI を作成できます。ただし、インスタンスのルートボリュームのスナップショットを作成する場合は、インスタンスを AMI として登録してから G2 インスタンスを起動すると、`Client.UnsupportedOperation` エラーが発生します。独自の AMI から G2 インスタンスを起動するには、コンソールを使用して G2 インスタンスから AMI を作成 ( インスタンスを選択して [Actions]、[Create Image] の順にクリック ) するか、[create-image](#) ( AWS CLI ) または [ec2-create-image](#) ( Amazon EC2 CLI ) を使用して G2 インスタンスから AMI を作成する必要があります。

## Linux での NVIDIA ドライバのインストール

GPU インスタンスには、適切な NVIDIA ドライバが必要です。インストールする NVIDIA ドライバは、インスタンスで実行するカーネルに対してコンパイルされる必要があります。

Amazon は、公式の各カーネル更新のために、NVIDIA カーネルドライバの互換性のある更新済みビルドを提供しています。Amazon が提供するバージョン以外の NVIDIA ドライババージョンを使用する場合、あるいは公式の Amazon ビルド以外のカーネルを使用する場合は、インストールするドライバのバージョンとの競合を回避するために、Amazon が提供する NVIDIA パッケージをシステムからアンインストールする必要があります。

このコマンドを使用して、Amazon が提供する NVIDIA パッケージをアンインストールします。

```
$ sudo yum erase nvidia cudatoolkit
```

Amazon が提供する CUDA ツールキットパッケージは、NVIDIA ドライブに依存します。NVIDIA パッケージをアンインストールすると、CUDA ツールキットが消去されます。NVIDIA ドライブをインストールした後に、CUDA ツールキットを再インストールする必要があります。

NVIDIA ドライブは、<http://www.nvidia.com/Download/Find.aspx> からダウンロードできます。NVIDIA GRID K520 ( G2 インスタンス )、または Linux 64 ビットシステム用の Tesla M-Class M2050 ( CG1 インスタンス ) のドライバを選択します。ドライバのインストールと設定の詳細については、NVIDIA ウェブサイトのドライバのダウンロードページで [ADDITIONAL INFORMATION] タブを開き、[README] リンクをクリックします。

## NVIDIA ドライバを手動でインストールします。

Amazon Linux AMI 用のドライバをインストールするには

1. `kernel-devel` パッケージがインストールされており、現在実行しているカーネルのバージョンと一致することを確認してください。

```
$ yum install kernel-devel-`uname -r`
```

2. NVIDIA ドライバをインストールするための自己インストールスクリプトを実行します。以下に例を示します。

```
$ /root/NVIDIA-Linux-x86_64_319.60.run
```

3. インスタンスを再起動します。詳細については、[インスタンスの再起動 \(p. 349\)](#) を参照してください。
4. ドライバが機能していることを確認します。次のコマンドのレスポンスに、インストールされた NVIDIA ドライババージョンおよび GPU に関する詳細が表示されます。

```
$ /usr/bin/nvidia-smi -q -a
```

## Windows での NVIDIA ドライバのインストール

Windows インスタンスに NVIDIA ドライバをインストールするには、リモートデスクトップを使用して、管理者としてインスタンスにログオンします。NVIDIA ドライバは、<http://www.nvidia.com/Download/Find.aspx> からダウンロードできます。使用しているバージョンの Windows Server 用の NVIDIA GRID K520 ( G2 インスタンス ) または Tesla M-Class M2050 ( CG1 インスタンス ) のドライバを選択します。ドライバをダウンロードしたフォルダを開き、インストールファイルをダブルクリックして起動します。ドライバをインストールする手順にしたがい、必要に応じてインスタンスを再起動します。GPU が正しく動作していることを確認するには、デバイスマネージャをチェックします。

リモートデスクトップを使用すると、WDDM ドライバモデルを使用している GPU は、高速化されていないリモートデスクトップディスプレイドライバに置き換えられます。GPU ハードウェアにアクセスするには、VNC などの異なるリモートアクセスツールを使用する必要があります。また、AWS Marketplace からいずれかの GPU AMI を使用することもできます。これは、3D 高速化をサポートするリモートアクセスツールを提供するためです。

## Amazon EBS 最適化インスタンス

### Abstract

Amazon EBS 最適化インスタンスを使用して、Amazon EBS ボリュームに最善のパフォーマンスを提供します。

Amazon EBS 最適化インスタンスは、最適化された設定スタックを使用し、追加の専用の容量を Amazon EBS I/O に提供します。この最適化により、Amazon EBS I/O とインスタンスからのその他のトラフィックの間での競合を最小化して、Amazon EBS ボリュームで最良のパフォーマンスが提供されます。

Amazon EBS 最適化インスタンスを使用すると、専用の容量について、安価な時間単位の料金を追加でお支払いいただきます。

Amazon EBS 最適化により、Amazon EBS ボリュームでプロビジョニングされた IOPS を、インスタンスで完全に利用することが可能になります。Amazon EBS 最適化インスタンスは、Amazon EBS に対して所定のスループットを実現するものであり、インスタンスタイプに応じて 500 ~ 2,000 Mbps の範囲内で選択できます。Amazon EBS 最適化インスタンスにアタッチした場合、Provisioned IOPS ボリュームは、プロビジョニングされた 10 パーセント内のパフォーマンスを、1 年の 99.9 パーセントの時間にわたり提供します。詳細については、[Provisioned IOPS ボリューム \(p. 569\)](#) を参照してください。

次の表では、Amazon EBS 最適化インスタンスとして起動できるインスタンスタイプ、インスタンスが提供する専用 Amazon EBS スループット、および Amazon EBS 接続で実行できる IOPS (1 秒あたりの入力/出力操作数) の最大値を示します。

インスタンスタイプ	専用 EBS スループット ( Mbps ) *	最大 16K IOPS**
c1.xlarge	1,000	8,000
c3.xlarge	500	4,000
c3.2xlarge	1,000	8,000
c3.4xlarge	2,000	16,000
g2.2xlarge	1,000	8,000
i2.xlarge	500	4,000
i2.2xlarge	1,000	8,000
i2.4xlarge	2,000	16,000
m1.large	500	4,000
m1.xlarge	1,000	8,000
m2.2xlarge	500	4,000
m2.4xlarge	1,000	8,000
m3.xlarge	500	4,000

インスタンスタイプ	専用 EBS スループット ( Mbps ) *	最大 16K IOPS**
m3.2xlarge	1,000	8,000
r3.xlarge	500	4,000
r3.2xlarge	1,000	8,000
r3.4xlarge	2,000	16,000

Amazon EBS 最適化インスタンスを起動するには、起動ウィザードで [Launch as EBS-optimized instance] オプションを選択します。選択したインスタンスタイプが Amazon EBS 最適化インスタンスとして起動できない場合、このオプションは使用できません。

AWS CLI を使用して Amazon EBS 最適化インスタンスを起動するには、`run-instances` コマンドを使用し、`--ebs-optimized` オプションを指定します。

Amazon EC2 CLI を使用して Amazon EBS 最適化インスタンスを起動するには、`ec2-run-instances` コマンドを使用し、`--ebs-optimized` オプションを指定します。

## プレイメントグループ

### Abstract

密結合ノード間通信には、単一のアベイラビリティゾーンを使ったインスタンスの論理グループである、プレイメントグループを使用します。

プレイメントグループは、単一のアベイラビリティゾーン内のインスタンスを論理的にグループ化したものです。プレイメントグループを使用すると、アプリケーションが低レイテンシーの 10 Gbps ネットワークに参加できるようになります。ネットワークレイテンシーが低い場合、ネットワークスループットが高い場合、またはその両方の場合にメリットがあるアプリケーションでは、プレイメントグループの使用をお勧めします。プレイメントグループで、最も低いレイテンシーと最も高いネットワークパフォーマンス (1 秒あたりパケット数) を実現するためには、拡張ネットワークをサポートするインスタンスタイプを選択します。詳細については、「[拡張ネットワーク \(p. 559\)](#)」を参照してください。

最初にプレイメントグループを作成してから、そのプレイメントグループ内で複数のインスタンスを起動します。単一のリクエストで起動するインスタンス数は、プレイメントグループ内で必要な数にすることをお勧めします。後でプレイメントグループ内にさらにインスタンスを追加しようとすると、容量不足のエラーを受け取る可能性が高くなります。

プレイメントグループ内のインスタンスを停止して再起動しても、そのインスタンスは同じプレイメントグループ内で実行されます。ただし、インスタンスに対して十分な容量がない場合、起動は失敗します。

プレイメントグループ内のインスタンスを起動するときに容量エラーを受け取った場合は、プレイメントグループ内のインスタンスを停止して再起動してから、もう一度起動してください。

### Topics

- [プレイメントグループの制限 \(p. 129\)](#)
- [プレイメントグループへのインスタンスの起動 \(p. 129\)](#)
- [プレイメントグループを削除する \(p. 130\)](#)

## プレースメントグループの制限

プレースメントグループには次の制限があります。

- プレースメントグループを、複数のアベイラビリティゾーンにまたがって設定することはできません。
- プレースメントグループ名に指定する名前は、AWS アカウント内で固有である必要があります。
- プレースメントグループにインスタンスを起動するときを使用できるインスタンスタイプは、次に示すもののみです。
  - コンピューティングを最適化: c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge | cc2.8xlarge
  - GPU: cg1.4xlarge | g2.2xlarge
  - メモリを最適化: cr1.8xlarge | r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge
  - ストレージを最適化: hi1.4xlarge | hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge
- プレースメントグループをマージすることはできません。代わりに、一方のプレースメントグループでインスタンスを終了してから、他方のプレースメントグループ内でインスタンスを再起動する必要があります。
- プレースメントグループはピア接続された VPC にわたって使用できますが、ピア接続された VPC のインスタンス間では完全な二分帯域幅が得られません。VPC のピア接続の詳細については、『*Amazon Virtual Private Cloud ユーザーガイド*』の「[VPC Peering](#)」を参照してください。

## プレースメントグループへのインスタンスの起動

プレースメントグループに起動するインスタンス専用の AMI を作成することをお勧めします。

コンソールを使用してプレースメントグループにインスタンスを起動するには

1. Amazon EC2 コンソールを開きます。
2. インスタンス用の AMI を作成します。
  - a. Amazon EC2 ダッシュボードで、[Launch Instance] をクリックします。ウィザードを完了したら、[Launch] をクリックします。
  - b. インスタンスに接続します (詳しくは、[インスタンスへの接続 \(p. 334\)](#) を参照してください)。
  - c. ソフトウェアとアプリケーションをインスタンスにインストールし、データをコピーするか、追加の Amazon EBS ボリュームをアタッチします。
  - d. ナビゲーションペインで、[Instances] をクリックしてインスタンスを選択し、[Actions] をクリックして、[Create Image] をクリックします。[Create Image] ダイアログボックスで要求された情報を入力し、[Create Image] をクリックします。
  - e. (オプション) それ以上使用しない場合、このインスタンスを終了できます。
3. プレースメントグループを作成します。
  - a. ナビゲーションペインで、[Placement Groups] をクリックします。
  - b. [Create Placement Group] をクリックします。
  - c. [Create Placement Group] ダイアログボックスで、プレースメントグループに対して、使用中の AWS アカウント内で一意の名前を入力し、[Create] をクリックします。

プレースメントグループのステータスが `available` であれば、インスタンスをプレースメントグループに起動できます。

4. インスタンスをプレースメントグループに起動します。
  - a. ナビゲーションペインの [Instances] をクリックします。
  - b. [Launch Instance] をクリックします。指示どおりにウィザードを完了し、次の項目を選択するように注意します。
    - 作成した AMI
    - 必要なインスタンスの数
    - 作成したプレースメントグループ

コマンドラインを使用してプレースメントグループを作成するには

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

- `create-placement-group` ( AWS CLI )
- `ec2-create-placement-group` ( Amazon EC2 CLI )
- `New-EC2PlacementGroup` ( AWS Tools for Windows PowerShell )

お好みに応じて、`ec2-create-image` コマンドを使用して AMI を作成し、`ec2-create-placement-group` コマンドを使用してプレースメントグループを作成し、`ec2-run-instances` コマンドを使用してインスタンスをプレースメントグループに起動できます。

## プレースメントグループを削除する

置き換えが必要になったり、プレースメントグループが不要になったりした場合は、プレースメントグループを削除できます。プレースメントグループを削除する前に、プレースメントグループに起動したすべてのインスタンスを終了する必要があります。

コンソールを使用してプレースメントグループを削除するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. プレースメントグループのすべてのインスタンスを選択して終了します ( インスタンスを終了する前に、そのインスタンスがプレースメントグループ内にあることを確認できます。そのためには、詳細ペインで [Placement Group] の値をチェックします )。
4. ナビゲーションペインで、[Placement Groups] をクリックします。
5. プレースメントグループを選択し、[Delete Placement Group] をクリックします。
6. 確認を求めるメッセージが表示されたら、[Yes, Delete] をクリックします。

コマンドラインを使用してプレースメントグループを削除するには

次のいずれかのコマンドセットを使用できます。これらのコマンドラインインターフェイスについての詳細は、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `terminate-instances` および `delete-placement-group` ( AWS CLI )
- `ec2-terminate-instances` および `ec2-delete-placement-group` ( Amazon EC2 CLI )
- `Stop-EC2Instance` および `Remove-EC2PlacementGroup` ( AWS Tools for Windows PowerShell )

## インスタンスのサイズ変更

### Abstract

ワークロードに合わせてインスタンスのサイズを変更します。

ニーズが変わるにつれて、インスタンスの利用率が高すぎたり ( インスタンスタイプが小さすぎる )、低すぎたりする ( インスタンスタイプが大きすぎる ) ことに気付く場合があります。このような場合は、インスタンスのサイズを変更できます。例えば、`t1.micro` インスタンスがワークロードに対して小さすぎる場合は、`m1.small` インスタンスに変更できます。

インスタンスのサイズ変更のプロセスは、次のようにルートデバイスボリュームのタイプによって異なります。

- インスタンスのルートデバイスが Amazon EBS ボリュームの場合は、インスタンスタイプを変更することで、インスタンスのサイズを簡単に変更できます。
- インスタンスのルートデバイスがインスタンスストアボリュームである場合は、新しいインスタンスに移行する必要があります。

インスタンスのルートデバイスタイプを判断するには、Amazon EC2 コンソールを開き、[Instances] をクリックしてインスタンスを選択し、詳細ペインで [Root device type] の値をチェックします。この値は `ebs` または `instance store` のどちらかです。

ルートデバイスボリュームの詳細については、[ルートデバイスのストレージ \(p. 57\)](#) を参照してください。

### Topics

- [Amazon EBS-Backed インスタンスのサイズ変更 \(p. 131\)](#)
- [Instance Store-Backed インスタンスのサイズ変更 \(p. 132\)](#)

## Amazon EBS-Backed インスタンスのサイズ変更

Amazon EBS-Backed インスタンスタイプを変更するには、そのインスタンスを停止する必要があります。インスタンスを停止して開始すると、インスタンスは新しいハードウェアに移動されます。インスタンスが EC2-Classic で実行している場合、新しいパブリック IP アドレスとプライベート IP アドレスが与えられ、インスタンスに関連付けられている Elastic IP アドレスの関連付けが解除されます。したがって、インスタンスでホストしているアプリケーションをユーザーが中断なく継続して使用できるようにするには、インスタンスの再起動後、Elastic IP アドレスを再度関連付ける必要があります。詳細については、[インスタンスの停止と起動 \(p. 346\)](#) を参照してください。

AWS Management Console を使って Amazon EBS-Backed インスタンスのサイズを変更するには、次の手順を行います。

Amazon EBS-Backed インスタンスのサイズを変更するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Instances] をクリックし、インスタンスを変更します。
3. [EC2-Classic] インスタンスに関連付けられた Elastic IP アドレスがある場合は、詳細ペインに表示された Elastic IP アドレスとインスタンス ID を書き留めます。
4. [Actions] をクリックして、[Stop] をクリックします。
5. 確認ダイアログボックスで [Yes, Stop] をクリックします。インスタンスが停止するまで、数分かかる場合があります。

[EC2-Classic] インスタンスの状態が `stopped` になると、詳細ペインの [Elastic IP]、[Public DNS]、[Private DNS]、および [Private IPs] の各フィールドは空白になり、古い値がインスタンスと関連付けられなくなったことを示します。

6. インスタンスが選択されている状態で [Actions] をクリックし、[Change Instance Type] をクリックします。インスタンスの状態が `stopped` ではない場合、このアクションは無効になるので注意してください。
7. [Change Instance Type] ダイアログボックスの [Instance Type] リストで、必要なインスタンスのタイプを選択し、[Apply] をクリックします。
8. 停止されているインスタンスを再起動するには、インスタンスを選択して、[Actions] をクリックし、次に [Start] をクリックします。
9. 確認ダイアログボックスで、[Yes, Start] をクリックします。インスタンスが `running` 状態になるまで、数分かかる場合があります。

[EC2-Classic] インスタンスの状態が `running` になると、詳細ペインの [Public DNS]、[Private DNS]、および [Private IPs] の各フィールドには、インスタンスに割り当てられた新しい値が含まれます。

10. [EC2-Classic] インスタンスに関連する Elastic IP アドレスがある場合は、次のようにして再び関連付ける必要があります。
  - a. ナビゲーションペインで、[Elastic IPs] をクリックします。
  - b. インスタンスを停止する前に書き留めた Elastic IP アドレスを選択します。
  - c. [Associate Address] をクリックします。
  - d. インスタンスを停止する前に書き留めたインスタンス ID を選択し、[Associate] をクリックします。

## Instance Store-Backed インスタンスのサイズ変更

現在のインスタンスからイメージを作成し、そのイメージから必要なインスタンスタイプで新しいインスタンスを起動した後で、不要になった元のインスタンスを終了できます。インスタンスでホストしているアプリケーションをユーザーが中断なく継続して使用できるようにするには、現在のインスタンスに関連付けられた Elastic IP アドレスを新しいインスタンスに関連付ける必要があります。

Instance Store-Backed インスタンスのサイズを変更するには

1. (オプション) サイズ変更を実行するインスタンスに Elastic IP アドレスが関連付けられている場合は、サイズ変更した後でインスタンスに関連付けることができるように Elastic IP アドレスを書き留めます。
2. Instance Store-Backed インスタンスから AMI を作成するには、「[Instance Store-Backed Linux AMI の作成 \(p. 81\)](#)」に記載された前提条件と手順に従います。インスタンスから新しい AMI を作成したら、この手順に戻ります。
3. ナビゲーションペインで Amazon EC2 コンソールを開き、[AMIs] を選択します。フィルタリストで [Owned by me] を選択し、前のステップで作成したイメージを選択します。[AMI Name] は、イメージを登録したときに指定した名前であり、[Source] は Amazon S3 バケットです。



### Note

前のステップで作成した AMI が表示されない場合は、AMI を作成したリージョンがコンソールに表示されていることを確認します。

4. [Launch] をクリックします。起動ウィザードでオプションを指定するときは、必要な新しいインスタンスタイプを必ず指定します。インスタンスが `running` 状態になるまで、数分かかる場合があります。

5. (オプション) 起動に使用したインスタンスに Elastic IP アドレスが関連付けられている場合は、次のようにして新しいインスタンスに関連付ける必要があります。
  - a. ナビゲーションペインで、[Elastic IPs] をクリックします。
  - b. この手順の最初に残った Elastic IP アドレスを選択します。
  - c. [Associate Address] をクリックします。
  - d. 新しいインスタンスのインスタンス ID を選択し、[Associate] をクリックします。
  
6. (オプション) 不要になった場合は、起動したインスタンスを終了できます。インスタンスを選択し、この手順の最初に残ったインスタンス ID と一致しているかをチェックして、正しいインスタンスを終了しようとしていることを確認します。[Actions] をクリックし、次に [Terminate] をクリックします。

# スポットインスタンス

## Abstract

スポットインスタンスと呼ばれる未使用の Amazon EC2 のコンピューティング能力に入札し、コストを大幅に下げることができます。

アプリケーションを実行するタイミングに融通が利く場合は、スポットインスタンスと呼ばれる未使用の Amazon EC2 のコンピューティング能力に入札し、コストを大幅に下げることができます。Amazon EC2 で定められているこれらのインスタンスのスポット価格は、スポットインスタンス能力の需要と供給に応じて定期的に変動します。

スポットインスタンスを使用するには、1時間ごとに各インスタンスに払うことができる上限価格を指定するスポットインスタンスリクエスト（入札）を行います。お客様の最高入札価格が現在のスポット価格よりも高い場合、リクエストが実行され、インスタンスを終了させるまで、またはスポット価格がお客様の上限価格を超えるまで、インスタンスが実行されます。市場価格が上昇していても、お客様の入札価格が市場価格と同じになった場合、インスタンスは終了します。これは容量に対する需要が上昇したり、供給が変動した場合に生じる可能性があります。



### Note

Amazon EC2 スポットインスタンスは、Amazon Linux/UNIX、SUSE Linux Enterprise Server、Windows ( SQL Server なし ) の各プラットフォームで実行できます。AWS 無料利用枠プログラムでは、スポットインスタンスを使用できません。

お客様の最大入札価格よりも、時間あたりの支払いが少なくなる場合が多くあります。リクエストが行われ、インスタンスで利用可能な供給量が変化するために、スポット価格は定期的に調整されます。最高入札価格の方が高いかどうかに関係なく、全員がその期間に対する同じスポット価格を支払い、お客様の1時間ごとの最高入札価格よりも多く支払うことはありません。

製品と料金の詳細については、以下のページを参照してください。

- [Amazon EC2 スポットインスタンス](#)
- [AWS サービス料金の概要](#)
- [Amazon EC2 On-Demand Instances Pricing](#)

## 動画クイックルック: スポットインスタンスの使用を開始する

以下の動画は、AWS マネジメントコンソールを使用して初めてスポットインスタンスを起動する方法について説明しています。この動画には入札方法、インスタンスが満たされたときの判断方法、インスタンスのキャンセル方法についての指示が含まれています。[スポットインスタンスの使用を開始する](#)

## スポットインスタンスの使用を開始するためのチェックリスト

スポットインスタンスの使用を開始するには、以下のリソースを使用する必要があります。

- [スポットインスタンスの使用を開始する \(p. 135\)](#)
  - [スポットインスタンス価格履歴の表示 \(p. 138\)](#)
  - [スポットインスタンスリクエストの作成 \(p. 140\)](#)
  - [実行中のスポットインスタンスの検索 \(p. 144\)](#)

- [スポットインスタンスリクエストのキャンセル](#) (p. 146)
- [スポットインスタンスの基本](#) (p. 148)
  - [スポットリクエストの発行](#) (p. 149)
  - [スポットインスタンスリクエストのタグ付け](#) (p. 160)
  - [スポットインスタンスのプロビジョニング、価格設定、中断を理解する](#) (p. 161)
  - [スポットインスタンスデータを中断から保護する](#) (p. 164)
- [ウォークスルー: AWS サービスでのスポットインスタンスの使用](#) (p. 165)
  - [Auto Scaling によるスポットインスタンスの管理](#) (p. 166)
  - [CloudFormation テンプレートを使用してスポットインスタンスを起動する](#) (p. 182)
  - [スポットインスタンスで Amazon Elastic MapReduce のジョブフローを起動](#) (p. 183)
  - [Amazon Virtual Private Cloud でのスポットインスタンスの起動](#) (p. 183)
- [高度なタスク](#) (p. 186)
  - [スポットインスタンスのデータフィードの購読](#) (p. 186)
  - [AWS Java SDK を使用したスポットインスタンスのプログラミング](#) (p. 190)
  - [スポットインスタンスでのクラスターの開始](#) (p. 216)

## スポットインスタンスの最新情報

以下は、スポットインスタンスの最新情報の概要です。

- [スポットインスタンスの制限](#) (p. 150)

## スポットインスタンスの使用を開始する

### Abstract

スポットインスタンスの使用を開始する前に知っておく必要があること、および使用を開始するために必要な前提条件について説明します

このセクションでは、スポットインスタンスの使用を開始します。使用を開始する前に把握しておく必要があること、および使用を開始するために必要な前提条件について説明します。特定のスポットインスタンスタスクに使用するツールについて説明し、メインタスクに進みます。

スポットインスタンスを初めて使用する場合は、この Amazon EC2 製品の利点を最大限に活用できるように「[スポットインスタンスを使用するための前提条件](#) (p. 136)」を参照してください。Amazon EC2 を使用しており、続行できる場合は、次のリストのいずれかの項目をクリックして先に進みます。

### Topics

- [スポットインスタンスを使用するための前提条件](#) (p. 136)
- [スポットインスタンス用に AWS Management Console を使用する](#) (p. 136)
- [スポットインスタンス用に AWS CLI ツールおよび EC2 CLI ツールを使用する](#) (p. 138)
- [スポットインスタンス用に AWS Java SDK を使用する](#) (p. 138)
- [スポットインスタンス価格履歴の表示](#) (p. 138)
- [スポットインスタンスリクエストの作成](#) (p. 140)
- [実行中のスポットインスタンスの検索](#) (p. 144)
- [スポットインスタンスリクエストのキャンセル](#) (p. 146)

スポットインスタンスを要求する前に、アプリケーションが以下のタスクを実行するように Amazon マシンイメージ (AMI) を設定することを検討してください。

- 起動時に必要なタスクを自動的に実行する（インスタンスは通知なしに非同期で開始するため）。例えば、バッチプロセスを使用する場合は、AMI を設定して Amazon Simple Queue Service ( Amazon SQS ) キューからジョブをプルすることが可能です。
- インスタンス終了の影響を受けない場所に定期的に重要なデータを格納する。例えば、Amazon Simple Storage Service ( Amazon S3 )、Amazon SimpleDB、または Amazon Elastic Block Store ( Amazon EBS ) を使用してデータを格納することが可能です。



#### Important

スポットインスタンスでは Amazon EBS-Backed AMI を使用できますが、Amazon EBS ルートデバイスを持つ AMI から起動されたスポットインスタンスを停止および起動することはできない点に注意してください。Amazon EBS に関する詳細は、[Amazon Elastic Block Store \( Amazon EBS \)](#) を参照してください。

- 適切な終了処理を行う。

AMI 作成の詳細については、「[独自の AMI の作成 \(p. 56\)](#)」を参照してください。

## スポットインスタンスを使用するための前提条件

Amazon EC2 スポットインスタンスを使用する場合は、Amazon EC2 のアカウントと認証情報の作成に関する情報を提供する「[Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#)」の手順を読んで該当する作業を完了しておいてください。

また、スポットインスタンスの作業には、さまざまなツールを使用できます。AWS Management Console、AWS コマンドラインインターフェイス ( AWS CLI )、Amazon EC2 コマンドラインインターフェイスツール ( EC2 CLI )、Amazon EC2 アプリケーションプログラミングインターフェイス ( EC2 API ) のいずれを選択する場合でも、スポット価格履歴の評価、スポットインスタンスリクエストの送信 ( 入札とも言う )、スポットリクエストとスポットインスタンスの管理などの作業に使用できる、スポットインスタンス用のツールがあります。AWS SDK を使用して、アプリケーションの使用、開発、および管理を行うこともできます。詳細については、「[Tools for Amazon Web Services](#)」を参照してください。

このトピックでは、スポットインスタンスの作業に使用することができる、AWS Management Console、AWS CLI ツール、Amazon EC2 CLI ツールについての概要を示します。次のガイドの手順を読んで該当する作業を完了していることを前提としています。これらの手順では、次のツールで使用する環境の設定方法について説明しています。

- AWS Management Console—[AWS Management Console の使用を開始する](#)
- AWS CLI—[AWS Command Line Interface ユーザーガイド](#)
- Amazon EC2 CLI—[Amazon EC2 ツールを設定する](#)

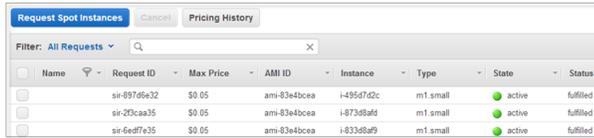
Amazon EC2 API の使用については、「[API リクエストを行う \(p. 721\)](#)」を参照してください。API アクションについては、[Amazon Elastic Compute Cloud API Reference](#) を参照してください。

## スポットインスタンス用に AWS Management Console を使用する

AWS Management Console において、EC2 コンソールは、スポットインスタンスリクエストのタスクに特化したツールを備えています。EC2 コンソールはまた、リクエストが受理されたときに起動されるインスタンスを管理する場合に使用できる一般ツールも備えています。

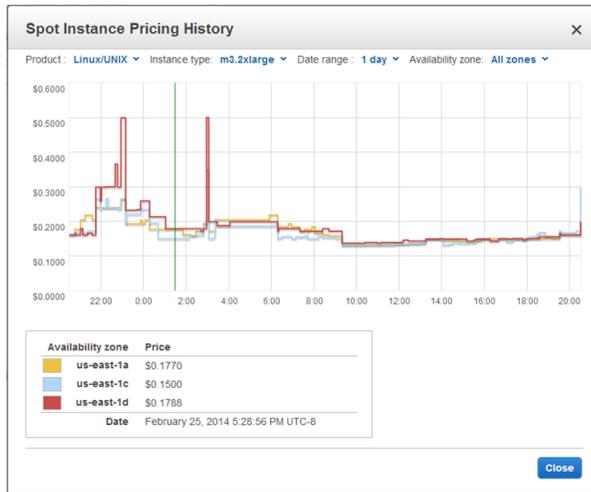
- [Spot Requests] ページから、スポットインスタンスリクエストの主な操作を行います。

## Amazon Elastic Compute Cloud ユーザーガイド スポットインスタンスの使用を開始する

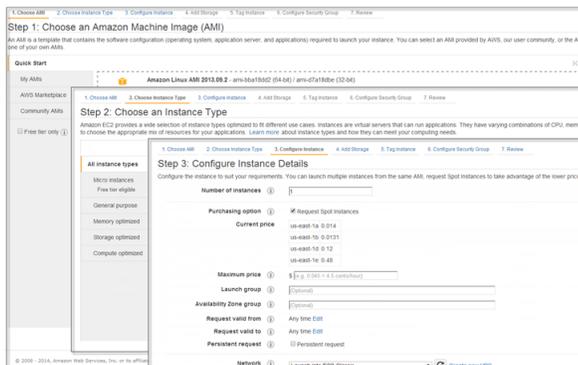


Name	Request ID	Max Price	AMI ID	Instance	Type	State	Status
sr-6976e32	sr-6976e32	\$0.05	ami-83e4bcea	i-495d7d2c	m1.small	active	fulfilled
sr-2f3caa35	sr-2f3caa35	\$0.05	ami-83e4bcea	i-87368af6	m1.small	active	fulfilled
sr-6ed7e35	sr-6ed7e35	\$0.05	ami-83e4bcea	i-83368af9	m1.small	active	fulfilled

- [Spot Instance Pricing History] では定義した期間で利用可能ゾーンの特定スポットインスタンスタイプの価格設定パターンを確認できます。



- [Request Spot Instances] ページを使用して、スポットインスタンスリクエストを送信し、リクエストが成功したときに起動するインスタンスの詳細を指定します。



Step 1: Choose an Amazon Machine Image (AMI)

Step 2: Choose an Instance Type

Step 3: Configure Instance Details

Number of instances: 1

Purchasing option:  If Request Spot Instances

Current price: us-east-1a: \$0.214, us-east-1c: \$0.1511, us-east-1d: \$0.12, us-ami-1e: \$0.48

Maximum price: \$1.00 (100% of 1.00)

Launch group:  OnDemand

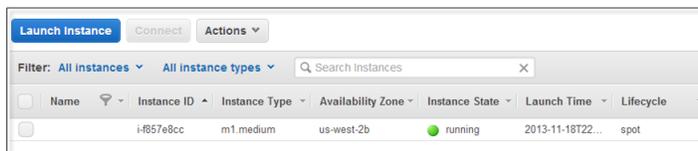
Availability Zone group:  OnDemand

Request valid from: Any time Edit

Request valid to: Any time Edit

Persistent request:  Persistent request

- [Instances] ページを使用して、スポットリクエストが成功したときに起動するインスタンスを管理します。



Name	Instance ID	Instance Type	Availability Zone	Instance State	Launch Time	Lifecycle
	i-857e9cc	m1.medium	us-west-2b	running	2013-11-18T22...	spot

## スポットインスタンス用に AWS CLI ツールおよび EC2 CLI ツールを使用する

AWS CLI ツールおよび Amazon EC2 CLI ツールには、スポットリクエストの管理用に特別に設計されたコマンドがあります。以下の表は、スポットリクエストのタスクで使用するコマンドを一覧しています。スポットリクエストが受理されると起動するインスタンスを管理するには、オンデマンド EC2 インスタンスで使用するのと同じコマンドを使用します。

タスク	AWS CLI	Amazon EC2 CLI
スポットインスタンス価格履歴の表示 (p. 138)。	<code>aws ec2 describe-spot-price-history</code>	<code>ec2-describe-spot-price-history</code>
実行中のスポットインスタンスの検索 (p. 144)。	<code>aws ec2 describe-spot-instance-requests</code>	<code>ec2-describe-spot-instance-requests</code>
スポットインスタンスリクエストの作成 (p. 140)。	<code>aws ec2 request-spot-instances</code>	<code>ec2-request-spot-instances</code>
スポットインスタンスのデータフィードの購読 (p. 188)。	<code>aws ec2 create-spot-datafeed-subscription</code>	<code>ec2-create-spot-datafeed-subscription</code>
データフィードのファイル名と形式 (p. 187)。	<code>aws ec2 describe-spot-datafeed-subscription</code>	<code>ec2-describe-spot-datafeed-subscription</code>
スポットインスタンスのデータフィードの削除 (p. 189)。	<code>aws ec2 delete-spot-datafeed-subscription</code>	<code>ec2-delete-spot-datafeed-subscription</code>
スポットインスタンスリクエストのキャンセル (p. 146)。	<code>aws ec2 cancel-spot-instance-requests</code>	<code>ec2-cancel-spot-instance-requests</code>

## スポットインスタンス用に AWS Java SDK を使用する

Java 開発者は *Java 用 AWS SDK* に進み、スポットインスタンスに関する Java チュートリアルを利用することができます。

- チュートリアル: Amazon EC2 スポットインスタンス (p. 191)
- チュートリアル: Amazon EC2 スポットリクエストの高度な管理 (p. 200)

## スポットインスタンス価格履歴の表示

### Abstract

スポットインスタンスの価格履歴を見て、スポット価格の入札を計画します。

単発のスポットリクエストが確実に受理されるようにするには、特定の価格を上回る価格で入札する必要があります。この特定の価格をスポット価格と言います。入札価格がスポット価格を超えている場合はスポットインスタンスが起動します。スポット価格が上昇して入札価格を超えた場合、スポットインスタンスは終了します。スポットリクエストが迅速に受理されるように、現在のスポット価格を上回る価格で入札してもかまいません。ただし、スポットインスタンスの入札価格を指定する前に、スポット価格履歴を確認することをお勧めします。同じインスタンスタイプ、同じオペレーティングシステム、そして同じ利用可能ゾーンを共有するスポットインスタンスのプールについては、過去 90 日間のスポット価格履歴を確認することができます。

例えば、us-east-1 リージョンで起動する Linux/UNIX t1.micro インスタンスに入札を行うとします。このスポットプールの過去の価格を確認するには、AWS マネジメントコンソールの [Spot Instance Pricing

History] ページ、DescribeSpotPriceHistory API アクション、または ec2-describe-spot-price-history CLI コマンドを使用して該当する値を指定します。特定の Availability Zone でスポットインスタンスを起動する必要がある場合は、スポット価格履歴を取得するときに目的の Availability Zone を指定します。

スポット価格履歴を確認した後、過去に 75 パーセントのスポットインスタンスアップタイムが確保されている価格で入札します。または、現在のスポット価格の 2 倍の価格で入札します。この場合、過去においては 99 パーセントのアップタイムが確保されています。ただし、入札を行う場合、スポット価格の過去の実績は今後の結果を保証するものではないので注意が必要です。スポット価格はリアルタイムの需要と供給の関係に基づいて変化するので、過去に特定のスポット価格または価格パターンを生成した状況が将来も繰り返されるとは限りません。



#### Note

Amazon EC2 を使用するための前提条件を設定しておきます。設定していない場合は、「[スポットインスタンスを使用するための前提条件 \(p. 136\)](#)」を参照してください。

2011-05-15 より前の API バージョンを使用する場合、DescribeSpotPriceHistory アクションまたは ec2-describe-spot-price-history コマンドはリージョンでの指定された期間における最低価格を返します。価格は古い順に返されます。

## AWS マネジメントコンソール

スポット価格履歴を表示するには

1. [Amazon EC2 コンソール](#) のナビゲーションペインで、[Spot Requests] をクリックします。

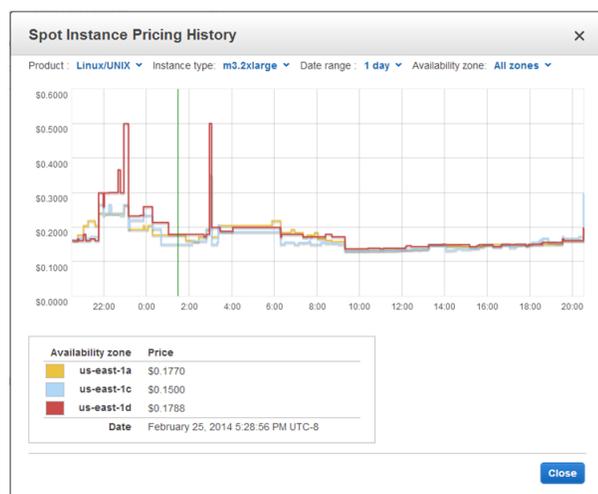
[Spot Requests] ペインが右側に開きます。スポットリクエストが存在する場合は、リスト表示されます。

2. ペイン上部の [Pricing History] をクリックします。

コンソールに [Spot Instance Pricing History] ページが表示されます。

3. 特定の Availability Zone のスポット価格履歴を表示する場合は、[Availability Zone] リストをクリックして、Availability Zone を選択します。

[Spot Instance Pricing History] ページに、すべてのゾーンまたは選択したゾーンのスポットインスタンスの価格設定履歴が表示されます。



4. スポット料金の詳細を表示するには、グラフで特定の時点までカーソルを移動します。カーソルとともに垂直線が移動し、さまざまなアベイラビリティゾーンのスプレッド価格と、カーソルと垂直線でポイントされている日時が、グラフの下のテーブルに表示されます。
5. 価格履歴をガイドとして使用して、必要な期間にインスタンスが実行されると考えられる価格を選択します。

## Amazon EC2 コマンドラインインターフェイス ( CLI ) ツール

スポット価格履歴を表示するには

1. 次のコマンドを入力します。

```
$ ec2-describe-spot-price-history -H --instance-type m1.xlarge
```

次のような出力が Amazon EC2 から返されます。

```
SPOTINSTANCEPRICE 0.384000 2011-05-25T11:37:48-0800 m1.xlarge
Windows us-east-1a
SPOTINSTANCEPRICE 0.384000 2011-05-25T11:37:48-0800 m1.xlarge
Windows us-east-1d
...
SPOTINSTANCEPRICE 0.242000 2011-04-18T14:39:14-0800 m1.xlarge SUSE
Linux us-east-1d
SPOTINSTANCEPRICE 0.242000 2011-04-18T14:39:14-0800 m1.xlarge SUSE
Linux us-east-1a
```

この例では、m1.xlarge インスタンスタイプの価格は、0.242 USD から 0.384 USD の間です。

2. 価格履歴をガイドとして使用して、必要な期間にインスタンスが実行されると考えられる価格を選択します。

### Tip

必要なインスタンスタイプまたは日付のみが含まれるように、スポット履歴データをフィルタリングすることができます。結果をフィルタリングする方法の詳細については、『*Amazon Elastic Compute Cloud コマンドラインリファレンス*』の「[ec2-describe-spot-price-history](#)」を参照してください。

API を使用したスポット価格履歴の表示

API を使用してスポット価格履歴を表示するには、[DescribeSpotPriceHistory](#) アクションを使用します。

次は何をしますか？

- [スポットインスタンスリクエストの作成 \(p. 140\)](#)
- [実行中のスポットインスタンスの検索 \(p. 144\)](#)
- [Amazon Virtual Private Cloud でのスポットインスタンスの起動 \(p. 183\)](#)

## スポットインスタンスリクエストの作成

Abstract

入札価格を決定した後、スポットインスタンスのリクエストを作成します。

入札価格を決定すると、スポットインスタンスをリクエストできる状態になります。AWS マネジメントコンソール、API、または CLI を使用して、希望するスポットインスタンスの数、使用するインスタンスタイプ、および支払うことができる入札価格を指定できます。

一度に複数のスポットインスタンスをリクエストした場合、Amazon EC2 では一意の ID (例えば、sir-1a2b3c4d) を使用して個々のスポットインスタンスリクエストが作成されます。結果、各リクエストのステータスを別々に追跡することができます。スポットリクエストの追跡については、「[入札ステータスコードによるスポットリクエストの追跡 \(p. 153\)](#)」を参照してください。

このセクションでは、AWS マネジメントコンソールまたは CLI を使用して Amazon EC2 スポットインスタンスのリクエストを作成する方法について説明します。API を使用してスポットインスタンスリクエストを作成するには、[RequestSpotInstances](#) アクションを使用します。スポットインスタンスの要求について詳しくは、次のトピックを参照してください。

- [スポットインスタンスの制限 \(p. 150\)](#)
- [スポットインスタンスリクエストのタイプ \(p. 152\)](#)
- [起動グループおよびアベイラビリティゾーンでのスポットインスタンスの起動 \(p. 152\)](#)
- [スポットインスタンスのリクエストの状態 \(p. 151\)](#)
- [Amazon Virtual Private Cloud でのスポットインスタンスの起動 \(p. 183\)](#)



#### Note

Amazon EC2 を使用するための前提条件を設定しておきます。設定していない場合は、「[スポットインスタンスを使用するための前提条件 \(p. 136\)](#)」を参照してください。

## AWS マネジメントコンソール

スポットインスタンスリクエストを作成するには

1. [Amazon EC2 コンソール](#) のナビゲーションペインで、[Spot Requests] をクリックします。
2. [Request Spot Instances] をクリックします。

スポットインスタンスを設定するために指定するスポットインスタンスオプションを除き、スポットインスタンスを要求するプロセスはオンデマンドインスタンスの起動プロセスと同じです。[Instances] ページで [Launch Instance] をクリックして同じ手順を実行します。Amazon マシンイメージ (AMI) とインスタンスタイプを選択し、インスタンスの詳細を設定します。詳細については、[インスタンスの起動 \(p. 324\)](#) を参照してください。

3. [Spot Requests] ページから [Configure Instance Details] 画面に移動した場合、[Purchasing Option] として [Request Spot Instances] がデフォルトで選択されています。スポットインスタンスリクエスト関連のオプションが表示されるのは、[Request Spot Instances] オプションが選択されている場合だけです。

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances (i) 1

Purchasing option (i)  Request Spot Instances

Current price (i) us-west-2a 0.004  
us-west-2b 0.004  
us-west-2c 0.004

Maximum price (i) \$ [e.g. 0.045 = 4.5 cents/hour]

Launch group (i) [Optional]

Availability Zone group (i) [Optional]

Request valid from (i) Any time Edit

Request valid to (i) Any time Edit

Persistent request (i)  Persistent request

スポットインスタンスに支払う [Maximum price] を指定します。これは必須オプションです。

リージョンの各アベイラビリティゾーンの現在のスポット価格が一覧表示されることに注意してください。この情報を参考にして、リクエストが認められるように価格を指定します。入札価格がスポット価格を超えた場合、インスタンスが起動されて実行されます。



**Note**

リクエストを送信した後では、スポット価格などのスポットリクエストのパラメータを変更できません。

スポット価格の傾向を調べるには、[Spot Requests] ページの [Pricing History] をクリックします。詳細については、[スポットインスタンス価格履歴の表示 \(p. 138\)](#) を参照してください。

4. オプションで定義できる詳細設定がいくつかあります。スポットリクエストを定義して、スポットインスタンスを設定するために必要なオプションを選択します。

オプション	説明
[Request valid from] および [Request valid to]	<p>スポットリクエストの有効期間を定義します。[Request valid from] から始まり、[Request valid to] までの有効期間によって、リクエストがアクティブである期間が指定されます。デフォルトでは、スポットリクエストは作成されてから、受理されるかお客様によってキャンセルされるまで、受理の対象となります。ただし、これらのオプションを使用して有効期間を制限することができます。</p> <p> <b>Note</b></p> <p>指定した終了時間は、このリクエストによって起動されるスポットインスタンスには適用されません。この終了時間は、スポットインスタンスリクエストにのみ適用されます。</p>
[Persistent request]	<p>リクエストがワンタイムか、持続かを決定します。デフォルトでは、ワンタイムリクエストになります。ワンタイムリクエストは、一度しか受理されません。永続リクエストは、同じリクエストで実行されているスポットインスタンスがない限り、常に受理の対象となります。</p>
[Launch group]	<p>同じ起動グループの一連のリクエストがグループ化されます。起動グループのすべてのリクエストには同時に開始し、終了するインスタンスが含まれています。スポットリクエストでの起動グループを使用する詳細については、<a href="#">起動グループおよびアベイラビリティゾーンでのスポットインスタンスの起動 (p. 152)</a> を参照してください。</p>

オプション	説明
[Availability Zone group]	同じ利用可能ゾーンの一連のリクエストがグループ化されます。利用可能ゾーングループを共有するすべてのリクエストは、同じ利用可能ゾーンでスポットインスタンスを起動します。スポットリクエストでのアベイラビリティゾーンの使用の詳細については、 <a href="#">起動グループおよびアベイラビリティゾーンでのスポットインスタンスの起動 (p. 152)</a> を参照してください。

- 通常、インスタンスを起動する場合と同様に、ウィザードを続行します。詳細については、[インスタンスの起動 \(p. 324\)](#) を参照してください。

リクエストに複数インスタンスが指定されている場合、Amazon EC2 ではインスタンスごとに別のスポットインスタンスリクエストが作成されます。リクエストは、[Spot Requests] ページに表示されます。

Name	Request ID	Max Price	AMI ID	Instance	Type	State	Status
	sr-97d6a32	\$0.05	ami-83e4bcea	i-45d742c	m1.small	active	fulfilled
	sr-203caa35	\$0.05	ami-83e4bcea	i-873d8afd	m1.small	active	fulfilled
	sr-6ed7e35	\$0.05	ami-83e4bcea	i-833d8af9	m1.small	active	fulfilled

## Amazon EC2 コマンドラインインターフェイス ( CLI ) ツール

スポット価格リクエストを作成するには

- 次のコマンドを入力します。

```
$ ec2-request-spot-instances
--price price
--user-data data
--instance-count count
--type one-time / persistent
[--valid-from timestamp]
[--valid-until timestamp]
[--launch-group launchgroup]
(run-instances-arguments)
```

以下に例を示します。

```
$ ec2-request-spot-instances --price 0.32 ami-1234abcd --key MyKeypair --group webserv --instance-type m1.small --instance-count 10 --type one-time
```

次のような出力が Amazon EC2 から返されます。

```
SPOTINSTANCEREQUEST    sir-09fb0a04    0.32    one-time    Linux/UNIX
  open    2010-04-06T10:03:09+0200    ami-1234abc    m1.small
MyKeypair    webserv    monitoring-disabled
...
```

リクエストに複数インスタンスが指定されている場合、Amazon EC2 ではインスタンスごとに別のスポットインスタンスリクエストが作成され、各インスタンスの ID は異なります ( sir-09fb0a04 など )。

API を使用したスポットインスタンスの要求

API を使用してスポットインスタンスを要求するには、[RequestSpotInstances](#) アクションを使用します。

次は何をしますか？

- [実行中のスポットインスタンスの検索](#) (p. 144)
- [スポットインスタンスリクエストのキャンセル](#) (p. 146)
- [中断の計画](#) (p. 164)
- [Amazon Virtual Private Cloud でのスポットインスタンスの起動](#) (p. 183)
- [スポットインスタンスリクエストのタグ付け](#) (p. 160)
- [スポットインスタンスのデータフィードの購読](#) (p. 186)

## 実行中のスポットインスタンスの検索

### Abstract

アクティブなスポットリクエストに関連付けられた実行中のスポットインスタンスを検索します。

スポット価格が入札価格を下回ると、スポットリクエストによってスポットインスタンスが起動されます。スポットインスタンスは、その最高入札価格がスポット価格を上回ることができなくなるまで、またはお客様がご自分でスポットインスタンスを終了するまで実行されます ( 入札価格がスポット価格と厳密に等しい場合、スポットインスタンスは実行されたままとなる可能性があります )。スポットリクエストが受理され、インスタンスが実行されている場合、スポットリクエストは *active* ( *open*、*closed*、または *canceled* とは対照的 ) になります。このセクションでは、お客様のアクティブなスポットリクエストに関連付けられた実行中のスポットインスタンスを検索する方法について説明します。



### Note

Amazon EC2 を使用するための前提条件を設定しておきます。設定していない場合は、「[スポットインスタンスを使用するための前提条件](#) (p. 136)」に進みます。

## AWS マネジメントコンソール

実行中のスポットインスタンスを検索するには

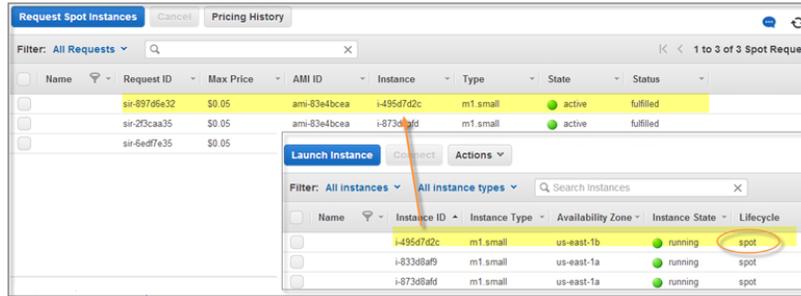
1. [Amazon EC2 コンソール](#) のナビゲーションペインで [Instance] をクリックします。

実行中のインスタンスの一覧がコンソールに表示されます。

Name	Instance ID	Instance Type	Availability Zone	Instance State	Launch Time	Lifecycle
	i-f857e8cc	m1.medium	us-west-2b	running	2013-11-18T22...	spot

2. スポットリクエストによって起動したインスタンスを識別するには、[Instances] ページ ( ここで、[Lifecycle] 列にインスタンスがスポットとして一覧される ) のインスタンス ID を [Spot Requests]

ページのインスタンス ID と照合します ( 列の表示をオンにしなければならない場合があります ( ページ右上の [Show/Hide] をクリック ) ) 。



## Amazon EC2 コマンドラインインターフェイス ( CLI ) ツール

実行中のスポットインスタンスを検索するには

1. `ec2-describe-spot-instance-requests` を使用します。スポットインスタンスリクエストが受理されていると ( インスタンスが起動している )、インスタンス ID が応答に表示されます。

```
$ ec2-describe-spot-instance-requests
SPOTINSTANCEREQUEST sir-e1471206 0.09 one-time Linux/UNIX
active 2010-09-13T16:50:44-0800
i-992cf7dd ami-813968c4 m1.small MyKey default
monitoring-disabled
```

2. あるいは、`ec2-describe-instances` をフィルタ `--filter instance-lifecycle=spot` と併用することもできます。Windows システムのコマンドラインインターフェイスツールを使用している場合は、引用符 ( `--filter "instance-lifecycle=spot"` ) を使用しなければならない場合があります。フィルタについての詳細は、「[リソースのリスト表示とフィルタリング \(p. 660\)](#)」を参照してください。

```
$ ec2-describe-instances --filter instance-lifecycle=spot
```

Amazon EC2 は以下のような出力を返します。

```
RESERVATION r-b58651f1 111122223333 default
INSTANCE i-992cf7dd ami-813968c4 ec2-184-72-8-111.us-west-1.compute.amazonaws.com ip-10-166-105-139.us-west-1.compute.internal
running MyKey 0 m1.small 2010-09-13T23:54:40+0000
us-west-1a aki-a13667e4 ari-a33667e6
monitoring-disabled 184.72.8.111 10.166.105.139 ebs
spot sir-e1471206 paravirtual
```

API を使用した実行中のスポットインスタンスの検索

API を使用して実行中のスポットインスタンスについての情報を取得するには、[DescribeSpotInstanceRequests](#) アクションを使用します。

次は何をしますか？

- [スポットインスタンスリクエストのキャンセル \(p. 146\)](#)

- [Amazon Virtual Private Cloud でのスポットインスタンスの起動 \(p. 183\)](#)
- [永続ルート EBS パーティション \(p. 165\)](#)
- [スポットインスタンスリクエストのタグ付け \(p. 160\)](#)
- [スポットインスタンスのデータフィードの購読 \(p. 186\)](#)

## スポットインスタンスリクエストのキャンセル

### Abstract

スポットリクエストが必要なくなったときは、open または active 状態のスポットインスタンスリクエストをキャンセルします。

スポットリクエストが不要になった場合には、それをキャンセルすることができます。お客様は open または active 状態のスポットインスタンスリクエストしかキャンセルできません。リクエストがまだ受理されておらず、インスタンスが起動されていない場合、スポットリクエストは open 状態にあります。リクエストが受理され、結果としてスポットインスタンスが起動された場合、スポットリクエストは active 状態にあります。スポットリクエストが active 状態にあり、そのスポットリクエストに関連付けられた実行中のスポットインスタンスがある場合、リクエストをキャンセルしてもインスタンスは自動的に終了しません。実行中のスポットインスタンスは手動で終了する必要があります。



### Note

Amazon EC2 を使用するための前提条件を設定しておきます。設定していない場合は、「[スポットインスタンスを使用するための前提条件 \(p. 136\)](#)」に進みます。

スポットリクエストは、open および active 状態のみならず、他の状態になる可能性もあります。詳細については、「[スポットインスタンスのリクエストの状態 \(p. 151\)](#)」を参照してください。これにより、各リクエストのステータスを別々に追跡することができます。スポットリクエストの追跡については、「[入札ステータスコードによるスポットリクエストの追跡 \(p. 153\)](#)」を参照してください。

永続スポットリクエストは、スポットインスタンスが中断または終了されると必ずスポットインスタンスの再起動を試みます。スポットリクエストとそのスポットインスタンスを両方も終了する場合は、必ずスポットリクエストをキャンセルしてからスポットインスタンスを終了します。こうすることで、リクエストは新しいインスタンスを起動しなくなります。

## AWS マネジメントコンソール

スポットインスタンスリクエストをキャンセルするには

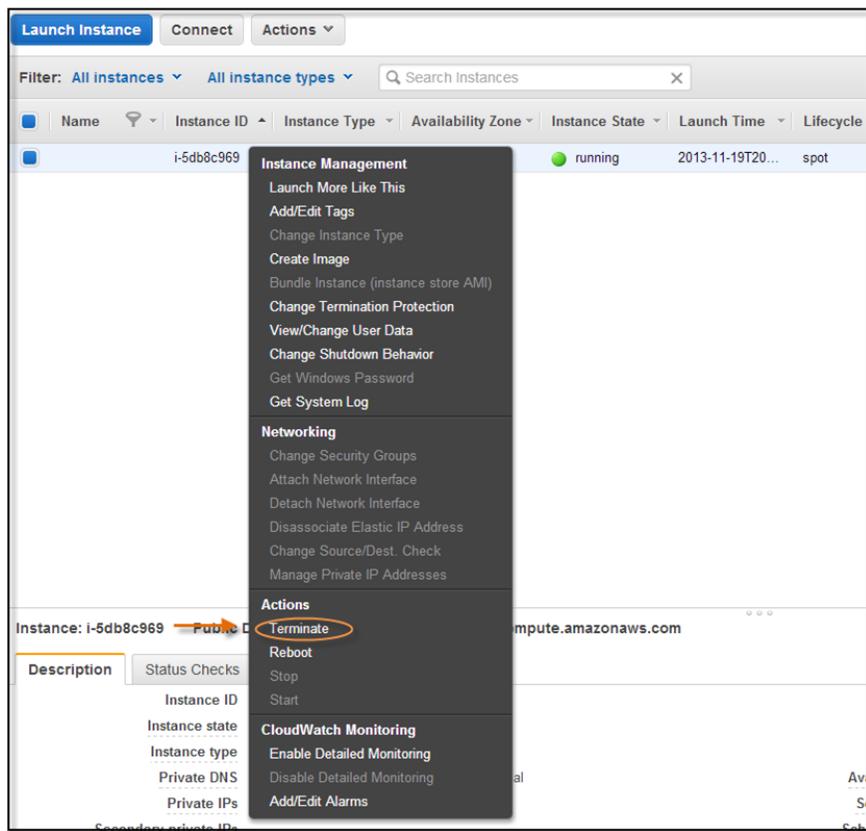
1. [Amazon EC2 コンソール](#)のナビゲーションペインで、[Spot Requests] をクリックします。

コンソールに、スポットインスタンスリクエストの一覧が表示されます。

2. キャンセルするスポットインスタンスを選択し、[Cancel] をクリックします。

Name	Request ID	Max Price	AMI ID	Instance	Type	State	Status
	sir-897d6e32	\$0.05	ami-83e4bcea	i-495d7d2c	m1.small	active	fulfilled
	sir-2f3caa35	\$0.05	ami-83e4bcea	i-873d8afd	m1.small	active	fulfilled
	sir-6ed7e35	\$0.05	ami-83e4bcea	i-833d8af9	m1.small	active	fulfilled

3. 関連付けられたスポットインスタンスの実行を継続する必要がない場合は、該当するスポットインスタンスをキャンセルすることができます。[Instances] ページでスポットインスタンスを選択し、右クリックして [Terminate] を選択します。



## Amazon EC2 コマンドラインインターフェイス ( CLI ) ツール

スポットインスタンスリクエストをキャンセルするには

1. スポットインスタンスリクエストを確認するには、次のコマンドをします。

```
$ ec2-describe-spot-instance-requests
```

次のような出力が Amazon EC2 から返されます。

```
SPOTINSTANCEREQUEST  sir-09fb0a04  0.04  one-time  Linux/UNIX  closed
2010-04-06T10:03:09+0200  ami-b232d0db  m1.small  gsg-keypair  default
monitoring-disabled
```

2. リクエストをキャンセルするには、次のコマンドを入力します。

```
$ ec2-cancel-spot-instance-requests sir-09fb0a04
```

次のような出力が Amazon EC2 から返されます。

```
SPOTINSTANCEREQUEST  sir-09fb0a04  canceled
```

3. キャンセルしたスポットリクエストに関連付けられているインスタンスを終了するには、次のコマンドを入力します。

```
$ ec2-terminate-instances i-48cd4829
```

次のような出力が Amazon EC2 から返されます。

```
INSTANCE      i-48cd4829      running shutting-down
```



#### Tip

特定の EC2 インスタンスタイプのみが返されるように、スポットインスタンスリクエストの一覧をフィルタリングすることができます。結果をフィルタリングする方法の詳細については、『*Amazon Elastic Compute Cloud コマンドラインリファレンス*』の「[ec2-describe-spot-instance-requests](#)」を参照してください。

#### API を使用したスポットインスタンスリクエストのキャンセル

API を使用してスポットインスタンスリクエストをキャンセルするには、[CancelSpotInstanceRequests](#) アクションを使用します。

次は何をしますか？

- [スポットインスタンスリクエストの作成 \(p. 140\)](#)
- [スポットインスタンス価格履歴の表示 \(p. 138\)](#)
- [実行中のスポットインスタンスの検索 \(p. 144\)](#)
- [Amazon Virtual Private Cloud でのスポットインスタンスの起動 \(p. 183\)](#)

## スポットインスタンスの基本

### Abstract

成功するスポットリクエストの作成と管理、また、中断や容量の変化などの不測事態に対する計画の基礎を示します。

お客様のアプリケーションが中断を許容し、変化する計算能力および価格を動的に処理することができれば、Amazon EC2 スポットインスタンスはアプリケーションのコストを最適化し、アプリケーションの計算規模を拡大するための強力なツールとなり得ます。スポットインスタンスは通常のオンデマンドインスタンスの場合と比較して大抵は大幅な節約となるので、計算ジョブの時間短縮およびコスト全体の削減の両方を実現するためにスポットインスタンスを利用できる場合があります。

このセクションでは、知っておく必要がある基本事項を説明します。この基本事項を把握することにより、良好なスポットリクエストを作成し、スポットインスタンスを効果的に管理し、中断や容量の変化などの不測の事態を考慮することが可能になります。

- スポットインスタンスリクエスト（入札とも呼ばれる）では、必要なスポットインスタンスの種類、およびスポットインスタンスに対して支払できる上限価格を定義します。入札の指定方法によって、リクエストが受理されるかどうか、およびスポットインスタンスが正常に起動するかどうかが決まります。詳細については、「[スポットリクエストの発行 \(p. 149\)](#)」を参照してください。
- タグを使用してスポットインスタンスを管理できます。詳細については、「[スポットインスタンスリクエストのタグ付け \(p. 160\)](#)」を参照してください。

- スポット価格、使用可能なスポット容量、入札価格の間の相互作用により、Amazon EC2 スポットインスタンスの実行期間や中断のタイミングが影響を受けます。詳細については、[スポットインスタンスのプロビジョニング、価格設定、中断を理解する \(p. 161\)](#) を参照してください。
- スポットインスタンスの利用可能性は、Amazon EC2 の容量に応じて大きく変動します。スポットインスタンスを使用する場合は、中断を予測および考慮し、中断がアプリケーション与える影響を緩和する必要があります。詳細については、「[スポットインスタンスデータを中断から保護する \(p.164\)](#)」を参照してください。

製品と料金の詳細については、以下のページを参照してください。

- [Amazon EC2 スポットインスタンス](#)
- [AWS サービス料金の概要](#)
- [Amazon EC2 On-Demand Instances Pricing](#)

## スポットリクエストの発行

### Abstract

スポットリクエストを実行して管理する際に、知っておくと役に立つ入札の基本について説明します。

スポットインスタンスリクエスト (または入札) は、Amazon EC2 スポットインスタンスのコア要素です。スポットリクエストの指定 (価格、利用可能ゾーン、起動グループ、スケジュール) により、リクエストの受理と、スポットインスタンスの正常な起動が左右されます。



#### Note

リクエストを送信した後では、スポット価格などのスポットリクエストのパラメータを変更できません。

このセクションでは、スポットリクエストの管理を行う上で知っておくと役に立つ入札の基本について説明します。

- [スポットインスタンスの制限 \(p. 150\)](#)

スポットインスタンスを使用するときに、スポット入札価格制限、およびリージョンあたりのスポットインスタンスリクエスト制限という 2 つの制限カテゴリについて説明します。

- [スポットリクエストのカスタマイズ \(p. 151\)](#)

スポットリクエストをカスタマイズする場合に指定できるオプションについて説明します。

- [入札ステータスコードによるスポットリクエストの追跡 \(p. 153\)](#)

スポット入札ステータス情報と、この情報を使用してスポットリクエストがどうなっているか理解する方法について説明します。

スポットリクエストの受理までにどれくらい時間をかけられるか、スポットインスタンスをどれくらい継続して実行するかに応じて、スポットインスタンスの入札を行う際に考慮しなければならない要素が多くあります。スポットリクエストに含める要件が受理の可能性に影響する場合があります。満たさなければならない要件が多い場合、リクエストが受理されるまでの時間が長くなる場合があります。

支払うことができる最大スポットインスタンス価格 (入札価格) もまた、スポットインスタンスが実行する時間の長さに影響します。インスタンスが可能な限り長い時間実行するようにし、中断の可能性を低く抑える場合は、高い入札価格を送信することをお勧めします。スポットの中断にはそれほど関心がなく、単にスポットを低コストで利用したいだけならば、低い入札価格を送信すべきです。

必ずしも入札価格を支払うわけではありません。例えば、0.500 USD で入札した場合、スポット価格が期間中 1 時間あたり 0.300 USD であると、1 時間あたり 0.300 USD を支払えばよいことになります。スポット価格が低下すれば、支払う価格はさらに少なくなります。スポット価格が上昇すれば、新しい価格 (最大で入札価格) を支払います。スポット価格が上昇して、入札価格を上回ると (または、別の容量制約が検出されると)、スポットインスタンスは中断します。



#### Note

高額で入札したからといって、スポットインスタンスが中断されないという保証はありません。また、スポット価格は長期間にわたって高価格にとどまることはない、ということを想定しないでください。長期間のスポット供給が厳しい場合は、持続期間のスポット価格を上げて、最高入札者の入札価格に連動させることができます。

スポットインスタンスのリクエストの作成については、「[スポットインスタンスリクエストの作成 \(p. 140\)](#)」を参照してください。

## スポットインスタンスの制限

### Abstract

スポットインスタンスに関する 2 種類の制限、入札価格制限とリクエスト制限について説明します。

スポットインスタンスを使用するときは、入札価格制限とリクエスト制限という 2 種類の制限について考慮する必要があります。

- スポット入札価格制限は、スポットインスタンスのリクエストを行うときに入札できる上限価格です。
- スポットリクエスト制限は、リクエストできるリージョンごとのインスタンスの数です。

### スポット入札価格制限

2013 年 12 月 20 日から、Amazon EC2 に、スポットインスタンスを入札できる最大金額に対するデフォルトの制限が導入されました。デフォルトの入札価格制限は、スポットインスタンスのお客様が高い入札価格によって予想外の請求を受けることがないように設けられています。また、この制限は、スポット価格が過剰に上昇する可能性を抑えることも目的とされています。

デフォルトの入札制限は、オンデマンド価格の 4 倍に設定されます。例えば、us-east-1 の Linux m3.2xlarge インスタンスのスポットインスタンスリクエストを送信する場合、そのオンデマンド価格が 1 時間あたり 0.900 USD であるとする、1 時間あたり最高 3.600 USD まで入札できます。



#### Note

新しいデフォルト制限を超える入札が現在ある場合は、既存の入札をキャンセルし、2013 年 12 月 20 日になる前に、制限内の入札を始めることをお勧めします。この期日を過ぎると、アカウントの入札制限を超えるスポットインスタンスリクエストは処理されない可能性があります。

アプリケーションでデフォルトの入札価格制限を超える入札を行う必要がある場合は、制限を引き上げるリクエストを [spot-limit-increase@amazon.com](mailto:spot-limit-increase@amazon.com) に送信してください。制限引き上げリクエストには、以下の情報を含める必要があります。

- AWS アカウント番号。
- スポットインスタンスの使用法およびスポット中断の管理方法。
- 要求する入札価格制限 (インスタンスタイプ、リージョン、製品プラットフォームを指定)。

オンデマンドインスタンス価格については、[Amazon EC2 料金表](#)を参照してください。

## スポットリクエスト制限

既定では、リージョン内のスポットインスタンスのリクエストの総数は、合計5に制限されています。新しいAWSアカウントの場合、さらに少ない可能性があります。現在、I2 インスタンスタイプおよびHS1 インスタンスタイプはスポットで使用できません。また、すべてのリージョンに提供されていないインスタンスタイプもあります。(インスタンスのタイプについては、「[インスタンスタイプ \(p. 109\)](#)」を参照してください)。

リージョン内で追加のスポットリクエストを行う必要がある場合は、[Amazon EC2 インスタンス申請フォーム](#)に記入してください。アカウントのスポットインスタンス制限の引き上げをリクエストしていることを [Use Case Description] で指定します。申請フォームで指定したリージョンに対して制限数の増加が適用されます。



### Note

スポットのリクエストが制限内であっても、入札価格がスポット価格を上回っていない場合、またはスポット容量が不足する場合、スポットインスタンスが終了する可能性があることに注意してください。

## スポットリクエストのカスタマイズ

### Abstract

ニーズに対応するインスタンスを取得できるように、スポットリクエストのオプションをカスタマイズします。

複数のオプションを使用してスポットリクエストをカスタマイズすれば、ニーズに対応するインスタンスを取得できます。これらのオプションは次のトピックで説明します。

- [スポットインスタンスリクエストのタイプ \(p. 152\)](#)
- [起動グループおよびアベイラビリティゾーンでのスポットインスタンスの起動 \(p. 152\)](#)

選択したオプションは、スポットリクエストの成功と、これらのリクエストによって起動されるスポットインスタンスの存続期間とに影響します。

### スポットインスタンスのリクエストの状態

スポットインスタンスリクエストは、次に示す状態のいずれかになります。

- Open – リクエストは受理されていません。
- Active – リクエストは現在アクティブ ( 受理された ) 状態であり、リクエストに関連付けられたスポットインスタンスが存在します。
- Failed – 指定されているパラメータが適切でないためリクエストは失敗しました。
- Closed – リクエストは完了した ( スポットインスタンスは起動し、その後、中断または終了した ) が、または指定した期間内に受理されませんでした。
- Canceled – リクエストをキャンセルしたが、入札リクエストが有効期間を過ぎているために、リクエストがキャンセルされています。

状況によっては、リクエストがクローズまたはキャンセルされてもインスタンスが実行し続ける場合があります。

スポットインスタンスリクエストを送信する際に指定する [Valid Until] オプションは、そのリクエストにのみ適用されます。この期限は、リクエストによって起動するスポットインスタンスには適用されません。さらに、入札リクエストが有効期間を過ぎたか、またはお客様が手動でキャンセルしたことが原因でスポットインスタンスリクエストがキャンセルされても、今キャンセルされたリクエストによって以前に起動されたスポットインスタンスは自動的に終了しません。スポットインスタンス

サービスは、スポット価格がお客様の価格以上である場合にのみ実行中のインスタンスを終了します。ただし、インスタンスは手動で任意のタイミングで終了できます。

スポットインスタンスリクエストのステータスと、入札ステータスコードによって起動されたインスタンスのステータスを追跡できます。スポットリクエストの追跡については、「[入札ステータスコードによるスポットリクエストの追跡 \(p. 153\)](#)」を参照してください。

### スポットインスタンスリクエストのタイプ

作成できるスポットインスタンスリクエストのタイプは2つです。1つはワンタイムリクエストで、もう1つは永続リクエストです。ワンタイムリクエストは次の条件のいずれかが満たされるまでアクティブ状態のままです。リクエストされたインスタンスのすべてが起動した。リクエストが有効期限切れになった。お客様がリクエストをキャンセルした。例えば、3つのインスタンスにワンタイムリクエストを作成した場合、3つのすべてのインスタンスが起動した時点でリクエスト完了とみなされます。

持続スポットインスタンスリクエストは、リクエストが以前に受理されている場合でも、期限が切れるかキャンセルするまではアクティブです。たとえば、スポット価格が \$0.300 のときに1つのインスタンスに対して永続スポットインスタンスリクエストを作成したとします。お客様の最大入札価格が \$0.300 を上回る場合は、Amazon EC2 によりインスタンスが起動され、インスタンスの実行が継続されます。スポット価格が最大入札価格を超え、インスタンスが終了された場合、お客様のスポットインスタンスリクエストはアクティブ状態のままであり、スポット価格が最大入札価格を下回ると、Amazon EC2 により、新たにインスタンスが起動されます。

ワンタイムリクエストと永続リクエストの両方を使用した場合は、スポット価格を上回ることができなくなるまで、お客様がインスタンスを終了するまで、またはインスタンスが自ら終了するまで、インスタンスは実行し続けます。上限価格がスポット価格とまったく同じである場合、インスタンスは続行する場合としない場合があります（空き容量によって異なります）。

### 起動グループおよびアベイラビリティゾーンでのスポットインスタンスの起動

スポットインスタンスが同時にまたは同じアベイラビリティゾーンで起動するように選択することができます。

Amazon EC2 に、スポットインスタンスリクエスト内のすべてのインスタンスが同時に受理される場合にのみインスタンスを起動するように伝えるには、スポットインスタンスリクエストで *起動グループ* を指定します。1つのアベイラビリティゾーンですべてのインスタンスを同時に起動するには、*アベイラビリティゾーングループ* を指定します。次のツールを使用して、これらのオプションを選択および指定することができます。

- [AWS Management Console の \[Request Spot Instances\] ウィザード](#)
- [aws ec2 request-spot-instances](#) AWS CLI コマンド
- [ec2-request-spot-instances](#) Amazon EC2 CLI コマンド
- [RequestSpotInstances](#) Amazon EC2 API アクション

これらのオプションは便利ですが、必要な場合にのみ実装してください。これらのオプションにより、スポットインスタンスリクエストが受理される機会が減る可能性があります。

#### 起動グループ オプションの使用

Amazon EC2 に、スポットインスタンスリクエスト内のインスタンスがすべて受理される場合にのみインスタンスを起動するように伝えるには、AWS Management Console を使用してスポットインスタンスリクエストで [Launch Group] を指定したり、AWS CLI コマンドまたは Amazon EC2 コマンドを使用して `--launch-group` オプションを指定したり、Amazon EC2 API アクションを使用して `LaunchGroup` オプションを指定したりします。

起動グループの指定により、スポットインスタンスが終了する可能性が増えることに注意してください。起動グループの1つのインスタンスが、スポットサービスによって終了する場合（スポット価格が入札価格を超える場合など）、グループ内のすべてのインスタンスが終了します。



#### Note

起動グループは、スポットサービスによって認識および管理される機能です。起動グループ内の1つ以上のインスタンスがお客様やスポットサービス以外のサービスによって終了される場合、起動グループ内のその他のインスタンスは終了されません。

#### アベイラビリティゾーングループオプションの使用

Amazon EC2 に、1つのアベイラビリティゾーン内のインスタンスを起動するように伝えるには、AWS Management Console を使用してスポットインスタンスリクエストで [Availability Zone group] を指定したり、AWS CLI コマンドまたは Amazon EC2 コマンドを使用して `--availability-zone-group` オプションを指定したり、Amazon EC2 API アクションを使用して `AvailabilityZoneGroup` オプションを指定したりします。

特定の利用可能ゾーンのスポットインスタンスの入札価格を特定するには、「[スポットインスタンス価格履歴の表示 \(p. 138\)](#)」を参照して、入札を行う利用可能ゾーンの価格履歴表示します。指定した利用可能ゾーンの価格履歴が、最新の価格セットから順に表示されます。利用可能ゾーンを指定しない場合、期間中のすべての利用可能ゾーンの最低価格が最新の価格セットから順に表示されます。



#### Note

`describe Spot price history` コマンドは 2 つのデータを返します。指定したアベイラビリティゾーングループ内の、指定した時間範囲内のインスタンスタイプの価格と価格が変更された時刻です。価格は指定した期間内で有効な価格であり、レスポンスでは単に価格が変更された最終時刻が表示されます。

2011-05-15 API バージョンより前の Amazon EC2 API アクション (`DescribeSpotPriceHistory`) または Amazon EC2 CLI コマンド (`ec2-describe-spot-price-history`) を使用すると、指定した期間のリージョンの最低価格が時系列順、つまり古いものから新しいものの順に返されます。

スポット価格を確認する際には、過去の実績は今後の結果を保証するものではないことに注意してください。スポット価格はリアルタイムの需要と供給の関係に基づいて変化するので、過去に特定のスポット価格または価格パターンを生成した状況が将来も繰り返されるとは限りません。

利用可能ゾーンについての詳細は、[リージョンとアベイラビリティゾーンに関する概念 \(p. 8\)](#) を参照してください。

## 入札ステータスコードによるスポットリクエストの追跡

### Abstract

使用および入札を戦略的に計画するために、入札ステータスコードでスポットインスタンスリクエストを追跡します。

#### Topics

- [スポットリクエストのライフサイクル \(p. 154\)](#)
- [スポット入札ステータスコードのリファレンス \(p. 158\)](#)

入札ステータスは、Amazon EC2 スポットインスタンスリクエストの追跡、スポットインスタンスの使用計画の策定、および戦略的な入札の実行に役に立ちます。スポット入札ステータス情報は、ステータスコード、更新時刻、およびステータスメッセージで構成されます。同時に、スポット入札ステータ

ス情報は、スポットリクエストの処理を決定する場合にも役に立ちます。例えば、入札ステータスによって、スポットリクエストがまだ受理されていない理由や、スポットリクエストの受理を妨げている制約の一覧を確認できます。



**Note**

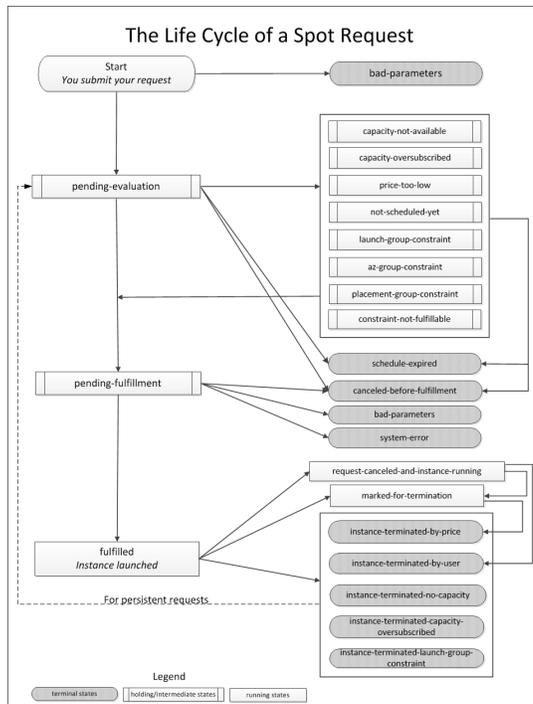
入札とは、価格に条件付きで結び付けられたリクエストです。これらの用語は、このドキュメントでは置き換えて使用することができます。

任意の Amazon EC2 スポットインスタンスツールを使用して、スポットリクエストに関するステータス情報を取得することができます。スポット入札ステータスは、新しいスポットリクエストと、すでにオープンしている入札リクエストで使用できます。ただし、Amazon EC2 CLI ツールまたは API ツールを使用して入札ステータス情報を取得する場合は、バージョン 2012-10-01 以降の API ツールを使用する必要があります。次の表では、Amazon EC2 ツールを使用した入札ステータス情報の取得方法について要約します。

ツール	使用アイテム	取得内容
Amazon EC2 コンソールの [Spot Requests] ページ	ステータス列 下部ペイン、ヒント	ステータスコード ステータスメッセージ
Amazon EC2 CLI	<code>ec2-request-spot-instances</code> または <code>ec2-describe-spot-instance-requests</code>	ステータスコード、ステータスメッセージ、 および更新時刻
Amazon EC2 API	<code>RequestSpotInstances</code> または <code>DescribeSpotInstanceRequests</code>	ステータスコード、ステータスメッセージ、 および更新時刻

**スポットリクエストのライフサイクル**

次の図では、スポットリクエストがライフサイクルでたどり得る経路を示します。この図に従って、リクエストの作成から終了状態に至るまでの、ライフサイクルの各ステップについて説明します。



スポットインスタンスを要求すると、そのリクエストは評価プロセスで評価されます。このプロセスの各ステップ（スポット入札リクエストのライフサイクルとも呼ばれます）では、特定のイベントによって後続のリクエスト状態が決まります。ライフサイクルの図では、各ステップはノードとして表現され、各ノードのステータスコードはスポットリクエストおよびスポットインスタンスのステータスを示します。

1. 評価保留 – スポットインスタンスリクエストを行うと、そのリクエストはすぐに `pending-evaluation` 状態になります。

ステータスコード	リクエストの状態	インスタンスの状態
<code>pending-evaluation</code>	<code>open</code>	<code>(none)</code>

スポット入札リクエストがこの `pending-evaluation` 状態にならない唯一の理由は、1つ以上のパラメータが無効であるということです (`bad-parameters`)。

ステータスコード	リクエストの状態	インスタンスの状態
<code>bad-parameters</code>	<code>closed</code>	<code>(none)</code>

2. 保留評価ホールディング状態 – 指定した1つ以上のオプションが有効であるものの条件を満たしていない場合、またはスポット容量の制約が検出された場合に、スポットインスタンスは `holding` 状態にあるステータスコードをレポートします。リクエストによって定義されるオプションは、リクエストが受理される可能性に影響します。例えば、現在のスポット価格を下回る最大入札価格を指定した場合、またはスポットインスタンスが1つの利用可能ゾーンで同時に起動されるようにリクエストした場合、スポットリクエストは、スポット価格が入札価格を下回るまで、または利用可能ゾーンの制約条件が満たされるまで、スポットリクエストは `holding` 状態のままです。

次の表では、入札が *holding* 状態にあつて制約条件が満たされるのを待機しているときにレポートされるステータスコードを一覧します。

ステータスコード	リクエストの状態	インスタンスの状態
capacity-not-available	open	(none)
capacity-oversubscribed	open	(none)
price-too-low	open	(none)
not-scheduled-yet	open	(none)
launch-group-constraint	open	(none)
az-group-constraint	open	(none)
placement-group-constraint	open	(none)
constraint-not-fulfillable	open	(none)

3. 保留評価/受理終了状態 – 入札に対して指定した有効期限が経過した場合、入札を自分でキャンセルした場合、またはシステムエラーが発生した場合には、リクエストが保留受理フェーズに到達する前であっても、スポットインスタンスリクエストが *terminal* 状態になる場合があります。



#### Note

スポットリクエストの *canceled* は、必ずしもスポットインスタンスの終了を意味するものではありません。対照的に、スポットリクエストの *closed* の場合は、リクエストはもはや考慮されず、スポットインスタンスが終了することを意味します。

次の表では、保留評価または保留受理の後で入札が *terminal* 状態になったときにレポートされるステータスコードを示します。

ステータスコード	リクエストの状態	インスタンスの状態
schedule-expired	closed	(none)
canceled-before-fulfillment*	canceled	(none)
bad-parameters	failed	(none)
system-error	closed	(none)

\* このステータスは、ユーザーによって開始されるキャンセルによってもたらされます。

4. 保留受理 – 指定した制約条件 (存在する場合) が満たされ、入札価格が現在のスポット価格以上である場合、スポットリクエストは *pending-fulfillment* 状態になります。

ステータスコード	リクエストの状態	インスタンスの状態
pending-fulfillment	open	(none)

この時点で、Amazon EC2 スポットサービスは要求されたインスタンスを提供するよう準備します。この時点でプロセスが停止した場合、これは、インスタンスが起動する前にユーザーによってキャンセルされたため、または予期しないシステムエラーが発生したためであると考えられます (「保留評価/受理終了状態」を参照)。

5. 受理済み – スポットインスタンスに対するすべての指定が満たされると、スポットリクエストが受理されます。Amazon EC2 スポットサービスによってスポットインスタンスが起動されます。起動には数分かかる場合があります。

ステータスコード	リクエストの状態	インスタンスの状態
fulfilled	active	starting up → running

この時点で、インスタンスが起動し、入札リクエストは `active` になります。入札価格がスポット価格以上であり、インスタンスタイプに対する予備のスポット容量が Amazon EC2 に存在し、お客様がインスタンスを終了しない限りは、インスタンスは実行し続けます。



#### Note

入札価格が市場価格と等しい場合、リクエストは `fulfilled` 状態にとどまりますが、市場価格が上昇した時点でスポットインスタンスが終了するというリスクがあります。場合によっては、入札価格がスポット価格と等しくても、その時点でのスポットインスタンスの申し込み超過が原因で、スポットインスタンスが強制退去させられる可能性があります。そのような場合は、`instance-terminated-capacity-oversubscribed` ステータスコードが返されます。詳細については、「[受理済み終了状態](#)」を参照してください。

6. 受理済み終了状態 – スポット価格または容量の変化により、スポットサービスがスポットインスタンスを終了させる必要が生じた場合、スポットリクエストは `terminal` 状態でステータスコードをレポートします。お客様がスポットリクエストをキャンセルまたはスポットインスタンスを終了した場合も、`terminal` 状態ステータスコードが返されます。

次の表では、入札が受理された後で、インスタンスが `terminal` 状態になったときにレポートされるステータスコードを一覧します。

ステータスコード	リクエストの状態	インスタンスの状態
<code>request-canceled-and-instance-running</code>	<code>canceled[c]</code>	実行中
<code>marked-for-termination</code>	<code>closed</code>	実行中
<code>instance-terminated-by-price</code>	<code>closed[a]</code>	(Spot) terminated
<code>instance-terminated-by-user</code> (または <code>spot-instance-terminated-by-user</code> 、 これは廃止される)	<code>closed[b]</code> キャンセル済み [b]	terminated[c]
<code>instance-terminated-no-capacity</code>	<code>closed[a]</code>	(Spot) terminated
<code>instance-terminated-capacity-oversubscribed</code>	<code>closed[a]</code>	(Spot) terminated
<code>instance-terminated-launch-gap-constraint</code>	<code>closed[a]</code>	(Spot) terminated

[a] スポット入札リクエストが永続リクエストである場合、リクエストの状態は `open` になります。次の項目「[永続リクエスト](#)」を参照してください。

[b] インスタンスを先に終了した場合は、スポットインスタンスが終了したとスポットサービスによって検出されるまでに遅延が生じる場合があります。永続リクエストの場合、ユーザーによって終了されたインスタンスについては、スポットリクエストが `pending-evaluation` 状態になるまで、いくらかの時間がかかる可能性があります。このステータスコードを使用すると、リクエストの状態はクローズまたはキャンセルになります。インスタンスは終了するが、入札を継続する場合は、

リクエストの状態は *closed* になります。インスタンスを終了し、入札をキャンセルする場合、リクエストの状態は *canceled* になります。

[c] ユーザーによってアクションが開始されます。

7. 永続リクエスト – スポットインスタンスが ( ユーザーまたは Amazon EC2 スポットサービスによって ) 終了するとき、永続スポットリクエストを指定していれば、リクエストは再度考慮されます。このリクエストは *pending-evaluation* 状態に戻ります。スポット入札リクエストは、当初評価されたのと同じ評価プロセスにかけられ、永続スポットリクエストに対して全く新しいスポットインスタンスが起動します。

## スポット入札ステータスコードのリファレンス

次のリストには、スポット入札ステータスコードと、コードはスポットリクエストにとって何を意味するのかの説明が含まれます。

- **az-group-constraint**

スポットサービスは、同じ利用可能ゾーンでお客様が要求したインスタンスをすべて起動できるとは限りません。

- **bad-parameters**

スポットリクエストの 1 つ以上のパラメータが無効か、形式が不正です ( 例えば、指定した AMI が存在していません )。入札ステータスメッセージによって、どのパラメータが無効かを確認できません。

- **canceled-before-fulfillment**

スポットリクエストが受理される前にユーザーがスポットリクエストをキャンセルしました。

- **capacity-not-available**

要求したインスタンスに使用できる容量が存在しません。

- **capacity-oversubscribed**

入札価格がお客様の入札価格以上であるスポットリクエストの数が、このプールで使用可能な容量を超えています。

- **constraint-not-fulfillable**

1 つ以上の制約条件が無効であるため、スポットリクエストを受理できません。例えば、無効な *from* または *to*、あるいは無効な利用可能ゾーンを指定しました。この入札ステータスメッセージは、どの制約条件が無効化を知らせます。

- **fulfilled**

入札リクエストは受理されました。したがって、スポットリクエストはアクティブであり、Amazon EC2 スポットサービスがスポットインスタンスを起動中 ( または起動済み ) です。

- **instance-terminated-by-user ( または *spot-instance-terminated-by-user* )**

受理済みのスポットインスタンスを終了させたので、入札ステータスは *closed* 状態になり ( 永続入札でない場合 )、インスタンスは *terminated* 状態になりました。



### Note

ステータスコード *instance-terminated-by-user* および *spot-instance-terminated-by-user* は、同じステータスを表します。コード *spot-instance-terminated-by-user* は廃止されます。

- **instance-terminated-by-price**

スポット価格が上昇し、入札価格を上回りました。リクエストが永続入札の場合、プロセス (ライフサイクル) が再び開始され、入札が再び評価保留となります。

- **instance-terminated-capacity-oversubscribed**

入札価格がお客様の入札価格以上であるスポットリクエストの数が、このプールで使用可能な容量を超えたので、インスタンスは終了しました。つまり、スポット価格は変化していないものの、お客様の価格がスポット価格と等しかったので、インスタンスは中断されました。



**Note**

特定の入札価格のリクエストが複数存在し、その中の一部のみを終了させる必要がある場合、スポットサービスは、終了するリクエストをランダムに選択します。

- **instance-terminated-launch-group-constraint**

起動グループ内のインスタンスの1つが終了したので、起動グループの制約条件が満たされなくなりました。

- **instance-terminated-no-capacity**

インスタンスで使用できるスポット容量がなくなりました。

- **launch-group-constraint**

スポットサービスは、お客様が同時に要求したインスタンスをすべて起動できるわけではありません。



**Note**

起動グループのすべてのリクエストには同時に開始し、終了するインスタンスが含まれています。スポットリクエストでの起動グループの使用の詳細については、「[起動グループおよびアベイラビリティゾーンでのスポットインスタンスの起動 \(p. 152\)](#)」を参照してください。

- **marked-for-termination**

スポットインスタンスに終了のためのマークが付けられます。

- **not-scheduled-yet**

スポットリクエストは、スケジュール設定された日付になるまで評価されません。

- **pending-evaluation**

スポットインスタンスリクエストを作成するとすぐに、スポットインスタンスリクエストは保留評価状態となりシステムはリクエストのパラメータを評価します。

- **pending-fulfillment**

スポットリクエストは、システムがスポットインスタンスの準備を試みている間、保留受理状態となります。

- **placement-group-constraint**

現在、指定した配置グループに新しいスポットインスタンスを追加することはできないので、まだスポットリクエストを受理することはできません。

- **price-too-low**

入札価格がスポット価格を下回っているために受理できない入札リクエストがあります。この場合、インスタンスは起動されず、入札はオープンのままになります。

- **request-canceled-instance-running**

スポットインスタンスの実行中にスポットリクエストをキャンセルしました。そのため、リクエストはキャンセルされましたが、インスタンスはまだ実行されています。

- `schedule-expired`

スポットリクエストは、有効な *to* 日までに受理されていないので、有効期限切れとなりました。

- `system-error`

予期しないシステムエラーが発生しました。これが反復性の問題である場合は、カスタマーサポートに連絡してください。

## スポットインスタンスリクエストのタグ付け

### Abstract

リクエストの分類と管理に役立つように、選択したメタデータでスポットインスタンスリクエストにタグを付けます。

スポットインスタンスリクエストを分類および管理しやすくするため、任意のメタデータでタグ付けすることができます。スポットインスタンスリクエストのタグ付けは、他の Amazon EC2 リソースにタグ付けするのと同じ方法で行います。キーと値を指定してタグを作成し、その組み合わせをスポットインスタンスリクエストに割り当てます。

スポットインスタンスリクエストに作成するタグは、そのリクエストにのみ適用されます。このタグはリクエストで起動されるインスタンスには伝播しません。リクエストが受理され、スポットインスタンスが起動すると、スポットインスタンスリクエストに使用したタグがスポットインスタンスに適用されていないことがわかります。タグ付けされたリクエストから起動するスポットインスタンスを分類するには、インスタンスに個別のタグのセットを作成する必要があります。

スポットインスタンスリクエストを初めて作成したときにタグ付けすることも、作成した後にタグを割り当てることもできます。スポットインスタンスリクエストにタグ付けするには、次のツールを使用します。

- AWS Management Console の EC2 コンソールの [Spot Requests] ページ
- `create-tags` AWS CLI コマンド
- `ec2-create-tags` Amazon EC2 CLI コマンド
- `CreateTags` API アクション

AmazonEC2でタグを使用する方法の詳細については、「[AmazonEC2リソースにタグを付ける \(p. 662\)](#)」を参照してください。

スポットインスタンスリクエストのタグを作成するには

1. AWS Management Console を使用してスポットインスタンスリクエストを作成する場合、Amazon マシンイメージ (AMI) およびインスタンスタイプを指定する手順を最初に実行します。詳細については、[スポットインスタンスリクエストの作成 \(p. 140\)](#) を参照してください。

インスタンスの構成やストレージの詳細を指定した後に、[Next: Tag Instances] をクリックします。

2. [Tag Spot Request] の手順では、タグに使用するキー名と値を入力して、[Review and Launch] をクリックします。

### Step 5: Tag Spot Request

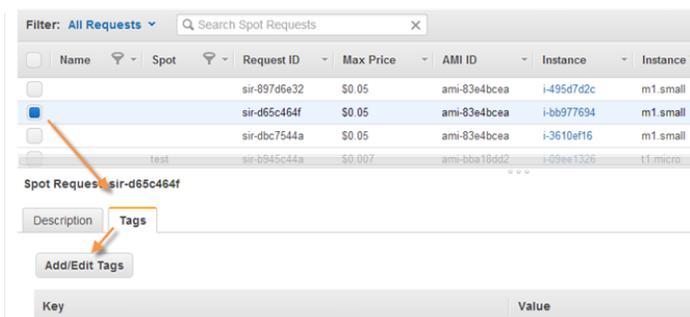
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#)  
Note that these tags will be applied to this Spot Instance request and not to any instances launched to fulfill this request.

Key (127 characters maximum)	Value (255 characters maximum)
Spot	Test

Create Tag (Up to 10 tags maximum)

別のタグを作成する場合は [Create Tag] をクリックします。

3. 既存のスポットインスタンスリクエストにタグを作成する場合、[Spot Requests] ページで、タグ付けするスポットリクエストを選択し、下のペインの [Tags] タブで、[Add/Edit Tags] をクリックします。



[Add/Edit Tags] ボックスで、[Create Tag] をクリックし、タグに使用するキー名と値を入力してから [Save] をクリックします。

#### Add/Edit Tags

Apply tags to your resources to help organize and identify them.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value
<input type="text"/>	<input type="text"/>

Create Tag Cancel Save

先ほどタグ付けしたスポットインスタンスリクエストから起動されるスポットインスタンスにタグを適用するには、同じツールとコマンドを使用してインスタンスに個別のタグのセットを作成する必要があります。インスタンスにタグを追加する方法の詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 662\)](#)」を参照してください。

## スポットインスタンスのプロビジョニング、価格設定、中断を理解する

### Abstract

スポットインスタンスがどのようにプロビジョニングされ、価格が設定されて、中断されるのかを理解します。

Amazon EC2 スポットインスタンスを所有している場合、スポットインスタンスが実行される期間や中断されるタイミング、価格の設定方法について知りたいとお考えになるでしょう。これらの要素は、入札時に設定した価格、Amazon EC2 によって設定されたスポット価格、使用可能なスポット容量に関係します。これらの要素の相互作用方法を理解すると、スポットインスタンスをいっそう効果的に使用できるはずです。

- [スポット市場の供給と需要 \(p. 162\)](#)
- [中断について理解する \(p. 163\)](#)

## スポット市場の供給と需要

スポットインスタンスは、供給と需要によって変化する価格で利用できるスペアの EC2 インスタンスです。スポットインスタンスを要求するときは、支払うことができる上限価格を設定します ( 入札 )。AWS は、各インスタンスプールにおいてスポットで提供できるスペア Amazon EC2 の容量と、各プールに対して行われた入札を継続的に評価し、利用できるスポット容量を最高額の入札者に対してプロビジョニングします。プールのスポット価格は、そのプールで履行された最低の入札に設定されます ( インスタンスプールは、同じリージョン、アベイラビリティゾーン、インスタンスタイプ、オペレーティングシステム、ネットワークプラットフォームを共有するインスタンスのグループです )。つまり厳密に言えば、スポット価格とは、単一のスポットインスタンスに対するスポットリクエストを満たすために、入札する必要のある最低価格です。

次の図では、このスポットプロビジョニングおよび価格設定メカニズムの仕組みを示します。

スポットインスタンスが 5 個だけ販売されているスポットプールに入札するものとします ( 例えば、us-east-1a リージョンおよびアベイラビリティゾーン、m1.small インスタンスタイプ、Linux オペレーティングシステム、VPC ネットワークプラットフォーム )。お客様の入札価格は 0.100 USD で、これは現在のスポット価格でもあります。現在の入札の価格は次のとおりです。



### Note

入札は高いものから順番に並んでいます。入札価格が同じ入札はランダムな順序で表示されています。

入札	入札価格	
1.	1.000 USD	
2.	1.000 USD	
3.	0.100 USD	
4.	0.100 USD	お客様の入札
5.	0.100 USD	最後に落札された入札がスポット価格を設定します
-----	-----	スポット容量カットオフ
6.	0.050 USD	

この表では、入札 1~5 は落札されます。最後に落札された入札 5 によってスポット価格が 0.100 USD に設定されます。入札 6 は落札されません。入札価格が同じ 0.100 USD である入札 3~5 はランダムな順序で並んでいます。

ここで、このスポットプールで利用できるスペアの Amazon EC2 容量が 3 に低下するとします。スポットプールは次のようになります。

入札	入札価格	
1.	1.000 USD	
2.	1.000 USD	

入札	入札価格	
3.	0.100 USD	最後に落札された入札がスポット価格を設定します
— — — —	— — — —	スポット容量カットオフ
4.	0.100 USD	お客様の入札
5.	0.100 USD	
6.	0.050 USD	

入札 1~3 は落札されます。最後に落札された入札 3 によってスポット価格が 0.100 USD に設定されます。同じ 0.100 USD の入札 4~5 は落札されません。ご覧のように、スポット価格は変わりませんが、0.100 USD の入札のうち、4 (お客様の入札) と 5 の 2 つがスポット提供縮小のために中断されています。

まだ単一インスタンスのスポットリクエストを落札したい場合は、現在のスポット価格 0.100 USD より高い価格で入札する必要があります。0.101 USD でスポットインスタンスを入札した場合は落札されて、入札 3 は中断され、スポット価格は 0.101 USD になります。2.000 USD で入札すると入札 3 は中断されて、スポット価格は 1.000 USD (入札 2 の価格) になります。このシナリオでは、どんなに高い価格で入札しても 3 個より多くのインスタンスは落札できないことに注意してください。

スポットに使用できるスペア Amazon EC2 の容量がゼロになると、すべての入札は中断されます。

 Note

つまり、どんなに高い価格で入札しても、スポットインスタンスは中断されます。スポット市場に価格の競合がある場合 (通常は、少量の使用可能なスポット容量について)、スポット価格が予想以上に上昇し、長期間高止まりする可能性があるため、支払可能な額より高い価格では入札しないでください。

## 中断について理解する

### Abstract

中断時にはどうすればよいかを学習します。

スポットインスタンスを使用するときは、常に中断に備える必要があります。インスタンスはいつでも中断される可能性があり、高額なスポット入札によって保護されるわけではありません。したがって、データのバックアップシステムを用意しておく必要があります。インスタンスのデータはデフォルトでは保存されません。インスタンスが中断されると失われます。

スポットインスタンスが中断される場合には 3 つの理由があります。

- スポット価格 – スポットインスタンスに対して支払うことができる上限価格の入札価格が現在のスポット価格より低い場合、実行中のスポットインスタンスは中断されます。

Amazon EC2 は、各アベイラビリティゾーンの各インスタンスタイプに対してスポット価格を設定します。スポット価格は、すべてのお客様が特定の期間スポットインスタンスを実行するために支払う価格です。スポット価格は、インスタンスの需要と供給の関係に基づいて変動します。しかしお客様が、スポットリクエストで指定した上限価格を超える金額を支払うことはありません。スポット価格が、お客様の上限価格を超過した場合は、お客様のインスタンスは Amazon EC2 によってシャットダウンされます。

- 容量 – 十分なスペア Amazon EC2 容量がない場合、入札価格が最低のものから順番にスポットインスタンスが中断されます。同じ入札価格で複数のインスタンスが存在する場合、インスタンスが中断される順番はランダムに決定されます。

- リクエスト制約–同じ起動グループや同じアベイラビリティゾーングループといった制約のあるスポットインスタンスをリクエストした場合、制約を満たすことができなくなった場合、これらのインスタンスはグループとして中断されることがあります。

スポットインスタンスの中断の原因は、お客様への請求方法に影響を与えます。お客様がご自分でスポットインスタンスを中断した場合は、1時間まるまる使用したかどうかに関係なく、1時間分の使用料金が請求されます。Amazon EC2 がスポットインスタンスを中断した場合は、中断された時間の料金は請求されません。新しいスポットインスタンスのリクエストが送信されてスポットインスタンス起動されるまで、アカウントが請求を受けることはありません。

## スポットインスタンスデータを中断から保護する

### Abstract

別のユーザーがお客様より高い入札を提出したとき、中断されるスポットインスタンスのデータを保護してください。

Amazon EC2 スポットインスタンスに対するお客様の需要は刻一刻と大幅に変化する可能性があります。また、スポットインスタンスの利用可能性も利用可能な Amazon EC2 の容量に応じて大きく変化する可能性があります。したがって、別のユーザーがお客様より高い価格で入札した場合や、当社で予備の容量を再要求する必要がある場合、スポットインスタンスは中断される可能性が常にあります。



### Note

どんなに高い価格で入札しても、スポットインスタンスが中断されるリスクは常についてまわります。オンデマンド価格を上回る価格で入札しないこと、および中断を許容できないアプリケーションに対してスポットを使用しないことを強く推奨します。

次のセクションでは、中断を考慮し、軽減する方法について説明します。

- [中断の計画 \(p. 164\)](#)
- [永続ルート EBS パーティション \(p. 165\)](#)

## 中断の計画

### Abstract

スポットインスタンスを中断する必要がある場合に、即時シャットダウンに対応するようアプリケーションを計画してください。

### Topics

- [動画クイックルック: 中断の管理 \(p. 164\)](#)

スポットインスタンスはいつでも終了する可能性があるため、スポットインスタンスで実行されるアプリケーションは正常に終了する必要があります。Amazon はインスタンスが問題なく終了するよう試みますが、アプリケーションが強制終了される場合にも対応できるように準備しておく必要があります。

アプリケーションをテストするには、オンデマンドインスタンスとしてアプリケーションを起動し、終了します。

### 動画クイックルック: 中断の管理

以下の動画は、お客様がスポットインスタンスの中断を管理する方法の一例を示しています。 [スポットインスタンスの中断を管理する方法](#)

中断を管理する方針についての詳細は、以下のセクションを参照してください。

- [スポットインスタンスで Amazon Elastic MapReduce のジョブフローを起動 \(p. 183\)](#)
- [Auto Scaling を使用してスポットインスタンスに関する通知を取得する \(p. 179\)](#)
- [スポットインスタンスでのクラスターの開始 \(p. 216\)](#)

## 永続ルート EBS パーティション

### Abstract

スポットインスタンスが中断されるときに、データが失われずに保持されるよう、データをルート EBS パーティションに格納します。

スポットインスタンスが終了すると失われるおそれがあるデータを保存するには、Amazon Elastic Block Store ( Amazon EBS ) が有効です。



#### Important

スポットインスタンスでは Amazon EBS-Backed AMI を使用できますが、Amazon EBS ルートデバイスを持つ AMI から起動されたスポットインスタンスを停止および起動することはできない点に注意してください。Amazon EBS に関する詳細は、[Amazon Elastic Block Store \( Amazon EBS \)](#) を参照してください。

スポットインスタンスデータの永続性を設定するには、起動するスポットインスタンスを既存の Amazon Elastic Block Store ( Amazon EBS ) スナップショットにマッピングします。*delete-on-termination* フラグを *false* に設定します。これは、スポットインスタンスの終了時に Amazon EC2 が Amazon EBS ボリュームを削除しないことを示しています。

以下の指定内容でスポットリクエストを作成してみましょう。

- 入札価格は 0.500 USD
- m1.xlarge インスタンスタイプのインスタンスが 1 つ
- スポットインスタンスの終了時に削除しないスナップショットに対するブロックデバイスマッピング

この例は CLI または API ツールのどちらかを使用して行います。CLI を使用した場合のリクエストの例を次に示します。

```
$ ec2-request-spot-instances -p 0.5 -t m1.xlarge -n 1 -b '/dev/sdb=snap-a123bcde:20:false' ami-8e1fece7
```

詳細については、以下のトピックを参照してください。

- [ブロックデバイスマッピング \(p. 643\)](#)
- [ec2-request-spot-instances](#)
- [RequestSpotInstances](#)

## ウォークスルー: AWS サービスでのスポットインスタンスの使用

### Abstract

AWS が提供する他のサービスと合わせて、Amazon EC2 スポットインスタンスを使用する方法を説明します。

#### Topics

- [Auto Scaling によるスポットインスタンスの管理 \(p. 166\)](#)
- [CloudFormation テンプレートを使用してスポットインスタンスを起動する \(p. 182\)](#)
- [スポットインスタンスで Amazon Elastic MapReduce のジョブフローを起動 \(p. 183\)](#)
- [Amazon Virtual Private Cloud でのスポットインスタンスの起動 \(p. 183\)](#)

AWS サービスはスポットインスタンスと一緒に使用できます。このセクションでは、Amazon EC2 スポットインスタンスが Auto Scaling、Amazon Elastic MapReduce および Amazon Virtual Private Cloud (Amazon VPC) などの他のサービスとどのように連携するかについて説明します。

## Auto Scaling によるスポットインスタンスの管理

### Abstract

Auto Scaling でスポットインスタンスを管理し、需要に基づいて容量をスケールリングします。

#### Topics

- [Auto Scaling をスポットインスタンスと一緒に管理するためのツール \(p. 167\)](#)
- [Auto Scaling でスポットインスタンスを起動する \(p. 168\)](#)
- [Auto Scaling により起動されたインスタンスの情報を取得する \(p. 171\)](#)
- [スポットインスタンスの入札価格の更新 \(p. 176\)](#)
- [スポット入札リクエストの予定設定 \(p. 178\)](#)
- [Auto Scaling を使用してスポットインスタンスに関する通知を取得する \(p. 179\)](#)

Auto Scaling 機能を使用してスポットインスタンスを管理できます。Auto Scaling を使用すると、Auto Scaling に入札させて、お客様の需要に基づいて必要な容量を拡大または縮小できます。

また、Auto Scaling のスケジューリング機能を使用することで、スポットインスタンスの入札時期や起動時期をより詳細に制御できます。また、インスタンスの起動や終了など、指定されたイベントが発生するたびに通知を送信する Amazon Simple Notification Service (Amazon SNS) -backed Auto Scaling 機能も使用できます。Auto Scaling の予定されているアクションを使用して、設定した時間に入札期限が切れるように設定できます。

スポットインスタンスで Auto Scaling を使用する際は、親しみのあるスポットインスタンスツールではなく、一部の Auto Scaling ツールを使用しなければならない可能性があります。また、使用できないスポットインスタンスオプションもあります。以下に違いの概要を記載します。

- 入札価格を設定する。Auto Scaling を使用してスポットインスタンスを起動する場合、Auto Scaling の起動設定で、入札価格を設定します。
- スポット価格と入札価格。これは現在の動作と同じです。スポットインスタンスの市場価格がお客様のインスタンス実行の入札価格を上回ると、Amazon EC2 はインスタンスを終了します。
- 入札価格を変更する。入札価格を変更する場合は、新しい起動設定を作成し、その設定を Auto Scaling グループに関連付ける必要があります。既存の起動設定を更新することはできません。
- 新しい入札価格とインスタンスの実行。新しい起動設定を作成して入札価格を変更すると、既に起動されているインスタンスは、実行されているインスタンスの入札価格が、スポットインスタンスに対する現在の市場価格より高い限り、継続して実行されます。
- スポットと Auto Scaling インスタンスの終了。インスタンスの入札価格がスポット価格より低くなると、Amazon EC2 はスポットインスタンスを終了します。Auto Scaling は、関連付けられている起動設定、インスタンスが実行されている請求時間の長さ、起動されているアベイラビリティゾーンを含む、基準の組み合わせに基づいて、インスタンスを終了します。Auto Scaling がインスタンスを終了する場合の詳細については、「[Auto Scaling Instance Termination](#)」を参照してください。
- スポットインスタンスを維持する。インスタンスが終了すると、Auto Scaling は、お客様が指定したご希望の容量を維持するため、そのインスタンスに代わる別のインスタンスを起動しようとします。

ただし、Auto Scaling が別のインスタンスの起動に成功するかどうかは、スポット価格と入札価格との比較に基づきます。入札価格がスポット価格を上回る場合、インスタンスは起動されますが、スポット価格が入札価格を上回る場合、その時点ではインスタンスは起動されません。

- 永続的入札。Auto Scaling は、Auto Scaling グループと起動設定が維持されている限り、入札を続行します。Auto Scaling グループは、お客様が維持したいインスタンス数を指定し、起動設定はスポットインスタンスの入札価格を指定します。

Auto Scaling の詳細については、『[Auto Scaling 開発者ガイド](#)』を参照してください。Auto Scaling を迅速に開始するには、『[Auto Scaling 入門ガイド](#)』を参照してください。

## Auto Scaling をスポットインスタンスと一緒に管理するためのツール

### Abstract

スポットインスタンスで Auto Scaling を管理するには、これらのツールを使用します。

Auto Scaling 機能をスポットインスタンスと一緒に使用するには、Auto Scaling コマンドラインインターフェイス ( CLI ) ツールを使用します。現在、AWS マネジメントコンソールは、Auto Scaling オブジェクトの作成と管理をサポートしていません。ただし、AWS マネジメントコンソールで、入札に関するほとんどの詳細と Auto Scaling によって起動された Amazon EC2 インスタンスを確認できます。

Auto Scaling CLI ツールを使用するには、パッケージをダウンロードして解凍し、Amazon EC2 CLI ツールの設定と同じように、コンピュータにツールを設定します。Auto Scaling CLI ツールと Amazon EC2 CLI ツールは別々のツールパッケージとして出荷されています。

次の表は、Auto Scaling CLI ツールの開始に役立つリソースの一覧です。[Auto Scaling 開発者ガイドの Auto Scaling Tools ツール](#)を参照してください。

### Auto Scaling CLI ツールを使用するためのリソース

タスク	リソース
Auto Scaling コマンドラインインターフェイスツールのダウンロード	アマゾン ウェブ サービスのサイトの <a href="#">Auto Scaling Command Line Tool</a>
Auto Scaling コマンドラインインターフェイスツールの設定とインストール	『 <a href="#">Auto Scaling 開発者ガイド</a> 』の「 <a href="#">Install the Command Line Interface</a> 」
Readme のインストールの手順を読む	<code>Install_Drive\AutoScaling-n.n.nn.n\README</code>

Auto Scaling CLI コマンドに関するヘルプを表示するには

Auto Scaling CLI ツールをインストールした後、コマンドラインにクエリを行うことによりコマンドに関する詳細を取得できます。

1. コマンドラインを開きます。
2. インストールドライブ\AutoScaling-n.n.nn.n\bin に移動します。
3. `as-cmd` を入力して、[Enter] を押します。

コマンドラインは、Auto Scaling CLI コマンドの一覧と各コマンドが実行する内容の短い説明を返します。

例えば、以下はこのセクションで使用する Auto Scaling CLI コマンドの一部です。

コマンド	説明
<code>as-create-launch-config</code>	指定された属性で、新しい起動設定を作成します。
<code>as-create-auto-scaling-group</code>	指定された名前とその他の属性で、新しい Auto Scaling グループを作成します。
<code>as-describe-auto-scaling-groups</code>	グループが存在する場合は、Auto Scaling グループの詳細を表示します。
<code>as-describe-auto-scaling-instances</code>	インスタンスが存在する場合は、Auto Scaling インスタンスの詳細を表示します。
<code>as-describe-scaling-activities</code>	グループに属する一連のまたはすべてのアクティビティの詳細を表示します。
<code>as-delete-auto-scaling-group</code>	グループにインスタンスがなく、進行中のスケールアップアクティビティがない場合は、指定された Auto Scaling グループは削除されます。

ドキュメントでコマンドシンボルを示す場合に使用する一般的な規則については、「[Document Conventions](#)」を参照してください。

4. `as-command-name --help` と入力します。

コマンドラインは、指定された Auto Scaling CLI コマンドの説明、構文情報、使用法の例を返します。

Amazon EC2 CLI ツールについては、以下を参照してください。

- Amazon EC2 CLI ツール。『*Amazon Elastic Compute Cloud コマンドラインリファレンス*』の「[Amazon Elastic Compute Cloud CLI Reference](#)」。
- スポットインスタンス固有の Amazon EC2 CLI ツール。 [スポットインスタンス用に AWS CLI ツールおよび EC2 CLI ツールを使用する \(p. 138\)](#)
- スポットインスタンスで使用可能なその他の Amazon EC2 ツール。 [スポットインスタンスの使用を開始する \(p. 135\)](#)

## Auto Scaling でスポットインスタンスを起動する

### Abstract

Auto Scaling 起動設定とスポットインスタンスを起動する Auto Scaling グループを作成します。

このセクションでは、Auto Scaling 起動設定とスポットインスタンスを起動する Auto Scaling グループを作成します。Auto Scaling コマンドラインインターフェイス (CLI) を使用して起動設定と Auto Scaling グループを作成し、これらおよびこれらが起動する Amazon EC2 インスタンスに関する情報を確認して取得します。

### 前提条件

起動設定や Auto Scaling グループの作成方法がよくわからない場合は、『*Auto Scaling 開発者ガイド*』の「[Basic Auto Scaling Configuration](#)」の手順を確認することをお勧めします。基本的なシナリオを使用して、ほとんどの Auto Scaling シナリオに必要なインフラストラクチャから開始してください。

お使いのコンピュータに Auto Scaling CLI ツールがインストールされていない場合は、インストールしてから次のチュートリアルを行ってください。詳細は、*Amazon Elastic Compute Cloud ユーザーガイド*

ドの「[Tools for Managing Auto Scaling with Spot Instances](#)」を参照してください。また、*Auto Scaling 開発者ガイド*の「[Install the Command Line Interface](#)」も参照できます。

このシナリオでは、次のタスクを実行します。

- [手順 1: 起動設定を作成する \(p. 169\)](#)
- [手順 2: Auto Scaling グループを作成する \(p. 170\)](#)

既に起動設定と Auto Scaling グループがある場合は、Auto Scaling を使用して実行できる別の関連するスポットインスタンスタスクがあります。

- [スポットインスタンスの入札価格の更新 \(p. 176\)](#)
- [スポット入札リクエストの予定設定 \(p. 178\)](#)
- [Auto Scaling を使用してスポットインスタンスに関する通知を取得する \(p. 179\)](#)
- [高度なタスク \(p. 186\)](#)

Auto Scaling の詳細については、「[What is Auto Scaling?](#)」(『*Auto Scaling 開発者ガイド*』)を参照してください。Auto Scaling を使用したシナリオの詳細については、『*Auto Scaling 開発者ガイド*』の「[Using Auto Scaling](#)」を参照してください。

### 手順 1: 起動設定を作成する

Auto Scaling 起動設定とは、新しい Amazon EC2 インスタンスの起動に必要なパラメータを含むテンプレートです。Auto Scaling グループに 1 度にアタッチできる起動設定は 1 つだけです。Auto Scaling を使用してインスタンスを起動するには、起動設定と Auto Scaling グループを作成する必要があります。

スポットインスタンスの入札を実行するには、`as-create-launch-config` Auto Scaling CLI コマンドを `--spot-price` オプションで使用します。インスタンスに支払う上限価格を指定します。この価格は、Auto Scaling によってインスタンスの入札に使用されますが、この価格は必ずしもインスタンスが起動されたときに、お客様が支払う価格ではありません。お客様が支払うのはスポット価格です。例えば、`m1.small` インスタンスに対して、`0.05 USD` で入札したとします。`m1.small` スポットインスタンスの現在の市場価格が `0.03 USD` であるか、`0.05 USD` 未満であれば、落札となり、1 時間あたり `0.03 USD` という現在の価格で課金されます。実際、現在の入札価格が市場価格を上回る限り、入札は遂行され、スポットインスタンスが起動されます。

スポット価格の履歴を確認すると、入札価格に関するガイダンスが得られます。これは、AWS Management Console、CLI または API を使用すれば可能です。詳細については、「[スポットインスタンス価格履歴の表示 \(p. 138\)](#)」を参照してください。

`as-create-launch-config` コマンドは次の引数をとります。

```
as-create-launch-config LaunchConfigurationName --image-id value --instance-type value [--spot-price value] [--iam-instance-profile value] [--block-device-mapping "key1=value1,key2=value2..." ] [--monitoring-enabled|--monitoring-disabled] [--kernel value ] [--key value ] [--ramdisk value] [--group value[,value...] ] [--user-data value] [--user-data-file value] [General Options]
```

必要なオプションは、起動設定名、イメージ ID、およびインスタンスタイプだけです。このウォークスルーでは、以下を指定してください。

- 起動設定名 = `spot1c-5cents`



## Note

Auto Scaling がインスタンスを起動する場合、オンデマンドインスタンスとスポットインスタンスの違いは区別されません。スポットインスタンスを識別するには、スポットと入札価格を含む起動設定名を割り当てることを検討してください。

- イメージ ID = ami-e565ba8c

イメージ ID はインスタンスの起動に使用される Amazon マシンイメージ (AMI) を識別します。AMI がなく、適切な AMI を探す場合は、「[適切な AMI の検索 \(p. 60\)](#)」を参照してください。

- インスタンスタイプ = m1.small
- スポット価格 = \$0.05

このパラメータはオプションです。スポットインスタンスを起動する場合は、スポットインスタンスの入札に使用するスポット入札価格を指定する必要があります。スポットインスタンス入札価格は、US ドルで指定する必要があります。

コマンドは、次の例のようになります。

```
as-create-launch-config spotlc-5cents --image-id ami-e565ba8c --instance-type m1.small --spot-price "0.05"
```

次の例のような確認が表示されます。

```
OK-Created launch config
```

## 手順 2: Auto Scaling グループを作成する

Auto Scaling グループとは、類似した特性を共有し、特定のスケールアクションを適用する Amazon EC2 インスタンスのコレクションです。Auto Scaling グループを使用して、インスタンス数を自動的に拡大、縮小するか、固定数のインスタンスを維持できます。Auto Scaling グループに 1 度にアタッチできる起動設定は 1 つだけです。Auto Scaling を使用して Amazon EC2 インスタンスを起動するには、起動設定と Auto Scaling グループを作成する必要があります。

as-create-auto-scaling-group コマンドは次の引数をとります。

```
as-create-auto-scaling-group AutoScalingGroupName --availability-zones value[,value...] --launch-configuration value --max-size value --min-size value [--default-cooldown value] [--desired-capacity value] [--grace-period value] [--health-check-type value] [--load-balancers value[, value]] [--placement-group value] [--vpc-zone-identifier value] [General Options]
```

このコマンドでは、Auto Scaling グループの名前、起動設定、1 つ以上の利用可能ゾーン、最小グループサイズ、および最大グループサイズを指定する必要があります。選択した利用可能ゾーンは、Auto Scaling インスタンスの物理的な場所を決定します。最小および最大グループサイズでは、Auto Scaling グループが持つ最小および最大インスタンス数を Auto Scaling に指示します。

希望する容量は、as-create-auto-scaling-group コマンドの重要な構成要素です。オプションのパラメータですが、希望する容量は、最初に行われるインスタンス数を Auto Scaling に指示します。Auto Scaling グループで実行するインスタンス数を調整するには、--desired-capacity の値を変更します。--desired-capacity を指定しないと、この値は最小グループサイズと同じになります。

as-create-auto-scaling-group コマンドの構文の詳細については、[Auto Scaling 開発者ガイドの Basic Auto Scaling Configuration](#) を参照してください。また、コマンドライン

`as-create-auto-scaling-group --help` からヘルプ情報を取得できます。詳細については、[Auto Scaling CLI ツールを使用するためのリソース \(p. 167\)](#) を参照してください。

このウォークスルーでは、コマンドにこれらの値を指定します。

- Auto Scaling グループ名 = `spotasg`



#### Note

Auto Scaling がインスタンスを起動する場合、オンデマンドインスタンスとスポットインスタンスの違いは区別されません。スポットインスタンスを識別するには、スポットインスタンスを起動する Auto Scaling グループに、スポット固有の名前を割り当てることを検討してください。

- 起動設定名 = `spot1c-5cents`
- 利用可能ゾーン = `us-east-1a,us-east-1b`
- 最大サイズ = 5
- 最小サイズ = 1
- 希望する容量 = 3

コマンドは、次の例のようになります。

```
as-create-auto-scaling-group spotasg --launch-configuration spot1c-5cents --availability-zones "us-east-1a,us-east-1b" --max-size 5 --min-size 1 --desired-capacity 3
```

次の例のような確認が表示されます。

```
OK-Created AutoScalingGroup
```

次は何をしますか。

- [Auto Scaling により起動されたインスタンスの情報を取得する \(p. 171\)](#)
- [スポットインスタンスの入札価格の更新 \(p. 176\)](#)
- [スポット入札リクエストの予定設定 \(p. 178\)](#)
- [Auto Scaling を使用してスポットインスタンスに関する通知を取得する \(p. 179\)](#)
- [高度なタスク \(p. 186\)](#)

Auto Scaling の詳細については、「[What is Auto Scaling?](#)」(『[Auto Scaling 開発者ガイド](#)』) を参照してください。

## Auto Scaling により起動されたインスタンスの情報を取得する

### Abstract

Auto Scaling によって起動された起動設定、Auto Scaling グループ、および Amazon EC2 インスタンスに関する情報を取得します。

Auto Scaling CLI ツールを使用すると、起動設定、Auto Scaling グループ、および Auto Scaling によって起動された Amazon EC2 インスタンスに関する情報を取得できます。

このセクションでは、次の Auto Scaling CLI コマンドを使用して、Auto Scaling が実行し起動したスポット価格の入札と、スポットインスタンスに関する情報を取得します。

- `as-describe-scaling-activities` - このコマンドが返す Auto Scaling アクティビティに関する情報を使用して、Auto Scaling によって送信された入札のステータスを確認できます。
- `as-describe-auto-scaling-groups` - このコマンドが返す Auto Scaling グループに関する情報を使用して、Auto Scaling によって仕様どおりにスポットインスタンスが起動されていることを確認できます。

Auto Scaling がお客様のために作成している入札のステータスを確認するには

`as-describe-scaling-activities` コマンドは、指定された Auto Scaling グループに対して Auto Scaling が実行したアクティビティを一覧します。

以下に基本的な構文を示します。

```
as-describe-scaling-activities [ActivityIds [ActivityIds ...]]  
[--auto-scaling-group value] [--max-records value] [General Options]
```

Auto Scaling グループの指定およびアクティビティ ID の指定はオプションです。Auto Scaling グループを指定しないと、このコマンドではアカウントに関連付けられたすべての Auto Scaling グループに関するすべてのアクティビティが返されます。Auto Scaling グループを指定すると、その Auto Scaling グループに関するアクティビティのみが一覧表示されます。

このシナリオでは、入札状況の確認に `as-describe-scaling-activities` コマンドを使用しています。Auto Scaling グループが 1 つしかなく ( `spotasg` )、すべてのアクティビティを一覧表示とします。

1. コマンドラインを開いて、Auto Scaling CLI ツールディレクトリの `bin` フォルダに移動します。
2. コマンド `as-describe-scaling-activities --auto-scaling-group spotasg --headers` を入力します。

次の例のような情報が表示されます。

ACTIVITY NAME	ACTIVITY-ID	MESSAGE	END-TIME	GROUP-
ACTIVITY	31bcbb67-7f50-4b88-ae7e-e564a8c80a90	WaitingForSpotInstanceId Placed Spot instance request: sir-fc8a3014. Waiting for instance(s)		spotasg
ACTIVITY	770bbeb5-407c-404c-a826-856f65db1c57	WaitingForSpotInstanceId Placed Spot instance request: sir-69101014. Waiting for instance(s)		spotasg
ACTIVITY	597e4ebd-220e-42bc-8ac9-2bae4d20b8d7	Successful	2012-05-23T17:40:22Z	spotasg

この応答では、入札が実行され、入札のうちの 1 つが成功し、Auto Scaling は他に 2 つの入札のために待機していることがわかります。



#### Note

`as-describe-scaling-activities` コマンドによって、失敗したアクティビティが含まれる一覧が返された場合は、起動設定で指定したデータを確認してください。以下に例を示します。

- Amazon マシンイメージ (AMI) が有効でなくなっている可能性があります。
- 起動設定で指定された入札価格が、スポット価格を下回っています。

3. `as-describe-scaling-activities` コマンドを後で再び実行した場合は、以下の例に示すような情報が返される可能性があります。

ACTIVITY NAME	ACTIVITY-ID CODE	END-TIME	GROUP-
ACTIVITY Successful	90630906-b40f-41a6-967a-cd6534b2dfca	2012-06-01T02:32:15Z	spotasg
ACTIVITY Successful	a1139948-ad0c-4600-9efe-9dab8ce23615	2012-06-01T00:48:02Z	spotasg
ACTIVITY Successful	33001e70-6659-4494-a817-674d1b7a2f58	2012-06-01T02:31:11Z	spotasg

出力には、一覧表示されたアクティビティの成功が示されます。`spotasg`は、スポット入札価格で起動設定を使用する Auto Scaling グループだとわかっているため、アクティビティは Auto Scaling が実行した入札を示すものと推測できます。

4. アクティビティとインスタンスの詳細を入手するには、`--show-xml` の `as-describe-scaling-activities` オプションを使用します。次のコマンドを入力します：  
`as-describe-scaling-activities --auto-scaling-group spotasg --show-xml`

次の例のような情報が表示されます。

```
<DescribeScalingActivitiesResponse xmlns="http://autoscaling.&api-do
main;/doc/2011-01-01/">
  <DescribeScalingActivitiesResult>
    <NextToken>b5a3b43e-10c6-4b61-8e41-2756db1fb8f5</NextToken>
    <Activities>
      <member>
        <StatusCode>Successful</StatusCode>
        <Progress>0</Progress>
        <ActivityId>90630906-b40f-41a6-967a-cd6534b2dfca</ActivityId>
        <StartTime>2012-06-01T00:48:21.942Z</StartTime>
        <AutoScalingGroupName>spotasg</AutoScalingGroupName>
        <Cause>At 2012-06-01T00:48:21Z a difference between desired and ac
tual capacity changing the desired capacity, increasing the capacity from
2 to 3.</Cause>
        <Details>{}</Details>
        <Description>Launching a new EC2 instance: i-fe30d187</Description>

        <EndTime>2012-06-01T02:32:15Z</EndTime>
      </member>
      <member>
        <StatusCode>Successful</StatusCode>
        <Progress>0</Progress>
        <ActivityId>a1139948-ad0c-4600-9efe-9dab8ce23615</ActivityId>
        <StartTime>2012-06-01T00:47:51.293Z</StartTime>
        <AutoScalingGroupName>spotasg</AutoScalingGroupName>
        <Cause>At 2012-06-01T00:47:51Z an instance was taken out of service
in response to a system health-check.</Cause>
        <Details>{}</Details>
        <Description>Terminating EC2 instance: i-88ce28f1</Description>
        <EndTime>2012-06-01T00:48:02Z</EndTime>
      </member>
      <member>
        <StatusCode>Successful</StatusCode>
        <Progress>0</Progress>
        <ActivityId>33001e70-6659-4494-a817-674d1b7a2f58</ActivityId>
        <StartTime>2012-06-01T00:46:19.723Z</StartTime>
```

```
<AutoScalingGroupName>spotasg</AutoScalingGroupName>
  <Cause>At 2012-06-01T00:46:19Z a difference between desired and ac
tual capacity changing the desired capacity, increasing the capacity from
2 to 3.</Cause>
  <Details>{}</Details>
  <Description>Launching a new EC2 instance: i-2c30d155</Description>

  <EndTime>2012-06-01T02:31:11Z</EndTime>
</member>
...
</Activities>
</DescribeScalingActivitiesResult>
<ResponseMetadata>
  <RequestId>d02af4bc-ad8f-11e1-85db-83e1968c7d8d</RequestId>
</ResponseMetadata>
</DescribeScalingActivitiesResponse>
```

XML 出力には、スポットと Auto Scaling アクティビティの詳細が表示されます。

- Cause: At 2012-06-01T00:48:21Z a difference between desired and actual capacity changing the desired capacity, increasing the capacity from 2 to 3. Description: Launching a new EC2 instance: i-fe30d187

インスタンスが終了し、インスタンス数が希望する容量を下回ると、Auto Scaling は新しいインスタンスを起動して、実行中のインスタンスの合計数が希望する容量で指定したレベルに戻るようになります。

- Cause: At 2012-06-01T00:47:51Z an instance was taken out of service in response to a system health-check. Description: Terminating EC2 instance: i-88ce28f1

Auto Scaling は、Auto Scaling グループでインスタンスのヘルスステータスをモニタリングすることによって希望するインスタンス数を維持します。インスタンスが *unhealthy* である、または終了したという通知を Auto Scaling が受信した場合、Auto Scaling は *unhealthy* なインスタンスに代わる別のインスタンスを起動します。詳細については、『[Auto Scaling 開発者ガイド](#)』の「[Configure Health Checks for Your Auto Scaling group](#)」を参照してください。



#### Note

Auto Scaling では、スケーリングアクティビティの結果ではない、インスタンス終了の原因を提供します。これには、スポット価格が入札価格を上回ったために終了したインスタンスも含まれます。

スポット価格がインスタンスの入札価格を上回ったことによって終了したインスタンスを置き換えようとするとき、Auto Scaling は現在の起動設定で指定された入札を実行し、別のインスタンスを起動して、希望する容量を維持しようとします。

Auto Scaling が仕様に従ってスポットインスタンスを起動していることを確認するには

`as-describe-auto-scaling-groups` を使用します。コマンドによって、グループと起動したインスタンスの詳細が表示されます。`as-describe-auto-scaling-groups` コマンドに関する詳細については、『[Auto Scaling 開発者ガイド](#)』の「[Verify Auto Scaling Group Creation](#)」を参照してください。

1. コマンドラインを開いて、Auto Scaling CLI ツールディレクトリの `bin` フォルダに移動します。
2. コマンド `as-describe-auto-scaling-groups spotasg --headers` を入力します。



## Note

--headers オプションでは、返されているデータの内容がわかるカラム名が表示されません。

次の例のような情報が表示されます。

AUTO-SCALING-GROUP	GROUP-NAME	LAUNCH-CONFIG	AVAILABILITY-ZONES		
MIN-SIZE	MAX-SIZE	DESIRED-CAPACITY			
AUTO-SCALING-GROUP	spotasg	spotlc-5cents	us-east-1b,us-east-1a		
1	5	3			
INSTANCE	INSTANCE-ID	AVAILABILITY-ZONE	STATE	STATUS	LAUNCH-CONFIG
INSTANCE	i-2c30d155	us-east-1a	InService	Healthy	spotlc-5cents
INSTANCE	i-fe30d187	us-east-1a	InService	Healthy	spotlc-5cents
INSTANCE	i-c630d1bf	us-east-1a	InService	Healthy	spotlc-5cents

指定したとおり、Auto Scaling が us-east-1a で 3 つのインスタンスを起動し、すべて実行されていることがわかります。

- また、Auto Scaling によって起動されたスポットインスタンスに関する詳細は、`as-describe-auto-scaling-instances` コマンドを使用して表示することもできます。

以下に基本的な構文を示します。

```
as-describe-auto-scaling-instances [InstanceIds [InstanceIds ...]]  
[--max-records value] [General Options]
```

`InstanceIds` の指定は、オプションです。このコマンドを指定すると、インスタンスが存在する場合、そのインスタンスに関する情報が表示されます。`InstanceIds` を指定しないと、このコマンドには、Auto Scaling アカウントに関連付けられたすべてのインスタンスに関する情報が返されます。

このウォークスルーでは、起動設定と Auto Scaling グループを 1 つ作成し、すべてのスポットインスタンスに関する詳細を確認すると仮定しています。

コマンドは、次の例のようになります。

```
as-describe-auto-scaling-instances --headers
```

取得できる情報は次の例のようになります。

INSTANCE	INSTANCE-ID	GROUP-NAME	AVAILABILITY-ZONE	STATE	STATUS
INSTANCE	i-2c30d155	spotasg	us-east-1a	InService	HEALTHY
	spotlc-5cents				
INSTANCE	i-c630d1bf	spotasg	us-east-1a	InService	HEALTHY
	spotlc-5cents				
INSTANCE	i-fe30d187	spotasg	us-east-1a	InService	HEALTHY
	spotlc-5cents				

次は何をしますか。

- [スポットインスタンスの入札価格の更新 \(p. 176\)](#)
- [スポット入札リクエストの予定設定 \(p. 178\)](#)

- [Auto Scaling を使用してスポットインスタンスに関する通知を取得する \(p. 179\)](#)
- [高度なタスク \(p. 186\)](#)

## スポットインスタンスの入札価格の更新

### Abstract

新しい起動設定を作成してスポットインスタンスの入札価格を更新します。

Auto Scaling の起動設定は変更できません。スポットインスタンスに対する入札価格を変更する場合は、新しい起動設定を作成する必要があります。

一連のスポットインスタンスを起動して、長時間中断せずに実行される可能性を高くしたい場合は、入札価格を高くします。これには、このウォークスルーで実行した手順を使用して、新しい起動設定を作成する必要があります ( 詳細については、「[手順 1: 起動設定を作成する \(p. 169\)](#)」を参照してください )。

次の値を指定します。

- 起動設定名 = spotlc-7cents
- イメージ ID = ami-e565ba8c



#### Note

AMI がなく、適切な AMI を探す場合は、「[適切な AMI の検索 \(p. 60\)](#)」を参照してください。

- インスタンスタイプ = m1.small
- スポット価格 = 0.07 USD

コマンドは、次の例のようになります。

```
as-create-launch-config spotlc-7cents --image-id ami-e565ba8c --instance-type m1.small --spot-price "0.07"
```

次の例のような確認が表示されます。

```
OK-Created launch config
```

新しい起動設定の作成に成功したら、その新しい起動設定を指定する新しい Auto Scaling グループを作成します。

コマンドは、次の例のようになります。

```
as-create-auto-scaling-group spotasg-7cents --launch-configuration spotlc-7cents --availability-zones "us-east-1a,us-east-1b" --max-size 5 --min-size 10 --desired-capacity 3
```

次の例のような確認が表示されます。

Amazon Elastic Compute Cloud ユーザーガイド  
ウォークスルー: AWS サービスでのスポットインスタンス  
の使用

```
OK-Created AutoScalingGroup
```

スポット入札のステータスと、Auto Scaling がお客様に代わって発行した入札の一覧を表示するには、Auto Scaling グループを作成した後すぐに `as-describe-scaling-activities` を実行します。

コマンドは、次の例のようになります。

```
as-describe-scaling-activities --headers
```

受理されていない入札が存在する場合は、次の例のような情報が返されます。

ACTIVITY NAME	ACTIVITY-ID CODE	MESSAGE	END-TIME	GROUP-
ACTIVITY	5879cc50-1e40-4539-a754-1cb084flaecd	WaitingForSpotInstanceId Placed Spot instance request: sir-93828812. Waiting for instance(s)		spotasg-
ACTIVITY	777fbelb-7a24-4aaf-b7a9-d368d0511878	WaitingForSpotInstanceId Placed Spot instance request: sir-016cf812. Waiting for instance(s)		spotasg-
ACTIVITY	f4b00f81-eaea-4421-80b4-a2e3a35cc782	WaitingForSpotInstanceId Placed Spot instance request: sir-cf60ea12. Waiting for instance(s)		spotasg-
ACTIVITY	31bcbb67-7f50-4b88-ae7e-e564a8c80a90	WaitingForSpotInstanceId Placed Spot instance request: sir-fc8a3014. Waiting for instance(s)		spotasg
ACTIVITY	770bbeb5-407c-404c-a826-856f65db1c57	WaitingForSpotInstanceId Placed Spot instance request: sir-69101014. Waiting for instance(s)		spotasg
ACTIVITY	597e4ebd-220e-42bc-8ac9-2bae4d20b8d7	Successful	2012-05-23T17:40:22Z	spotasg
ACTIVITY	ecal58b4-a6f9-4ec5-a813-78d42c1738e2	Successful	2012-05-23T17:40:22Z	spotasg
ACTIVITY	1a5bd6c6-0b0a-4917-8cf0-eee1044a179f	Successful	2012-05-23T17:22:19Z	spotasg
ACTIVITY	c285bf16-d2c4-4ae8-acad-7450655facb5	Successful	2012-05-23T17:22:19Z	spotasg
ACTIVITY	127e3608-5911-4111-906e-31fb16d43f2e	Successful	2012-05-23T15:38:06Z	spotasg
ACTIVITY	bfb548ad-8bc7-4a78-a7db-3b41f73501fc	Successful	2012-05-23T15:38:07Z	spotasg
ACTIVITY	82d2b9bb-3d64-46d9-99b6-054a9ecf5ac2	Successful	2012-05-23T15:30:28Z	spotasg
ACTIVITY	95b7589b-f8ac-49bc-8c83-514bf664b4ee	Successful	2012-05-23T15:30:28Z	spotasg
ACTIVITY	57bbf77a-99d6-4d94-a6db-76b2307fb9de	Successful	2012-05-23T15:16:34Z	spotasg
ACTIVITY	cdef758e-0be2-416e-b402-0ef521861039		2012-05-23T15:16:17Z	spotasg

Successful				
ACTIVITY	d7e0a3ed-7067-4583-8a87-1561b3de2aed	2012-05-23T14:51:46Z	spotasg	Successful
ACTIVITY	da5471ab-482c-4680-b430-99e4173d2bd7	2012-05-23T14:52:48Z	spotasg	Successful
ACTIVITY	78701f3d-a747-46e1-8b0f-8aff22834f46	2012-05-23T14:38:17Z	spotasg	Successful
ACTIVITY	274d4772-3614-4f5c-8858-026b33635be3	2012-05-23T14:38:16Z	spotasg	Successful
ACTIVITY	1024abb2-bf84-4fae-b717-a398bac91c4f	2012-05-23T14:22:39Z	spotasg	Successful

入札は、sir-93828812 や sir-016cf812 のような値で表現されます。

スポットインスタンスに新しい入札価格を設定する新しい起動設定を作成するときに、既に別の価格に基づいて実行されているスポットインスタンスがあると、これらのインスタンスは引き続き実行され、スポット価格が実行中のインスタンスの基となる入札価格を超えた場合のみ、そのインスタンスは終了します。

次は何をしますか？

- [スポット入札リクエストの予定設定 \(p. 178\)](#)
- [Auto Scaling を使用してスポットインスタンスに関する通知を取得する \(p. 179\)](#)
- [高度なタスク \(p. 186\)](#)

## スポット入札リクエストの予定設定

### Abstract

Auto Scaling を使用してスポット入札リクエストをスケジュールし、履歴で価格が低下する時間帯を利用します。

Auto Scaling は特定のインスタンス数を特定の時間に起動するように設定できます。この機能は、例えば、履歴で価格が低い時間帯を活用したり、特定の時間にスポットインスタンスを終了させたい場合に便利です。

Auto Scaling CLI コマンド `as-put-scheduled-update-group-action` を使用して、スケジュールを設定します。以下に基本的な構文を示します。

```
as-put-scheduled-update-group-action ScheduledActionName --auto-scaling-group value [--desired-capacity value] [--end-time value][--max-size value][--min-size value] [--recurrence value][--start-time value][--time value][General Options]
```

このシナリオでは、次の値を使用します。

- 予定されているアクション名: as-spotbid-schedule
- Auto Scaling グループ: spotasg
- 開始時間: 2012-05-15T19:10:00Z
- 終了時間: 2012-05-15T19:15:00Z
- 希望する容量:20

コマンドは、次の例のようになります。

```
as-put-scheduled-update-group-action as-spotbid-schedule --auto-scaling-group
spotasg --desired-capacity 20 --start-time 2012-05-15T19:10:00Z --end-time 2012-
05-15T19:15:00Z
```

次の例のような確認が表示されます。

```
OK-Put Scheduled Update Group Action
```

予定されているアクションをチェックするには、`as-describe-scheduled-actions` を実行します。

次の例のような情報を受け取ります。

```
UPDATE-GROUP-ACTION spotasg as-spotbid-schedule 2012-05-15T19:10:00Z 20
```

次は何をしますか。

- [Auto Scaling により起動されたインスタンスの情報を取得する \(p. 171\)](#)
- [スポットインスタンスの入札価格の更新 \(p. 176\)](#)
- [Auto Scaling を使用してスポットインスタンスに関する通知を取得する \(p. 179\)](#)
- [高度なタスク \(p. 186\)](#)

## Auto Scaling を使用してスポットインスタンスに関する通知を取得する

### Abstract

Auto Scaling を使用してインスタンスが中断および起動されるときに通知を受け取ります。

Auto Scaling を設定することで、インスタンスが終了および起動したときに通知が送信されるようにすることができます。スポット価格が上昇して、入札価格を上回った場合、Amazon EC2 はその入札価格に基づいて起動されたインスタンスを終了します。スポットインスタンスが終了すると、Auto Scaling は入札を送信し、代替インスタンスを起動して、Auto Scaling グループに指定された容量を確保しようとします。Auto Scaling は、インスタンスの起動または終了イベントが発生したときに通知するように設定できます。

スポットインスタンスの通知を取得する方法には 2 つあります。

- Auto Scaling
- AWS SDK サンプルコードツール

### AWS Java SDK のスポット通知サンプル

AWS Java SDK を使用して、スポットインスタンスの使用状況をモニタリングするアプリケーションをユースケースに合わせてカスタマイズして開発することもできます。スポット通知のサンプルアプリケーションは、Amazon EC2 インスタンスのステータス、スポットインスタンスのリクエストおよびスポット価格の変動をモニタリングする技術を示す Java アプリケーションです。アプリケーションは文書化されており、[How to Track Spot Instance Activity with the Spot-Notifications Sample Application](#) から無料でダウンロードできます。アプリケーションは目的に合わせて自由に変更したり、スポットインスタンスのモニタリング用のアプリケーション開発のガイドとして使用できます。詳細については、[AWS SDK for Java](#) を参照してください。

Auto Scaling グループを設定して、スポットインスタンスに関する通知が送信されるようにする

ウォークスルーのこの部分では、Amazon SNS を設定して、E メール通知が送信されるようにする方法を説明します。

これを行うには、以下のことが必要です。

**Amazon リソース名 (ARN)**。Amazon Simple Notification Service (Amazon SNS) トピックを作成すると、ARNが生成されます。トピックとは、メッセージを送信し、通知を受信する通信チャネルです。発行者と受信者が相互に意思疎通するためのアクセスポイントを提供します。Eメールアドレスなどのエンドポイントでは、トピックに対して発行されたメッセージを受信できるように、トピックをサブスクライブする必要があります。トピックを作成するには、*Amazon Simple Notification Service 入門ガイド* の [トピックの作成](#) を参照してください。

**Auto Scaling グループ** このウォークスルーで作成した Auto Scaling グループを使用します。

**通知の設定**。Auto Scaling グループを設定して、`as-put-notification-configuration` CLI コマンドまたは `PutNotificationConfiguration` API アクションを呼び出すと、指定されたイベントが発生した場合に通知が送信されます。通知設定の手順については、このセクションで後述します。コマンドの詳細については、*Auto Scaling API リファレンス* の [PutNotificationConfiguration](#) を参照してください。

**Auto Scaling 通知タイプの一覧**。これらが通知の送信を生じさせる通知タイプです。

Auto Scaling を使用して通知を設定する

1. Amazon SNS トピックを作成して ARN が作成されたら、通知設定を行う準備が整っています (トピックを作成するには、*Amazon Simple Notification Service 入門ガイド* の [トピックの作成](#) を参照してください)。

Auto Scaling グループを設定して、指定されたイベントが発生したときに通知を送信するには、`as-put-notification-configuration` CLI コマンドを使用します。

`as-put-notification-configuration` コマンドは次の引数をとります。

```
as-put-notification-configuration AutoScalingGroupName --notification-types value --topic-arn topic-ARN [General Options]
```

Auto Scaling グループ名、ARN、および通知タイプを指定する必要があります。

この例では、以下を指定してください。

- Auto Scaling グループ名: `spotasg`

どの Auto Scaling グループについて通知を取得するかを指定します。このウォークスルーでは、`spotasg` Auto Scaling グループに対してインスタンスが起動され、終了されたときに Auto Scaling に通知を送信させます。

- ARN: `arn:placeholder:MyTopic`



#### Note

ARN とは、AWS リソース独自の識別子です。ARN プレースホルダを独自の ARN で置き換えます。

- 通知タイプ: `autoscaling:EC2_Instance_Launch`, `autoscaling:EC2_Instance_Terminate`  
通知タイプは、発生した場合に Auto Scaling に E メールを送信させるイベントです。

コマンドプロンプトを開いて、`as-put-notification-configuration` コマンドを入力します。

```
as-put-notification-configuration spotasg--topic-arn arn:placeholder:MyTopic
--notification-types autoscaling:EC2_INSTANCE_LAUNCH, autoscaling:EC2_IN
STANCE_TERMINATE
```

Auto Scaling は以下を返します。

```
OK-Put Notification Configuration
```

これで、インスタンスが起動または終了されたときに、arn:placeholder:MyTopic ARN でエンドポイントが受信する通知を送信する通知設定ができました。

## 2. 通知設定を確認します。

指定されたイベントが発生したときに、Auto Scaling グループに関連する通知アクションを確認するには、as-describe-notification-configurations を使用します。

as-describe-notification-configurations コマンドは次の引数をとります。

```
as-describe-notification-configurations [--auto-scaling-groups
value[,value...]] [--maxrecords value] [General Options]
```

Auto Scaling グループを指定した場合は、リストした Auto Scaling グループのあらゆる通知設定の完全なリストがこのコマンドによって返されます。Auto Scaling グループ名を指定しないと、サービスはすべての Auto Scaling グループの詳細を返します。取得するページが複数ある場合、このコマンドはトークンも返します。次のページを取得するには、返されたトークンを next-token 引数として使用し、このアクションを再度呼び出します。このウォークスルーでは、以下を指定してください。

- Auto Scaling グループ名: spotasg

コマンドプロンプトを開いて、as-describe-notification-configurations コマンドを入力します。

```
as-describe-notification-configurations --auto-scaling-groups spotasg -headers
```

Auto Scaling は以下を返します。

```
NOTIFICATION-CONFIG GROUP-NAME TOPIC-ARN NOTIFICATION-TYPE-NAME
NOTIFICATION-CONFIG spotasg arn:placeholder:spotasg autoscaling:EC2_IN
STANCE_LAUNCH
NOTIFICATION-CONFIG spotasg arn:placeholder:spotasg autoscaling:EC2_IN
STANCE_TERMINATE
```

これで、spotasg Auto Scaling グループに対する通知設定が行われたことが確認できました。

Auto Scaling グループで指定したように、希望する容量に達するか、または維持するために Auto Scaling がインスタンスを起動すると、Amazon SNS はそのインスタンスに関する情報が記載された通知を E メールアドレスに送信します。Eメールの受信箱でこの情報を確認してから、as-describe-auto-scaling-group を実行して、Auto Scaling グループの現在のインスタンスに関する情報を取得し、Eメールに一覧表示されたインスタンスが実際に存在することを確認できます。

次は何をしますか。

- [Auto Scaling により起動されたインスタンスの情報を取得する \(p. 171\)](#)
- [スポットインスタンスの入札価格の更新 \(p. 176\)](#)
- [スポット入札リクエストの予定設定 \(p. 178\)](#)

- [高度なタスク \(p. 186\)](#)

## CloudFormation テンプレートを使用してスポットインスタンスを起動する

### Abstract

CloudFormation テンプレートを使用して、スポットインスタンスと、アプリケーションに必要な AWS リソースを起動します。

AWS CloudFormation テンプレートを使用して、スポットインスタンスと、アプリケーションに必要な AWS リソースを起動することができます。テンプレートとは、グループとしてデプロイする AWS リソースのスタックと呼ばれるコレクションについて説明するテキストファイルです。AWS CloudFormation テンプレートの詳細については、[Learn Template Basics](#) を参照してください。AWS CloudFormation の使用の詳細については、「[AWS CloudFormation ユーザーガイド](#)」を参照してください。

テンプレートは最初から記述することも、[AWS CloudFormation Sample Template Library](#) のテンプレートの例を使用することもできます。AWS CloudFormation の次のテンプレートは、スポットインスタンスを活用しています。

- [Asynchronous Queue-Based Processing](#)

このテンプレートでは、キューにある作業 (メッセージ) をモニタリングするオートスケールされたワーカーを作成します。アプリケーションはキューにある作業の量に基づいてオートスケールされません。作業があると、Auto Scaling は拡張し、作業がないと、Auto Scaling は縮小します。各メッセージには、実行するコマンド/スクリプト、入力ファイルの場所、結果の出力場所が含まれます。

このテンプレートをダウンロードするには、[Asynchronous Queue-Based Processing](#) にアクセスしてください。

- [Bees with Machine Guns](#)

このテンプレートはロードバランサー、コントローラー、ロードバランサーの背後のインスタンスを作成し、攻撃を起動して開始し、Amazon Simple Storage Service ( Amazon S3 ) にログを保存してからシャットダウンします (有効にされている場合)。このテンプレートでは、インスタンスは Amazon Linux AMI を使用し、設定には、既に AWS Identity と Access Management ( IAM ) SSL 認証があることが必要です。このテンプレートを変更して、SSL 依存関係を削除できます。



### Note

このテンプレートを起動するには、このタスク専用で作成した新しい SSH 秘密キーが必要です。テンプレートによって作成されたインスタンスにログインして、この秘密キーの情報を指定する必要があります。

このテンプレートをダウンロードするには、[Bees with Machine Guns](#) にアクセスしてください。

- [Grid Computing for Spot Instances](#)

このテンプレートでは、Star Cluster を使用して Amazon EC2 インスタンスのクラスターの起動とブートストラップを行い、スポット価格設定を使用して、高いパフォーマンスのコンピュータによるタスクを実行します。使用するインスタンス数、インスタンスサイズおよびスポット価格のみを提供する必要があります。

このテンプレートをダウンロードするには、[Grid Computing for Spot Instances](#) にアクセスしてください。

これらのテンプレートから、AWS Management Console、AWS CloudFormation コマンドラインインタールフェイスツール、または AWS CloudFormation API を使用して、スタックを作成できます。

これらのテンプレートを使用する前に、以下を実行する必要があります。

1. AWS アカウントを保有して、AWS CloudFormation にサインアップする。
2. 使用するテンプレートを決定する。
3. スタックに必要なパラメータを入力する。
4. スタックを作成する。

AWS CloudFormation を使い始める場合は、[AWS CloudFormation ユーザーガイド](#) の [Getting Started with AWS CloudFormation](#) を参照してください。

## スポットインスタンスで Amazon Elastic MapReduce のジョブフローを起動

### Abstract

スポットインスタンスで Amazon Elastic MapReduce ジョブフローを開始し、コンピュータクラスターのセットアップと管理を簡素化します。

スポットインスタンスで Amazon Elastic MapReduce のジョブフローを起動できます。Amazon EMR とは、データ分析ツールです。コンピュータクラスター、ソースデータ、計算ツールを簡潔に設定して管理し、高度なデータ処理ジョブを簡潔に実装するのに役立ちます。詳細については、[What is Amazon EMR?](#) を参照してください。

さらに、Amazon EMR クラスターおよび AWS Data Pipeline でスポットインスタンスを使用できます。詳細については、「[Amazon EC2 Spot Instances with Amazon EMR Clusters and AWS Data Pipeline](#)」を参照してください。

### 動画クイックルック: Amazon Elastic MapReduce でのスポットインスタンスの使用

以下の動画は、Amazon EMR でのスポットインスタンスの動作について説明し、AWS Management Console からスポットインスタンスのジョブフローを起動するプロセスを順番に案内します。[Amazon Elastic MapReduce でのスポットインスタンスの使用](#)

## Amazon Virtual Private Cloud でのスポットインスタンスの起動

### Abstract

Amazon Virtual Private Cloud の仮想ネットワーキング環境でスポットインスタンスを起動します。

Amazon Virtual Private Cloud ( Amazon VPC ) を使用すると、アマゾン ウェブ サービス ( AWS ) クラウドの、隔離されたプライベートなセクションに仮想ネットワーク環境を定義することができます。この仮想ネットワークは完全に制御でき、高度なセキュリティ機能やネットワークアクセスコントロールをインスタンスレベルおよびサブネットレベルで使用することができます。Amazon VPC の詳細については、『[Amazon Virtual Private Cloud ユーザーガイド](#)』を参照してください。

スポットインスタンスを使用する際に Amazon VPC の機能を活用するには、VPC でインスタンスが起動することをスポットリクエストで指定します。VPC でスポットインスタンスが起動することを確認するには。

- AWS アカウントでデフォルトの VPC がサポートされているかどうかを特定します。詳細については、「[Detecting Your Supported Platforms and Whether You Have a Default VPC](#)」を参照してください。
- AWS アカウントでデフォルト VPC をサポートしている場合は、その VPC でインスタンスを起動するかどうかを判断します。スポットインスタンスを要求する際に追加の手順を実行する必要はありません。[Request Spot Instances] ウィザードの [Network] フィールドに VPC ID が一覧表示されます。

- AWS アカウントでデフォルト VPC をサポートしている場合は、スポットインスタンスに対して別の VPC を使用するかどうかを判断します。新しい VPC を作成し、作成した VPC の ID を指定することができます。
- AWS アカウントでデフォルト VPC がサポートされていない場合は、VPC を作成し、その作成した VPC の ID を指定します。

デフォルト VPC の詳細については、「[デフォルトの VPC の基本](#)」を参照してください。VPC の設定の詳細については、*Amazon Virtual Private Cloud 入門ガイド* の [Getting Started with Amazon VPC](#) を参照してください。

## 動画クイックルック: Amazon VPC でスポットインスタンスを起動

以下の動画は、AWS マネジメントコンソールを使用して初めて Amazon VPC でスポットインスタンスを起動する方法について説明しています。この動画には、VPC サブネットの作成、入札の実行、インスタンスが満たされるタイミングの決定、インスタンスのキャンセルについての指示が含まれます。  
[Amazon VPC でスポットインスタンスを起動](#)

## Amazon VPC でスポットインスタンスを起動するプロセスについて

VPC で起動するスポットインスタンスリクエストを作成するプロセスは、Amazon EC2 の VPC 部分以外でスポットインスタンスリクエストを作成する場合に従うプロセスと同じですが、次のような違いがあります。

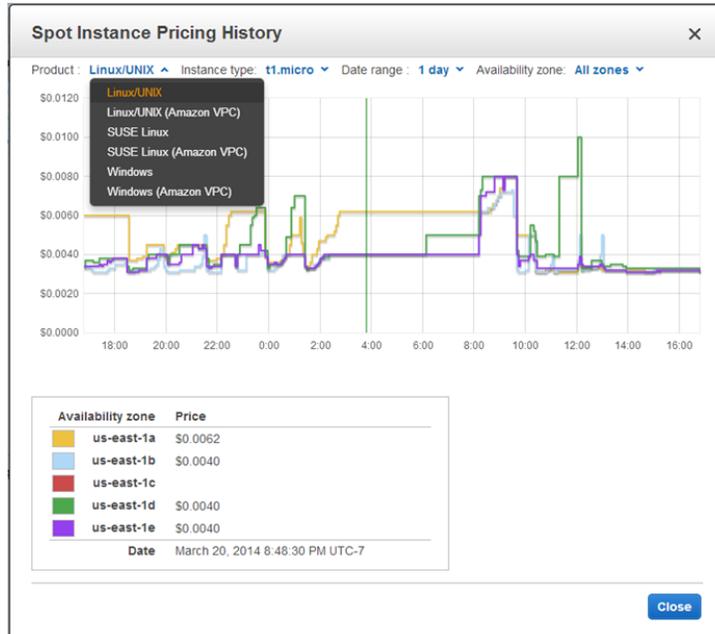
入札は VPC のスポットインスタンスのスポット価格履歴に基づいて行います。

`describe-spot-price-history` AWS CLI コマンドまたは `ec2-describe-spot-price-history` Amazon EC2 CLI コマンドを使用する場合、(*Amazon VPC*) を *product-description* に追加します。

AWS CLI を使用した場合、コマンドは次のようになります。

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge --product-description "Linux/UNIX (Amazon VPC)" --start-time 2014-01-06T07:08:09 --end-time 2014-01-06T08:09:10
```

AWS Management Console を使用して、[Spot Instance Pricing History] ページを確認し、Amazon EC2 と Amazon VPC の両方で実行されている Amazon EC2 インスタンスのスポット価格設定履歴を表示します。



スポットインスタンスを起動する VPC を指定します。

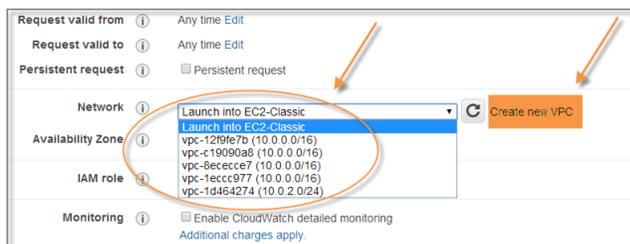
[aws ec2 request-spot-instances](#) AWS CLI コマンドまたは [ec2-request-spot-instances](#) Amazon EC2 CLI コマンドを使用する場合、スポットインスタンスを起動する VPC サブネットの ID を指定します。

AWS CLI を使用した場合、コマンドは次のようになります。

```
$ aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type  
"one-time" --launch-specification "{\"ImageId\":\"ami-a43909e1\",\"Instance  
Type\":\"  
m1.small\",\"SubnetId\":\"subnet-d50bfefc\"}"
```

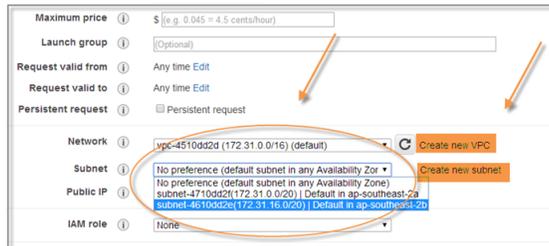
AWS Management Console を使用して、Amazon EC2 コンソールの [Spot Requests] ページから Request Spot Instances ウィザードを起動する場合、[Configure Instance Details] ページの [Network] ドロップダウンから VPC を選択します。[Network] では、アカウントがデフォルト VPC をサポートするかどうかにより表示される項目が異なります。(詳細については、「[Detecting Your Supported Platforms and Whether You Have a Default VPC](#)」を参照してください。)

- アカウントが EC2-Classic で、デフォルト VPC をサポートしていない場合、スポットリクエストを作成する前に VPC が存在するようであれば、[Network] に VPC オプションが表示されます。たとえば、us-east-1 にスポットインスタンスをリクエストしており、VPC を作成済みの場合、[Network] には、そのリージョン内のアベイラビリティゾーンで使用できる VPC サブネットの一覧が表示されます。



VPCがない場合は、[Create new VPC] をクリックします。新しいタブに [VPC Management Console] が開きます。[Create VPC] をクリックし、ウィザードにしたがって新しい VPC を作成します。

- アカウントがデフォルト VPC をサポートしている場合、[Network] には、デフォルト VPC の一覧が表示されます。[Subnet] ドロップダウンリストで、スポットインスタンスをリクエストしているリージョン内のアベイラビリティゾーンのデフォルトサブネットを確認することもできます。新しい VPC や新しいサブネットを作成することもできます。



特定の価格の安いアベイラビリティゾーンでスポットインスタンスを開始する場合、スポットインスタンスリクエストにサブネットを指定する必要があります。

選択するサブネットによって、スポットインスタンスが起動するアベイラビリティゾーンが決まります。スポットインスタンスリクエストを送信する際にサブネットとアベイラビリティゾーンを指定しない場合、スポットインスタンスは、使用できるアベイラビリティゾーンのサブネットに配置されます。アベイラビリティゾーンが最低スポット価格になるとは限りません。

アカウントがデフォルト VPC をサポートし、スポットリクエストでサブネットとアベイラビリティゾーンを指定しない場合、配置システムによって選ばれるデフォルト VPC は、最低スポット価格のアベイラビリティゾーンとは異なる可能性があります。

Amazon VPC の使用の詳細については、[Amazon Virtual Private Cloud ユーザーガイド](#) を参照してください。

## 高度なタスク

### Abstract

スポットインスタンスで実行できる高度なタスクの一覧を示します。

スポットインスタンスリクエストが作成され、スポットインスタンスの作業が完了したところで、いくつかの高度なタスクについて説明します。

- [スポットインスタンスのデータフィードの購読 \(p. 186\)](#)
- [AWS Java SDK を使用したスポットインスタンスのプログラミング \(p. 190\)](#)
- [スポットインスタンスでのクラスターの開始 \(p. 216\)](#)

その他にこのセクションで説明するべき高度なタスクがありましたら、[spot-instance-feedback@amazon.com](mailto:spot-instance-feedback@amazon.com) までご意見をお寄せください。

## スポットインスタンスのデータフィードの購読

### Abstract

スポットインスタンスのデータフィードを購読して、スポットインスタンスの使用状況を監視します。

スポットインスタンスの使用状況をモニタリングするには、Amazon Simple Storage Service ( Amazon S3 ) に使用状況のログを保存する、スポットインスタンスデータフィードを購読します。

このセクションでは、データフィードの内容、およびスポットインスタンスのデータフィードの作成方法について説明します。データフィードはアカウントごとに 1 つ作成することができます。

## スポットインスタンスデータフィードの概要

スポットインスタンスの料金について理解しやすくするため、Amazon EC2 では、スポットインスタンスの使用状況と料金表の詳細を記載したデータフィードにアクセスすることができます。このデータフィードは、選択した Amazon S3 バケットに送信されます。

Amazon S3 バケットにデータフィードが送信されるようにするには、Amazon EC2 API を使用してスポットインスタンスデータフィード購読を作成する必要があります。この購読を作成すると、データフィードのファイルを送信する Amazon S3 バケット、および衝突を回避するために使用するファイル名プレフィックスを指定することができます。

### データフィードのファイル名と形式

スポットインスタンスのデータフィードのファイル名には次の形式を使用します ( UTC の日付と時刻を使用 )。

```
{Bucket}.s3.amazonaws.com/{Optional Prefix}/{AWS Account ID}.{YYYYY}-{MM}-{DD}-{HH}.n.{Unique ID}.gz
```

例えば、バケット名が `myawsbucket` であり、プレフィックス `myprefix` を指定した場合、ファイル名は次のようになります。

```
myawsbucket.s3.amazonaws.com/myprefix/111122223333.2010-03-17-20.001.pwBdGTJG.gz
```

データフィードファイルは、通常、1 時間に 1 回バケットに届き、各使用時は、通常、単一のデータファイルでカバーされます。このファイルは、バケットに配信される前に圧縮 ( gzip ) されます。ファイルが非常に大きい場合は、指定した時間の使用状況に関するファイルを複数書き込むことができます ( 例えば、ある時間のファイルコンテンツが圧縮前に 50 MB を超える場合 )。



#### Note

ある時間に対してスポットインスタンスが実行されていない場合、その時間のデータフィードファイルは届きません。

スポットインスタンスのデータフィードファイルはタブで区切られています。データファイルの各行が、1 インスタンス時間に対応しています。各行には、以下の表のフィールドが含まれています。

フィールド	説明
Timestamp	そのインスタンス時間に対して請求される価格を決定するために使用されるタイムスタンプ。
UsageType	請求の対象となっている使用タイプおよびインスタンスタイプを示します。m1.small スポットインスタンスの場合、このフィールドは [SpotUsage] に設定されます。それ以外のインスタンスタイプの場合、このフィールドは [SpotUsage:{instance-type}] に設定されます。例えば、[SpotUsage:c1.medium] のようになります。

フィールド	説明
Operation	請求の対象となっている製品を示します。Linux/UNIX スポットインスタンスの場合、このフィールドは [RunInstances] に設定されます。Microsoft Windows の場合、このフィールドは [RunInstances:0002] に設定されます。スポット使用状況は、利用可能ゾーンに従ってグループ化されます。
InstanceID	このインスタンス時間を生成したスポットインスタンスのインスタンス ID。
MyBidID	このインスタンス時間を生成したリクエストのスポットインスタンスリクエスト ID。
MyMaxPrice	このスポットインスタンスリクエストに指定された上限価格。
MarketPrice	タイムスタンプフィールドに指定された時間のスポット価格。
Charge	このインスタンス時間に請求される価格。
Version	このレコードのデータフィールドファイル名に含まれるバージョン。

### データフィールドのための Amazon S3 の準備

データフィールドの購読時に、Amazon EC2 にデータフィールドのファイルを保存するバケットを通知します。データフィールドを受信登録する前に、Amazon S3 バケットの選択時には以下の点を検討してください。

- 指定するバケットには Amazon S3 FULL\_CONTROL 権限が必要です。

バケット所有者には、デフォルトでこの権限があります。バケット所有者でない場合は、バケット所有者がお客様の AWS アカウントに FULL\_CONTROL 権限を与える必要があります。

- データフィールド購読を作成すると、Amazon EC2 は指定されたバケットの ACL を更新して AWS データフィールドアカウントに読み書き権限を与えます。
- 各データフィールドファイルには、独自の ACL があります (バケットの ACL とは別です)。

バケット所有者には、データファイルの FULL\_CONTROL 権限があります。データフィールドアカウントには読み書き権限があります。

- データフィールドアカウントの権限を削除しても、データフィールドは無効になりません。

これらの権限を削除してもデータフィールドを無効にしない (コントロール API で行う) 場合は、次にデータフィールドアカウントでバケットのデータファイルへの書き込みが必要になったときに、再度これらの権限を与えます。

- データフィールドの購読を削除しても、Amazon EC2 ではバケットまたはデータファイルのデータフィールドアカウントの読み書き権限は削除されません。

読み書き権限の削除は自分で行う必要があります。

### スポットインスタンスのデータフィールドの購読

#### Amazon EC2 コマンドラインインターフェイス (CLI) ツール

スポットインスタンスのデータフィールドを購読するには

- 次のコマンドを入力します。

```
$ ec2-create-spot-datafeed-subscription --bucket myawsbucket [--prefix prefix]
]
```

次のような出力が Amazon EC2 から返されます。

```
SPOTDATAFEEDSUBSCRIPTION      111122223333      myawsbucket      prefix
Active
```

## API

スポットインスタンスのデータフィードを購読するには

- 次のクエリリクエストを作成します。

```
https://ec2.amazonaws.com/
?Action=CreateSpotDatafeedSubscription
&Bucket=myawsbucket
&Prefix=my-spot-subscription
&...auth parameters...
```

以下に、応答の例を示します。

```
<CreateSpotDatafeedSubscriptionResponse xmlns="http://ec2.amazon
aws.com/doc/2014-02-01/">
  <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>
  <spotDatafeedSubscription>
    <ownerId>999988887777</ownerId>
    <bucket>myawsbucket</bucket>
    <prefix>my-spot-subscription</prefix>
    <state>Active</state>
    <fault>/</fault>
  </spotDatafeedSubscription>
</CreateSpotDatafeedSubscriptionResponse>
```

## スポットインスタンスのデータフィードの削除

### コマンドラインツール

スポットインスタンスのデータフィードを削除するには

- データフィードを削除するには、次のコマンドを入力します。

```
$ ec2-delete-spot-datafeed-subscription
```

リクエストが成功した場合、出力は空です。

## API

スポットインスタンスのデータフィードを削除するには

- 次のクエリリクエストを作成します。

```
https://ec2.amazonaws.com/  
?Action=DeleteSpotDatafeedSubscription  
&...auth parameters...
```

以下に、応答の例を示します。購読が削除されたことが確認されます。

```
<DeleteSpotDatafeedSubscriptionResponse xmlns="http://ec2.amazon  
aws.com/doc/2014-02-01/">  
  <requestId>59dbff89-35bd-4eac-99ed-be587EXAMPLE</requestId>  
  <return>true</return>  
</DeleteSpotDatafeedSubscriptionResponse>
```

## AWS Java SDK を使用したスポットインスタンスのプログラミング

### Abstract

AWS SDK for Java でスポットインスタンスをプログラミングする方法について、以下の Java コードの例をひとつおご確認ください。

AWS Java SDK を使用してスポットインスタンスをプログラミングできます。

- [チュートリアル: Amazon EC2 スポットインスタンス \(p. 191\)](#)
- [チュートリアル: Amazon EC2 スポットリクエストの高度な管理 \(p. 200\)](#)

## チュートリアル: Amazon EC2 スポットインスタンス

### 概要

スポットインスタンスとは、Amazon Elastic Compute Cloud ( Amazon EC2 ) の未使用キャパシティに対してお客様から価格を提示していただき、入札価格がその時点のスポット価格を上回っている限りお客様がインスタンスを実行できるというシステムです。Amazon EC2 のスポット価格は、需要と供給に基づいて定期的に変動しますが、お客様の入札価格がその価格以上ならば、空いているスポットインスタンスにアクセスできます。オンデマンドインスタンスやリザーブドインスタンスと同様に、スポットインスタンスは計算キャパシティを増やしたいときの選択肢の 1 つとなります。

スポットインスタンスを利用すると、Amazon EC2 でバッチ処理、科学研究、画像処理、動画エンコーディング、データと Web のクローリング、財務分析、テストなどのコストの大幅削減を期待できます。加えて、スポットインスタンスは、大量の追加計算キャパシティが必要であるけれどもその緊急性が低いという場合にも適しています。

スポットインスタンスを使用するには、スポットインスタンスリクエストを提出し、このときにインスタンス時間当たりいくらまで支払えるかを指定します。これが入札価格です。入札価格がその時点のスポット価格を超えている場合は、リクエストが受理されてインスタンスを実行できるようになります。このインスタンスの実行は、お客様がインスタンスを終了した時点と、スポット価格が入札価格を上回った時点のいずれか早い方までとなります。

次の 2 点にご注意ください。

- 時間当たりの支払い金額が入札価格を下回ることもよくあります。Amazon EC2 のスポット価格は、提出されるリクエストや空きインスタンスの変動に応じて、定期的に変更されます。お客様それぞれの入札価格の方が上かどうかがかわらず、どのお客様もその期間の同一のスポット価格をお支払いいただきます。したがって、お客様が支払う金額は入札価格を下回ることもありますが、入札価格を超えることはありません。
- スポットインスタンスを実行しているときに、お客様の入札価格がその時点のスポット価格以上ではなくなった場合は、そのインスタンスは終了となります。つまり、お客様の作業負荷とアプリケーションが、この投機的能力を利用するのに充分な柔軟性を有しているか確認することが大切です。

スポットインスタンスは稼働中、他の Amazon EC2 インスタンスとまったく同じように動作します。そして他の Amazon EC2 インスタンスと同様に、スポットインスタンスは必要なくなった場合に終了することができます。お客様がインスタンスを終了した場合は、使用時間の端数分についても料金をいただきます ( オンデマンドやリザーブドのインスタンスと同様です )。ただし、スポット価格がお客様の入札価格を超えたためにインスタンスが Amazon EC2 によって終了させられた場合は、使用時間の端数分の料金は発生しません。

このチュートリアルでは、Java プログラミング言語を使用して以下のことを行う方法の概要を説明します。

- スポットリクエストを提出する
- スポットリクエストが受理されたかどうかを判断する
- スポットリクエストをキャンセルする
- 関連するインスタンスの終了

### 前提条件

このチュートリアルを使用するには、アマゾン ウェブ サービス ( AWS ) にサインアップする必要があります。まだ AWS にサインアップしていない場合は、[アマゾン ウェブ サービスのウェブサイト](#)にアクセスし、ページ右上にある [Create an AWS Account] をクリックしてください。さらに、[AWS Java SDK](#) のインストールも必要です。

Eclipse 開発環境を使用している場合は、[AWS Toolkit for Eclipse](#) を使用することをお勧めします。AWS Toolkit for Eclipse には、最新バージョンの AWS SDK for Java が含まれています。

### ステップ 1: 認証情報のセットアップ

このサンプルコードの使用を開始するには、AwsCredentials.properties ファイルに認証情報を入力する必要があります。具体的には、シークレットキーとアクセスキーを入力する必要があります。

アクセスキーとシークレットアクセスキーをコピーして AwsCredentials.properties ファイルに貼り付けます。

これで設定が完了したので、例に示すコードを使用できるようになります。

### ステップ 2: セキュリティグループのセットアップ

セキュリティグループとは、ファイアウォールとしての役割を果たすものであり、インスタンスのグループに対してどのトラフィックの送受信を許可するかを制御します。デフォルトでは、インスタンスの起動時にセキュリティグループは何も設定されていません。つまり、着信 IP トラフィックは、どの TCP ポートであってもすべて拒否されます。したがって、ここでは、スポットリクエストを提出する前に、必要なネットワークトラフィックを許可するセキュリティグループをセットアップすることになります。このチュートリアル目的に合わせて、ここでは新しいセキュリティグループを「GettingStarted」という名前で作成します。このグループでは、自分のアプリケーションを実行する IP アドレスからの Secure Shell (SSH) トラフィックを許可します。新しいセキュリティグループをセットアップするには、次に示すコードサンプルをインクルードするか実行する必要があります。このコードは、セキュリティグループをプログラムからセットアップするためのものです。

AmazonEC2 クライアントオブジェクトを作成した後で、CreateSecurityGroupRequest オブジェクトを作成し、「GettingStarted」という名前と、セキュリティグループの説明を指定します。その後で、ec2.createSecurityGroup API を呼び出してグループを作成します。

このグループにアクセスできるようにするために、ipPermission オブジェクトを作成します。IP アドレス範囲は、ローカルコンピュータのサブネット (CIDR 表現) で設定します。IP アドレスの「/10」というサブフィックスが、指定した IP アドレスのサブネットを示します。また、ipPermission オブジェクトを設定して TCP プロトコルとポート 22 (SSH) を指定します。最後のステップは、ec2.authorizeSecurityGroupIngress を呼び出すことです。このときに、作成したセキュリティグループの名前と ipPermission オブジェクトを指定します。

```
1
  // Retrieves the credentials from an AwsCredentials.properties file.
  AwsCredentials credentials = null;
  try {
5     credentials = new PropertiesCredentials(
        GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
    } catch (IOException e1) {
        System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
        System.out.println(e1.getMessage());
10    System.exit(-1);
    }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);
15
    // Create a new security group.
    try {
        CreateSecurityGroupRequest securityGroupRequest = new CreateSecurityGroupRequest("GettingStartedGroup", "Getting Started Security Group");
```

```

        ec2.createSecurityGroup(securityGroupRequest);
20 } catch (AmazonServiceException ase) {
    // Likely this means that the group is already created, so ignore.
    System.out.println(ase.getMessage());
}

25 String ipAddr = "0.0.0.0/0";

    // Get the IP of the current host, so that we can limit the Security
    // Group by default to the ip range associated with your subnet.
    try {
30     InetAddress addr = InetAddress.getLocalHost();

        // Get IP Address
        ipAddr = addr.getHostAddress()+"/10";
    } catch (UnknownHostException e) {
35 }

    // Create a range that you would like to populate.
    ArrayList<String> ipRanges = new ArrayList<String>();
    ipRanges.add(ipAddr);
40

    // Open up port 22 for TCP traffic to the associated IP
    // from above (e.g. ssh traffic).
    ArrayList<IpPermission> ipPermissions = new ArrayList<IpPermission> ();
    IpPermission ipPermission = new IpPermission();
45 ipPermission.setIpProtocol("tcp");
    ipPermission.setFromPort(new Integer(22));
    ipPermission.setToPort(new Integer(22));
    ipPermission.setIpRanges(ipRanges);
    ipPermissions.add(ipPermission);
50

    try {
        // Authorize the ports to the used.
        AuthorizeSecurityGroupIngressRequest ingressRequest =
            new AuthorizeSecurityGroupIngressRequest("GettingStartedGroup", ip
Permissions);
55     ec2.authorizeSecurityGroupIngress(ingressRequest);
    } catch (AmazonServiceException ase) {
        // Ignore because this likely means the zone has
        // already been authorized.
        System.out.println(ase.getMessage());
60 }

```

このコードサンプル全体を見るには、`CreateSecurityGroupApp.java` コードサンプルを参照してください。このアプリケーションを実行して新しいセキュリティグループを作成する必要があるのは1回のみです。

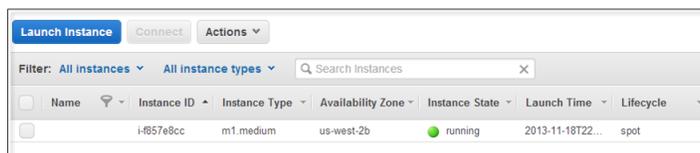
セキュリティグループは、AWS Toolkit for Eclipse を使用して作成することもできます。詳細については、[ツールキットのドキュメント](#)を参照してください。

### ステップ 3: スポットリクエストを提出する

スポットリクエストを提出するには、最初に、使用するインスタンスタイプ、Amazon マシンイメージ (AMI)、最高入札価格を決定する必要があります。前のステップで設定したセキュリティグループも指定する必要があります。これは、必要に応じてインスタンスにログインできるようにするためです。

複数のインスタンスタイプから選択できます。インスタンスタイプの一覧については、「[インスタンスタイプ \(p. 109\)](#)」を参照してください。このチュートリアルでは、最も低価格のインスタンスタイプである t1.micro を使用します。次に、使用する AMI のタイプを決定します。ここでは、ami-8c1fece5 を使用します。これは、このチュートリアルの執筆時点で最新の Amazon Linux AMI です。最新の AMI は時間の経過とともに変化することがありますが、最新バージョンの AMI は次の方法でいつでも確認できます。

1. AWS マネジメントコンソールにログインし、EC2 タブをクリックし、EC2 コンソールダッシュボードからインスタンスの 1 つを起動します。



2. AMI が表示されているウィンドウで、ウィザードに示す AMI ID を使用します。DescribeImages API を使用することもできますが、このコマンドの利用方法は、このチュートリアルでは取り上げません。



スポットインスタンス入札のアプローチは多数あります。さまざまなアプローチの概要については、[スポットインスタンスの入札](#)の動画をご覧ください。ただし、ここでは初めての方のために、3つの一般的な戦略について説明します。その3つとは、「コストがオンデマンド価格より低くなるように入札する」、「計算処理の結果の価値に基づいて入札する」、「できるだけ早く計算キャパシティを獲得できるように入札する」です。

- コストをオンデマンドよりも低くする 実行完了までに何時間も、あるいは何日間もかかるバッチ処理ジョブがあるとします。ただし、いつ開始していつ完了するかについては、特に決められていないものとします。このジョブを完了するためのコストを、オンデマンドインスタンスを使用する場合よりも低くできるかどうかを考えます。インスタンスタイプ別のスポット価格の履歴を、AWS マネジメントコンソールまたは Amazon EC2 API を使用して調べます。詳細については、「[スポットインスタンス価格履歴の表示 \(p. 138\)](#)」を参照してください。使用したいインスタンスタイプの、特定の Availability Zone での価格履歴を分析した後は、入札のアプローチとして次の2つも考えられます。
  - スポット価格の範囲の上限（ただしオンデマンド価格よりは下）で入札します。このようにすれば、この1回限りのスポットリクエストが受理される可能性が高くなり、ジョブが完了するまで連続して実行できるからです。
  - または、価格範囲の下限で入札し、1つの永続リクエストで次々とインスタンスを起動するよう計画を立てます。これらのインスタンスの実行時間を合計すると、ジョブを完了するのに十分な長さとなり、合計コストも低くなります。（この作業を自動化する方法については、このチュートリアルで後ほど説明します）
- 結果の価値以上は支払わない データ処理ジョブを実行するとします。このジョブの結果の価値は判明しており、計算コストに換算してどれくらいになるかもわかっています。使用するインスタンスタイプのスポット価格履歴の分析が完了した後で、入札価格を選択します。計算時間のコストがこのジョブの結果の価値を上回ることがないように、価格を決定します。永続リクエストを作成し、スポット価格が入札価格以下となったときに断続的に実行するよう設定します。
- 計算キャパシティをすぐに獲得する 追加のキャパシティが突然、短期間だけ必要になることがあり、オンデマンドインスタンスではそのキャパシティを獲得できないとします。使用するインスタンスタイプのスポット価格履歴の分析が完了した後で、履歴の価格の最大値を超える価格で入札します。こ

のようにすれば、リクエストがすぐに受理される可能性が高まり、計算が完了するまで連続して計算できるようになります。

入札価格を選択すると、スポットインスタンスをリクエストできる状態になります。ここでは、このチュートリアル目的に合わせて、オンデマンド価格 (0.03 USD) で入札します。これは、受理される可能性を最大にするためです。利用できるインスタンスのタイプと、インスタンスのオンデマンド価格を調べるには、Amazon EC2 の価格表ページを参照してください。スポットインスタンスをリクエストするには、これまでに選択したパラメータを使用してリクエストを作成します。初めに、`RequestSpotInstanceRequest` オブジェクトを作成します。このリクエストオブジェクトには、起動したいインスタンスの数と入札価格が必要です。さらに、リクエストの `LaunchSpecification` を設定する必要があります。この内容は、インスタンスタイプ、AMI ID、および使用するセキュリティグループです。リクエストの内容が入力されたら、`AmazonEC2Client` オブジェクトの `requestSpotInstances` メソッドを呼び出します。次の例は、スポットインスタンスをリクエストする方法を示しています。

```
1
// Retrieves the credentials from a AWScredentials.properties file.
AWScredentials credentials = null;
try {
5   credentials = new PropertiesCredentials(
        GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
    } catch (IOException e1) {
        System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
        System.out.println(e1.getMessage());
10    System.exit(-1);
    }

// Create the AmazonEC2Client object so we can call various APIs.
AmazonEC2 ec2 = new AmazonEC2Client(credentials);
15
// Initializes a Spot Instance Request
RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

// Request 1 x t1.micro instance with a bid price of $0.03.
20 requestRequest.setSpotPrice("0.03");
   requestRequest.setInstanceCount(Integer.valueOf(1));

// Setup the specifications of the launch. This includes the
// instance type (e.g. t1.micro) and the latest Amazon Linux
25 // AMI id available. Note, you should always use the latest
// Amazon Linux AMI id or another of your choosing.
LaunchSpecification launchSpecification = new LaunchSpecification();
launchSpecification.setImageId("ami-8c1fece5");
launchSpecification.setInstanceType("t1.micro");
30
// Add the security group to the request.
ArrayList<String> securityGroups = new ArrayList<String>();
securityGroups.add("GettingStartedGroup");
launchSpecification.setSecurityGroups(securityGroups);
35
// Add the launch specifications to the request.
requestRequest.setLaunchSpecification(launchSpecification);

// Call the RequestSpotInstance API.
```

```
40 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(requestRequest);
```

このコードを実行すると、新しいスポットインスタンスリクエストが発行されます。他にも、スポットリクエストの設定に使用できるオプションがあります。詳細については、「[チュートリアル: Amazon EC2 スポットリクエストの高度な管理 \(p. 200\)](#)」または Java SDK の [RequestSpotInstances API](#) を参照してください。



#### Note

スポットインスタンスが実際に起動されるとお客様への課金が発生するので、料金を抑えるために、リクエストを作成した場合はキャンセルし、インスタンスを起動した場合は終了してください。

### ステップ 4: スポットリクエストの状態を特定する

次に、最後のステップに進む前にスポットリクエストの状態が「アクティブ」になるのを待つようにするコードを作成する必要があります。スポットリクエストの状態を特定するには、[describeSpotInstanceRequests](#) メソッドをポーリングすることによって、モニタリング対象のスポットリクエスト ID の状態を調べます。

ステップ 2 で作成したリクエスト ID は、`requestSpotInstances` リクエストへのレスポンスに埋め込まれています。次のコード例は、リクエスト ID を `requestSpotInstances` レスポンスから取り出して `ArrayList` への入力に使用する方法を示しています。

```
1
    // Call the RequestSpotInstance API.
    RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(requestRequest);
    List<SpotInstanceRequest> requestResponses = requestResult.getSpotInstanceRequests();
5
    // Setup an arraylist to collect all of the request ids we want to
    // watch hit the running state.
    ArrayList<String> spotInstanceRequestIds = new ArrayList<String>();

10 // Add all of the request ids to the hashset, so we can determine when they
    // hit the
    // active state.
    for (SpotInstanceRequest requestResponse : requestResponses) {
        System.out.println("Created Spot Request: "+requestResponse.getSpotInstanceRequestId());
        spotInstanceRequestIds.add(requestResponse.getSpotInstanceRequestId());
15 }
```

リクエスト ID をモニタリングするには、`describeSpotInstanceRequests` メソッドを呼び出してリクエストの状態を特定します。その後で、リクエストが「オープン」状態でなくなるまでループを繰り返します。状態が、例えば「アクティブ」ではなく、「オープン」以外かどうかをモニタリングするのは、リクエストが直接「クローズ済み」に遷移することもあるからです (リクエストの引数に問題がある場合)。次に示すコード例では、このことを実現する具体的な方法を示します。

```
1
    // Create a variable that will track whether there are any
```

```
// requests still in the open state.
boolean anyOpen;
5
do {
    // Create the describeRequest object with all of the request ids
    // to monitor (e.g. that we started).
    DescribeSpotInstanceRequestsRequest describeRequest = new DescribeSpot
InstanceRequestsRequest();
10    describeRequest.setSpotInstanceRequestIds(spotInstanceRequestIds);

    // Initialize the anyOpen variable to false - which assumes there
    // are no requests open unless we find one that is still open.
    anyOpen=false;

15    try {
        // Retrieve all of the requests we want to monitor.
        DescribeSpotInstanceRequestsResult describeResult = ec2.describeS
potInstanceRequests(describeRequest);
        List<SpotInstanceRequest> describeResponses = describeResult.get
SpotInstanceRequests();
20
        // Look through each request and determine if they are all in
        // the active state.
        for (SpotInstanceRequest describeResponse : describeResponses) {
            // If the state is open, it hasn't changed since we attempted
25            // to request it. There is the potential for it to transition
            // almost immediately to closed or canceled so we compare
            // against open instead of active.
            if (describeResponse.getState().equals("open")) {
                anyOpen = true;
30                break;
            }
        }
    } catch (AmazonServiceException e) {
        // If we have an exception, ensure we don't break out of
35        // the loop. This prevents the scenario where there was
        // blip on the wire.
        anyOpen = true;
    }

40    try {
        // Sleep for 60 seconds.
        Thread.sleep(60*1000);
    } catch (Exception e) {
        // Do nothing because it woke up early.
45    }
} while (anyOpen);
```

このコードを実行すると、スポットインスタンスリクエストは完了するか失敗し、失敗の場合はエラーが画面に出力されます。どちらの場合も、次のステップに進んで、アクティブなリクエストがある場合はクリーンアップし、実行中のインスタンスがある場合は終了してください。

#### ステップ 5: スポットリクエストとインスタンスをクリーンアップする

最後に、リクエストとインスタンスをクリーンアップする必要があります。未完了リクエストのキャンセルと、インスタンスの終了の両方を行うことが重要です。リクエストをキャンセルするだけではインスタンスは終了しないので、引き続きお客様への課金が発生することになります。インスタンスを終了

すると、スポットリクエストがキャンセルされることもあります。場合によっては（例えば、持続的入札を使用した場合）、インスタンスを終了しただけでは、リクエストが再度受理されるのを停止できないことがあります。したがって、アクティブな入札のキャンセルと実行中インスタンスの終了の両方を行うことをお勧めします。

次のコードは、リクエストをキャンセルする方法を示しています。

```

1
  try {
    // Cancel requests.
    CancelSpotInstanceRequestsRequest cancelRequest = new CancelSpotInstanceRequestsRequest(spotInstanceRequestIds);
5    ec2.cancelSpotInstanceRequests(cancelRequest);
  } catch (AmazonServiceException e) {
    // Write out any exceptions that may have occurred.
    System.out.println("Error canceling instances");
    System.out.println("Caught Exception: " + e.getMessage());
10   System.out.println("Response Status Code: " + e.getStatusCode());
    System.out.println("Error Code: " + e.getErrorCode());
    System.out.println("Request ID: " + e.getRequestId());
  }

```

稼働中のインスタンスを終了するには、そのインスタンスを起動したリクエストに関連付けられているインスタンスIDが必要です。次のコード例は、前に示したインスタンスをモニタリングするためのコードに `ArrayList` を追加したものです。この中に、`describeInstance` レスポンスに関連付けられているインスタンスIDを格納します。

```

1
  // Create a variable that will track whether there are any requests
  // still in the open state.
  boolean anyOpen;
5
  // Initialize variables.
  ArrayList<String> instanceIds = new ArrayList<String>();

  do {
10    // Create the describeRequest with all of the request ids to
    // monitor (e.g. that we started).
    DescribeSpotInstanceRequestsRequest describeRequest = new DescribeSpotInstanceRequestsRequest();
    describeRequest.setSpotInstanceRequestIds(spotInstanceRequestIds);

15    // Initialize the anyOpen variable to false, which assumes there
    // are no requests open unless we find one that is still open.
    anyOpen = false;

    try {
20      // Retrieve all of the requests we want to monitor.
      DescribeSpotInstanceRequestsResult describeResult = ec2.describeSpotInstanceRequests(describeRequest);
      List<SpotInstanceRequest> describeResponses = describeResult.getSpotInstanceRequests();

      // Look through each request and determine if they are all
25      // in the active state.
      for (SpotInstanceRequest describeResponse : describeResponses) {

```

```

        // If the state is open, it hasn't changed since we
        // attempted to request it. There is the potential for
        // it to transition almost immediately to closed or
30    // canceled so we compare against open instead of active.
        if (describeResponse.getState().equals("open")) {
            anyOpen = true;
            break;
        }
35
        // Add the instance id to the list we will
        // eventually terminate.
        instanceIds.add(describeResponse.getInstanceId());
    }
40    } catch (AmazonServiceException e) {
        // If we have an exception, ensure we don't break out
        // of the loop. This prevents the scenario where there
        // was blip on the wire.
        anyOpen = true;
45    }

    try {
        // Sleep for 60 seconds.
        Thread.sleep(60*1000);
50    } catch (Exception e) {
        // Do nothing because it woke up early.
    }
} while (anyOpen);

```

この `ArrayList` に格納されているインスタンス ID を使用して、稼働中のインスタンスを終了します。コードは次のとおりです。

```

1    try {
        // Terminate instances.
        TerminateInstancesRequest terminateRequest = new TerminateInstances
Request(instanceIds);
5        ec2.terminateInstances(terminateRequest);
    } catch (AmazonServiceException e) {
        // Write out any exceptions that may have occurred.
        System.out.println("Error terminating instances");
        System.out.println("Caught Exception: " + e.getMessage());
10       System.out.println("Response Status Code: " + e.getStatusCode());
        System.out.println("Error Code: " + e.getErrorCode());
        System.out.println("Request ID: " + e.getRequestId());
    }
}

```

### ステップの集約

これまでに説明したステップは、よりオブジェクト指向的なアプローチをとって1つに集約することができます。このステップとは、EC2 クライアントの初期化、スポットリクエストの提出、スポットリクエストがオープン状態でなくなったかどうかの特定、および未完了のスポットリクエストや関連するインスタンスのクリーンアップです。これらのすべてを実行する、`Requests` というクラスを作成します。

さらに、`GettingStartedApp` というクラスも作成します。ここにメインメソッドがあり、ここで高レベルの関数呼び出しを実行します。具体的には、既に説明した `Requests` オブジェクトを初期化します。スポットインスタンスリクエストを提出します。その後は、スポットリクエストが「アクティブ」状態になるまで待ちます。最後に、リクエストとインスタンスをクリーンアップします。

ソースコード全体を [GitHub](#) からダウンロードできます。

これで、AWS Java SDK を使用したスポットインスタンスソフトウェア開発の入門チュートリアルは終了です。

## 次のステップ

Java 開発者向けの「[チュートリアル: Amazon EC2 スポットリクエストの高度な管理 \(p. 200\)](#)」に進むことをお勧めします。

## チュートリアル: Amazon EC2 スポットリクエストの高度な管理

### Abstract

チュートリアル - スポットインスタンスを使用して Amazon EC2 のコストを削減する方法を示します。

### 概要

スポットインスタンスとは、Amazon Elastic Compute Cloud ( Amazon EC2 ) の未使用キャパシティに対してお客様から価格を提示していただき、入札価格がその時点のスポット価格を上回っている限りお客様がインスタンスを実行できるというシステムです。Amazon EC2 のスポット価格は、需要と供給に応じて定期的に変動します。お客様の入札価格がスポット価格以上ならば、空きスポットインスタンスを使用できます。オンデマンドインスタンスやリザーブドインスタンスと同様に、スポットインスタンスは計算キャパシティを増やしたいときの選択肢の 1 つとなります。

スポットインスタンスを利用すると、Amazon EC2 でバッチ処理、科学研究、画像処理、動画エンコーディング、データと Web のクローリング、財務分析、テストなどのコストの大幅削減を期待できます。加えて、スポットインスタンスは、大量の追加計算キャパシティが必要であるけれどもその緊急性が低いという場合にも適しています。

このチュートリアルでは、スポットリクエストの高度な機能のいくつかについて概要を説明します。例えば、スポットリクエスト作成時の詳細なオプション、スポットインスタンス起動の代替方法、インスタンスを管理する方法などです。このチュートリアルは、スポットインスタンスに関連する高度なトピックをすべて取り上げるものではありません。代わりに、スポットリクエストとスポットインスタンスの管理のために一般的に使用されている方法のいくつかについて、コードのサンプルを紹介します。

### 前提条件

このチュートリアルを使用するには、アマゾン ウェブ サービス ( AWS ) にサインアップする必要があります。まだ AWS にサインアップしていない場合は、[アマゾン ウェブ サービスのウェブサイト](#) にアクセスし、ページ右上にある [サインアップ] をクリックします。さらに、[AWS SDK for Java](#) のインストールも必要です。

Eclipse 開発環境を使用している場合は、[AWS Toolkit for Eclipse](#) を使用することをお勧めします。Toolkit for Eclipse には、最新バージョンの AWS SDK for Java が含まれています。

### ステップ 1: 認証情報のセットアップ

このサンプルコードの使用を開始するには、`AwsCredentials.properties` ファイルに認証情報を入力する必要があります。具体的には、`secretKey` と `accessKey` を入力する必要があります。

アクセスキー ID および秘密アクセスキーを `AwsCredentials.properties` ファイルにコピーして貼り付けます。

## ステップ 2: セキュリティグループのセットアップ

さらに、セキュリティグループを設定する必要があります。セキュリティグループとは、ファイアウォールとしての役割を果たすものであり、インスタンスのグループに対してどのトラフィックの送受信を許可するかを制御します。デフォルトでは、インスタンスの起動時にセキュリティグループは何も設定されていません。つまり、着信 IP トラフィックは、どの TCP ポートであってもすべて拒否されます。したがって、ここでは、スポットリクエストを提出する前に、必要なネットワークトラフィックを許可するセキュリティグループをセットアップすることにします。このチュートリアルの目的に合わせて、ここでは新しいセキュリティグループを「GettingStarted」という名前で作成します。このグループでは、自分のアプリケーションを実行する IP アドレスからの Secure Shell (SSH) トラフィックを許可します。新しいセキュリティグループをセットアップするには、次に示すコードサンプルをインクルードするか実行する必要があります。このコードは、セキュリティグループをプログラムからセットアップするためのものです。

AmazonEC2 クライアントオブジェクトを作成した後で、`CreateSecurityGroupRequest` オブジェクトを作成し、「GettingStarted」という名前と、セキュリティグループの説明を指定します。その後で、`ec2.createSecurityGroup` API を呼び出してグループを作成します。

このグループにアクセスできるようにするために、`ipPermission` オブジェクトを作成します。IP アドレス範囲は、ローカルコンピュータのサブネット (CIDR 表現) で設定します。IP アドレスの「/10」というサフィックスが、指定した IP アドレスのサブネットを示します。また、`ipPermission` オブジェクトを設定して TCP プロトコルとポート 22 (SSH) を指定します。最後のステップは、`ec2.authorizeSecurityGroupIngress` を呼び出すことです。このときに、作成したセキュリティグループの名前と `ipPermission` オブジェクトを指定します。

(次に示すコードは、最初のチュートリアルで使用したのと同じものです)

```
1 // Retrieves the credentials from an AWSCredentials.properties file.
  AWSCredentials credentials = null;
  try {
    credentials = new PropertiesCredentials(
5      GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
  } catch (IOException e1) {
    System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
    System.out.println(e1.getMessage());
    System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Create a new security group.
    try {
      CreateSecurityGroupRequest securityGroupRequest =
        new CreateSecurityGroupRequest("GettingStartedGroup", "Getting
Started Security Group");
      ec2.createSecurityGroup(securityGroupRequest);
20 } catch (AmazonServiceException ase) {
    // Likely this means that the group is already created, so ignore.
    System.out.println(ase.getMessage());
  }

25 String ipAddr = "0.0.0.0/0";

    // Get the IP of the current host, so that we can limit the security group
    // by default to the IP range associated with your subnet.
```

```

    try {
30      InetAddress addr = InetAddress.getLocalHost();

        // Get IP Address
        ipAddress = addr.getHostAddress()+"/10";
    } catch (UnknownHostException e) {
35 }

    // Create a range that you would like to populate.
    ArrayList<String> ipRanges = new ArrayList<String>();
    ipRanges.add(ipAddr);
40

    // Open up port 22 for TCP traffic to the associated IP from
    // above (e.g., ssh traffic).
    ArrayList<IpPermission> ipPermissions = new ArrayList<IpPermission> ();
    IpPermission ipPermission = new IpPermission();
45 ipPermission.setIpProtocol("tcp");
    ipPermission.setFromPort(new Integer(22));
    ipPermission.setToPort(new Integer(22));
    ipPermission.setIpRanges(ipRanges);
    ipPermissions.add(ipPermission);
50

    try {
        // Authorize the ports to the used.
        AuthorizeSecurityGroupIngressRequest ingressRequest =
            new AuthorizeSecurityGroupIngressRequest("GettingStartedGroup", ip
Permissions);
55      ec2.authorizeSecurityGroupIngress(ingressRequest);
    } catch (AmazonServiceException ase) {
        // Ignore because this likely means the zone has already
        // been authorized.
        System.out.println(ase.getMessage());
60 }

```

このコードサンプル全体を見るには、advanced.CreateSecurityGroupApp.java コードサンプルを参照してください。このアプリケーションを実行して新しいセキュリティグループを作成する必要のあるのは1回のみです。

セキュリティグループは、AWS Toolkit for Eclipse を使用して作成することもできます。詳細については、[ツールキットのドキュメント](#)を参照してください。

### スポットインスタンスリクエスト作成の詳細なオプション

「チュートリアル: Amazon EC2 スポットインスタンス (p. 191)」で説明したように、リクエストを作成してインスタンスタイプ、Amazon マシンイメージ (AMI)、および最高入札価格を指定する必要があります。

初めに、RequestSpotInstanceRequest オブジェクトを作成します。このリクエストオブジェクトには、必要なインスタンスの数と入札価格が必要です。さらに、リクエストの LaunchSpecification も設定する必要があります。この内容は、インスタンスタイプ、AMI ID、および使用するセキュリティグループです。リクエストの内容が入力されたら、AmazonEC2Client オブジェクトの requestSpotInstances メソッドを呼び出します。スポットインスタンスをリクエストする方法の例を次に示します。

(次に示すコードは、最初のチュートリアルで使用したのと同じものです)

```
1 // Retrieves the credentials from an AWSCredentials.properties file.
  AWSCredentials credentials = null;
  try {
    credentials = new PropertiesCredentials(
5      GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
  } catch (IOException e1) {
    System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
    System.out.println(e1.getMessage());
    System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

    // Request 1 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
20 requestRequest.setInstanceCount(Integer.valueOf(1));

    // Set up the specifications of the launch. This includes the
    // instance type (e.g., t1.micro) and the latest Amazon Linux
    // AMI id available. Note, you should always use the latest
25 // Amazon Linux AMI id or another of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
    launchSpecification.setImageId("ami-8c1f5e5");
    launchSpecification.setInstanceType("t1.micro");

30 // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
    securityGroups.add("GettingStartedGroup");
    launchSpecification.setSecurityGroups(securityGroups);

35 // Add the launch specification.
    requestRequest.setLaunchSpecification(launchSpecification);

    // Call the RequestSpotInstance API.
    RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(requestRequest);
```

### 永続リクエストと1回限りのリクエスト

スポットリクエストを作成するときは、複数の任意パラメータを指定できます。最初のパラメータは、そのリクエストが1回限りか永続的なものを指定するためのものです。デフォルトでは、リクエストは1回限りとなります。1回限りのリクエストが受理されるのは1回だけであり、リクエストしたインスタンスが終了すると、そのリクエストはクローズ済みとなります。永続リクエストは、同じリクエストで実行されているスポットインスタンスがない限り、常に受理の対象となります。リクエストのタイプを指定するには、スポットリクエストの Type を設定します。このことを行うコードを次に示します。

```
1 // Retrieves the credentials from an AWSCredentials.properties file.
  AWSCredentials credentials = null;
  try {
```

```

        credentials = new PropertiesCredentials(
5         GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
    } catch (IOException e1) {
        System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
        System.out.println(e1.getMessage());
        System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

    // Request 1 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
20 requestRequest.setInstanceCount(Integer.valueOf(1));

    // Set the type of the bid to persistent.
    requestRequest.setType("persistent");

25 // Set up the specifications of the launch. This includes the
    // instance type (e.g., t1.micro) and the latest Amazon Linux
    // AMI id available. Note, you should always use the latest
    // Amazon Linux AMI id or another of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
30 launchSpecification.setImageId("ami-8c1feca5");
    launchSpecification.setInstanceType("t1.micro");

    // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
35 securityGroups.add("GettingStartedGroup");
    launchSpecification.setSecurityGroups(securityGroups);

    // Add the launch specification.
    requestRequest.setLaunchSpecification(launchSpecification);
40

    // Call the RequestSpotInstance API.
    RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(requestRequest);

```

### リクエストの期間の制限

また、リクエストの有効期間もオプションで指定できます。その期間の開始時点と終了時点の両方を指定できます。デフォルトでは、スポットリクエストが受理の対象とみなされるのは、作成された時点から、そのリクエストが受理されるか作成者によってキャンセルされるまでの間となります。ただし、必要であれば、作成時に有効期間を指定できます。この期間を指定する方法の例を次のコードに示します。

```

1 // Retrieves the credentials from an AWSCredentials.properties file.
  AWSCredentials credentials = null;
  try {
    credentials = new PropertiesCredentials(
5     GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
  } catch (IOException e) {
    System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
    System.out.println(e.getMessage());
    System.exit(-1);
  }

```

```
ties"));
    } catch (IOException e1) {
        System.out.println("Credentials were not properly entered into AwsCre
dentials.properties.");
        System.out.println(e1.getMessage());
        System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
Request();

    // Request 1 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
20 requestRequest.setInstanceCount(Integer.valueOf(1));

    // Set the valid start time to be two minutes from now.
    Calendar cal = Calendar.getInstance();
    cal.add(Calendar.MINUTE, 2);
25 requestRequest.setValidFrom(cal.getTime());

    // Set the valid end time to be two minutes and two hours from now.
    cal.add(Calendar.HOUR, 2);
    requestRequest.setValidUntil(cal.getTime());
30

    // Set up the specifications of the launch. This includes
    // the instance type (e.g., t1.micro)

    // and the latest Amazon Linux AMI id available.
35 // Note, you should always use the latest Amazon
    // Linux AMI id or another of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
    launchSpecification.setImageId("ami-8c1fece5");
    launchSpecification.setInstanceType("t1.micro");
40

    // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
    securityGroups.add("GettingStartedGroup");
    launchSpecification.setSecurityGroups(securityGroups);
45

    // Add the launch specification.
    requestRequest.setLaunchSpecification(launchSpecification);

    // Call the RequestSpotInstance API.
50 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);
```

## Amazon EC2 スポットインスタンスリクエストのグループ化

スポットインスタンスリクエストには、いくつか異なる方法でグループ化するオプションがあります。ここでは、起動グループ、アベイラビリティゾーングループ、およびプレイズメントグループの利点について説明します。

リクエストしたスポットインスタンスがすべて同時に起動され、同時に終了するようにしたい場合は、起動グループを利用します。起動グループとは、1つにまとめる入札のグループに付けられるラベルで

す。同じ起動グループ内のインスタンスはすべて、同時に起動されて同時に終了します。なお、起動グループ内のインスタンスが受理済みの場合に、その同じ起動グループで起動される新しいインスタンスも受理されるという保証はありません。以下の例は、起動グループを設定する手順を示します。

```
1 // Retrieves the credentials from an AWSCredentials.properties file.
  AWSCredentials credentials = null;
  try {
    credentials = new PropertiesCredentials(
5      GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
  } catch (IOException e1) {
    System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
    System.out.println(e1.getMessage());
    System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

    // Request 5 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
20 requestRequest.setInstanceCount(Integer.valueOf(5));

    // Set the launch group.
    requestRequest.setLaunchGroup("ADVANCED-DEMO-LAUNCH-GROUP");

25 // Set up the specifications of the launch. This includes
    // the instance type (e.g., t1.micro) and the latest Amazon Linux
    // AMI id available. Note, you should always use the latest
    // Amazon Linux AMI id or another of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
30 launchSpecification.setImageId("ami-8c1fece5");
    launchSpecification.setInstanceType("t1.micro");

    // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
35 securityGroups.add("GettingStartedGroup");
    launchSpecification.setSecurityGroups(securityGroups);

    // Add the launch specification.
    requestRequest.setLaunchSpecification(launchSpecification);
40

    // Call the RequestSpotInstance API.
    RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(requestRequest);
```

1つのリクエスト内のすべてのインスタンスが同じアベイラビリティーゾーン内で起動されるようにする必要があるが、どのアベイラビリティーゾーンでもかまわない場合は、アベイラビリティーゾーングループを利用します。アベイラビリティーゾーングループとは、同じアベイラビリティーゾーンにまとめるインスタンスのグループに付けられるラベルです。同じアベイラビリティーゾーングループに属し、同時に受理されたインスタンスはすべて、同じアベイラビリティーゾーンで起動されます。アベイラビリティーゾーングループを設定する方法の例を次に示します。

```

1 // Retrieves the credentials from an AWSCredentials.properties file.
  AWSCredentials credentials = null;
  try {
    credentials = new PropertiesCredentials(
5      GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
  } catch (IOException e1) {
    System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
    System.out.println(e1.getMessage());
    System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

    // Request 5 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
20 requestRequest.setInstanceCount(Integer.valueOf(5));

    // Set the Availability Zone group.
    requestRequest.setAvailabilityZoneGroup("ADVANCED-DEMO-AZ-GROUP");

25 // Set up the specifications of the launch. This includes the instance
    // type (e.g., t1.micro) and the latest Amazon Linux AMI id available.
    // Note, you should always use the latest Amazon Linux AMI id or another
    // of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
30 launchSpecification.setImageId("ami-8c1fece5");
    launchSpecification.setInstanceType("t1.micro");

    // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
35 securityGroups.add("GettingStartedGroup");
    launchSpecification.setSecurityGroups(securityGroups);

    // Add the launch specification.
    requestRequest.setLaunchSpecification(launchSpecification);
40

    // Call the RequestSpotInstance API.
    RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(requestRequest);

```

リクエストするスポットインスタンスをどのアベイラビリティゾーンで起動したいかを指定できます。次のコードサンプルでは、アベイラビリティゾーンの設定方法を示します。

```

1 // Retrieves the credentials from an AWSCredentials.properties file.
  AWSCredentials credentials = null;
  try {
    credentials = new PropertiesCredentials(
5      GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
  } catch (IOException e1) {

```

```

        System.out.println("Credentials were not properly entered into AwsCre
dentials.properties.");
        System.out.println(e1.getMessage());
        System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
Request();

    // Request 1 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
20 requestRequest.setInstanceCount(Integer.valueOf(1));

    // Set up the specifications of the launch. This includes the instance
    // type (e.g., t1.micro) and the latest Amazon Linux AMI id available.
    // Note, you should always use the latest Amazon Linux AMI id or another
25 // of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
    launchSpecification.setImageId("ami-8c1fece5");
    launchSpecification.setInstanceType("t1.micro");

30 // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
    securityGroups.add("GettingStartedGroup");
    launchSpecification.setSecurityGroups(securityGroups);

35 // Set up the Availability Zone to use. Note we could retrieve the
    // Availability Zones using the ec2.describeAvailabilityZones() API.
    // For this demo, we will just use us-east-1a.
    SpotPlacement placement = new SpotPlacement("us-east-1b");

40 launchSpecification.setPlacement(placement);

    // Add the launch specification.
    requestRequest.setLaunchSpecification(launchSpecification);

45 // Call the RequestSpotInstance API.
    RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);

```

最後のプレイズメントグループは、ハイパフォーマンスコンピューティング (HPC) スポットインスタンス (クラスターコンピューティングインスタンスやクラスター GPU インスタンスなど) を使用する場合に指定できます。プレイズメントグループを利用すると、低レイテンシー、高帯域幅でインスタンス間を接続できます。以下の例は、プレイズメントグループを設定する手順を示します。

```

1 // Retrieves the credentials from an AWSCredentials.properties file.
    AWSCredentials credentials = null;
    try {
        credentials = new PropertiesCredentials(
5         GettingStartedApp.class.getResourceAsStream("AwsCredentials.proper
ties"));
    } catch (IOException e1) {
        System.out.println("Credentials were not properly entered into AwsCre

```

```

credentials.properties.");
        System.out.println(e1.getMessage());
        System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
Request();

    // Request 1 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
20 requestRequest.setInstanceCount(Integer.valueOf(1));

    // Set up the specifications of the launch. This includes the instance
    // type (e.g., t1.micro) and the latest Amazon Linux AMI id available.
    // Note, you should always use the latest Amazon Linux AMI id or another
25 // of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
    launchSpecification.setImageId("ami-8c1fece5");
    launchSpecification.setInstanceType("t1.micro");

30 // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
    securityGroups.add("GettingStartedGroup");
    launchSpecification.setSecurityGroups(securityGroups);

35 // Set up the placement group to use with whatever name you desire.
    // For this demo, we will just use "ADVANCED-DEMO-PLACEMENT-GROUP".
    SpotPlacement placement = new SpotPlacement();
    placement.setGroupName("ADVANCED-DEMO-PLACEMENT-GROUP");
    launchSpecification.setPlacement(placement);

40
    // Add the launch specification.
    requestRequest.setLaunchSpecification(launchSpecification);

    // Call the RequestSpotInstance API.
45 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);

```

このセクションで示したパラメータはいずれも任意パラメータです。もう1つ重要なことは、これらのパラメータのほとんどは（入札が永続的か1回限りかにかかわらず）、入札受理の可能性の低下につながりうるということです。したがって、これらのオプションは、そのオプションが必要な場合に限って使用することが重要です。これまでに示したコード例すべてを1つにまとめたものが `com.amazonaws.codesamples.advanced.InlineGettingStartedCodeSampleApp.java` クラスの中にあります。

### 中断または終了の後もルートパーティションを永続化する方法

スポットインスタンスの中断を管理する最も簡単な方法は、データのチェックポイントを作成して Amazon Elastic Block Store ( Amazon EBS ) ボリュームに保存するという処理を定期的に行うことです。チェックポイントを定期的を作成しておく、中断が発生したときでも、データが失われるのは最後のチェックポイント以降に作成された分だけになります（その間に他の非べき等アクションが実行されていないことを前提とします）。このプロセスを容易にするには、スポットリクエストを設定するときに、中断時や終了時にルートパーティションを削除しないことを指定します。このシナリオを実現する方法を示す新しいコードが、次の例に挿入されています。

追加されたコードの中では、BlockDeviceMapping オブジェクトを作成し、対応する Elastic Block Storage (EBS) を EBS オブジェクトに設定しています (このオブジェクトは、スポットインスタンスが終了しても削除しないよう設定済みです)。その後で、この BlockDeviceMapping をマッピングの ArrayList に追加し、起動指定の中でこのマッピングを指定します。

```

1 // Retrieves the credentials from an AWSCredentials.properties file.
  AWSCredentials credentials = null;
  try {
    credentials = new PropertiesCredentials(
5      GettingStartedApp.class.getResourceAsStream("AwsCredentials.properties"));
  } catch (IOException e1) {
    System.out.println("Credentials were not properly entered into AwsCredentials.properties.");
    System.out.println(e1.getMessage());
    System.exit(-1);
10 }

    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

15 // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstancesRequest();

    // Request 1 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
20 requestRequest.setInstanceCount(Integer.valueOf(1));

    // Set up the specifications of the launch. This includes the instance
    // type (e.g., t1.micro) and the latest Amazon Linux AMI id available.
    // Note, you should always use the latest Amazon Linux AMI id or another
25 // of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
    launchSpecification.setImageId("ami-8c1feca5");
    launchSpecification.setInstanceType("t1.micro");

30 // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
    securityGroups.add("GettingStartedGroup");
    launchSpecification.setSecurityGroups(securityGroups);

35 // Create the block device mapping to describe the root partition.
    BlockDeviceMapping blockDeviceMapping = new BlockDeviceMapping();
    blockDeviceMapping.setDeviceName("/dev/sda1");

    // Set the delete on termination flag to false.
40 EbsBlockDevice ebs = new EbsBlockDevice();
    ebs.setDeleteOnTermination(Boolean.FALSE);
    blockDeviceMapping.setEbs(ebs);

    // Add the block device mapping to the block list.
45 ArrayList<BlockDeviceMapping> blockList = new ArrayList<BlockDeviceMapping>();
    blockList.add(blockDeviceMapping);

    // Set the block device mapping configuration in the launch specifications.
    launchSpecification.setBlockDeviceMappings(blockList);

```

```

50
// Add the launch specification.
requestRequest.setLaunchSpecification(launchSpecification);

// Call the RequestSpotInstance API.
55 RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);

```

このボリュームがインスタンスの起動時に再度接続されるようにしたい場合は、ブロックデバイスマッピング設定を使用することもできます。別の方法としては、ルート以外のパーティションを接続する場合に、どの Amazon EBS ボリュームをスポットインスタンス再開後にインスタンスに接続するかを指定できます。このようにするには、スナップショット ID を `EbsBlockDevice` オブジェクトで指定し、代替デバイス名を `BlockDeviceMapping` オブジェクトで指定します。ブロックデバイスマッピングを利用すると、インスタンスのブートストラップが容易になります。

ルートパーティションを使用して重要なデータのチェックポイントを作成しておく、インスタンスの中断の可能性を管理するうえで大いに役立ちます。中断の可能性を管理するその他の方法については、[中断の管理についての動画](#)をご覧ください。

### スポットリクエストとインスタンスにタグを付加する方法

EC2 リソースにタグを追加すると、クラウドインフラストラクチャの管理を簡略化できます。タグとは、メタデータの形を取るものであり、わかりやすい名前を付けるのに使用できます。また、検索がしやすくなり、複数ユーザー間での共同作業にも役立ちます。タグは、プロセスのスクリプトや各部分の自動化にも使用できます。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 662\)](#)」を参照してください。

使用するリソースにタグを追加するには、リソースをリクエストした後でタグを付ける必要があります。具体的には、スポットリクエストの提出完了後または `RunInstances` 呼び出しの実行完了後にタグを追加する必要があります。次のコード例で、タグの追加方法を示します。

```

1 /*
   * Copyright 2010-2011 Amazon.com, Inc. or its affiliates. All Rights Re
served.
   *
   * Licensed under the Apache License, Version 2.0 (the "License").
5  * You may not use this file except in compliance with the License.
   * A copy of the License is located at
   *
   * http://aws.amazon.com/apache2.0
   *
10 * or in the "license" file accompanying this file. This file is distributed
   * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
   * express or implied. See the License for the specific language governing
   * permissions and limitations under the License.
   */
15 package com.amazonaws.codesamples.advanced;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

20 import com.amazonaws.AmazonServiceException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.PropertiesCredentials;
import com.amazonaws.codesamples.getting_started.GettingStartedApp;
25 import com.amazonaws.services.ec2.AmazonEC2;

```

```

import com.amazonaws.services.ec2.AmazonEC2Client;
import com.amazonaws.services.ec2.model.CancelSpotInstanceRequestsRequest;
import com.amazonaws.services.ec2.model.CreateTagsRequest;
import com.amazonaws.services.ec2.model.DescribeSpotInstanceRequestsRequest;
30 import com.amazonaws.services.ec2.model.DescribeSpotInstanceRequestsResult;
import com.amazonaws.services.ec2.model.LaunchSpecification;
import com.amazonaws.services.ec2.model.RequestSpotInstancesRequest;
import com.amazonaws.services.ec2.model.RequestSpotInstancesResult;
import com.amazonaws.services.ec2.model.SpotInstanceRequest;
35 import com.amazonaws.services.ec2.model.Tag;
import com.amazonaws.services.ec2.model.TerminateInstancesRequest;

/**
 * Welcome to your new AWS Java SDK based project!
40 *
 * This class is meant as a starting point for your console-based application
that
 * makes one or more calls to the AWS services supported by the Java SDK,
such as EC2,
 * SimpleDB, and S3.
 *
45 * In order to use the services in this sample, you need:
 *
 * - A valid Amazon Web Services account. You can register for AWS at:
 *   https://aws-portal.amazon.com/gp/aws/developer/registration/in
dex.html
 *
50 * - Your account's Access Key ID and Secret Access Key:
 *   http://aws.amazon.com/security-credentials
 *
 * - A subscription to Amazon EC2. You can sign up for EC2 at:
 *   http://aws.amazon.com
55 *
 */

public class InlineTaggingCodeSampleApp {

60 /**
 * @param args
 */
public static void main(String[] args) {
//=====//
65 //===== Submitting a Request =====//
//=====//

// Retrieves the credentials from an AWSCredentials.properties file.
AWSCredentials credentials = null;
70 try {
credentials = new PropertiesCredentials(
GettingStartedApp.class.getResourceAsStream("AwsCredentials.proper
ties"));
} catch (IOException e1) {
System.out.println("Credentials were not properly entered into
AwsCredentials.properties.");
75 System.out.println(e1.getMessage());
System.exit(-1);
}
}

```

```

80    // Create the AmazonEC2Client object so we can call various APIs.
    AmazonEC2 ec2 = new AmazonEC2Client(credentials);

    // Initializes a Spot Instance request
    RequestSpotInstancesRequest requestRequest = new RequestSpotInstances
Request();

85    // Request 1 x t1.micro instance with a bid price of $0.03.
    requestRequest.setSpotPrice("0.03");
    requestRequest.setInstanceCount(Integer.valueOf(1));

    // Set up the specifications of the launch. This includes
90    // the instance type (e.g., t1.micro) and the latest Amazon
    // Linux AMI id available. Note, you should always use the
    // latest Amazon Linux AMI id or another of your choosing.
    LaunchSpecification launchSpecification = new LaunchSpecification();
    launchSpecification.setImageId("ami-8c1fece5");
95    launchSpecification.setInstanceType("t1.micro");

    // Add the security group to the request.
    ArrayList<String> securityGroups = new ArrayList<String>();
    securityGroups.add("GettingStartedGroup");
100    launchSpecification.setSecurityGroups(securityGroups);

    // Add the launch specifications to the request.
    requestRequest.setLaunchSpecification(launchSpecification);

105    //=====
    //===== Getting the Request ID from the Request =====
    //=====

    // Call the RequestSpotInstance API.
110    RequestSpotInstancesResult requestResult = ec2.requestSpotInstances(re
questRequest);
    List<SpotInstanceRequest> requestResponses = requestResult.getSpotIn
stanceRequests();

    // Set up an arraylist to collect all of the request ids we want to
    // watch hit the running state.
115    ArrayList<String> spotInstanceRequestIds = new ArrayList<String>();

    // Add all of the request ids to the hashset, so we can
    // determine when they hit the active state.
    for (SpotInstanceRequest requestResponse : requestResponses) {
120        System.out.println("Created Spot Request: "+requestResponse.getSpot
InstanceRequestId());
        spotInstanceRequestIds.add(requestResponse.getSpotInstanceRequestId());
    }

    //=====
125    //===== Tag the Spot Requests =====
    //=====

    // Create the list of tags we want to create.
    ArrayList<Tag> requestTags = new ArrayList<Tag>();
130    requestTags.add(new Tag("keyname1", "value1"));

    // Create a tag request for the requests.

```

```

CreateTagsRequest createTagsRequest_requests = new CreateTagsRequest();
createTagsRequest_requests.setResources(spotInstanceRequestIds);
135 createTagsRequest_requests.setTags(requestTags);

// Try to tag the Spot Request submitted.
try {
    ec2.createTags(createTagsRequest_requests);
140 } catch (AmazonServiceException e) {
    // Write out any exceptions that may have occurred.
    System.out.println("Error terminating instances");
    System.out.println("Caught Exception: " + e.getMessage());
    System.out.println("Reponse Status Code: " + e.getStatusCode());
145 System.out.println("Error Code: " + e.getErrorCode());
    System.out.println("Request ID: " + e.getRequestId());
}

//=====//
150 //===== Determining the State of the Spot Request =====//
//=====//

// Create a variable that will track whether any
// requests are still in the open state.
155 boolean anyOpen;

// Initialize variables.
ArrayList<String> instanceIds = new ArrayList<String>();

160 do {
    // Create the describeRequest with tall of the request
    // id to monitor (e.g., that we started).
    DescribeSpotInstanceRequestsRequest describeRequest = new DescribeS
potInstanceRequestsRequest();
    describeRequest.setSpotInstanceRequestIds(spotInstanceRequestIds);
165

    // Initialize the anyOpen variable to false - which assumes there are
    no requests open unless
    // we find one that is still open.
    anyOpen = false;

170 try {
    // Retrieve all of the requests we want to monitor.
    DescribeSpotInstanceRequestsResult describeResult = ec2.describeS
potInstanceRequests(describeRequest);
    List<SpotInstanceRequest> describeResponses = describeResult.get
SpotInstanceRequests();

175 // Look through each request and determine if they are all
// in the active state.
    for (SpotInstanceRequest describeResponse : describeResponses) {
        // If the state is open, it hasn't changed since we
        // attempted to request it. There is the potential
180 // for it to transition almost immediately to closed or
        // canceled so we compare against open instead of active.
        if (describeResponse.getState().equals("open")) {
            anyOpen = true;
            break;
185 }
    }
}

```

```

        // Add the instance id to the list we will
        // eventually terminate.
        instanceIds.add(describeResponse.getInstanceId());
190     }
    } catch (AmazonServiceException e) {
        // If we have an exception, ensure we don't break out
        // of the loop. This prevents the scenario where there
        // was a blip on the wire.
195     anyOpen = true;
    }

    try {
        // Sleep for 60 seconds.
200     Thread.sleep(60*1000);
    } catch (Exception e) {
        // Do nothing because it woke up early.
    }
} while (anyOpen);
205

//=====//
//===== Tag the Spot Instances =====//
//=====//

210 // Create the list of tags we want to create.
ArrayList<Tag> instanceTags = new ArrayList<Tag>();
instanceTags.add(new Tag("keyname1","value1"));

// Create a tag request for instances.
215 CreateTagsRequest createTagsRequest_instances = new CreateTagsRequest();
createTagsRequest_instances.setResources(instanceIds);
createTagsRequest_instances.setTags(instanceTags);

// Try to tag the Spot Instance started.
220 try {
    ec2.createTags(createTagsRequest_instances);
} catch (AmazonServiceException e) {
    // Write out any exceptions that may have occurred.
    System.out.println("Error terminating instances");
225 System.out.println("Caught Exception: " + e.getMessage());
    System.out.println("Reponse Status Code: " + e.getStatusCode());
    System.out.println("Error Code: " + e.getErrorCode());
    System.out.println("Request ID: " + e.getRequestId());
}
230

//=====//
//===== Canceling the Request =====//
//=====//

235 try {
    // Cancel requests.
    CancelSpotInstanceRequestsRequest cancelRequest = new CancelSpotIn
stanceRequestsRequest(spotInstanceRequestIds);
    ec2.cancelSpotInstanceRequests(cancelRequest);
} catch (AmazonServiceException e) {
240 // Write out any exceptions that may have occurred.
    System.out.println("Error canceling instances");
    System.out.println("Caught Exception: " + e.getMessage());
    System.out.println("Reponse Status Code: " + e.getStatusCode());

```

```

245     System.out.println("Error Code: " + e.getErrorCode());
        System.out.println("Request ID: " + e.getRequestId());
    }

    //=====//
    //===== Terminating any Instances =====//
    //=====//
250     try {
        // Terminate instances.
        TerminateInstancesRequest terminateRequest = new TerminateInstances
Request(instanceIds);
        ec2.terminateInstances(terminateRequest);
255     } catch (AmazonServiceException e) {
        // Write out any exceptions that may have occurred.
        System.out.println("Error terminating instances");
        System.out.println("Caught Exception: " + e.getMessage());
        System.out.println("Reponse Status Code: " + e.getStatusCode());
260     System.out.println("Error Code: " + e.getErrorCode());
        System.out.println("Request ID: " + e.getRequestId());
    }
} // main
}

```

タグとは、インスタンスのクラスターの管理を容易にするための第一歩です。Amazon EC2 リソースにタグを追加する方法の詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 662\)](#)」を参照してください。

### ステップの集約

これまでに説明したステップは、よりオブジェクト指向的なアプローチをとって1つのクラスに集約し、利便性を高めることができます。Requests という名前のクラスをインスタンス化すると、これらのアクションを実行できます。さらに、GettingStartedApp というクラスも作成します。ここにメインメソッドがあり、ここで高レベルの関数呼び出しを実行します。

ソースコード全体を [GitHub](#) からダウンロードできます。

これで、AWS SDK for Java を使用してスポットインスタンスソフトウェアを開発するための、高度なリクエスト機能のチュートリアルは終了です。

## スポットインスタンスでのクラスターの開始

### Abstract

クラスターをセットアップし、スポットインスタンスでグリッドコンピューティングを使用すると、複数のインスタンスを利用して並列計算を実行します。

グリッドは、ユーザーが複数のインスタンスを活用して並列計算を実行できるようにするための、分散したコンピューティングの形態です。この種類のアーキテクチャはスポットインスタンスに組み込まれた伸縮自在性や、より高いコスト効率で作業を終わらせるための価格が安いという利点があるため、[Numerate](#)、[Scribd](#)、[バルセロナ大学/メルボルン大学](#)といったお客様がスポットインスタンスを使用したグリッドコンピューティングを利用しています。

開始するには、ユーザーがジョブと呼ばれる別々の単位に作業を分類し、その作業を「マスターノード」として送信します。これらのジョブはキューに入り、「スケジューラー」と呼ばれるプロセスによって「ワーカーノード」と呼ばれるグリッドの他のインスタンスに作業が分散されます。ワーカーノードによって結果が計算されると、マスターノードに通知され、ワーカーノードが次の処理をキューから取得できるようになります。ジョブが失敗するか、インスタンスが中断すると、スケジューラープロセスによってジョブが自動的に再度キューに入ります。

アプリケーションのアーキテクトを行う際は、ジョブに含める作業の分量を適切にすることが重要です。処理にかかる時間に基づいて、ジョブを論理グループに分類することをお勧めします。ワークロードを再度処理しなければならない場合に追加コストが発生しないように、通常は、ワークロードのサイズを 1 時間以内で作成する必要があります (もしインスタンスが中断された場合、その 1 時間に対しコストがかかりません)。

多くのお客様が Oracle Grid Engine や UniCloud といったグリッドスケジューラーを使用してクラスターを設定しています。長時間実行されるワークロードがある場合、マスターノードをオンデマンドまたはリザーブドインスタンスで実行し、ワーカーノードをスポットまたはオンデマンド、リザーブド、スポットインスタンスの混合で実行する方法が最善です。または、1時間以内のワークロードがあるか、テスト環境で実行している場合、すべてのインスタンスをスポットで実行することもできます。設定に関係なく、中断の可能性があるインスタンスが自動的に再追加されるスクリプトを作成することをお勧めします。StarCluster のような既存ツールがこのプロセスの管理に役立つ場合もあります。

### 動画クイックルック: スポットでクラスターを起動する方法

AWS ソリューションプロバイダ [BioTeam](#) の Chris Dagdigian (Chris Dagdigian) 氏が、StarCluster を使用して 10~15 分で Amazon EC2 スポットインスタンスにゼロからクラスターを起動する方法の概要を説明します。StarCluster は、MIT の研究室で開発された、新しい Oracle Grid Engine クラスターの設定を容易にするオープンソースのツールです。この動画では、インストール、設定、およびクラスターでの簡単なジョブの実行のプロセスを順を追って説明します。加えて、Chris はスポットインスタンスを活用するため、より迅速な作業の完了、および 50~66% の節約が実現する可能性があります。 [スポットでクラスターを起動する方法](#)

## リザーブドインスタンス

Amazon Elastic Compute Cloud ( Amazon EC2 ) リザーブドインスタンス価格モデルを使用すると、EC2 インスタンスのために処理能力を予約でき、平均インスタンスコストを抑えることができます。リザーブドインスタンスを使用する場合は、処理能力の予約に対して低額の1回限りの予約金を支払うと、インスタンスの時間当たりの使用料金について大幅な割引を受けることができます。予約済み処理能力を使用するときは、購入した処理能力と同じ構成で EC2 インスタンスを起動します。インスタンスの起動方法は関係ありません。オンデマンドインスタンスとして直接起動したり、Auto Scaling 経由で起動したり、あるいは別のプロセスを使用して起動することが可能です。アマゾン ウェブ サービス ( AWS ) では、お客様の処理能力予約に対応する割引された時間料金が自動的に適用されます。リザーブドインスタンスを保有している限り、EC2 インスタンスの割引された時間料金が課金されます。

リザーブドインスタンスの期間が終了しても、中断せずに EC2 インスタンスを使い続けることができます。ただし、その時点からはオンデマンド料金で課金されます。オンデマンドインスタンスの起動については、[インスタンスの起動 \(p. 324\)](#) を参照してください。Auto Scaling を使用したインスタンスの起動については、[Auto Scaling 開発者ガイド](#) を参照してください。

リザーブドインスタンスにより、お客様がご自身のハードウェアを所有している場合でも、オンデマンドのインスタンスのみを実行している場合でも、かなりの節約になるばかりでなく、必要な能力を必要となときに利用できることが保証されます。

Amazon EC2 リザーブドインスタンスを購入するには、インスタンスタイプ ( m1.small など )、プラットフォーム ( Linux/UNIX、Windows、Windows と SQL Server )、場所 ( リージョンおよびアベイラビリティゾーン )、および期間 ( 1 年または 3 年 ) を選択する必要があります。特定の Linux/UNIX プラットフォームでリザーブドインスタンスを実行する場合、予約済み処理能力を購入するときに、そのプラットフォームを指定する必要があります。その後、購入したリザーブドインスタンスを使用する準備ができたら、その Linux/UNIX プラットフォームを実行する Amazon マシンイメージ ( AMI ) と、購入時に指定したその他すべての仕様を選択する必要があります。

たとえば、Singapore リージョンで 1 年間の *SUSE Linux*、m1.medium リザーブドインスタンスが必要な場合、Singapore リージョンの 1 年間の *Linux/UNIX*、m1.medium リザーブドインスタンスを購入しても処理能力の予約や料金メリットは得られません。リザーブドインスタンスの購入とインスタンスの起動の両方で、同じ *SUSE Linux* 製品プラットフォームを指定する必要があります。

製品の料金については、次のページを参照してください。

- [AWS サービス料金の概要](#)
- [Amazon EC2 On-Demand Instances Pricing](#)
- [Amazon EC2 Reserved Instance Pricing](#)

## リザーブドインスタンスの概要

Amazon EC2 リザーブドインスタンスの使用を開始する際に役立つ情報を以下に示します。

- まず、リザーブドインスタンスの使用を開始できるように、登録、サインアップ、ツールのインストールなどの前提条件となるタスクを完了します。詳細については、「[リザーブドインスタンスの使用を開始する \(p. 219\)](#)」を参照してください。
- リザーブドインスタンスを売買する前に、「[リザーブドインスタンスを使用するためのステップ \(p. 220\)](#)」を読んで詳細を学ぶことができます。
- リザーブドインスタンスは、標準として 1 年および 3 年の期間を AWS から購入でき、標準以外の期間はサードパーティの再販業者から リザーブドインスタンスマーケットプレイス を通じて購入できます。
- インスタンスの予定される使用頻度に最もよく合う価格モデルを選択して、リザーブドインスタンスのコストを最適化してください。詳細については、「[使用計画に基づいてリザーブドインスタンスを選択する \(p. 225\)](#)」を参照してください。

- リザーブインスタンスの料金メリットに関する情報が用意されています。詳細については、「[リザーブインスタンスの料金メリットについて \(p. 231\)](#)」を参照してください。
- リザーブインスタンス料金範囲 および割引料金の利用に関する情報が用意されています。詳細については、「[リザーブインスタンス料金範囲を理解する \(p. 226\)](#)」を参照してください。
- 未使用のリザーブインスタンスは、リザーブインスタンスマーケットプレイス で販売することができます。リザーブインスタンスマーケットプレイス では、不要になったリザーブインスタンスを持つ販売者が、追加の処理能力を求める購入者を見つけることができます。リザーブインスタンスマーケットプレイス を通じて売買されたリザーブインスタンスは、他のリザーブインスタンス同様に動作します。詳細については、「[リザーブインスタンスマーケットプレイス \(p.233\)](#)」を参照してください。

リザーブインスタンスおよび リザーブインスタンスマーケットプレイス を使用するための要件をまとめたチェックリストについては、[リザーブインスタンスの要件チェックリスト \(p. 288\)](#) を参照してください。

## 次は何をしますか。

- 知識:
  - [リザーブインスタンスの使用を開始する \(p. 219\)](#)
  - [リザーブインスタンスを購入する \(p. 236\)](#)
  - [リザーブインスタンスマーケットプレイス での販売 \(p. 261\)](#)
  - [Amazon VPC でのリザーブインスタンスの使用 \(p. 221\)](#)
- 開始:
  - [購入者になる \(p. 237\)](#)
  - [リザーブインスタンスを購入する \(p. 238\)](#)
  - [リザーブインスタンスに関する情報を取得する \(p. 246\)](#)
  - [リザーブインスタンスの変更 \(p. 251\)](#)
  - [販売者として登録する \(p. 262\)](#)
  - [リザーブインスタンスの出品 \(p. 267\)](#)

## リザーブインスタンスの使用を開始する

### Topics

- [設定 \(p. 219\)](#)
- [リザーブインスタンスを使用するためのステップ \(p. 220\)](#)
- [Amazon VPC でのリザーブインスタンスの使用 \(p. 221\)](#)
- [リザーブインスタンスを使用するためのツール \(p. 222\)](#)

Amazon Elastic Compute Cloud ( Amazon EC2 ) リザーブインスタンスを使用すると、インスタンスのための処理能力を予約でき、低価格コンピューティングのメリットが得られます。リザーブインスタンスを使用すれば、低額の1回限りの予約金を支払うことによって、インスタンスの時間当たりの使用料金について大幅な割引を受けることができます。リザーブインスタンスにより、お客様が自身のハードウェアを所有している場合でも、オンデマンドのインスタンスのみを実行している場合でも、かなりの節約になるばかりでなく、必要な能力を必要なときに利用できることが保証されます。このトピックでは、リザーブインスタンスの使用を開始するために必要な基本情報を説明します。

## 設定

リザーブインスタンスの使用を開始する前に、以下のタスクを実行する必要があります。

- サインアップする。

リザーブドインスタンスを使用するには、Amazon EC2 のアカウントおよび認証情報によるサインアップに関する情報を提供している [Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#) で説明されている手順を読み、実行する必要があります。

- ツールをインストールする。

Amazon EC2 ツール – AWS Management Console、Amazon EC2 コマンドラインインターフェイス (CLI) ツール、または Amazon EC2 API – を使用することにより、EC2 リザーブドインスタンスを使用したり提供タイプを検索することができます。詳細については、「[リザーブドインスタンスを使用するためのツール \(p. 222\)](#)」を参照してください。

特定のツールを使用してリザーブドインスタンスの使用を開始するには、「[AWS マネジメントコンソール \(p. 222\)](#)」、「[コマンドラインインターフェイスツール \(p. 223\)](#)」、または「[API \(p. 224\)](#)」を参照してください。

## リザーブドインスタンスを使用するためのステップ

リザーブドインスタンスを使用するには、以下の 5 つのステップを実行します。Amazon EC2 リザーブドインスタンスを購入すると、それを起動できるようになります。所有しているリザーブドインスタンスは表示および変更でき、未使用のリザーブドインスタンスはリザーブドインスタンスマーケットプレイスで販売できます (制限事項があります。詳細については、「[リザーブドインスタンスの要件チェックリスト \(p. 288\)](#)」を参照してください)。このセクションでは、リザーブドインスタンスの購入、起動、表示、変更、および販売について説明します。

これらのタスクはいずれも、AWS Management Console、Amazon EC2 CLI ツール、または Amazon EC2 API を使用して実行できます。始める前に、前提条件となるアカウントとツールを設定する必要があります。詳細については、「[設定 \(p. 219\)](#)」を参照してください。

- 購入する。

- 予約する処理能力を決定します。インスタンスを予約するための以下の基準を指定します。
  - プラットフォーム (例: Linux/UNIX)。



### Note

特定の Linux/UNIX プラットフォームでリザーブドインスタンスを実行する場合、予約済み処理能力を購入するときに、そのプラットフォームを指定する必要があります。そして購入した予約済み処理能力を使用してインスタンスを起動するときに、その Linux/UNIX プラットフォームを実行する Amazon マシンイメージ (AMI) と、購入時に指定したその他すべての仕様を選択する必要があります。

- インスタンスタイプ (例: m1.small)。
  - 処理能力を予約する期間 (1 年または 3 年)。
  - シングルテナントハードウェア (共有でなく専用テナンシー) で実行するインスタンスのために処理能力を予約する場合は、テナンシーの指定。
  - インスタンスを実行するリージョンおよびアベイラビリティゾーン。
- 支払額、およびインスタンスの予定される実行頻度に最も適した提供タイプを選択します。
    - 重度使用
    - 中度使用
    - 軽度使用

これらの提供タイプについては、[使用計画に基づいてリザーブドインスタンスを選択する \(p. 225\)](#) を参照してください。

- 指定した基準を満たす提供タイプを探します。

d. 要件を満たす提供タイプを購入します。

詳細については、「[リザーブインスタンスを購入する \(p. 238\)](#)」を参照してください。

2. 起動する。

予約済みインスタンスを使用するには、リザーブインスタンスと同じ認証情報 (リザーブインスタンスを購入したときに指定したものと同一リージョン、アベイラビリティゾーン、インスタンスタイプ、およびプラットフォーム) で、オンデマンドの EC2 インスタンスを起動します。ステップ 1 を参照してください。



#### Note

リザーブインスタンスの価格メリットおよび処理能力の予約は、所有している実行中の EC2 インスタンスのうち、まだ予約の対象になっていないすべてのインスタンスに対して自動的に適用されます。

詳細については、「[インスタンスの起動 \(p. 324\)](#)」を参照してください。

3. 表示する。

所有している、または自分のアカウントから使用可能なリザーブインスタンスは、表示して、インスタンスが指定どおりに実行されているかどうか確認できます。

詳細については、「[リザーブインスタンスに関する情報を取得する \(p. 246\)](#)」を参照してください。

4. 変更する。

リザーブインスタンスについては、同じリージョン内でのアベイラビリティゾーン間で移動する、同じインスタンスファミリー (M1 インスタンスファミリーなど) の別のインスタンスタイプに変更する、あるいは EC2-VPC と EC2-Classic の間でリザーブインスタンスのネットワークプラットフォームを変更するなどの変更を行えます。

詳細については、「[リザーブインスタンスの変更 \(p. 251\)](#)」を参照してください。

5. 不要なリザーブインスタンス処理能力を販売する。

a. [Seller Registration] ウィザードを使用して、リザーブインスタンスマーケットプレイスに販売者として登録します。詳細については、「[販売者として登録する \(p. 262\)](#)」を参照してください。



#### Note

リザーブインスタンスマーケットプレイスで販売できないお客様もあり、リザーブインスタンスマーケットプレイスで販売できないリザーブインスタンスも存在します。詳細については、「[リザーブインスタンスの要件チェックリスト \(p. 288\)](#)」を参照してください。

b. 販売しようとするリザーブインスタンスの価格を決定します。詳細については、「[リザーブインスタンスの価格決定 \(p. 267\)](#)」を参照してください。

c. リザーブインスタンスの一覧を表示します。詳細については、「[リザーブインスタンスの出品 \(p. 267\)](#)」を参照してください。

d. お客様のリザーブインスタンスが売れたときに、どのように支払いを受けるかを確認します。詳細については、「[支払いを受け取る \(p. 287\)](#)」を参照してください。

## Amazon VPC でのリザーブインスタンスの使用

Amazon Virtual Private Cloud (Amazon VPC) でリザーブインスタンスを起動するには、デフォルトの VPC をサポートするアカウントを持っているか、Amazon VPC リザーブインスタンスを購入する必要があります。

お客様のアカウントが VPC をサポートしていない場合は、*Amazon VPC* を名前に含むプラットフォームを選択して Amazon VPC リザーブドインスタンスを購入する必要があります。詳細については、「[Detecting Your Supported Platforms and Whether You Have a Default VPC](#)」を参照してください。Amazon VPC の詳細については、「[Amazon VPC とは](#)」を参照してください（「*Amazon Virtual Private Cloud ユーザーガイド*」）。

お客様のアカウントがデフォルトの VPC をサポートしていない場合、使用可能なプラットフォームの一覧には、*Amazon VPC* を名前に含むものは表示されません。これは、すべてのプラットフォームがデフォルトのサブネットを持っているためです。この場合、予約し、料金を支払った処理能力と同じ構成のインスタンスを起動すると、そのインスタンスはデフォルトの VPC 内で起動し、処理能力の予約と請求のメリットが適用されます。デフォルトの VPC の詳細については、「[Your Default VPC and Subnets](#)」を参照してください（「*Amazon Virtual Private Cloud ユーザーガイド*」）。

また、専用インスタンステナンシーを指定すると、ホストハードウェアレベルで物理的に切り離されたリザーブドインスタンスを購入することもできます。詳細については、「[Amazon Virtual Private Cloud ユーザーガイド](#)」の「[Using EC2 Dedicated Instances Within Your VPC](#)」を参照してください。

## リザーブドインスタンスを使用するためのツール

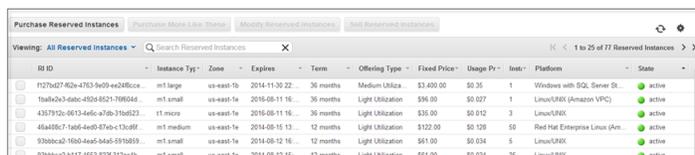
[AWS マネジメントコンソール](#) (p. 222)、[コマンドラインインターフェイスツール](#) (p. 223)、または [API](#) (p. 224) を使用することにより、使用可能な Amazon EC2 リザーブドインスタンスの一覧表示または検索、リザーブドインスタンス処理能力の購入、リザーブドインスタンスの管理、未使用のリザーブドインスタンスの販売を実行できます。

CLI ツールを使用する場合は、まず、「[Setting Up the Amazon EC2 Tools](#)」で説明されている手順を読み、完了する必要があります。このトピックでは、CLI ツールを使用するための環境を設定する手順について説明します。

## AWS マネジメントコンソール

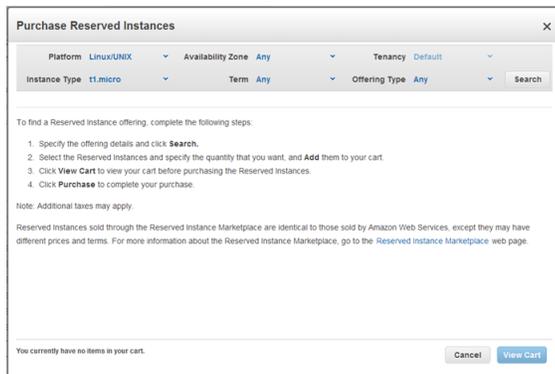
AWS Management Console には、リザーブドインスタンスのタスクのために特別に設計されたツールが用意されています。これは、Amazon EC2 コンソールにあります。さらに、リザーブドインスタンスを使用するとき起動するインスタンスの管理に使用できる一般的なツールもあります。

- [Reserved Instances] ページで、リザーブドインスタンスを操作します。



RI ID	Instance Type	Zone	Expires	Term	Offering Type	Fixed Price	Usage Pr	Insts	Platform	State
f127b27-62a-47c3-9e03-e24f0ce	m1.large	us-east-1b	2014-11-30 22:00	36 months	Medium Utiliza	\$3,400.00	\$0.35	1	Windows with SQL Server S...	active
1a8b2a3-d4bc-450a-8521-7090d4d	m1.small	us-east-1e	2016-08-11 16:00	36 months	Light Utilization	\$36.00	\$0.027	1	Linux/UNIX (Amazon VPC)	active
4357932-0613-446c-a7db-31ba8223	t1.micro	us-east-1e	2016-08-11 16:00	36 months	Light Utilization	\$36.00	\$0.012	3	Linux/UNIX	active
45a40927-1a6b-4a4b-87ba-c131d9f	m1.medium	us-east-1e	2014-09-15 13:00	12 months	Light Utilization	\$122.00	\$0.129	50	Red Hat Enterprise Linux (Am...	active
93b0bc02-1030-4a4b-84ab-83919a859	m1.small	us-east-1e	2014-09-12 16:00	12 months	Light Utilization	\$61.00	\$0.034	5	Linux/UNIX	active
93b0bc02-1a17-4603-8239-312a4db	m1.small	us-east-1e	2014-09-12 16:00	12 months	Light Utilization	\$61.00	\$0.034	25	Linux/UNIX	active

- [Purchase Reserved Instances] ページを使用して、購入するリザーブドインスタンスの詳細を指定します。



- [Instances] ページを使用して、リザーブドインスタンスを起動し、管理します。

## コマンドラインインターフェイスツール

リザーブドインスタンスを購入、またはリザーブドインスタンスマーケットプレイスで販売するには、それらのタスク用に特別に設計された Amazon EC2 コマンドラインインターフェイス (CLI) ツールを使用することができます。リザーブドインスタンスが起動されているときにインスタンスを管理するには、他の Amazon EC2 インスタンスに使用する場合と同じコマンドを CLI ツールで使用します。

次の表は、CLI ツールをリザーブドインスタンスのタスクで使用するときのコマンドの一覧です。

タスク	CLI
購入したリザーブドインスタンスの一覧を表示する	<code>ec2-describe-reserved-instances</code>
リザーブドインスタンスを変更します。	<code>ec2-modify-reserved-instances</code>
リザーブドインスタンスに対して行った変更を表示します。	<code>ec2-describe-reserved-instances-modifications</code>
購入できるリザーブドインスタンスの提供タイプを表示する	<code>ec2-describe-reserved-instances-offerings</code>
販売するリザーブドインスタンスの一覧をリザーブドインスタンスマーケットプレイス内に作成します。	<code>ec2-create-reserved-instances-listing</code>
リザーブドインスタンスの一覧の詳細をリザーブドインスタンスマーケットプレイスで表示します。	<code>ec2-describe-reserved-instances-listings</code>
リザーブドインスタンスを購入する	<code>ec2-purchase-reserved-instances-offering</code>

タスク	CLI
リザーブインスタンスマーケットプレイス内の有効なリザーブインスタンスの一覧をキャンセルします。	<code>ec2-cancel-reserved-instances-listing</code>

CLI コマンドについては、「[Amazon Elastic Compute Cloud コマンドラインリファレンス](#)」を参照してください。

## API

リザーブインスタンスを購入するには、これらのタスク用に特別に設計された API 呼び出しを使用します。他の Amazon EC2 インスタンスに使用する API 呼び出しと同じものを使用して、リザーブインスタンスの起動時にインスタンスを管理できます。

次の表は、リザーブインスタンスのタスクで使用する API 呼び出しの一覧です。

タスク	API
購入したリザーブインスタンスの一覧を表示する	<code>DescribeReservedInstances</code>
リザーブインスタンスを変更します。	<code>ModifyReservedInstances</code>
リザーブインスタンスに対して行った変更を表示します。	<code>DescribeReservedInstancesModifications</code>
購入できるリザーブインスタンスの提供タイプを表示する	<code>DescribeReservedInstancesOfferings</code>
販売するリザーブインスタンスの一覧をリザーブインスタンスマーケットプレイス内に作成します。	<code>CreateReservedInstancesListing</code>
リザーブインスタンスの一覧の詳細をリザーブインスタンスマーケットプレイスで表示します。	<code>DescribeReservedInstancesListings</code>
リザーブインスタンスを購入する	<code>PurchaseReservedInstancesOffering</code>
リザーブインスタンスマーケットプレイス内の有効なリザーブインスタンスの一覧をキャンセルします。	<code>CancelReservedInstancesListing</code>

API アクションについては、「[Amazon Elastic Compute Cloud API Reference](#)」を参照してください。

## リザーブインスタンスの基礎

ここでは、リザーブインスタンスの利点を最大限に利用し、効果的に使用および管理するのに役立つ基本概念について説明します。

- [使用計画に基づいてリザーブインスタンスを選択する \(p. 225\)](#) – インスタンスの予定される使用頻度に最もよく合う価格モデルを選択します。
- [リザーブインスタンス料金範囲を理解する \(p. 226\)](#) – リージョンでのリザーブインスタンスの合計前払いの表示価格が 250,000 USD 以上の場合は、リージョンのリザーブインスタンス料金範囲割引を利用します。

- [リザーブドインスタンスの料金メリットについて \(p. 231\)](#) – リザーブドインスタンスの料金面のメリットがどのように適用されるかを学習します。
- [リザーブドインスタンスマーケットプレイス \(p. 233\)](#) – リザーブドインスタンスマーケットプレイスが AWS をご利用するお客様一に提供する柔軟性を理解します。これにより、ニーズの変化に応じてリザーブドインスタンスの余りを販売したり、標準期間未満でリザーブドインスタンスを他の AWS カスタマーから購入したりできます。

## 使用計画に基づいてリザーブドインスタンスを選択する

インスタンスの予定される使用頻度に基づいて、リザーブドインスタンスの料金構造を選択できます。インスタンスの予想される利用度に対応するために、**重度使用**、**中度使用**、**軽度使用** の 3 種類を用意しています。(これらのリザーブドインスタンスの種類を使用するには、API バージョン 2011-11-01 以降が必要です。)

**重度使用** リザーブドインスタンスを使用すると、一定の基本的な処理能力を持つ負荷を有効にしたり、安定した状態の負荷を実行できます。重度使用 リザーブドインスタンスは、必要な前払いの義務が最も高額です。しかし、3 年間の約 35% 以上の時間に渡ってリザーブドインスタンスを実行する予定がある場合は、すべての提供タイプの中で節約額が最も大きくなる可能性があります (インスタンス利用率が 100% の場合、最大でオンデマンド価格の約 65%)。重度使用 リザーブドインスタンスを利用した場合、他のリザーブドインスタンス提供タイプとは違って、予約金を 1 回支払えば、インスタンスが実行されているかどうかにかかわらず期間中は低額の使用料が時間単位で適用されます。

**中度使用** リザーブドインスタンスは、インスタンスを長い時間使用する場合に、予約金を低く抑えたいとき、またはインスタンスが停止したらすぐに支払いを中止できるようにしたいときに最適です。中度使用 は、リザーブドインスタンス期間の約 19% から 35% の時間に渡ってリザーブドインスタンスを実行する予定の場合にコスト効果の高い選択肢です。このオプションを使用すると、オンデマンド料金から最大 59% を節約できます。中度使用 リザーブドインスタンスの予約金は 軽度使用 リザーブドインスタンスをわずかに上回りますが、インスタンス実行時の時間当たりの使用料は低く抑えられます。これは、API バージョン 2011-11-01 よりも前のリザーブドインスタンスの提供タイプと同等です。

**軽度使用** リザーブドインスタンスは、1 日に数時間、週に数日間のみ実行される定期的な作業負荷に最適です。軽度使用 リザーブドインスタンスでは、予約金を 1 回支払えば、インスタンスの実行時に時間単位で割引使用料が課金されます。インスタンスの実行時間がリザーブドインスタンス期間の 11% から 19% になるとコスト節減が始まり、リザーブドインスタンスの全期間を通してオンデマンド料金から最大 49% を節約できます。



### Note

**軽度使用** および **中度使用** リザーブドインスタンスを使用する場合は、1 回だけの前払い料金を支払い、その後はインスタンスを使用したときにのみ時間単位の料金を支払います。**重度使用** リザーブドインスタンスを利用する場合は、1 回だけの前払い料金を支払い、リザーブドインスタンスの期間内におけるすべての時間について、インスタンスを使用するかどうかに関係なく時間単位の料金を支払う契約を結びます。

リザーブドインスタンスの購入に対する割引利用料金は、インスタンスのタイプおよびアベイラビリティゾーンに指定に関連付けられています。リザーブドインスタンスを購入した結果として割引が適用されている EC2 インスタンスの実行を停止し、その時点でリザーブドインスタンスの期間がまだ終了していない場合、その残り期間内に同じ仕様の別のインスタンスを起動すれば、起動したそのインスタンスに引き続き割引が適用されます。

次の表は、リザーブドインスタンスの提供タイプごとの違いをまとめたものです。

## リザーブドインスタンスの提供タイプ

提供タイプ	前払いコスト	使用料	メリット
重度使用	高	時間当たりの使用料が最も低く、リザーブドインスタンスの使用状況にかかわらず期間全体に適用されます。	リザーブドインスタンスの使用が3年間で全体の約35%を超える場合に、全体的なコストを最も抑えることができます。
中度使用	中	インスタンスの使用時間に応じて使用料が時間単位で課金されます。オフになっているインスタンスは課金されないため、使用していないインスタンスはオフにすることをお勧めします。	変動の大きい負荷、または使用量が3年間で全体の約19%から35%の中程度であると予想される場合に適しています。
軽度使用	低	時間当たりの使用料で課金されます。3種類のうち最も高い料金設定ですが、課金されるのは、リザーブドインスタンスを使用しているときだけに限られます。オフになっているインスタンスは課金されないため、使用していないインスタンスはオフにすることをお勧めします。	常時実行する場合は全体的なコストが最も高くなりますが、まれにしかリザーブドインスタンスを使用しない(3年間で全体の約11%から19%)場合はこのオプションが適しています。

## リザーブドインスタンス料金範囲を理解する

リージョンの Amazon Elastic Compute Cloud ( EC2 ) リザーブドインスタンス料金範囲割引を利用するには、リージョンでのリザーブドインスタンスの前払いの定価総額が 250,000 USD 以上である必要があります。割引料金範囲が適用されると、そのアカウントは、以降、その範囲レベル内で行われるすべてのリザーブドインスタンス購入の前払い料金および使用料に対して、自動的に割引を受けます。

ここでは、リザーブドインスタンスの料金範囲および料金範囲割引を利用する方法について説明します。

- [リザーブドインスタンス料金範囲とは \(p. 226\)](#)
- [現在の制限 \(p. 227\)](#)
- [料金範囲レベルの決定 \(p. 227\)](#)
- [料金範囲の割引はどのように適用されますか? \(p. 229\)](#)

### リザーブドインスタンス料金範囲とは

次の表では、各料金範囲の条件、およびリージョンのアクティブなリザーブドインスタンスの合計前払い表示価格がその料金範囲内になったときに適用される割引額を表示します。

範囲レベル	リージョンのアクティブなリザーブインスタンスの合計前払い表示価格	範囲内で購入されたリザーブインスタンスの前払い料金および使用料に適用される割引
範囲 0	\$0 - \$249,999	標準のリザーブインスタンスの前払い料金と使用料。割引なし。
範囲 1	\$250,000 - \$1,999,999	5% 割引
範囲 2	2,000,000 ~ 5,000,000 USD	10% 割引
範囲 3	5,000,000 USD 超	<a href="#">お問い合わせ</a>

## 現在の制限

現在、リザーブインスタンスの料金範囲には以下の制限が適用されます。

- Amazon EC2 リザーブインスタンス料金範囲割引に適用される購入は、Amazon EC2 リザーブインスタンスの購入のみです。Amazon EC2 リザーブインスタンスの料金範囲およびそれに関連する割引は、Amazon EC2 リザーブインスタンスの購入にのみ適用されます。
- Amazon EC2 リザーブインスタンス料金範囲は、SQL Server Standard または SQL Server Web を使用する Windows 用のリザーブインスタンスには適用されません。
- 料金範囲に基づく割引の一部として購入された Amazon EC2 リザーブインスタンスは、リザーブインスタンスマーケットプレイスで販売することはできません。リザーブインスタンスマーケットプレイスの詳細については、「[リザーブインスタンスマーケットプレイス \(p. 233\)](#)」を参照してください。

リザーブインスタンスおよび リザーブインスタンスマーケットプレイス を使用するための要件をまとめたチェックリストについては、「[リザーブインスタンスの要件チェックリスト \(p. 288\)](#)」を参照してください。

## 料金範囲レベルの決定

特定のリージョンで適用されるリザーブインスタンス料金範囲を確認するには、リザーブインスタンスの合計前払い表示価格の合計と、料金範囲に必要な合計前払い表示価格を比較します。表示価格は、割引が適用されていないリザーブインスタンス料金のことで、AWS Management Console または AWS マーケティングウェブサイトに表示されています。リザーブインスタンスを購入した後でリザーブインスタンスの料金が引き下げられた場合、割引前のリザーブインスタンス料金にはその値下げ分が反映されない可能性があることにご注意ください。

表示価格は支払価格と同じではありません。支払価格は、リザーブインスタンスに対してお客様が実際に支払った金額です（支払価格は、AWS Management Console のリザーブインスタンスツール、コマンドラインインターフェイス (CLI)、または API を使用したときに表示される固定価格と同じです）。リザーブインスタンスが割引ありで購入された場合、支払価格は表示価格より安くなります。リザーブインスタンスが割引なしで購入された場合、支払価格は表示価格と同じになります。リザーブインスタンスを割引なしで購入した場合は、コンソール、CLI、または API の固定価格の値を使用して、購入が該当する料金範囲を確認できます。



### Note

Amazon EC2 コンソールでは、右上の [Show/Hide] をクリックして [Fixed Price] 列の表示を有効にする必要があるかもしれません。

リザーブインスタンスを割引なしで購入したとすると、リージョンのすべてのリザーブインスタンスの合計固定価格を計算することによって、アカウントの料金範囲を決定できます。コンソールを使用してこれを行うには、[Fixed Price] 列の金額の合計を計算します。

CLI、Amazon EC2 CLI、または Amazon EC2 API を使用して、`aws ec2 describe-reserved-instances` AWS CLI コマンド、`ec2-describe-reserved-instances` EC2 コマンド、または `DescribeReservedInstances` アクションによって返される **固定価格の値の合計** を計算することによって、アカウントの料金範囲を決定できます。

AWS CLI コマンドを使用すると、次の例のようになります。

```
$ aws ec2 describe-reserved-instances
```

Amazon EC2 が次の例のような出力を返します。

```
{
  "ReservedInstances": [
    {
      "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342fexample",
      "OfferingType": "Medium Utilization",
      "AvailabilityZone": "us-west-1a",
      "End": "2016-08-14T21:34:34.000Z",
      "ProductDescription": "Linux/UNIX",
      "UsagePrice": 0.01,
      "RecurringCharges": [],
      "Start": "2013-08-15T21:34:35.086Z",
      "State": "active",
      "FixedPrice": 82.0,
      "CurrencyCode": "USD",
      "Duration": 94608000,
      "InstanceTenancy": "default",
      "InstanceType": "t1.micro",
      "InstanceCount": 10
    },
    {
      "ReservedInstancesId": "d16f7a91-d8b4-4b0f-bef9-aef08example",
      "OfferingType": "Medium Utilization",
      "AvailabilityZone": "us-west-1c",
      "End": "2016-08-14T21:34:34.000Z",
      "ProductDescription": "Linux/UNIX",
      "UsagePrice": 0.01,
      "RecurringCharges": [],
      "Start": "2013-08-15T21:34:35.455Z",
      "State": "active",
      "FixedPrice": 82.0,
      "CurrencyCode": "USD",
      "Duration": 94608000,
      "InstanceTenancy": "default",
      "InstanceType": "t1.micro",
      "InstanceCount": 10
    },
    ...
  ]
}
```

Amazon EC2 CLI コマンドを使用する例については、「[ec2-describe-reserved-instances](#)」を参照してください。Amazon EC2 API アクションを使用する例については、「[DescribeReservedInstances](#)」を参照してください。

## 料金範囲の割引はどのように適用されますか？

リージョンのリザーブインスタンスの1回の購入でしきい値を超える場合は、そのご購入分のうち、割引範囲のしきい値を超える部分が割引になります。範囲と割引価格ポイントについては、前のセクションの表を参照してください。



### Note

Amazon EC2 リザーブインスタンスの購入は、Amazon EC2 リザーブインスタンスの料金範囲を決定する唯一の購入であり、Amazon EC2 リザーブインスタンス料金範囲は Amazon EC2 リザーブインスタンスの購入に対してのみ適用されます。

割引範囲のしきい値をまたがるリザーブインスタンスの購入割引に影響を与える例を次に示します。現在、us-east-1 リージョンに 200,000 USD 相当のアクティブなリザーブインスタンスを持っているとします。表示価格が 1,000 USD のリザーブインスタンスを 75 個購入すると、購入額の合計は 75,000 USD です。これにより、アクティブなリザーブインスタンスに対して支払った総額は 275,000 USD になります。割引料金のしきい値は 250,000 USD であるため、新しく購入したリザーブインスタンスの最初の 50,000 USD については割引にはなりません。残りの 25,000 USD は割引料金のしきい値を超えているので、5% (1,250 USD) の割引が適用されます。つまり、購入分の残り (25 インスタンス) に対しては 23,750 USD が課金され、その 25 個のリザーブインスタンスの使用料には割引が適用されます。(ただし、前払い - 割引なし - の合計表示価格はまだ 275,000 USD であることに注意してください。)

リージョンでのアクティブなリザーブインスタンスの合計前払い表示価格が割引料金範囲に該当した場合、そのリージョンでのリザーブインスタンスは、以降、すべての購入が割引料金で課金されます。合計表示価格が割引範囲の金額を上回っている限り、そのリージョンで購入されるリザーブインスタンスには、すべて割引が適用されます。合計表示価格が割引範囲の金額を下回った場合は (一部のリザーブインスタンスの有効期限が切れた場合など)、以降、そのリージョンで購入されるリザーブインスタンスには割引が適用されません。ただし、もともと割引料金範囲に該当して購入されたりリザーブインスタンスに対しては、引き続き割引が適用されます。

アカウントが一括請求アカウントに含まれる場合は、リザーブインスタンス料金範囲のメリットを得られます。一括請求アカウントでは、一括請求アカウントに含まれる同一リージョン内のすべてのアクティブなリザーブインスタンスアカウントの定価が、1つの定価にまとめられます。一括請求アカウントのアクティブなリザーブインスタンスの定価総額が割引範囲レベルに達すると、以降 (その一括請求アカウントの定価総額が割引範囲の金額を超えている限り)、一括請求アカウント内のアカウントで購入されたリザーブインスタンスには割引が適用されます。

以下に、リザーブインスタンスの購入が一括請求でどのように扱われるかを示します。2つのアカウント A および B があり、これらが一括請求アカウントの一部であるとして、一括請求アカウントのアクティブなリザーブインスタンスはすべて1つのリージョン内にあります。アカウント A には 135,000 USD 相当のリザーブインスタンスが、アカウント B には 115,000 USD 相当のリザーブインスタンスが含まれます。アカウント A と B の一括請求の前払いコストは合計で 250,000 USD です。250,000 USD は割引料金のしきい値です。つまり、アカウント A または B のいずれか、または両方がリザーブインスタンスを追加購入すると、その新しい購入に対しては 5% の割引が適用されます。したがって、アカウント B が定価 15,000 USD でリザーブインスタンスを購入した場合、一括請求アカウントでは、新しいリザーブインスタンスにして 14,250 USD (15,000 USD から 5% 割引分の 750 USD を引いた 14,250 USD) のみが課金され、アカウント B の新しいリザーブインスタンスの使用料には割引が適用されます。

一括請求アカウントに適用されるリザーブインスタンスのメリットについては、「[リザーブインスタンスと一括請求 \(コンソリデेटィッドビルギング\)](#) (p. 232)」を参照してください。

## 割引範囲の価格での購入

リザーブドインスタンスを購入すると、割引範囲に該当する部分のリザーブドインスタンスに対応する割引があれば、Amazon EC2によって自動的に適用されます。特に何かを行う必要はなく、どのAmazon EC2 ツールを使用しても購入できます。

- AWS Management Console: [Amazon EC2 コンソール](#)の [Reserved Instances] ページの [Purchase Reserved Instances] ボタンをクリックします。
- AWS CLI の場合: `aws ec2 purchase-reserved-instances-offering` コマンドを使用します。
- Amazon EC2 CLI の場合: `ec2-purchase-reserved-instances-offering` コマンドを使用します。
- Amazon EC2 API の場合: `PurchaseReservedInstancesOffering` アクションを呼び出します。

お客様の購入が割引料金範囲に到達した場合、コンソール、`aws ec2 describe-reserved-instances` コマンド、`ec2-describe-reserved-instances` コマンド、または `DescribeReservedInstances` アクションでは、その購入が複数のエントリとして表示されます。表示された購入のうち、通常のリザーブドインスタンス料金が課金される部分のエントリと、それぞれの割引料金が課金される部分のエントリが表示されます。

この結果、購入 CLI コマンドまたは API アクションによって返されるリザーブドインスタンス ID は、新しいリザーブドインスタンスの実際の ID とは異なるものになります。詳細については、[リザーブドインスタンス ID \(p. 231\)](#) を参照してください。

例えば、あるリージョン内に表示価格総額 245,000 USD のリザーブドインスタンスがあるとします。そして、さらに定価総額 10,000 USD のリザーブドインスタンスを購入します。5% 割引範囲のしきい値は 250,000 USD です。したがって、追加購入により割引が適用されることとなります。取引が完了すると、コンソール、CLI、または API によって表示される一覧には 2 つの予約が示されます。1 つは割引が適用されない 5,000 USD に対する予約で、もう 1 つは割引料金 4,750 USD に対する予約です。割引料金は 5,000 USD の 5% で、250 USD になります。

アクティブなリザーブドインスタンスの合計定価額が割引料金範囲に達した後のご購入には、割引料金が適用されます。ただし、リザーブドインスタンスの一部の期限が切れて、アクティブなリザーブドインスタンスの定価総額が割引料金範囲レベル未満になった場合、次回のご購入には、リザーブドインスタンスの小売価格が適用されます。前の例を使って説明します。現在、定価総額 254,750 USD のリザーブドインスタンスがあるとします。250,000 USD は割引なしの料金で、4,750 USD は割引料金です。2 か月後、リザーブドインスタンスの 20,000 USD 分の有効期限が切れて、総額が 234,750 USD になり、5% の割引範囲を下回りました。それ以降に 15,000 USD のリザーブドインスタンスを購入する場合は、小売価格が適用されます。ただし、以前割引料金で購入した 4,500 USD のリザーブドインスタンスは、引き続き割引料金で課金されます。

重要なのは、定価は、リザーブドインスタンス購入時の割引のない料金で、コンソール、AWS CLI、Amazon EC2 CLI、Amazon EC2 API の **固定価格** の値は、購入したすべてのリザーブドインスタンスに対する **支払価格** であるという点です。割引範囲は、**表示価格** に基づいています。

リザーブドインスタンスを購入すると、次の 4 つのいずれかの状況になります。

- 1 つのリージョンでのリザーブドインスタンスの購入が、まだ割引しきい値より下である。これは割引を受けないことを意味します。
- 1 つのリージョンでのリザーブドインスタンスの購入により、最初の割引範囲のしきい値を超える。この場合、購入が処理されている間、新たに購入したリザーブドインスタンスは pending 状態になります。この購入では、割引のない料金と割引料金の 2 つの異なる料金でリザーブドインスタンスが購入されます。重要なのは、`purchase` コマンドを使用したときに CLI または API から返されるリザーブドインスタンス ID は、購入の完了時に実際に作成される新しいリザーブドインスタンス ID とは異なるということです。これらの ID は、`describe` コマンドを使用したときに返されます。この相違の説明については、次のセクション [リザーブドインスタンス ID \(p. 231\)](#) を参照してください。
- リージョン内のリザーブドインスタンスの購入全体が 1 つの割引範囲に完全に含まれる。この場合、購入が処理されている間、新たに購入したリザーブドインスタンスは pending 状態になります。リ

ザーブドインスタンスは該当する割引レベルで購入されます。重要なのは、前のシナリオと同じように、返されるリザーブドインスタンス ID は、購入の完了時に実際に作成される新しいリザーブドインスタンス ID とは異なるということです。これらの ID は、describe コマンドを使用したときに返されます。この相違の説明については、次のセクション [リザーブドインスタンス ID \(p. 231\)](#) を参照してください。

- 1つのリージョンでのリザーブドインスタンスの購入が、下の割引範囲から上の割引範囲まで及ぶ。前のシナリオと同じように、購入が処理されている間、新たに購入したリザーブドインスタンスは pending 状態になります。この購入では、1 番目の低い方の割引範囲の料金と高い方の割引範囲の料金の2つの異なる料金でリザーブドインスタンスが購入されます。重要なのは、返されるリザーブドインスタンス ID は、購入の完了時に実際に作成される新しいリザーブドインスタンス ID とは異なるということです。この相違の説明については、次のセクション「[リザーブドインスタンス ID \(p. 231\)](#)」を参照してください。

## リザーブドインスタンス ID

購入の総額が1つまたは複数のリザーブドインスタンス割引料金範囲と重なる場合、購入コマンドによって返されるリザーブドインスタンス ID は、購入後に呼び出す describe コマンドによって返されるリザーブドインスタンス ID とは異なる可能性があります。リザーブドインスタンスサービスによって複数のリザーブドインスタンス ID が生成されます。これは、割引が適用されない範囲と最初の割引範囲、または複数の割引範囲に購入がまたがるためです。リザーブドインスタンスのうち1つの範囲に含まれる部分ごとに1つのリザーブドインスタンス ID が作成されます。

AWS CLI を使用する場合、aws ec2 purchase-reserved-instances-offering によって返されたリザーブドインスタンス ID `1ba8e2e3-edf1-43c3-b587-7742example` を認識しない aws ec2 describe-reserved-instances コマンドを実行すると、次のように出力されます。

```
{
  "ReservedInstances": []
}
```

この例では、購入コマンドによって生成されるリザーブドインスタンス ID は、購入の処理中に使用される暫定的な ID です。購入は非割引範囲（範囲 0）から1番目の割引範囲（範囲 1）に及んでいるので、リザーブドインスタンスサービスは実際には複数のリザーブドインスタンス ID（暫定的なリザーブドインスタンス ID、割引なしの定価で購入したリザーブドインスタンスのリザーブドインスタンス ID、5% 割引料金で購入したリザーブドインスタンスのリザーブドインスタンス ID）を生成します。

## リザーブドインスタンスの料金メリットについて

リザーブドインスタンスを購入すると、2つのメリットを得られます。将来起動する EC2 インスタンスの処理能力を予約できること、および時間当たりの料金の割引です。リザーブドインスタンスの起動および操作に関するその他のタスクについては、オンデマンド EC2 インスタンスと変わりありません。

このセクションでは、Amazon EC2 におけるリザーブドインスタンスの料金面でのメリットについて説明します。

### リザーブドインスタンスの料金メリットの適用

リザーブドインスタンスでは、使用可能な Amazon EC2 インスタンスの処理能力の予約に対して、購入の際に指定した仕様（製品プラットフォーム、インスタンスタイプ、アベイラビリティゾーンなど）に基づいて前払い料金を支払います。リザーブドインスタンスの購入後に、変更の要件を満たしており容量がある場合は、リザーブドインスタンスを変更できます。リザーブドインスタンスの変更の詳細については、「[リザーブドインスタンスの変更 \(p. 251\)](#)」を参照してください。（未使用のリザーブドインスタンスは、リザーブドインスタンスマーケットプレイスで販売することもできます。）

処理能力を予約できるほか、リザーブインスタンスを購入したアカウントに関連付けられているオンデマンド EC2 インスタンスの時間当たりの使用料金が割引になります。割引が適用されるには、オンデマンドインスタンスが、リザーブインスタンスの仕様と一致していなければなりません。

例えば、ユーザー A が次のオンデマンド EC2 インスタンスを実行しているとします。

- us-east-1a アベイラビリティゾーン内の 4 つの m1.small インスタンス
- us-east-1b アベイラビリティゾーン内の 4 つの c1.medium インスタンス
- us-east-1b アベイラビリティゾーン内の 2 つの c1.xlarge インスタンス

ユーザー A が次の 6 つのリザーブインスタンスを購入します。

- us-east-1a アベイラビリティゾーン内の 2 つの m1.small インスタンス
- us-east-1a アベイラビリティゾーン内の 3 つの c1.medium インスタンス
- us-east-1b アベイラビリティゾーン内の 1 つの c1.xlarge インスタンス

リザーブインスタンスを購入するときに、予約した処理能力に対する前払い料金を支払うことにより、ユーザー A は、指定した仕様のインスタンスを必要に応じて 6 個起動できます。さらに、毎月 6 個分のインスタンスの時間当たりの使用料金が割引になります。リザーブインスタンスの購入時に既にインスタンスを実行中だったため、実行中のオンデマンドインスタンスのうち、購入したリザーブインスタンスの仕様と合致するインスタンスに、自動的に時間当たりの割引料金が適用されます。

具体的には次のようになります。

- ユーザー A が購入した 2 個の m1.small リザーブインスタンスの割引使用料が、us-east-1a アベイラビリティゾーン内で実行中の 4 個の m1.small Amazon EC2 インスタンスのうち 2 個に適用されます。

us-east-1a アベイラビリティゾーン内の残り 2 個の EC2 インスタンスは、現在のオンデマンド料金で課金されます。

- ユーザー A が購入した 3 個の c1.medium リザーブインスタンスによる割引料金は適用されません。これらの c1.medium リザーブインスタンスは、現在 c1.medium Amazon EC2 インスタンスが実行されているアベイラビリティゾーンとは異なるゾーンで実行するように指定されているためです。

実行中の 4 個の c1.medium Amazon EC2 インスタンスは、現在のオンデマンド料金で課金されます。

ユーザー A が c1.medium EC2 インスタンスを us-east-1a アベイラビリティゾーンで起動すると、そのインスタンスにはリザーブインスタンスの割引料金が適用されます。

- ユーザー A が購入した 1 個の c1.xlarge リザーブインスタンスの割引使用料が、us-east-1b アベイラビリティゾーン内で実行中の 2 個の c1.xlarge Amazon EC2 インスタンスのうち 1 個に適用されます。

us-east-1b アベイラビリティゾーン内のもう 1 個の c1.xlarge EC2 インスタンスは、現在のオンデマンド料金で課金されます。

このシナリオでは、6 個のリザーブインスタンスを購入することで、既に実行されている 2 個の m1.small と 1 個の c1.xlarge のオンデマンド EC2 インスタンスに対する時間当たりのコストを節約しています。同時に、6 個のリザーブインスタンスを必要に応じて実行するための処理能力を確保しています。

## リザーブインスタンスと一括請求 ( コンソリデーティッドビルング )

購入アカウントが、1 つの一括請求 ( CB ) の支払いアカウントに請求される一連のアカウントの一部である場合、リザーブインスタンスの料金面でのメリットを広範囲に利用できます。一括請求を利用

すると、お客様のすべての請求を1つのアカウント、すなわちCB支払いアカウントを使って支払うことができます。また、すべてのサブアカウントの毎月の時間当たりの使用料がCB支払いアカウントにまとめられます。この請求は、さまざまな役割を持つチームやグループがある企業にとっては特に便利です。一括請求の詳細については、「[AWS Account Billing について](#)」の「[一括請求を参照してください](#)」。

リザーブインスタンスでは、リンクされているすべてのアカウントの使用料が毎月の利用時間に基づいてCB支払いアカウントに合算されます。したがって、請求書の計算には通常のリザーブインスタンスのロジックが適用されます。

例えば、アカウントが一括請求アカウントに含まれ、リザーブインスタンスの購入にお客様のアカウントを使用しているとします。リザーブインスタンスの前払い料金はCB支払アカウントから支払われ、割引はサブアカウント間で分配されます。総コストの分配は、サブアカウントそれぞれの使用量を、CBアカウント全体の使用量で割った比率を使って計算されます。しかし、処理能力の予約は、予約を行ったサブアカウント、この例ではお客様のアカウントに残ります。また、処理能力の予約は、購入時に指定した製品プラットフォーム、インスタンスタイプ、および利用可能ゾーンに対してのみ適用されます。

一括請求アカウントに適用されるリザーブインスタンス料金範囲の割引の詳細については、「[リザーブインスタンス料金範囲を理解する \(p. 226\)](#)」を参照してください。

## リザーブインスタンスマーケットプレイス

リザーブインスタンスマーケットプレイスは、AWS のお客様が購入した未使用の Amazon Elastic Compute Cloud ( Amazon EC2 ) リザーブインスタンスを他の企業や組織に売却できる、オンラインのマーケットプレイスです。お客様がリザーブインスタンスマーケットプレイスで購入する場合は、他の AWS ユーザー ( サードパーティとして出品 ) が販売している、さまざまな期間や価格設定のリザーブインスタンスから選ぶことができます。リザーブインスタンスマーケットプレイスでは、ニーズの変化に合わせて、余ったリザーブインスタンスを柔軟に販売することができます。例えば、インスタンスを新しい AWS リージョンに移動する、新しいインスタンスタイプに変更する、期限が切れる前に終了するプロジェクトの処理能力を売却する、などが考えられます。EC2 インスタンスをリザーブインスタンスマーケットプレイスで購入した場合も、容量の予約については、AWS で直接購入したリザーブインスタンスと同様です。



### Note

販売者になるための要件、予約した処理能力を販売できる時期など、いくつかの制限があります。リザーブインスタンスおよびリザーブインスタンスマーケットプレイスの制限と要件については、「[リザーブインスタンスの要件チェックリスト \(p. 288\)](#)」を参照してください。

購入者から見ると、このようなリザーブインスタンスとアマゾン ウェブ サービス ( AWS ) から直接購入したリザーブインスタンスの間にはいくつかの違いがあります。

- 期間。リザーブインスタンスマーケットプレイス でサードパーティ販売者から購入したリザーブインスタンスは、残り期間が完全な標準期間よりも短くなっています。AWS から購入できるリザーブインスタンスの完全な標準期間は、1年間または3年間です。
- 前払い価格。サードパーティのリザーブインスタンスは、さまざまな前払い価格で販売されます。使用料金または定期的に支払う料金は、リザーブインスタンスを最初にAWSから購入したときに設定された料金と同じ金額です。
- 範囲による割引。割引範囲のしきい値に到達したために割引価格で購入された Amazon EC2 リザーブインスタンスは、リザーブインスタンスマーケットプレイスで販売することはできません。リザーブインスタンスマーケットプレイスの詳細については、「[リザーブインスタンス料金範囲を理解する \(p. 226\)](#)」を参照してください。

販売者は、リザーブインスタンスの一部を出品するか、すべて出品するかを選択できます。また、受け取る前払い価格を選択できます。リザーブインスタンスがリザーブインスタンスマーケットプレイスに出品され、購入が可能になります。リザーブインスタンスマーケットプレイスで販売する各リザーブインスタンスに対しては、前払い価格の総額の12%がサービス料として課金されます。出品したリザーブインスタンスは、売れるまで使用することができます。販売が成立すると、予約済みの処理能力とそれに伴う割引料金は失われます。予約した処理能力が売れた後も、インスタンスを使用し続けることはできます。ただし、そのインスタンスに関して予約した処理能力が売れた時点からは、インスタンスのオンデマンド料金を支払う必要があります。



#### Note

リザーブインスタンスマーケットプレイスで販売できるのは、Amazon EC2 リザーブインスタンスのみです。Amazon Relational Database Service ( Amazon RDS ) や Amazon ElastiCache リザーブインスタンスなどの他の AWS リザーブインスタンスは、リザーブインスタンスマーケットプレイスでは販売できません。

リザーブインスタンスの詳細については、「[リザーブインスタンス \(p.218\)](#)」を参照してください。

## 購入者の概要

リザーブインスタンスマーケットプレイスは、AWS から提供されているものとは異なる期間のリザーブインスタンスを購入しようとしている場合に便利です。マーケットプレイスでは、特別なビジネスニーズに対応する設定のリザーブインスタンスを検索することもできます。

クイックスタート: リザーブインスタンスマーケットプレイスでの購入に関するビデオ

次の動画は、AWS マネジメントコンソールを使用したリザーブインスタンスマーケットプレイスでのリザーブインスタンスの購入方法についての説明です。[Getting Started Buying Reserved Instances in the リザーブインスタンスマーケットプレイス](#)

### 要件

リザーブインスタンスマーケットプレイスでリザーブインスタンスを購入するには、有効なアマゾンウェブサービス ( AWS ) アカウントが必要です。AWS アカウントの設定を行うことについては、「[Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#)」を参照してください。

Amazon EC2 リザーブインスタンスを以前に購入したことがあるユーザーにとって、リザーブインスタンスマーケットプレイスでリザーブインスタンスを購入する手順とツールは非常にわかりやすいものになっています。

### リザーブインスタンスを購入するためのステップ

リザーブインスタンスを購入するためのステップは、標準の AWS リザーブインスタンスでもリザーブインスタンスマーケットプレイスのインスタンスでも同じです。

1. 購入するリザーブインスタンスの詳細を指定します。
2. リザーブインスタンスマーケットプレイスに示されている一覧から、指定に基づいて目的のリザーブインスタンスを選択します。
3. 選択と購入の内容を確認します。

詳細については、「[リザーブインスタンスを購入する \(p. 238\)](#)」を参照してください。

## 販売者の概要

リザーブインスタンスマーケットプレイスは、リザーブインスタンスを所有しており、予約した処理能力の残りの期間を販売したい場合や、ビジネスにおいて、現在所有しているものと異なる設定のリザーブインスタンスを探している場合に便利です。マーケットプレイスでは、短期間のワークロー

ドを必要とし、標準の 1 年および 3 年とは異なるリザーブインスタンスを購入しようとしているビジネスに対して、自分のインスタンスを販売することができます。

リザーブインスタンスマーケットプレイスに出品することで、ビジネスニーズの変化に応じてリザーブインスタンスの設定を柔軟に変更できます。例えば、欧州 (アイルランド) リージョンで 3 年の m1.xlarge リザーブインスタンスを現在所有しているとします。今年は、顧客ベースがアジアにも拡大したため、アジアパシフィック (東京) リージョンで使用できる m1.large リザーブインスタンスが必要です。この場合、リザーブインスタンスマーケットプレイスを使用して、欧州 (アイルランド) リージョンで購入した m1.xlarge リザーブインスタンスの一部の残り期間を売却し、アジアパシフィック (東京) リージョンで処理能力を購入することができます。



#### Note

マーケットプレイスで販売するリザーブインスタンスごとに、前払い価格の総額の 12% がサービス料として AWS から課金されます。

クイックスタート: リザーブインスタンスマーケットプレイス での販売に関するビデオ

次の動画は、AWS マネジメントコンソールを使用したリザーブインスタンスマーケットプレイスでのリザーブインスタンスの販売方法についての説明です。この動画には、販売者としての登録手順と、インスタンスの出品手順が含まれています。[Getting Started Selling Reserved Instances in the リザーブインスタンスマーケットプレイス](#)

#### 要件

- 販売者として登録します。米国の法人与米国外の法人のいずれも、最初に販売者として登録することによってリザーブインスタンスマーケットプレイスで販売することができます。詳細については、「[販売者として登録する \(p. 262\)](#)」を参照してください。
- 税登録を行います。リザーブインスタンスの取引数が 200 件以上ある販売者や 20,000 USD 以上の出品を予定している販売者は、税務上の理由から、ビジネスに関する追加情報を提供する必要があります。詳細については、「[税金情報 \(p. 264\)](#)」を参照してください。
- 米国の銀行を指定します。AWS では、リザーブインスタンスの販売時に集金された金額をお支払いするために、お客様の銀行情報が必要です。住所が米国内の銀行でなければなりません。詳細については、「[銀行 \(p. 263\)](#)」を参照してください。

#### リザーブインスタンスを販売するためのステップ

販売者として登録し、すべての必須情報を指定すると、リザーブインスタンスマーケットプレイスでリザーブインスタンスを販売できるようになります。

1. 販売するリザーブインスタンスを選択します。
2. リザーブインスタンスを販売する希望価格を選択します。
3. リザーブインスタンスの一覧を表示します。

詳細については、「[リザーブインスタンスマーケットプレイスでの販売 \(p. 261\)](#)」を参照してください。

#### 次は何をしますか。

- 詳細:
  - [リザーブインスタンスの使用を開始する \(p. 219\)](#)
  - [リザーブインスタンスの料金メリットについて \(p. 231\)](#)
  - [リザーブインスタンス料金範囲を理解する \(p. 226\)](#)
- 開始:
  - [リザーブインスタンスに関する情報を取得する \(p. 246\)](#)

- [リザーブドインスタンスを購入する \(p. 238\)](#)
- [販売者として登録する \(p. 262\)](#)
- [リザーブドインスタンスの出品 \(p. 267\)](#)

## リザーブドインスタンスを購入する

アマゾン ウェブ サービスから 1 年または 3 年の期間の Amazon EC2 リザーブドインスタンスを、または不要になった EC2 リザーブドインスタンスを保有しているサードパーティの販売業者から EC2 リザーブドインスタンスを、それぞれ購入できます。サードパーティから購入したり リザーブドインスタンスマーケットプレイスで販売されるリザーブドインスタンスも、AWS から購入したり リザーブドインスタンスと同じように機能し、購入プロセスも同じです。ただ、サードパーティから購入されるリザーブドインスタンスは、残余期間が完全な長さではなく、異なる前払い価格で販売することができる点が異なります。

購入者にとっては、リザーブドインスタンスマーケットプレイスにより、必要なインスタンスタイプ、リージョン、期間の組み合わせに最も一致するリザーブドインスタンスを検索できることで選択肢が増え、柔軟な選択が可能になります。

リザーブドインスタンスをいったん購入すると、リザーブドインスタンスマーケットプレイスでサードパーティから購入したか、AWS から購入したかに関わらず、購入をキャンセルすることはできません。ただし、ニーズが変化したときは、リザーブドインスタンスを変更および売却できます。リザーブドインスタンスの変更の詳細については、[リザーブドインスタンスの変更 \(p. 251\)](#) を参照してください。リザーブドインスタンスの売却については、[リザーブドインスタンスマーケットプレイスでの販売 \(p. 261\)](#) を参照してください。

この購入者を対象としたガイドのセクションを次に示します。

- 「[購入の仕組み \(p. 236\)](#)」 – リザーブドインスタンスマーケットプレイスで購入を開始するために必要なことの概要を示します。
- 「[購入者になる \(p. 237\)](#)」 – リザーブドインスタンスマーケットプレイスで購入者になるために行う必要があることと、特定の情報が必要な理由について説明します。
- 「[リザーブドインスタンスを購入する \(p. 238\)](#)」 – 購入プロセスについて説明します。このプロセスには、リザーブドインスタンスマーケットプレイスを使用する場合に繰り返す可能性があるタスクが含まれます。
- 「[明細書 \(請求書\) を読む \(p. 245\)](#)」 – 請求書の意味を説明します。

リザーブドインスタンスマーケットプレイスに関する一般的な情報については、「[リザーブドインスタンスマーケットプレイス \(p. 233\)](#)」を参照してください。リザーブドインスタンスの売却については、「[リザーブドインスタンスマーケットプレイスでの販売 \(p. 261\)](#)」を参照してください。リザーブドインスタンスの基本的な情報については、「[リザーブドインスタンス \(p. 218\)](#)」を参照してください。

製品の料金については、次のページを参照してください。

- [Amazon EC2 Reserved Instance Pricing](#)
- [Amazon EC2 On-Demand Instances Pricing](#)
- [AWS サービス料金の概要](#)

## 購入の仕組み

リザーブドインスタンスマーケットプレイスの使用を開始するには、AWS のアカウントを作成する必要があります。リザーブドインスタンスマーケットプレイスで購入を行うには、購入するリザーブドインスタンスの詳細を AWS マネジメントコンソールの検索ウィザードで指定します。すると、リクエストと一致する選択肢が提示されます。選択したリザーブドインスタンスは、出品されている状態のまま購入しなければなりません。つまり、期間やインスタンスタイプなどを変更することはできません。

購入するリザーブドインスタンスを選択すると、AWS から選択内容にかかる全費用の見積もりが提供されます。購入を進める場合は、AWS により自動的に、購入価格に**限度価格**が設けられるため、リザーブドインスタンスの全費用が提示された見積額を超えることはありません。何らかの理由で価格が上がった場合は、AWS により自動的に前の画面が表示され、価格が変更されたために購入が完了できなかったことが通知されます。また、選択した内容と同じような内容で価格が安い出品がある場合は、高い方の選択肢ではなく、安い価格の出品が販売されます。

リザーブドインスタンスの料金範囲の割引は、AWS での購入にのみ適用されます。この種の割引は、サードパーティのリザーブドインスタンスの購入には適用されません。例えば、リザーブドインスタンスマーケットプレイスでサードパーティから 250,000 USD 相当のリザーブドインスタンスを購入し、その後で別のサードパーティのリザーブドインスタンスのセットを購入して、合計額が最初の料金範囲のしきい値を超えた場合、この範囲に対する 10% の割引は受けられません。しかし、Marketplace からの購入が価格のしきい値を超え、その後で AWS からリザーブドインスタンスを購入すると、その AWS からの購入には割引が適用されます。詳細については、「[リザーブドインスタンス料金範囲を理解する \(p. 226\)](#)」を参照してください。

リザーブドインスタンスの取引が完了したかどうかは、Amazon EC2 コンソールの [Reserved Instances] ページか、`ec2-describe-reserved-instances` コマンドまたは `DescribeReservedInstances` アクションの結果から判断できます。問題なく購入された場合、リザーブドインスタンスは `[pending-payment]` 状態から `[active]` 状態に移行します。

リザーブドインスタンスを購入し、支払いが失敗すると、コンソール、`ec2-describe-reserved-instances` コマンド、および `DescribeReservedInstances` アクションでは、購入しようとしたリザーブドインスタンスが `payment-pending` 状態から変化して `payment-failed` 状態になることを示して、この失敗したトランザクションが表示されます。

## 購入者になる

購入者になることは、単純で簡単です。アマゾン ウェブ サービス (AWS) アカウントを既にお持ちの場合は、すぐに購入を開始できます。詳細については、次のセクションを参照してください。

- [購入者が公開する情報を理解する \(p. 237\)](#)
- [リザーブドインスタンスを購入する \(p. 238\)](#)

(AWS) アカウントがない場合は、まずサインアップし、AWS でアカウントを作成する必要があります。詳細については、「[Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#)」を参照してください。リザーブドインスタンスのをご存じない場合は、「[リザーブドインスタンスの使用を開始する \(p. 219\)](#)」を参照してください。

### 購入者が公開する情報を理解する

購入者に関するいくつかの基本情報は、販売者に伝えられます。購入者の郵便番号と国の情報が、支払いレポートで販売者に提供されます。この情報を使用して、販売者は、国に支払う必要な取引税 (売上税や付加価値税など) を計算できます。さらに、AWS が販売者に E メールアドレスを提供する必要がある場合があります。これは、販売者が、販売に関する質問があり、それに関して連絡できるようにするためです (例えば税務上の質問など)。

同様の理由で、AWS は購入者の請求書に販売者の正式名を記載します。さらに、税金または税に関する理由から販売者の情報が必要な場合は、AWS カスタマーサービスに問い合わせることも可能です。

### 次のステップ

AWS でサインアップすると、リザーブドインスタンスマーケットプレイスでリザーブドインスタンスを購入できるようになります。

- 固有のビジネスニーズに対応するリザーブドインスタンスを検索するには、「[リザーブドインスタンスを購入する \(p. 238\)](#)」を参照してください。

- 請求書については、「[明細書 \( 請求書 \) を読む \( p. 245 \)](#)」を参照してください。
- 未使用のリザーブドインスタンスを販売する場合は、「[リザーブドインスタンスマーケットプレイスでの販売 \( p. 261 \)](#)」を参照してください。

リザーブドインスタンスの詳細については、「[リザーブドインスタンス \( p. 218 \)](#)」を参照してください。

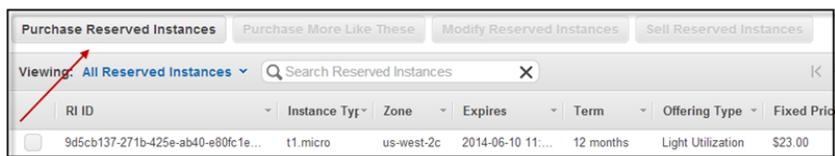
## リザーブドインスタンスを購入する

Amazon EC2 リザーブドインスタンスをサードパーティから リザーブドインスタンスマーケットプレイスで購入する手順は、基本的にアマゾンウェブサービス ( AWS ) からのリザーブドインスタンスの購入手順と同じです。AWS マネジメントコンソール、Amazon EC2 コマンドラインインターフェイス ( CLI ) ツール、または Amazon EC2 API を使用して、リザーブドインスタンスマーケットプレイスでリザーブドインスタンスを購入できます。

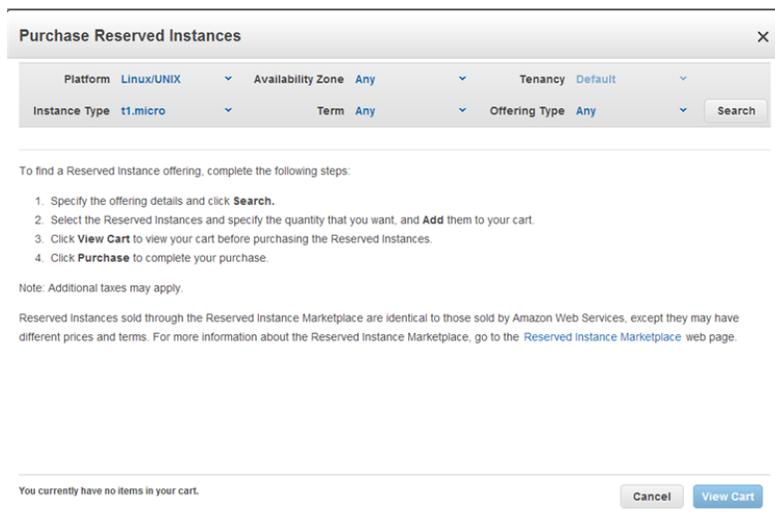
### AWS マネジメントコンソール

リザーブドインスタンスを検索および購入するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. [Navigation] ペインで [Reserved Instances] をクリックします。
3. [Reserved Instances] ページで、[Purchase Reserved Instances] をクリックします。



4. [Purchase Reserved Instances] ページで、購入するリザーブドインスタンスを指定 ( [Platform]、[Instance Type]、[Availability Zone]、[Term]、[Tenancy] などのわかりやすいフィルタを使用 ) して、[Search] をクリックします。



[Purchase Reserved Instances] ウィザードにより、検索条件を満たすリザーブドインスタンスの一覧が表示されます。

5. 目的のリザーブドインスタンスを選択し、購入する量を入力して、[Add to Cart] をクリックします。引き続き別のリザーブドインスタンスを選択してカートに追加することができます。

[Seller] 列には、販売者が [3rd Party] 販売者が [AWS] かが表示されます。[Term] 列には、販売者がサードパーティ販売者である場合、標準以外の期間が表示されます。[Purchase Reserved Instances] ウィザードのページの下には、常にカート内の現時点での合計が表示されます。

Seller	Term	Effective Rate	Upfront Price	Hourly Rate	Availability Zone	Offering Type	Quantity Available	Desired Quantity
AWS	12 months	\$0.082	\$122.00	\$0.068	us-west-2c	Light Utilization	Unlimited	1
AWS	12 months	\$0.082	\$122.00	\$0.068	us-west-2a	Light Utilization	Unlimited	1
AWS	12 months	\$0.082	\$122.00	\$0.068	us-west-2b	Light Utilization	Unlimited	1
3rd Party	7 months	\$0.064	\$180.00	\$0.028	us-west-2a	Heavy Utilization	1	1
AWS	36 months	\$0.061	\$192.00	\$0.054	us-west-2c	Light Utilization	Unlimited	1
AWS	36 months	\$0.061	\$192.00	\$0.054	us-west-2a	Light Utilization	Unlimited	1
AWS	36 months	\$0.061	\$192.00	\$0.054	us-west-2b	Light Utilization	Unlimited	1
AWS	12 months	\$0.074	\$277.00	\$0.042	us-west-2c	Medium Utilization	Unlimited	1
AWS	12 months	\$0.074	\$277.00	\$0.042	us-west-2a	Medium Utilization	Unlimited	1
AWS	12 months	\$0.074	\$277.00	\$0.042	us-west-2b	Medium Utilization	Unlimited	1
3rd Party	23 months	\$0.057	\$319.24	\$0.038	us-west-2a	Medium Utilization	1	1

Your cart: 2 Reserved Instances, Total Due Now: **\$314.00**  
Additional taxes may apply.

6. [View Cart] をクリックして、選択したリザーブドインスタンスの要約を確認します。

リザーブドインスタンスをさらにカートに追加する場合は、[Add More To Cart] をクリックします。カートからアイテムを削除するには、[Delete] をクリックします。

または、異なるセットのリザーブドインスタンスをやり直したり、検索する場合は、[Cancel] をクリックします。

Items to buy now	Upfront Price	Hourly Price	Quantity
Light Utilization Linux/UNIX m1.medium in us-west-2c for 12 months Payment Terms: Upfront + Every Hour Sold By: Amazon <a href="#">Delete</a>	\$122.00	\$0.068	1
Light Utilization Linux/UNIX m1.medium in us-west-2c for 36 months Payment Terms: Upfront + Every Hour Sold By: Amazon <a href="#">Delete</a>	\$192.00	\$0.054	1

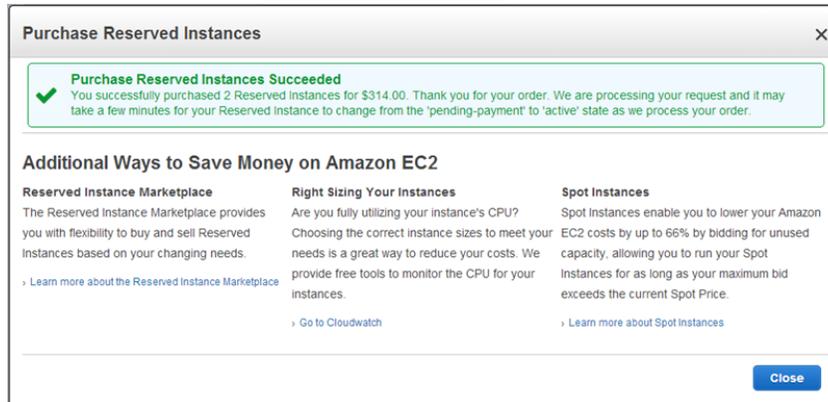
Your cart: 2 Reserved Instances, Total Due Now: **\$314.00**  
Additional taxes may apply.

7. 購入するリザーブドインスタンスをすべて追加して清算する場合は、[Purchase] をクリックします。



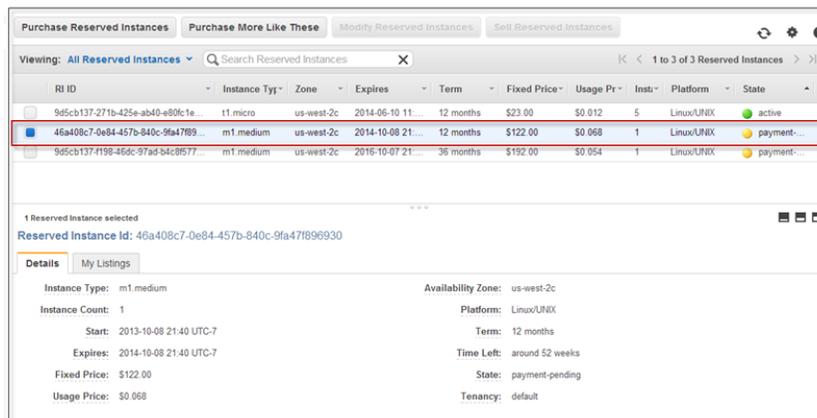
## Note

購入時に、選択した内容と同じような内容で価格が安い製品があった場合、AWS はその製品をその安い価格で販売します。



購入が完了します。

- 注文を確認するには、EC2 コンソールの [Reserved Instances] ページに移動します。



[Reserved Instances] ページに、お客様のアカウントに属するリザーブドインスタンスの一覧が、購入したばかりの新しいリザーブドインスタンスも含めて表示されます。

リザーブドインスタンスは、購入が完了するといつでも使用できます。つまり、リザーブドインスタンスの [State] が [payment-pending] から [Active] に変わったことになります。リザーブドインスタンスを使用するには、オンデマンドインスタンスと同じ方法でインスタンスを起動します。必ずリザーブドインスタンスに対して指定したものと同一基準を指定するようにしてください。AWS では自動的に低い時間料金が課金されます。インスタンスを再起動する必要はありません。

## Amazon EC2 CLI

リザーブドインスタンスを検索および購入するには

- ec2-describe-reserved-instances-offerings を使用して、仕様に一致するリザーブドインスタンスの出品の一覧を取得します。この例では、sa-east-1b アベイラビリティゾーン内で使用可能な m1.small、Linux/UNIX リザーブドインスタンスを確認します。

```
PROMPT> ec2-describe-reserved-instances-offerings -t m1.small -z sa-east-1b  
-d Linux/UNIX --headers
```

Amazon EC2 が次の例のような出力を返します。

```
PROMPT> ec2-describe-reserved-instances-offerings
Type Source ReservedInstancesOfferingId AvailabilityZone InstanceType Duration
  FixedPrice UsagePrice ProductDescription Currency InstanceTenancy Offering
Type
OFFERING AWS 4b2293b4-3236-49f5-978d-a74c3example sa-east-1b m1.small 3y
574.0 0.0 Linux/UNIX USD default Heavy Utilization
Type Frequency Amount
RECURRING-CHARGE Hourly 0.021
OFFERING AWS 3a98bf7d-07e1-4b33-8e11-e5314example sa-east-1b m1.small 3y
473.0 0.031 Linux/UNIX USD default Medium Utilization
OFFERING AWS 438012d3-5fc5-4e49-a88e-273edexample sa-east-1b m1.small 3y
203.0 0.055 Linux/UNIX USD default Light Utilization
OFFERING AWS d586503b-bb92-41fa-9065-e5b90example sa-east-1b m1.small 1y
372.94 0.0 Linux/UNIX USD default Heavy Utilization
Type Frequency Amount
RECURRING-CHARGE Hourly 0.03
OFFERING AWS ceb6a579-b235-41e2-9aad-15a23example sa-east-1b m1.small 1y
307.13 0.04 Linux/UNIX USD default Medium Utilization
OFFERING AWS 649fd0c8-4ffb-443d-824d-eae3fexample sa-east-1b m1.small 1y
131.63 0.07 Linux/UNIX USD default Light Utilization
OFFERING 3rd Party b6121943-9faf-4350-8047-bc6d4example sa-east-1b m1.small
10m - 0.032 Linux/UNIX USD default Medium Utilization
Type Count Price
PRICING_DETAIL 2 $1.2
OFFERING 3rd Party 08edcff2-8143-4c1d-b23c-e4c11example sa-east-1b m1.small
5m - 0.032 Linux/UNIX USD default Medium Utilization
Type Count Price
PRICING_DETAIL 19 $1.2
PRICING_DETAIL 4 $1.23
```

上の出力は、購入可能な出品全体の一部を示しています。



#### Tip

希望するタイプのリザーブインスタンス提供タイプだけを表示するように、一覧をフィルタリングできます。結果をフィルタリングする方法については、「[Amazon Elastic Compute Cloud コマンドラインリファレンス](#)」の「`ec2-describe-reserved-instances-offerings`」を参照してください。

2. 使用できるリザーブインスタンスの一覧から、要件を満たすリザーブインスタンスを購入します。リザーブインスタンスを購入するには、次のコマンドを使用します。

```
PROMPT> ec2-purchase-reserved-instances-offering --offering offering --in-  
stance-count count
```

次のような出力が Amazon EC2 から返されます。

```
PURCHASE af9f760e-c1c1-449b-8128-1342dexample 438012d3-80c7-42c6-9396-  
a209cexample
```

- 応答には、提供 ID と予約 ID が含まれます。
3. 今後の参照用に予約 ID を書き留め、保存します。
  4. 購入を確認します。

```
PROMPT> ec2-describe-reserved-instances
```

次のような出力が Amazon EC2 から返されます。

```
RESERVEDINSTANCE af9f760e-c1c1-449b-8128-1342dexample sa-east-1b  
m1.small ly 227.5 0.03 Linux/UNIX Active
```

リザーブドインスタンスは、購入の完了後いつでも実行できます。リザーブドインスタンスを実行するには、オンデマンドインスタンスと同じ方法で起動します。必ずリザーブドインスタンスに対して指定したものと同一基準を指定してください。AWS では自動的に低い時間料金が課金されます。

## Amazon EC2 API

リザーブドインスタンスを検索および購入するには

1. `DescribeReservedInstancesOfferings` を使用して、仕様に一致するリザーブドインスタンスの出品の一覧を取得します。この例では、Linux/UNIX が購入できるか、重度使用リザーブドインスタンスが購入可能かどうかを確認します。

```
https://ec2.amazonaws.com/?Action=DescribeReservedInstancesOfferings  
&MaxResults=50  
&ProductDescription=Linux%2FUNIX  
&OfferingType=Heavy+Utilization  
&AUTHPARAMS
```



### Note

Query API を使用している場合、“/” は “%2F” と示されます。

以下に、応答の例を示します。

```
<DescribeReservedInstancesOfferingsResponse xmlns='http://ec2.amazon  
aws.com/doc/2012-08-15/'>  
  <requestId>768e52ac-20f5-42b1-8559-e70e9example</requestId>  
  <reservedInstancesOfferingsSet>  
    <item>  
      <reservedInstancesOfferingId>d0280f9e-afcl-47f3-9899-  
c3a2cexample</reservedInstancesOfferingId>  
      <instanceType>m1.xlarge</instanceType>  
      <availabilityZone>us-east-1a</availabilityZone>  
      <duration>25920000</duration>  
      <fixedPrice>195.0</fixedPrice>  
      <usagePrice>0.0</usagePrice>  
      <productDescription>Linux/UNIX</productDescription>  
      <instanceTenancy>dedicated</instanceTenancy>  
      <currencyCode>USD</currencyCode>  
      <offeringType>Heavy Utilization</offeringType>  
      <recurringCharges>
```

```
<item>
  <frequency>Hourly</frequency>
  <amount>0.2</amount>
</item>
</recurringCharges>
<marketplace>true</marketplace>
<pricingDetailsSet>
  <item>
    <price>195.0</price>
    <count>1</count>
  </item>
  <item>
    <price>310.0</price>
    <count>1</count>
  </item>
  <item>
    <price>377.0</price>
    <count>1</count>
  </item>
  <item>
    <price>380.0</price>
    <count>1</count>
  </item>
</pricingDetailsSet>
</item>
<item>
  <reservedInstancesOfferingId>649fd0c8-7846-46b8-8f84-a6400example</reservedInstancesOfferingId>
  <instanceType>m1.large</instanceType>
  <availabilityZone>us-east-1a</availabilityZone>
  <duration>94608000</duration>
  <fixedPrice>1200.0</fixedPrice>
  <usagePrice>0.0</usagePrice>
  <productDescription>Linux/UNIX</productDescription>
  <instanceTenancy>default</instanceTenancy>
  <currencyCode>USD</currencyCode>
  <offeringType>Heavy Utilization</offeringType>
  <recurringCharges>
    <item>
      <frequency>Hourly</frequency>
      <amount>0.052</amount>
    </item>
  </recurringCharges>
  <marketplace>>false</marketplace>
  <pricingDetailsSet/>
</item>
</reservedInstancesOfferingsSet>
<nextToken>QUUVo/0S3X6nEBjsQZR/pRRlCPP/5Lrx79Wyxexample</nextToken>
</DescribeReservedInstancesOfferingsResponse>
```

2. 前述の例の購入可能なリザーブドインスタンスの一覧から、目的のリザーブドインスタンスを選択し、上限価格を指定します。

```
https://ec2.amazonaws.com/?Action=PurchaseReservedInstancesOffering
&ReservedInstancesOfferingId=d0280f9e-afcl-47f3-9899-c3a2cexample
&InstanceCount=1
```

```
&LimitPrice.Amount=200  
&AUTHPARAMS
```

以下に、応答の例を示します。

```
<PurchaseReservedInstancesOfferingResponse xmlns="http://ec2.amazon  
aws.com/doc/2012-08-15/" >  
  <requestId>59dbff89-35bd-4eac-99ed-be587example</requestId>  
  <reservedInstancesId>e5a2ff3b-7d14-494f-90af-0b5d0example</reservedIn  
stancesId>  
</PurchaseReservedInstancesOfferingResponse>
```

### 3. 購入を検証するには、新しいリザーブドインスタンスを確認します。

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstances  
&AUTHPARAMS
```

以下に、応答の例を示します。

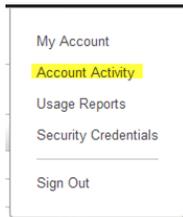
```
<DescribeReservedInstancesResponse xmlns='http://ec2.amazonaws.com/doc/2012-  
08-15/' >  
  <requestId>ebe3410a-8f37-441d-ae11-2e78eexample</requestId>  
  <reservedInstancesSet>  
    <item>  
      <reservedInstancesId>e5a2ff3b-7d14-494f-90af-0b5d0example</re  
servedInstancesId>  
      <instanceType>m1.xlarge</instanceType>  
      <availabilityZone>us-east-1a</availabilityZone>  
      <start>2012-08-23T15:19:31.071Z</start>  
      <duration>25920000</duration>  
      <fixedPrice>195.0</fixedPrice>  
      <usagePrice>0.0</usagePrice>  
      <instanceCount>1</instanceCount>  
      <productDescription>Linux/UNIX</productDescription>  
      <state>active</state>  
      <instanceTenancy>dedicated</instanceTenancy>  
      <currencyCode>USD</currencyCode>  
      <offeringType>Heavy Utilization</offeringType>  
      <recurringCharges>  
        <item>  
          <frequency>Hourly</frequency>  
          <amount>0.2</amount>  
        </item>  
      </recurringCharges>  
    </item>  
  </reservedInstancesSet>  
</DescribeReservedInstancesResponse>
```

リザーブドインスタンスは、購入の完了後いつでも実行できます。リザーブドインスタンスを実行するには、オンデマンド EC2 インスタンスと同じ方法で起動します。必ずリザーブドインスタンスに対して指定したものと同一基準を指定してください。AWS では自動的に低い時間料金が課金されます。

## 明細書 ( 請求書 ) を読む

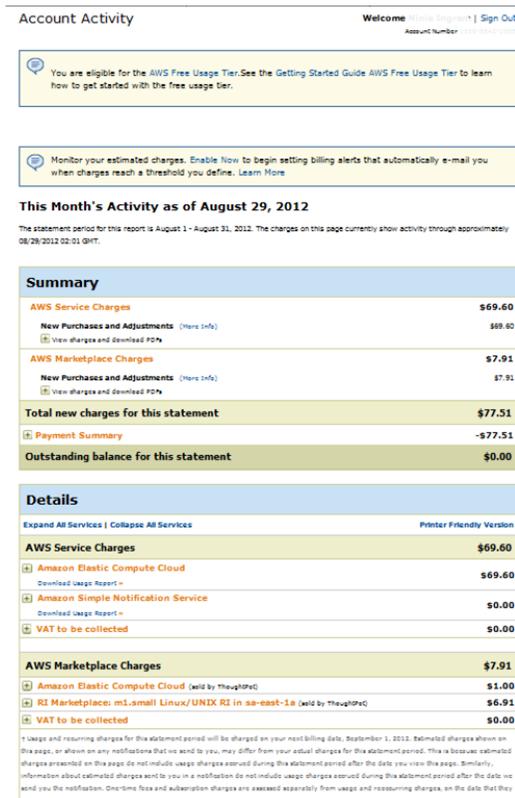
### アカウントアクティビティ

AWS マネジメントコンソールの [Account Activity] ページを表示することによって、アカウントへの請求および料金を知ることができます。このページにアクセスするには、アカウント名の横のドロップダウン矢印をクリックします。



[Account Activity] ページには、アカウントに課されたすべての料金が表示されます。これには、前払い料金、一括払い料金、および定期料金が含まれます。課金すべての概要と、課金の詳細なリストの両方を取得できます。

リザーブドインスタンスマーケットプレイスのサードパーティのリザーブドインスタンスの前払い料金は、[AWS Marketplace Charges] セクションに表示され、その横に販売者名が表示されます。ただし、これらのリザーブドインスタンスのすべての定期料金または使用量は、[AWS Service Charges] セクションに表示されます。



Account Activity Welcome [Alex Deegan](#) | [Sign Out](#)  
Account Number: [XXXXXXXXXX](#)

You are eligible for the [AWS Free Usage Tier](#). See the [Getting Started Guide: AWS Free Usage Tier](#) to learn how to get started with the free usage tier.

Monitor your estimated charges. [Enable Now](#) to begin setting billing alerts that automatically e-mail you when charges reach a threshold you define. [Learn More](#)

**This Month's Activity as of August 29, 2012**  
The statement period for this report is August 1 - August 31, 2012. The charges on this page currently show activity through approximately 08/29/2012 02:01 GMT.

Summary	
<b>AWS Service Charges</b>	<b>\$69.60</b>
New Purchases and Adjustments <small>(<a href="#">View Info</a>)</small>	\$69.60
<a href="#">View charges and download PDFs</a>	
<b>AWS Marketplace Charges</b>	<b>\$7.91</b>
New Purchases and Adjustments <small>(<a href="#">View Info</a>)</small>	\$7.91
<a href="#">View charges and download PDFs</a>	
<b>Total new charges for this statement</b>	<b>\$77.51</b>
<a href="#">Payment Summary</a>	-\$77.51
<b>Outstanding balance for this statement</b>	<b>\$0.00</b>

Details	
<a href="#">Expand All Services</a>   <a href="#">Collapse All Services</a> <span style="float: right;"><a href="#">Printer Friendly Version</a></span>	
<b>AWS Service Charges</b>	<b>\$69.60</b>
<a href="#">Amazon Elastic Compute Cloud</a>	\$69.60
<a href="#">Download Usage Report</a>	
<a href="#">Amazon Simple Notification Service</a>	\$0.00
<a href="#">Download Usage Report</a>	
<a href="#">VAT to be collected</a>	\$0.00
<b>AWS Marketplace Charges</b>	<b>\$7.91</b>
<a href="#">Amazon Elastic Compute Cloud (sold by ThoughtSpac)</a>	\$1.00
<a href="#">RI Marketplace: ml.small Linux/UNIX RI in sa-east-1 (sold by ThoughtSpac)</a>	\$6.91
<a href="#">VAT to be collected</a>	\$0.00

\* Usage and recurring charges for this statement period will be charged on your next billing date, September 1, 2012. Estimated charges shown on this page, or shown on any notifications that we send to you, may differ from your actual charges for this statement period. This is because estimated charges presented on this page do not include usage charges accrued during this statement period after the date you view this page. Similarly, information about admitted charges sent to you in a notification do not include usage charges accrued during this statement period after the date we send you the notification. One-time fees and subscription charges are assessed separately from usage and recurring charges, on the date that they occur.

料金をオンラインで表示できるほか、料金情報を示す PDF をダウンロードすることもできます。

**Summary**

AWS Service Charges			\$69.60
<b>New Purchases and Adjustments</b> (More Info)			\$69.60
<a href="#">View charges and download PDFs</a>			
	AWS Services: Subscription charge	August 24, 2012	\$1.20
	AWS Services: Subscription charge	August 24, 2012	\$2.40
	AWS Services: Subscription charge	August 22, 2012	\$6.00
	AWS Services: Subscription charge	August 27, 2012	\$34.00
	AWS Services: Subscription charge		
	AWS Services: Subscription charge		
	AWS Services: Subscription charge		
	AWS Services: Subscription charge		
	AWS Services: Subscription charge		
	AWS Services: Subscription charge		
<b>Total New Purchases and Adjustment</b>			
<b>AWS Marketplace Charges</b>			
<b>New Purchases and Adjustments</b> (More Info)			
<a href="#">View charges and download PDFs</a>			
	AWS Marketplace: Subscription charge		
	AWS Marketplace: Subscription charge		
	AWS Marketplace: Subscription charge		
	AWS Marketplace: Subscription charge		
	AWS Marketplace: Subscription charge		
	AWS Marketplace: Subscription charge		
	AWS Marketplace: Subscription charge		
<b>Total New Purchases and Adjustment</b>			
<b>Total new charges for this statement</b>			

**Amazon Web Services Statement**  
 Email or talk to us about your AWS account or bill, visit [aws.amazon.com/contact-us/](http://aws.amazon.com/contact-us/)

Account number: [REDACTED]

Statement Number: [REDACTED]  
 Statement Date: August 20, 2012

Bill to Address:  
 ATTN: [REDACTED]  
 [REDACTED]  
 [REDACTED]

**TOTAL AMOUNT DUE ON August 20, 2012 \$1.20**

This statement is for the billing period August 1 - August 31, 2012

Greetings from Amazon Web Services, we're writing to provide you with an electronic statement of your transactions on the AWS Marketplace. Additional information regarding your bill, individual service charge details, and your account history are available on the Account Activity Page.

**Summary**

<b>AWS Marketplace Charges</b>	\$1.20
1 x Amazon Elastic Compute Cloud (one time fee)	\$1.20
Charges	\$0.00
Credits	\$0.00
Tax *	\$0.00
<b>Total for this statement</b>	<b>\$1.20</b>

\* Details of services from Japan on which consumption tax is included are provided on the Account Activity Page, visit [aws.amazon.com/](http://aws.amazon.com/)  
 \* This statement is not an invoice. It is a statement of transactions on the AWS Marketplace. If you need an invoice, please contact the seller listed in the description of the transaction.

**Detail**

Amazon Elastic Compute Cloud sold by ThoughtPot	\$1.20
1 x Amazon Elastic Compute Cloud (one time fee)	\$1.20
Availability Zone - GRL2	
m1.small	
Charges	\$0.00
Estimated US sales tax to be collected	\$0.00

[Detail] セクションには、アベイラビリティゾーン、インスタンスタイプ、コスト、インスタンス数などのリザーブドインスタンスに関する情報が含まれています。また、購入したリザーブドインスタンスの販売者名も含まれます。

## リザーブドインスタンスに関する情報を取得する

状態、インスタンスタイプ、アベイラビリティゾーン、期間など、お客様のリザーブドインスタンスに関する情報は、予約してある処理能力をいつ使用するかを決定する際に役立ちます。お客様のアカウントで使用できるリザーブドインスタンスに関する情報は、購入や販売に使用したいいずれかの Amazon EC2 ツールを使用することによって確認できます。

## リザーブドインスタンスの状態

リザーブドインスタンスは、以下のいずれかの状態になります。

- Active – リザーブドインスタンスは、使用可能です。
- Payment-Pending – アマゾン ウェブ サービス (AWS) で、リザーブドインスタンスに対するお客様の支払が処理中です。リザーブドインスタンスは、状態が Active になると使用できます。
- Retired – リザーブドインスタンスは終了しました。この状態になる原因としては、以下のいずれかが考えられます。
  - AWS がお客様からの支払を受け取っていない。例えば、クレジットカードの処理を行えなかった場合などです。
  - リザーブドインスタンスの期限が切れた。
  - リザーブドインスタンスがキャンセルされた。

[Reserved Instance] ページの [State] 列に表示される状態情報は、[My Listings] タブの [Listing State] に表示される状態情報とは異なります。[State] 列には、リザーブドインスタンスの状態が表示されます。[Listing State] には、リザーブドインスタンスの一覧の状態が表示されます。[My Listings] タブには、お客様が リザーブドインスタンスマーケットプレイス で販売者になっている場合のみ情報が表示されます。詳細については、「[リザーブドインスタンスの出品状態 \(p. 286\)](#)」を参照してください。

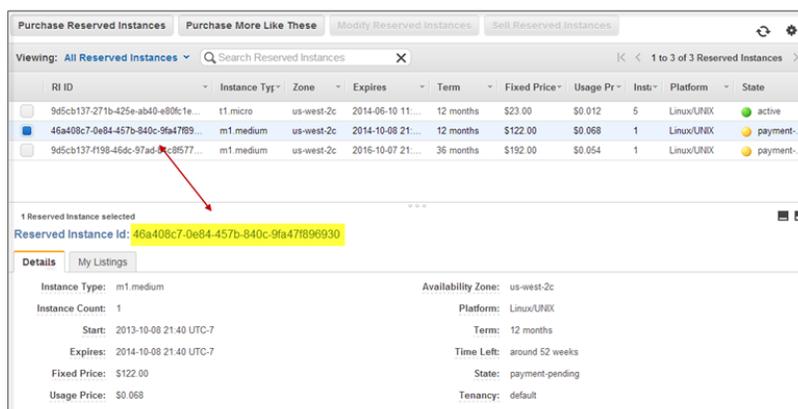
## AWS Management Console

出品したインスタンスを表示するには

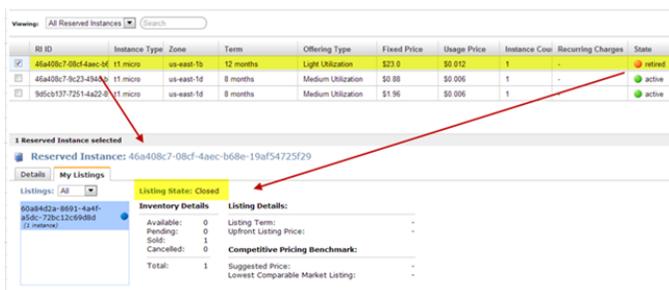
1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. [Navigation] ペインで [Reserved Instances] をクリックします。

[Reserved Instances] ページに、アカウントのインスタンスの一覧が表示されます。

3. リザーブドインスタンスを選択します。[Details] タブに、選択したインスタンスの詳細が表示されます。



4. リザーブドインスタンスマーケットプレイス で販売者になっている場合、リザーブドインスタンスの出品に関する情報が必要なときは、[My Listings] タブをクリックします。選択したリザーブドインスタンスの詳細が表示されます。



## Amazon EC2 CLI

出品したインスタンスを表示するには

- `ec2-describe-reserved-instances-listing` を実行して、出品の詳細を取得します。

```
PROMPT> ec2-describe-reserved-instances-listings
```

次のような出力が Amazon EC2 から返されます。

```
PROMPT> ec2-describe-reserved-instances-listings
Type ReservedInstancesListingId ReservedInstancesId CreateDate UpdateDate
Status StatusMessage
LISTING 615d8a10-8224-4c19-ba7d-b9aa0example 1ba8e2e3-d20d-44ec-b202-
fcb6aexample Wed Aug 22 09:02:58 PDT 2012 Wed Aug 22 14:24:26 PDT 2012 can
celled cancelled
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 1
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 10 $1.2
PRICE-SCHEDULE 9 $1.08
PRICE-SCHEDULE 8 $0.96
PRICE-SCHEDULE 7 $0.84
PRICE-SCHEDULE 6 $0.72
PRICE-SCHEDULE 5 $0.6
PRICE-SCHEDULE 4 $0.48
PRICE-SCHEDULE 3 $0.36
PRICE-SCHEDULE 2 $0.24
PRICE-SCHEDULE 1 $0.12
LISTING d5fa5166-83c3-40e4-abb2-b7298example 1ba8e2e3-d20d-44ec-b202-
fcb6aexample Wed Aug 22 14:31:55 PDT 2012 Wed Aug 22 14:42:40 PDT 2012 closed
closed
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 1
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 10 $0.9
PRICE-SCHEDULE 9 $0.81
PRICE-SCHEDULE 8 $0.72
PRICE-SCHEDULE 7 $0.63
PRICE-SCHEDULE 6 $0.54
PRICE-SCHEDULE 5 $0.45
PRICE-SCHEDULE 4 $0.36
PRICE-SCHEDULE 3 $0.27
PRICE-SCHEDULE 2 $0.18
PRICE-SCHEDULE 1 $0.09
....
LISTING 095c0e18-c9e6-4692-97e5-653e0example b847fa93-c736-4eae-bca1-
e3147example Tue Aug 28 18:21:07 PDT 2012 Tue Aug 28 18:21:07 PDT 2012 active
active
INSTANCE-COUNT available 1
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.2
PRICE-SCHEDULE 3 $1.2
PRICE-SCHEDULE 2 $1.2
PRICE-SCHEDULE 1 $1.2
```

## Amazon EC2 API

出品したインスタンスを表示するには

- `DescribeReservedInstancesListings` を呼び出して出品の詳細を取得します。

呼び出しは、次の例のようになります。

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstancesListings
&AUTHPARAMS
```

以下に、応答の例を示します。

```
<DescribeReservedInstancesListingsResponse>
  <requestId>cec5c904-8f3a-4de5-8f5a-ff7f9example</requestId>
  <reservedInstancesListingsSet>
    <item>
      <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
      <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re
servedInstancesId>
      <createDate>2012-08-30T17:11:09.449Z</createDate>
      <updateDate>2012-08-30T21:00:42.300Z</updateDate>
      <status>active</status>
      <statusMessage>active</statusMessage>
      <instanceCounts>
        <item>
          <state>Available</state>
          <instanceCount>2</instanceCount>
        </item>
        <item>
          <state>Sold</state>
          <instanceCount>1</instanceCount>
        </item>
        <item>
          <state>Cancelled</state>
          <instanceCount>0</instanceCount>
        </item>
        <item>
          <state>Pending</state>
          <instanceCount>0</instanceCount>
        </item>
      </instanceCounts>
      <priceSchedules>
        <item>
          <term>11</term>
          <price>2.5</price>
          <currencyCode>USD</currencyCode>
          <active>true</active>
        </item>
        <item>
          <term>10</term>
          <price>2.5</price>
          <currencyCode>USD</currencyCode>
          <active>false</active>
        </item>
      </priceSchedules>
    </item>
  </reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

```
<item>
  <term>9</term>
  <price>2.5</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>8</term>
  <price>2.0</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>7</term>
  <price>2.0</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>6</term>
  <price>2.0</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>5</term>
  <price>1.5</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>4</term>
  <price>1.5</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>3</term>
  <price>0.7</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>2</term>
  <price>0.7</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>1</term>
  <price>0.1</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
```

```
</reservedInstancesListingsSet>  
</DescribeReservedInstancesListingsResponse>
```

## リザーブドインスタンスの変更

コンピューティングニーズが変化したときは、リザーブドインスタンスを変更して、引き続き予約した容量を利用できます。同じリージョン内のアベイラビリティゾーン間でリザーブドインスタンスを移動したり、ネットワークプラットフォームを EC2-Classic と EC2-VPC の間で変更したりできます。さらに、Amazon EC2 API バージョン 2013-10-01 からは、利用可能な容量があり、Linux/UNIX 製品プラットフォーム（現時点で変更できる唯一のプラットフォーム）に対する予約である場合は、予約のインスタンスタイプを、同じファミリー（M1 インスタンスファミリーなど）のさらに大きいインスタンスまたは小さいインスタンスに変更できます。

変更しても、リザーブドインスタンスの有効な残りの期間は変わりません。終了日は同じです。リザーブドインスタンスの変更には手数料は必要なく、新しく課金されたり、請求書が届いたりすることはありません。

予約を変更した場合でも、リザーブドインスタンスによってカバーされるインスタンスは実行し続けます。ただし、変更後、リザーブドインスタンスの割引料金は、リザーブドインスタンスの新しいパラメータと一致するインスタンスに適用されるようになります。EC2 インスタンスのオンデマンド料金で請求されるようになり、変更されたリザーブドインスタンスの割引料金は適用されません。ただし、アカウントに他の該当する予約がある場合は、他のリザーブドインスタンスの料金で請求されます。

変更しても、リザーブドインスタンスの使用、購入、売却の方法には影響ありません。リザーブドインスタンスを購入するときは、製品プラットフォーム、インスタンスタイプ、ネットワークプラットフォーム、テナント属性、有効期間、提供タイプ、およびアベイラビリティゾーンを指定する必要があります。

リクエストのステータスは、コンソールで、あるいは `DescribeReservedInstancesModifications` API アクションまたは `ec2-describe-reserved-instances-modifications` CLI コマンドを呼び出して、表示できます。

ここでは変更のプロセスについて説明します。

- [変更プロセスを理解する \(p. 251\)](#) – 変更リクエストを送信するとどうなりますか？
- [変更のステータスの確認 \(p. 253\)](#) – 変更リクエストをどのようにして追跡しますか？
- [変更の要件 \(p. 254\)](#) – 変更できるのはどのリザーブドインスタンスですか？
- [予約のインスタンスタイプの変更 \(p. 255\)](#) – リザーブドインスタンスのインスタンスサイズを変更できるのはいつですか？
- [変更リクエストの送信 \(p. 257\)](#) – リザーブドインスタンスを変更するにはどうすればよいですか？

## 変更プロセスを理解する

リザーブドインスタンスは次のいずれかの方法で変更できます。

- 同じリージョン内のアベイラビリティゾーン間でリザーブドインスタンスを移動します。

リザーブドインスタンスのアベイラビリティゾーンを変更すると、元のゾーンでのインスタンスに対する予約した容量と割引料金の適用が停止し、新しいアベイラビリティゾーンでの適用が開始されます。

- リザーブドインスタンスのネットワークプラットフォームを、EC2-VPC と EC2-Classic の間で変更します。

リザーブドインスタンスのネットワークプラットフォームを変更すると、元のネットワークプラットフォームでのインスタンスに適用された予約容量が停止し、新しいネットワークプラットフォームでの適用が開始します。ただし、割引料金の適用は、残っているリザーブドインスタンスパラメータに一致する EC2-Classic および EC2-VPC の両方のインスタンスに対して続けられます。

- 同じインスタンスファミリー（たとえば M1 インスタンスファミリー）内でリザーブドインスタンスのインスタンスタイプをアップグレードまたはダウングレードします。

リザーブドインスタンスのインスタンスタイプを変更する場合は、予約のインスタンスファミリーでより大きいまたはより小さいインスタンスタイプを使用できること、および変更に必要な該当するリザーブドインスタンスがあることを確認する必要があります。

予約全体または予約のサブセットを変更できます。予約のサブセットを変更すると、Amazon EC2 は元のリザーブドインスタンスを複数の新しいリザーブドインスタンスに分割します。例えば、us-east-1a 内に 10 個のインスタンスがリザーブドインスタンスにあり、そのうち 5 個のインスタンスを us-east-1b に移動する場合、変更リクエストにより 2 つの新しいリザーブドインスタンス ( us-east-1a (元のゾーン) の 5 個のインスタンス用に 1 つと、us-east-1b の 5 個のインスタンス用に 1 つ ) が作成されます。

Amazon EC2 は、使用可能な容量に応じて、変更リクエストに可能な限り速やかに対応します。変更リクエストが完了するまで、リザーブドインスタンスに関連付けられている予約容量と割引料金は、変更前の予約パラメータに基づいて継続されます。

 Note

送信した後で保留中の変更リクエストをキャンセルまたは変更することはできません。変更の処理中、変更しているリザーブドインスタンスのステータスは *active ( pending modification )* です。変更が完了した後は、別の変更リクエストを送信して、実行した変更をロールバックできます。

リザーブドインスタンス変更リクエストが成功した場合。

- 変更後の予約がすぐに有効になり、変更リクエストが完了した時刻から、リザーブドインスタンスの割引料金が新しいインスタンスに適用されます。例えば、午後 9 時 15 分にリザーブドインスタンスの変更が成功した場合、割引料金は午後 9:時から新しいインスタンスに移ります ( 変更されたリザーブドインスタンスの発行日は DescribeReservedInstances API アクションまたは ec2-describe-reserved-instances CLI コマンドを使用して取得できます )。
- 変更後のリザーブドインスタンスの終了日は予約の元の終了日と同じです。有効期限のうち 16 か月が残っている 3 年の予約を正常に変更した場合、変更後の予約は 16 か月のリザーブドインスタンスであり、終了日は変更前のリザーブドインスタンスと同じです。

例えば、次の表で、変更 #1 は、RI af9f760... が正常に変更されて 2013-08-30 16:00 UTC-7 に終了し、それと同じ日時に新しい RI 46a408c... が作成されたことを示します。変更 #2 は、RI 46a408c... が正常に変更されて 2013-09-03 14:00 UTC-7 に終了し、それと同じ日時に新しい RI b847fa9... が作成されたことを示します。

RI ID	Start	Expires	Term	State
93bbca2-46a...	2013-08-28 10:01 UTC-7	2016-08-27 10:01 UTC-7	36 months	active
93bbca2-6a7...	2013-09-04 14:00 UTC-7	2014-08-30 15:45 UTC-7	12 months	active
93bbca2-c7a...	2013-08-30 14:00 UTC-7	2014-08-28 10:01 UTC-7	12 months	active
b847fa93-0b3...	2013-09-03 14:00 UTC-7	2014-08-30 15:45 UTC-7	12 months	active
Modification #2	2013-08-28 10:01 UTC-7	2013-08-30 14:00 UTC-7	12 months	retired
46a408c7-61d...	2013-08-30 16:00 UTC-7	2013-09-03 14:00 UTC-7	12 months	retired
Modification #1	2013-08-30 15:45 UTC-7	2013-08-30 16:00 UTC-7	12 months	retired
af9f760e-92bf...	2013-08-30 15:45 UTC-7	2013-08-30 16:00 UTC-7	12 months	retired
d16f7a91-b9ef...	2013-08-30 16:00 UTC-7	2013-09-04 14:00 UTC-7	12 months	retired

- 元の予約は終了します。その終了日は新しい予約の開始日であり、新しい予約の終了日はアクティブであったときの元のリザーブインスタンスの終了日と同じです。
- 変更後のリザーブインスタンスの固定価格は 0 USD であり、元のリザーブインスタンスの固定価格ではありません。



#### Note

変更後の予約の固定価格はアカウントに適用される割引範囲の計算に影響を与えません。割引範囲の計算は元の予約の固定価格に基づきます。

変更リクエストが失敗した場合。

- リザーブインスタンスは要求前の元のプロパティを維持します。
- リザーブインスタンスに対して別の変更リクエストを行うことができます。

要求のステータスは、変更しているリザーブインスタンスの状態を見ることで確認できます。詳細については、[変更のステータスの確認 \(p. 253\)](#) を参照してください。

## 変更のステータスの確認

変更リクエストの状態は、AWS Management Consoleの [State] フィールドに表示されます。また、DescribeReservedInstancesModifications API アクションを使用して変更リクエストについての詳細情報を取得することもできます。返される state によって示される要求の状態は in-progress、fulfilled、または failed です。

変更できるのはリザーブインスタンスが *active* の場合だけです。他の状態のリザーブインスタンスを変更することはできません。また、リザーブインスタンスマーケットプレイスに出品されているリザーブインスタンスも変更できません。詳細については、[変更の要件 \(p. 254\)](#) を参照してください。リザーブインスタンスが *active* 状態ではない場合、または変更できない場合は、AWS Management Consoleの [Modify Reserved Instances] ボタンが有効になりません。API を使用して *active* ではないリザーブインスタンスを変更した場合、エラーが発生します。

アベイラビリティゾーンとネットワークプラットフォームは変更できても、インスタンスタイプは変更できないリザーブインスタンスを選択した場合は、[Modify Reserved Instances] ボタンは有効であり、[Modify Reserved Instances] ページに進むことができます。ただし、インスタンスタイプを変更することはできません。詳細については、[予約のインスタンスタイプの変更 \(p. 255\)](#) を参照してください。

### 変更の状態

変更リクエストは処理中です。変更リクエストが処理中の間、変更対象のリザーブインスタンスのステータスは *active ( pending modification )* と表示されます。リザーブインスタンスがこの状態になっているのは短時間だけです。変更が成功したかどうかに応じて、状態は *active* に戻るか、または *retired* になります ( DescribeReservedInstancesModifications API アクションを使用している場合、変更リクエストのステータスは *processing* と表示されます )。

変更は成功しました。変更が成功した場合、変更対象のリザーブインスタンスは *retired* になり、新しいリザーブインスタンスは要求した変更設定で作成されます。新しいリザーブインスタンスのステータスは *active* です ( DescribeReservedInstancesModifications API アクションを使用している場合、変更リクエストのステータスは *fulfilled* と表示されます )。



#### Note

新しいリザーブインスタンスがアクティブ化されている短時間の間、元のリザーブインスタンスは *retired ( pending modification )* と表示されます。

変更は失敗しました。変更が完了しなかった場合、変更対象のリザーブドインスタンスは *active* 状態を返します ( `DescribeReservedInstancesModifications` API アクションを使用している場合、変更リクエストのステータスは *failed* と表示されます )。リザーブドインスタンスを変更できない理由については、[変更の要件 \(p. 254\)](#) を参照してください。

## 変更の要件

Amazon EC2 は、以下の条件を満たしている場合に変更リクエストを処理します。

- ターゲット設定用に十分なリザーブドインスタンス処理能力がある。
- 変更しているリザーブドインスタンスがアクティブである。
- リザーブドインスタンスで別の変更リクエストが保留中になっていない。

リザーブドインスタンスはどのような頻度でも変更できます。ただし、前の変更リクエストがまだ保留されている場合、つまり、前の変更リクエストが *active ( pending modification )* 状態である場合は、リザーブドインスタンスの変更リクエストを送信できません。変更の状態の詳細については、「[変更のステータスの確認 \(p. 253\)](#)」を参照してください。

- リザーブドインスタンスがリザーブドインスタンスマーケットプレイスには出品されていない。

リザーブドインスタンスマーケットプレイスに出品されているリザーブドインスタンスを変更するには、出品をキャンセルし、リザーブドインスタンスを変更した後で、再び出品します。

さらに、リザーブドインスタンスマーケットプレイスの出品を、購入する前または購入と同時に変更することはできません。ただし、マーケットプレイスの出品を購入した後で変更リクエストを送信することはできます。詳細については、[リザーブドインスタンスマーケットプレイス \(p. 233\)](#) を参照してください。

- 次の属性のみが変更されます。同じリージョンのアベイラビリティゾーン、同じインスタンスファミリー (たとえば M1 インスタンスファミリー) 内のインスタンスタイプ、およびネットワークプラットフォーム。
- リザーブドインスタンスを結合するときは、リザーブドインスタンスの終了日における 24 時間形式での時間の桁が同じである必要がありますが、分または秒の桁が同じである必要はありません。つまり、同じ日の 13:01 と 13:59 に終了するリザーブドインスタンスは結合できますが、12:59 と 13:01 に終了するリザーブドインスタンスは結合できません。
- インスタンスタイプ属性を変更する場合
  - 変更するリザーブドインスタンスは、Amazon Linux/UNIX 製品プラットフォーム (現時点でインスタンスタイプを変更できる唯一の製品プラットフォーム) のものでなければなりません。
  - リザーブドインスタンスの新しいインスタンスタイプは、元の予約と同じインスタンスファミリー (たとえば M1 インスタンスファミリー) に属している必要があります。
  - リザーブドインスタンスのインスタンスタイプをアップグレードする場合、ターゲットのリザーブドインスタンスの標準インスタンス数を、元の予約の標準インスタンス数と同じにする必要があります。

変更の対象として複数のリザーブドインスタンスを選択し、それらのリザーブドインスタンスうちの 1 つ以上がインスタンスタイプを変更できない製品プラットフォームのものである場合、[Modify Reserved Instances] ページには、選択したリザーブドインスタンスのインスタンスタイプを変更するオプションが表示されません。インスタンスタイプの変更の詳細については、「[予約のインスタンスタイプの変更 \(p. 255\)](#)」を参照してください。

- ターゲットのリザーブドインスタンス設定の属性は、アベイラビリティゾーン、インスタンスタイプ、およびネットワークプラットフォームの組み合わせが一意になるように変更する必要があります。

リザーブドインスタンスの変更方法については、[変更リクエストの送信 \(p. 257\)](#) を参照してください。

## 予約のインスタンスタイプの変更

特定の条件が満たされた場合、リザーブドインスタンスのインスタンスタイプを調整できます。複数のインスタンスサイズを持つインスタンスファミリーに Amazon Linux/UNIX に対する容量予約がある場合、異なるインスタンスタイプを同じファミリーに含めるリザーブドインスタンスの変更をリクエストできます。容量が存在し、変更によってリザーブドインスタンスのインスタンスサイズフットプリントが変わらない場合は、リクエストは正常に処理されます。

例えば、1つの m1.large インスタンスの予約を4つの m1.small インスタンスに分割したり、4つの m1.small インスタンスの予約を1つの m1.large インスタンスに結合したりできます。どちらの場合も、予約のインスタンスサイズフットプリントは変化しません。これに対し、2つの m1.small インスタンスの予約を1つの m1.large インスタンスに変更することはできません。これは、現在の予約の既存のインスタンスサイズフットプリントがリクエストされている予約より小さいためです。

インスタンスサイズのフットプリントは、インスタンスタイプの正規化係数と予約に含まれるインスタンスの数によって決まります。ここでは、リザーブドインスタンスのインスタンスタイプを変更する2つの方法、およびインスタンスタイプの正規化係数を使用して、可能なインスタンスタイプの変更を調べる方法について説明します。

- [インスタンス正規化係数を理解する \(p. 255\)](#)
- [インスタンスタイプのアップグレード \(p. 256\)](#)
- [インスタンスタイプのダウングレード \(p. 256\)](#)

### 例外

以下のインスタンスタイプのサイズは、ファミリー内に他のサイズがないため変更できません。

- t1.micro
- cc1.4xlarge
- cc2.8xlarge
- cg1.8xlarge
- cr1.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- g2.2xlarge

インスタンスタイプまたは製品プラットフォームを変更できないリザーブドインスタンスを変更しようとすると、[Modify Reserved Instances] ページに進むことはできますが、Amazon EC2 コンソールでの変更オプションは [Network] と [Availability Zone] に限定されます。

インスタンスタイプの変更は、リザーブドインスタンスの他の仕様詳細（リージョン、使用タイプ、テナント属性、製品、終了日時など）が一致し、利用できる容量がある場合にのみ許可されることに注意してください。さらに、そのような変更によって、請求される割引された時間単位使用料金、または予約の終了日は変更されません。

### インスタンス正規化係数を理解する

各リザーブドインスタンスにはインスタンスサイズのフットプリントがあります。リザーブドインスタンスのインスタンスタイプを変更する場合、インスタンスタイプがダウンサイジングされてもアップサイジングされても、フットプリントは維持されます。変更後の設定のフットプリントが元の設定のサイズと一致しない場合、変更リクエストは処理されません。

インスタンスタイプのフットプリントのサイズは、正規化係数を使用して計算できます。正規化係数は、インスタンスファミリー（たとえば M1 インスタンスファミリー）内のタイプのサイズに基づきま

す。正規化係数は同じインスタンスファミリー内でのみ意味を持ちます。異なるファミリーのインスタンスタイプに変更することはできません。

次の表では、インスタンスファミリー内で適用される正規化係数を示します。例えば、m1.small インスタンスの正規化係数は 1、m1.medium インスタンスの係数は 2、m1.large インスタンスの係数は 4 です。

インスタンスサイズ	正規化係数
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
8xlarge	64

各リザーブドインスタンスは、正規化されたインスタンスユニットの総数を保持しています。これは、インスタンスの数にインスタンスタイプの正規化係数を掛けたものと等しい値です。例えば、m1.medium の正規化係数は 2 なので、m1.medium インスタンス 4 個のリザーブドインスタンスは正規化されたインスタンスユニット 8 個に相当します。次のようにして値を求めることができます。

4 [個数] x 2 [正規化係数]

リザーブドインスタンスの標準インスタンスユニットの総数が同じである場合は、同じインスタンスファミリー（たとえば M1 インスタンスタイプ）内で、リザーブドインスタンスを異なるインスタンスサイズとして割り当てることができます。4 個の m1.medium インスタンスのリザーブドインスタンスがある場合、8 個の m1.small インスタンスの予約に変更できます（m1.small インスタンスの正規化係数は 1 なので、8 x 1）。変更後のリザーブドインスタンスでも、正規化されたインスタンスユニットの数またはインスタンスサイズフットプリントは変わりません。

## インスタンスタイプのアップグレード

あるインスタンスファミリー（たとえば M1 インスタンスファミリー）に属している複数の小さいインスタンスタイプの予約を統合して、同じファミリーの大きいインスタンスタイプの予約にすることができます。予約をアップグレード、つまりアップサイジングするには、大きいインスタンスタイプ統合するのに十分な数の小さいインスタンスタイプがあり、統合しても予約のインスタンスサイズフットプリント全体が変化しない必要があります。

例えば、4 個の m1.small インスタンス（4 個の正規化されたインスタンスユニット（4 x 1）と同等）を、1 個の m1.large インスタンス（やはり 4 個の正規化されたユニット（1 x 4）と同等）に変換できます。しかし、1 個の m1.small インスタンス（1 x 1）の予約を 1 個の m1.large インスタンス（1 x 4）に変換することはできません。この場合、2 つの予約は同等ではありません。

複数のインスタンスサイズを持ち、[例外 \(p. 255\)](#) リストに出品されていないインスタンスファミリーは、大きいインスタンスタイプサイズに変更またはアップグレードできません。

## インスタンスタイプのダウングレード

大きいインスタンスタイプのリザーブドインスタンスを、同じファミリー（たとえば M1 インスタンスファミリー）の小さいインスタンスタイプの複数のリザーブドインスタンスに変更できます。リザーブドインスタンスをダウングレード、つまりダウンサイジングすると、実際には大きい予約が複数の小さい

い予約に分割されます。インスタンスタイプのアップサイジングと同様に、予約のインスタンスサイズフットプリント全体が変わらない場合にのみダウンサイジングは成功します。

例えば、2 個の m1.large インスタンスの予約は 8 個の正規化されたインスタンスユニット (2 x 4) と同等です。容量があるものとする、これらを 4 個の m1.medium インスタンスのリザーブドインスタンスに変換できます。これもまた 8 個の正規化されたインスタンスユニット (4 x 2) と同等です。

複数のインスタンスサイズを持ち、[例外 \(p. 255\)](#) リストに出品されていないインスタンスファミリーは、小さいインスタンスタイプサイズに変更またはダウングレードできません。

Amazon EC2 のインスタンスタイプについては、[インスタンスタイプ \(p. 109\)](#) を参照してください。リザーブドインスタンス変更リクエストについては、[変更リクエストの送信 \(p. 257\)](#) を参照してください。

## 変更リクエストの送信

### Abstract

AWS Management Console、CLI、または API を使用して、リザーブドインスタンスを変更するリクエストを送信します。

AWS Management Console を使用して、以下の方法でリザーブドインスタンスを変更できます。

- 同じリージョンの別の Availability Zone に容量の予約のすべてまたは一部を移動します。
- リザーブドインスタンスのすべてまたは一部のネットワークプラットフォームを、EC2-Classic から EC2-VPC に (またはその逆に) 変更します。
- 容量があり、変更により予約容量の領域が保持される場合は、同じインスタンスファミリー (たとえば M1 インスタンスファミリー) 内でリザーブドインスタンスのインスタンスタイプのすべてまたは一部をアップグレードまたはダウングレードします。

AWS CLI ([modify-reserved-instances](#))、Amazon EC2 CLI ([ec2-modify-reserved-instances](#))、Amazon EC2 API ([ModifyReservedInstances](#))、および [AWS SDK for Java](#) を使用して、プログラムでこれらの変更を完了することもできます。

リザーブドインスタンスの変更を使用できるプログラミング ツールについての詳細は、「[リザーブドインスタンスを使用するためのツール \(p. 222\)](#)」を参照してください。

### [Modify Reserved Instances] ページを使用する

このセクションでは、変更を行うために、Amazon EC2 Console で [Modify Reserved Instances] ページを使用する方法について説明します。また、以下についても説明します。

- [個数とユニットの値 \(p. 257\)](#)
- [変更リクエストに関するメッセージ \(p. 260\)](#)

#### 個数とユニットの値

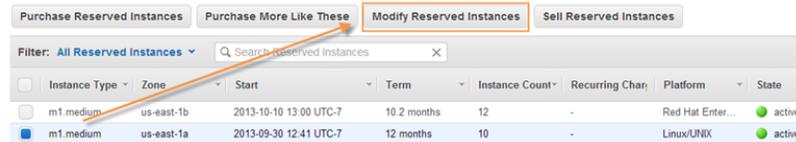
EC2 console の [Modify Reserved Instances] ページでは、ターゲット (または新しい) 設定に割り当てられたリザーブドインスタンス総数およびユニット数と、変更できるリザーブドインスタンスの総数が追跡されます。個数は、現在のインスタンスタイプのインスタンスのリテラル数です。ユニットは、インスタンスファミリー (たとえば M1 インスタンスファミリー) に対する予約の総インスタンスサイズを表します。ユニットは、個数にインスタンスタイプの正規化係数を掛けた値です。インスタンスの正規化係数の詳細については、「[インスタンス正規化係数を理解する \(p. 255\)](#)」を参照してください。

リザーブドインスタンスの各ターゲット設定行には個数とユニットの値があります。変更が可能なインスタンス数よりも多くまたは少なくインスタンスを指定した場合、割り当てられたリザーブドインスタンス総数は赤で表示されます。ターゲット設定に対して指定された総ユニット数が、変更できるリザーブドインスタンスの総ユニット数と等しくない場合、[Continue] をクリックすることはできません。変

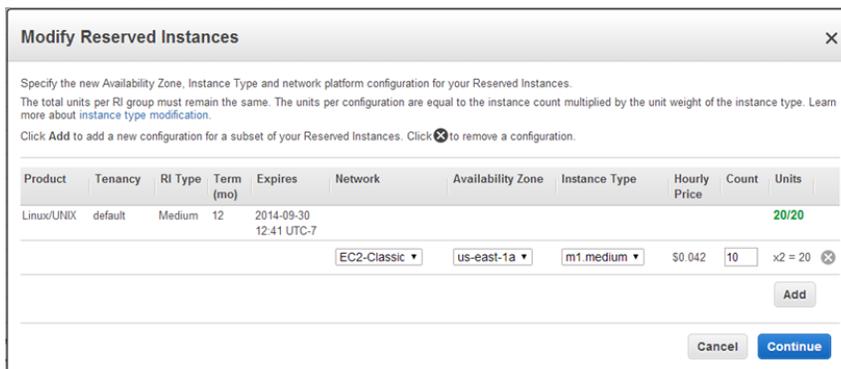
更が可能なすべてのリザーブドインスタンスに対して変更を指定すると、ユニットの値は緑で表示されます。

リザーブドインスタンスを変更するには

1. [Reserved Instances] ページで、変更するリザーブドインスタンスを 1 つ以上選択し、[Modify Reserved Instances] をクリックします。



[Modify Reserved Instances] ページに、選択したリザーブドインスタンスとそのインスタンスの現在の設定が表示されます。



2. リザーブドインスタンスを変更する方法を決定します。この例では、変更しないものが 2 つ、別のアベイラビリティゾーンに移行するものが 2 つ、EC2-Classic から EC2-VPC に変換するものが 2 つ、m1.medium から m1.large にインスタンスタイプをアップグレードするものが 2 つとして、リザーブドインスタンスを変更します。アップグレードには、1 つの m1.large インスタンスごとに、2 つの m1.medium インスタンスが必要であることを注意してください。

次の表は、実行する変更を示しています。

変更	ネットワーク	アベイラビリティゾーン	インスタンスタイプ	変更前の個数	変更後の個数	単位
変更しない (元の RI の一部を保持する)	EC2-Classic	us-east-1a	m1.medium	2	2	4
アベイラビリティゾーンの変更	EC2-Classic	us-east-1b	m1.medium	2	2	4
ネットワークプラットフォームの変更	EC2-VPC	us-east-1a	m1.medium	2	2	4
インスタンスタイプの 変更	EC2-Classic	us-east-1a	m1.large	4	2	8

変更	ネットワーク	アベイラビリティゾーン	インスタンスタイプ	変更前の個数	変更後の個数	単位
合計				10	8	20

- 2つのリザーブインスタンスに対して元の設定を保持するには（変更しない）、[Count] に 2 と入力します。[Network]、[Availability Zone]、[Instance Type] は元のまま保持します。[Units] は自動的に 4/20 に更新されます。

リザーブインスタンスに対して行う変更ごとに行を分けて表示するために、[Add] ボタンを使用して、行を追加します。

Product	Tenancy	RI Type	Term (mo)	Expires	Network	Availability Zone	Instance Type	Hourly Price	Count	Units
Linux/UNIX	default	Medium	12	2014-09-30 12:41 UTC-7						4/20
					EC2-Classical	us-east-1a	m1.medium	\$0.042	2	x2 = 4
Add										

- アベイラビリティゾーンを変更するには、[Add] をクリックして、[Availability Zone] リストで *us-east-1b* を選択します。[Count] に 2 と入力します。[Network] および [Instance Type] は元のまま保持します。[Units] は自動的に 8/20 に更新されます。

Product	Tenancy	RI Type	Term (mo)	Expires	Network	Availability Zone	Instance Type	Hourly Price	Count	Units
Linux/UNIX	default	Medium	12	2014-09-30 12:41 UTC-7						8/20
					EC2-Classical	us-east-1a	m1.medium	\$0.042	2	x2 = 4
					EC2-Classical	us-east-1b	m1.medium	\$0.042	2	x2 = 4
Add										

- ネットワークプラットフォームを変更するには、最初に、[Add] をクリックします。次に、[Network] リストで、*EC2-VPC* を選択します。[Count] に 2 と入力します。[Availability Zone] および [Instance Type] は元のまま保持します。[Units] は自動的に 12/20 に更新されます。

Product	Tenancy	RI Type	Term (mo)	Expires	Network	Availability Zone	Instance Type	Hourly Price	Count	Units
Linux/UNIX	default	Medium	12	2014-09-30 12:41 UTC-7						12/20
					EC2-Classical	us-east-1a	m1.medium	\$0.042	2	x2 = 4
					EC2-Classical	us-east-1b	m1.medium	\$0.042	2	x2 = 4
					EC2-VPC	us-east-1a	m1.medium	\$0.042	2	x2 = 4
Add										

- インスタンスタイプを変更するには、[Add] をクリックし、[Instance Type] リストで *m1.large* を選択します。[Count] に 2 と入力します。[Availability Zone] および [Network] は元のまま保持します。[Units] は自動的に 20/20 に更新されます。

Product	Tenancy	RI Type	Term (mo)	Expires	Network	Availability Zone	Instance Type	Hourly Price	Count	Units
Linux/UNIX	default	Medium	12	2014-09-30 12:41 UTC-7						20/20
					EC2-Classical	us-east-1a	m1.medium	\$0.042	2	x2 = 4
					EC2-Classical	us-east-1b	m1.medium	\$0.042	2	x2 = 4
					EC2-VPC	us-east-1a	m1.medium	\$0.042	2	x2 = 4
					EC2-Classical	us-east-1a	m1.large	\$0.084	2	x4 = 8
Add										

インスタンスタイプを変更する場合は、インスタンスタイプを大きくすることも小さくすることもできることを確認します。インスタンスタイプの変更の詳細については、「[予約のインスタンスタイプの変更 \(p. 255\)](#)」を参照してください。

この例では、m1.medium からより容量の大きい m1.large へとインスタンスタイプをアップグレードします。m1.large インスタンスタイプの正規化係数は 4 です。1 つの m1.large インスタンスタイプの取得には、4 つの m1.small インスタンスタイプまたは 2 つの m1.medium インスタンスタイプが必要です。したがって、残りの 4 つの m1.medium インスタンスタイプ (正規化された値は 8) は、2 つの m1.large インスタンスタイプ (同じく正規化された値は 8) にアップグレードできます。[Units] には、緑で 20/20 と表示されます。この演習において、変更が可能なすべてのリザーブドインスタンスに対して変更を指定したことを意味します。これで変更リクエストの手順に進むことができます。

インスタンスの正規化係数の詳細については、「[インスタンス正規化係数を理解する \(p. 255\)](#)」を参照してください。

7. 指定した設定 (インスタンスのアップグレードなど) を削除する場合、該当する行の [X] をクリックします。



#### Note

[Modify Reserved Instances] ページに表示されている設定変更の行が 1 つだけの場合、その行は削除できません。

8. ターゲット設定の指定を完了したときに、変更の設定を確認するには、[Submit Modifications] をクリックします。Amazon EC2 は変更のリクエストを確認します。要求した設定が固有で、指定したインスタンス数が変更できるリザーブドインスタンスの総数と一致する場合、Amazon EC2 がお客様のリクエストを処理していることが通知されます。

#### We have received your Modify Reserved Instances request.



The Reserved Instances in your request will be **active (pending modification)** until the modification completes. If the modification succeeds, your Reserved Instances will be **retired**, and new **active** ones will reflect the modifications; otherwise, your original Reserved Instances will return to the **active** state.



#### Note

この時点で、Amazon EC2 は変更リクエストのパラメータが有効であることを判定しており、Amazon EC2 は要求を処理できます。変更リクエストの処理の間、この画面が表示された後でのみ、容量チェックが行われます。変更リクエストは容量がないために後でまだ失敗する可能性があります。

## 変更リクエストに関するメッセージ

場合によっては、確認の代わりに変更リクエストが不完全または失敗したことを示すメッセージが表示されることがあります。メッセージの情報を参考にして、別の変更リクエストを再送信します。

- 選択されたリザーブドインスタンスに変更できないものがあります。

次のメッセージで、Amazon EC2 は変更できないリザーブドインスタンスを示します。このようなメッセージを受け取ったら、EC2 コンソールの [Reserved Instances] ページで容量予約についての詳細情報を確認します。



#### We have received your Modify Reserved Instances request, but some of your requests could not be processed.

The Reserved Instances in your request will be **active (pending modification)** until the modification completes. If the modification succeeds, your Reserved Instances will be **retired**, and new **active** ones will reflect the modifications; otherwise, your original Reserved Instances will return to the **active** state.

However, there was an error processing your request to modify the following Reserved Instances. Please try submitting your modification request for these Reserved Instances again.

- b847fa93-4da4-4d22-9401-6b791aaef686
- f127bd27-2d9c-4df5-ac3b-4fe42624f3d3

If the problem persists, contact AWS Customer Support.

- 変更リクエストの処理中にエラーが発生しました。

送信したリザーブインスタンス変更リクエストをすべて処理できません。変更しているリザーブインスタンスの数によっては、メッセージが異なる場合があります。

**Error**  
The error(s) below occurred while we were processing your Modify Reserved Instances request. If you cannot resolve them, please contact AWS Customer Support.  
Invalid value for 'targetReservedInstancesConfigurations': Configuration LeaseConfiguration(availabilityZone=us-east-1a, targetPlatform=EC2-Classical, instanceCount=1) is a duplicate.

**Error**  
The error(s) below occurred while we were processing your Modify Reserved Instances request. If you cannot resolve them, please contact AWS Customer Support.  
Invalid value for 'targetReservedInstancesConfigurations': Configuration LeaseConfiguration(availabilityZone=us-east-1a, targetPlatform=EC2-VPC, instanceCount=1) is a duplicate.  
Invalid value for 'targetReservedInstancesConfigurations': Configuration LeaseConfiguration(availabilityZone=us-east-1a, targetPlatform=EC2-VPC, instanceCount=1) is a duplicate.  
Invalid value for 'targetReservedInstancesConfigurations': Configuration LeaseConfiguration(availabilityZone=us-east-1a, targetPlatform=EC2-Classical, instanceCount=1) is a duplicate.

これらのメッセージで、Amazon EC2 は変更リクエストを処理できない理由を示します。例えば、変更しているリザーブインスタンスの1つ以上のサブセットに同じターゲット設定 (アベイラビリティゾーンとプラットフォームの組み合わせ) を指定したような場合です。リザーブインスタンスのインスタンス詳細が一致し、変更対象のリザーブインスタンスのすべてのサブセットのターゲット設定が一意であることを確認して、変更リクエストの再送信を試みます。

次は何をしますか？

- [リザーブインスタンスに関する情報を取得する \(p. 246\)](#)
- [リザーブインスタンスを購入する \(p. 236\)](#)
- [リザーブインスタンスマーケットプレイスでの販売 \(p. 261\)](#)

## リザーブインスタンスマーケットプレイスでの販売

リザーブインスタンスマーケットプレイスを使用すると、ニーズの変化に合わせて、余ったリザーブインスタンスを柔軟に販売することができます。例えば、インスタンスを別のアマゾンウェブサービス (AWS) リージョンに移動する、別のインスタンスタイプに変更する、リザーブインスタンスの期限が切れる前に終了するプロジェクトの処理能力を売却する、などが考えられます。(販売者になるための要件、予約した処理能力を販売できる時期など、いくつかの制限があります。リザーブインスタンスおよびリザーブインスタンスマーケットプレイスの制限と要件については、「[リザーブインスタンスの要件チェックリスト \(p. 288\)](#)」を参照してください。)

出品したリザーブインスタンスはすぐに、AWS の他のお客様が見ることのできる一覧に掲載されます。リザーブインスタンスは、出品されているインスタンスのタイプ、残存期間、時間料金に基づいてグループ化されます。このグループ化により、購入者は希望するリザーブインスタンスを容易に見つけることができます。お客様は、一覧からご自分の条件に最も合致するインスタンスを購入することを選択し、提示されている前払い価格と時間料金のバランスを判断します。

購入者のリクエストに対応するために、AWS は指定されたグループ内で前払い価格が最低のリザーブインスタンスを最初に販売します。次に、2 番目に低い価格のリザーブインスタンスを販売します。このようにして、購入者の購入注文すべてに対応するまで販売します。AWS はトランザクションを処理し、リザーブインスタンスの所有権を購入者に移します。販売者はリザーブインスタンスに対する現金の支払いを、振込によって直接各自の銀行口座で受け取ります。

リザーブインスタンスマーケットプレイスで販売を行うと、購入者の郵便番号と国情報が支払いレポートにより報告されます。この情報により、販売者は行政機関に支払う必要がある税金を計算できます。販売者としての商号も、購入者の購入請求に記載されます。AWS では、リザーブインスタンスマーケットプレイスでの販売に対して管理料 (前払い価格の総額の 12%) が課金されます。詳細については、[Amazon EC2 リザーブインスタンス製品ページ](#)の「[リザーブインスタンスマーケットプレイス](#)」を参照してください。

リザーブドインスタンスは、売却されるまでは、コントロール可能です。販売が成立すると、予約済みの処理能力と割引使用料金は失われます。予約済みの処理能力を売却した後も、インスタンスを使用し続けられますが、オンデマンド価格が請求されます。オンデマンド価格は、リザーブドインスタンスの売却時点から適用されます。したがって、使用するインスタンスに対するオンデマンド価格の課金を避けるには、別のリザーブドインスタンスを購入するか、販売する（または期限が切れる）ときにリザーブドインスタンスを終了します。

このトピックでは、リザーブドインスタンスマーケットプレイスでの販売手順を説明します。

- 「[販売者として登録する \(p. 262\)](#)」 – 販売者として登録し、米国に住所がある銀行を指定します。年間 200 件以上の取引計画がある場合、または 20,000 USD 相当のリザーブドインスタンスの販売計画がある場合、税金情報も提供する必要があります。
- 「[リザーブドインスタンスを販売する \(p. 266\)](#)」 – 有効期間が 1 か月以上あるアクティブなリザーブドインスタンスを出品します。また、1 か月よりも長い期間のリザーブドインスタンスを所有していなければなりません。
- 「[リザーブドインスタンスが売却された後 \(p. 286\)](#)」 – リザーブドインスタンスを販売する時期と、料金の受取方法を確認します。
- 「[クイックスタート: リザーブドインスタンスマーケットプレイスでの販売に関するビデオ \(p. 235\)](#)」 – リザーブドインスタンスマーケットプレイスですぐに販売を開始するために必要な情報を収集します。

リザーブドインスタンスの購入については、「[リザーブドインスタンスを購入する \(p. 236\)](#)」を参照してください。リザーブドインスタンスの基本的な情報については、「[リザーブドインスタンス \(p. 218\)](#)」を参照してください。

## リザーブドインスタンスマーケットプレイスでの販売に関する重要な注意

- リザーブドインスタンスマーケットプレイスで販売者になるためには、販売者として登録し、米国に住所がある銀行を指定する必要があります。詳細については、「[販売者として登録する \(p. 262\)](#)」を参照してください。
- リザーブドインスタンスを売却するには、そのインスタンスがアクティブであった期間が 30 日以上であることと、AWS が前払い料金を受領済みであることが必要です。
- 1 年間に 50,000 USD までのリザーブドインスタンスを販売できます。これよりも多くのリザーブドインスタンスを販売するには、「[Amazon EC2 インスタンス上限緩和申請](#)」フォームにご記入ください。
- 料金範囲に基づく割引価格で購入された Amazon EC2 リザーブドインスタンスは、リザーブドインスタンスマーケットプレイスで販売することはできません。割引価格範囲の詳細については、「[リザーブドインスタンス料金範囲を理解する \(p. 226\)](#)」を参照してください。

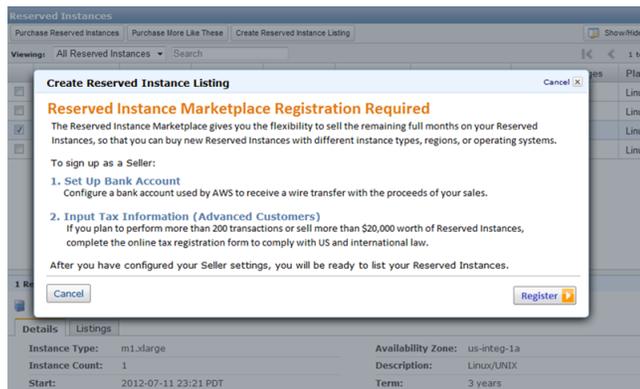
リザーブドインスタンスおよびリザーブドインスタンスマーケットプレイスを使用するための要件をまとめたチェックリストについては、「[リザーブドインスタンスの要件チェックリスト \(p. 288\)](#)」を参照してください。

## 販売者として登録する

### Topics

- [銀行 \(p. 263\)](#)
- [税金情報 \(p. 264\)](#)
- [販売者登録の確認 \(p. 265\)](#)
- [購入者との情報の共有 \(p. 265\)](#)
- [次のステップ \(p. 266\)](#)

リザーブドインスタンスマーケットプレイスで販売するための最初のタスクは、販売者として登録することです。



AWS アカウントをまだ作成していない場合は、リザーブドインスタンスマーケットプレイスへの登録の前に作成する必要があります。「[Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#)」で説明している手順を実行します。この手順では、Amazon EC2 アカウントの作成および認証情報に関する情報を提供します。

「[リザーブドインスタンスマーケットプレイス Seller Registration](#)」ウェブページから、登録処理を行うことができます。出品を作成しようとした時点でまだ登録されていない場合は、この販売者登録ページが表示されます。

登録とは、商号、銀行情報、納税識別番号を提供することです。通常、こうした情報を指定する必要がありますのは1回だけです。ただし、個人情報および銀行情報は「[リザーブドインスタンスマーケットプレイス Seller Registration](#)」ウェブページから更新できます。最初に登録したときに使用したアカウントを使って AWS にログインすると、個人情報と銀行情報のページに直接移動できます。

## 銀行

AWS では、リザーブドインスタンスの販売時に集金された金額をお支払いするために、お客様の銀行情報を必要とします。住所が米国内の銀行でなければなりません。

[Manage Bank Account] ページで、支払いを受け取る銀行に関する次の情報を提供します。

- 銀行口座の名義
- 支店コード

- アカウント番号
- 銀行口座の種類



#### Note

法人の銀行口座を使用する場合は、口座に関する情報を FAX ( 1-206-765-3424 ) で送信するように指示されます。

支払いを受け取るデフォルトの銀行口座は変更できます。最初に登録したときに使用したアカウントを使って [リザーブドインスタンスマーケットプレイス Seller Registration] ページにアクセスすると、個人情報と銀行情報のページに直接移動できます。

登録が完了すると、AWS が銀行口座を確認し、デフォルトの銀行として設定します。AWS が銀行の口座を確認するまで支払いを受け取ることはできません。銀行の確認は、最長 2 週間かかる場合があるため、新規口座の場合は、最長 2 週間の間、売却による支払いを受けられません。設定が済んでいる口座の場合、支払い完了まで通常およそ 2 日かかります。

## 税金情報

リザーブドインスタンスマーケットプレイス でのリザーブドインスタンスの販売には、消費税または付加価値税などの取引税がかかることがあります。取引関連の税金が適用されるかどうかについては、税務部、法務部、財務部、または経理部に確認する必要があります。お客様は、取引関連の税金を収集し、該当する税務署に納める役割を担います。

販売者登録手続きの一環として、Tax interview を行うことができます。次のいずれかに該当する場合は、この手続きを行うことをお勧めします。

- AWS でフォーム 1099-K を生成したい。
- リザーブドインスタンスについて、1 年間に 200 件を超える取引または 20,000 USD 以上の販売が予想される。1 件の取引には、1 つまたは複数のリザーブドインスタンスを含めることができます。登録時にこの手順をスキップし、後で取引が 199 件に達した場合、"You have reached the transaction limit for pre-tax. 次の URL で Tax Interview を記入してください。  
[http://portal.aws.amazon.com/ec2/ri/seller\\_registration?action=taxInterview](http://portal.aws.amazon.com/ec2/ri/seller_registration?action=taxInterview)
- お客様が米国以外の販売者である。この場合は、フォーム W-8BEN に電子的に記入する必要があります。

Tax interview を完了した場合、入力する税金情報は、お客様が米国の法人であるか米国外の法人であるかによって異なります。米国の販売者については、納税者番号の他に、事業所の連絡先情報を AWS に提供する必要があります。



税金登録手続きを完了すると、AWS によってフォーム 1099-K が提出され、税金口座がしきい値レベルに達した年の翌年の 1 月 31 日までに、米国の郵便によって上記フォームのコピーがお客様に届けられます。

Tax interview の記入を行う際は、以下に注意してください。

- AWS が提供する情報 ( このトピックの情報を含む ) は、税金、法律、またはその他の専門的なアドバイスではありません。IRS のレポート要件がビジネスに与える影響について情報が必要な場合、または他の疑問がある場合は、税金、法律、またはその他の専門家にお尋ねください。
- IRS のレポート要件をできるだけ効率的に満たすには、Tax interview の中で要求されたすべての質問に答え、情報を入力します。
- 答えを確認します。綴りを間違ったり、誤った税金識別番号を入力したりしないようにします。これらのミスがあると、誤った税金フォームが生成されます。

IRS の要件とフォーム 1099-K の詳細については、[IRS のウェブサイト](#)を参照してください。

## 販売者登録の確認

記入済みの販売者登録が受領されると、登録を確認する電子メールが届いて、リザーブドインスタンスマーケットプレイス での販売が可能になったことが伝えられます。

## 購入者との情報の共有

リザーブドインスタンスマーケットプレイス でリザーブドインスタンスを販売する際、AWS は米国の規制に準拠するために、お客様の正式な会社名を購入者のステートメントに示します。さらに、購入者から請求書またはその他の税金関連の理由について販売者に連絡したいとの要望が AWS カスタマーサービスにあった場合、AWS は、購入者が直接販売者に連絡できるよう販売者の E メールアドレスを購入者に提供する必要があります。

同様の理由で、販売者には購入者の郵便番号および国情報が支払いレポートによって提供されます。販売者として、この情報を、国に支払わなければならない取引税 ( 売上税や付加価値税など ) に添付する必要がある場合があります。

AWS は税金に関する助言を行うことはできませんが、お客様が追加情報を必要としているとお客様の税務専門家が判断した場合は、AWS カスタマーサポートにご連絡ください。

## 次のステップ

リザーブドインスタンスマーケットプレイスへの販売者としての登録が正常に完了したら、リザーブドインスタンスの販売を開始できます。リザーブドインスタンスマーケットプレイスで販売するには、次の手順を行う必要があります。

### 1. 販売するリザーブドインスタンスを決めます。

販売するアクティブなリザーブドインスタンスを特定し、売却するときの前払い価格を選択します。詳細については、「[リザーブドインスタンスの価格決定 \(p. 267\)](#)」を参照してください。

### 2. リザーブドインスタンスを出品します。

リザーブドインスタンスをリザーブドインスタンスマーケットプレイスのリストに含めます。詳細については、「[リザーブドインスタンスの出品 \(p. 267\)](#)」を参照してください。

### 3. 出品内容を確認します。

リザーブドインスタンスを監視して、出品内容を確認できます。詳細については、「[リザーブドインスタンスに関する情報を取得する \(p. 246\)](#)」を参照してください。

### 4. 出品内容のキャンセル、変更を行います。

出品内容を変更するには、まずキャンセルし、その後で再度出品します。詳細については、「[出品内容のキャンセル、変更を行います。 \(p. 276\)](#)」を参照してください。

リザーブドインスタンスの販売の詳細については、「[リザーブドインスタンスを販売する \(p. 266\)](#)」を参照してください。

## リザーブドインスタンスを販売する

### Topics

- [リザーブドインスタンスの価格決定 \(p. 267\)](#)
- [リザーブドインスタンスの出品 \(p. 267\)](#)
- [出品内容のキャンセル、変更を行います。 \(p. 276\)](#)

このセクションでは、リザーブドインスタンスマーケットプレイスにリザーブドインスタンスを出品して販売する方法について説明します。販売者として登録していない場合は、まず最初に登録する必要があります。詳細については、「[リザーブドインスタンスを販売する \(p. 266\)](#)」を参照してください。

登録済みの販売者の場合、販売するリザーブドインスタンスを1つまたは複数選択し、すべてを1件のリストにまとめて販売することができます。さらに、次の要件を満たしている場合は、インスタンスのタイプ、プラットフォーム、リージョン、アベイラビリティゾーンの設定に関係なく、すべてのタイプのリザーブドインスタンスを出品できます。

- 出品したリザーブドインスタンスの前払い料金が支払い済みであること。

つまり、出品できるのは、*Active* 状態のリザーブドインスタンスです。ただし、AWS が実際に前払い金を受け取る前でも、リザーブドインスタンスが *Active* な状態になっている可能性があることに注意してください。この場合、前払い金が支払われるまで、リザーブドインスタンスマーケットプレイスにリザーブドインスタンスを出品することはできません。

- 最低1か月以上、そのリザーブドインスタンスを保有していること。
- 出品するリザーブドインスタンスの残りの有効期間が1か月以上あること。

出品するリザーブドインスタンスの残りの期間は、切り捨てによって一番近い月数に変更されます。例えば、リザーブドインスタンスの残りの有効期間が9か月と13日の場合は、9か月分のリザーブドインスタンスを出品できます。

- 販売しようとするリザーブドインスタンスが、割引されたりプライベートのリザーブドインスタンスでないこと。これらのタイプのリザーブドインスタンスは、リザーブドインスタンスマーケットプレイスには出品できません。

リザーブドインスタンスおよびリザーブドインスタンスマーケットプレイスを使用するための要件をまとめたチェックリストについては、「[リザーブドインスタンスの要件チェックリスト \(p. 288\)](#)」を参照してください。

既存のリザーブドインスタンスの詳細情報は、次のいずれかのツールで取得できます。

- 「」の [EC2] コンソールの [Reserved Instances][AWS マネジメントコンソール \(p. 222\)](#) ページ。
- `ec2-describe-reserved-instances` CLI コマンド。
- `DescribeReservedInstances` API アクション。

リザーブドインスタンスの詳細については、「[リザーブドインスタンス \(p. 218\)](#)」を参照してください。

## リザーブドインスタンスの価格決定

リザーブドインスタンスマーケットプレイスの販売者の場合、リザーブドインスタンスに指定できるのは前払い料金のみです。前払い料金は、購入者がリザーブドインスタンスを購入する際に支払う一括払いの料金です。販売者は、使用料金や定期的な料金を指定することはできません。リザーブドインスタンスの購入者は、最初の購入時に設定された使用料または定期的な料金を支払うことになります。この使用料は、軽度使用 および 中度使用 リザーブドインスタンスに適用され、使用しているリザーブドインスタンス提供対象によって異なります。定期的な料金は 重度使用 に適用される時間あたりの料金で、リザーブドインスタンスを使用しているかどうかに関係なく支払いが発生します。使用率別の使用可能なリザーブドインスタンス製品の詳細については、「[使用計画に基づいてリザーブドインスタンスを選択する \(p. 225\)](#)」を参照してください。

### 価格決定スケジュールの設定

リザーブドインスタンスを販売する時期に基づいて、異なる前払い料金（価格）を設定できます。リザーブドインスタンスをただちに販売する場合の価格と、その後のタイミングでリザーブドインスタンスを販売した場合の価格を設定できます。リザーブドインスタンスの価値は時間が経つにつれて減少するため、AWSではデフォルトで、直線的に減少するような価格設定を行っています。つまり、価格は、1月ごとに同じ割合で低下します。価格の設定方法は、複数の中から選ぶことができます。

例えば、リザーブドインスタンスの期間が9か月残っている場合、残余期間9か月のリザーブドインスタンスに対して販売者が受け取る金額を指定することも、残余期間が5か月の場合、さらに1か月の場合の金額を指定することもできます。

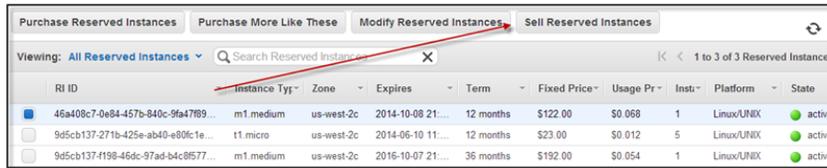
## リザーブドインスタンスの出品

AWS Management Console、Amazon EC2 CLI、または Amazon EC2 API を使用して、リザーブドインスタンスマーケットプレイスで販売するリザーブドインスタンスを出品できます。

### AWS マネジメントコンソール

リザーブドインスタンスマーケットプレイスでリザーブドインスタンスを出品するには

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. [Navigation] ペインで [Reserved Instances] をクリックします。  
  
[Reserved Instances] ページに、アカウントのインスタンスの一覧が表示されます。
3. マーケットプレイスに出品するリザーブドインスタンスを選択し、[Sell Reserved Instances] をクリックします。



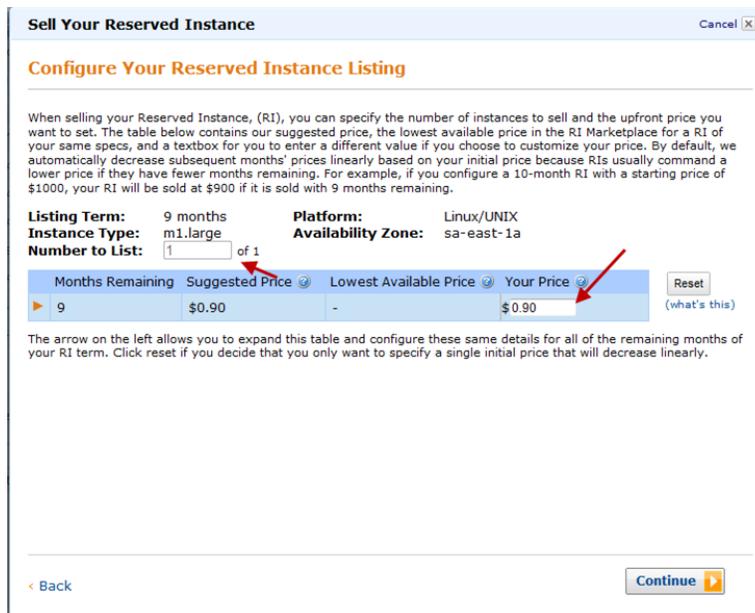
RI ID	Instance Type	Zone	Expires	Term	Fixed Price	Usage Pr	Instr	Platform	State
46a408c7-0e84-457b-940c-9fa47789...	m1.medium	us-west-2c	2014-10-08 21:...	12 months	\$122.00	\$0.068	1	Linux/UNIX	active
9d5cb137-271b-425e-ab09-e8061e...	t1.micro	us-west-2c	2014-06-10 11:...	12 months	\$23.00	\$0.012	5	Linux/UNIX	active
9d5cb137-f198-46dc-97ad-b4c8577...	m1.medium	us-west-2c	2016-10-07 21:...	36 months	\$192.00	\$0.054	1	Linux/UNIX	active



### Note

リザーブドインスタンスマーケットプレイス 販売者の登録処理が完了していない場合は、ここで完了するように要求されます。販売者の登録処理の詳細については、「[販売者として登録する \(p. 262\)](#)」を参照してください。

4. [Configure Your Reserved Instance Listing] ページの [Number to List] で、販売するインスタンスの数を設定し、[Your Price] で、残余期間に対する前払い価格を設定します。



**Sell Your Reserved Instance** Cancel X

### Configure Your Reserved Instance Listing

When selling your Reserved Instance, (RI), you can specify the number of instances to sell and the upfront price you want to set. The table below contains our suggested price, the lowest available price in the RI Marketplace for a RI of your same specs, and a textbox for you to enter a different value if you choose to customize your price. By default, we automatically decrease subsequent months' prices linearly based on your initial price because RIs usually command a lower price if they have fewer months remaining. For example, if you configure a 10-month RI with a starting price of \$1000, your RI will be sold at \$900 if it is sold with 9 months remaining.

**Listing Term:** 9 months      **Platform:** Linux/UNIX  
**Instance Type:** m1.large      **Availability Zone:** sa-east-1a

**Number to List:** 1 of 1

Months Remaining	Suggested Price	Lowest Available Price	Your Price
▶ 9	\$0.90	-	\$0.90

The arrow on the left allows you to expand this table and configure these same details for all of the remaining months of your RI term. Click reset if you decide that you only want to specify a single initial price that will decrease linearly.

< Back Continue ▶

[Months Remaining] 列の左にある矢印をクリックすると、残りの有効期間の変化によるリザーブドインスタンスの価値の変化の様子を確認できます。AWS のデフォルトでは、価格が直線形で低下するように設定されています。つまり、価格は毎月同じ額ずつ低下していきます。

5. 上級ユーザーが価格をカスタマイズする場合は、今後の月に異なる価格を入力できます。デフォルトの直線形の価格減少に戻すには、[Reset] をクリックします。出品の設定が終了したら、[Continue] をクリックします。

**Sell Your Reserved Instance** Cancel X

### Configure Your Reserved Instance Listing

When selling your Reserved Instance (RI), you can specify the number of instances to sell and the upfront price you want to set. The table below contains our suggested price, the lowest available price in the RI Marketplace for a RI of your same specs, and a textbox for you to enter a different value if you choose to customize your price. By default, we automatically decrease subsequent months' prices linearly based on your initial price because RIs usually command a lower price if they have fewer months remaining. For example, if you configure a 10-month RI with a starting price of \$1,000, your RI will be sold at \$900 if it is sold with 9 months remaining.

**Listing Term:** 9 months      **Platform:** Linux/UNIX  
**Instance Type:** m1.large      **Availability Zone:** sa-east-1a  
**Number to List:** 1 of 1

Months Remaining	Suggested Price	Lowest Available Price	Your Price
9 months	0.90	-	\$0.90
8 months	0.80	-	\$0.80
7 months	0.70	-	\$0.70
6 months	0.60	-	\$0.60
5 months	0.50	-	\$0.50
4 months	0.40	-	\$0.40
3 months	0.30	-	\$0.30

Reset (what's this)

The arrow on the left allows you to expand this table and configure these same details for all of the remaining months of your RI term. Click reset if you decide that you only want to specify a single initial price that will decrease linearly.

< Back Continue >

6. [Confirm Your Reserved Instance Listing] 画面に表示された出品詳細に問題がなければ、[List Reserved Instance] をクリックします。

出品が処理されたことの確認が通知されます。

**Sell Your Reserved Instance** Cancel X

It may take a few minutes before your Reserved Instance listing changes from pending to active. When active, your Reserved Instance will appear in the purchase search results grouped by its unique attributes, such as Availability Zone. You can use your Reserved Instance until it is sold. When your Reserved Instance listing sells, any running instance will now be charged at the On-Demand rate.

**Interested in Purchasing New Reserved Instances?**

**Find the Optimal Reserved Instance**  
Visit the Reserved Instance Marketplace to purchase a new Reserved Instance to replace the one you are selling. As usual, you can simply make a low, one-time payment to reserve compute capacity, and in turn, receive a significant discount on the hourly charge for that instance.

[Learn more about Reserved Instances](#)

**Right Sizing Your Instances**  
Are you fully utilizing your instance's CPU? Choosing the correct instance sizes to meet your needs is a great way to reduce your costs. We provide free tools to monitor the CPU for your instances.

[Go to CloudWatch to see your utilization](#)

Close >

7. リザーブドインスタンスの出品の詳細を表示するには、[Reserved Instances] ページで表示されるリザーブドインスタンスを選択し、[My Listings] タブをクリックします。

Viewing: All Reserved Instances

RI ID	Instance Type	Zone	Term	Offering Type	Fixed Price	Usage Price	Instance Count	Recurring Charges	State
46a408c7-08cf-4aec-b68e-19af54725f29	m1.micro	us-east-1b	12 months	Light Utilization	\$2.0	\$0.012	1	-	reserved
46a408c7-9c23-434a-...	m1.micro	us-east-1d	8 months	Medium Utilization	\$0.88	\$0.006	1	-	active
9d5c137-7251-4a22-...	m1.micro	us-east-1d	8 months	Medium Utilization	\$1.96	\$0.006	1	-	active

1 Reserved Instance selected

Reserved Instances: 46a408c7-08cf-4aec-b68e-19af54725f29

Details: **My Listings**

Listings: All

Listing State: Closed

Inventory Details	Listing Details
Available: 0	Listing Term: -
Pending: 0	Upfront Listing Price: -
Sold: 1	Competitive Pricing Benchmark:
Cancelled: 0	Suggested Price: -
Total: 1	Lowest Comparable Market Listing: -

## Amazon EC2 CLI

リザーブドインスタンスマーケットプレイス でリザーブドインスタンスを出品するには

1. `ec2-describe-reserved-instances` を呼び出して、リザーブドインスタンスのリストを取得します。

```
PROMPT> ec2-describe-reserved-instances --headers
```

次のような出力が Amazon EC2 から返されます。

```
PROMPT> ec2-describe-reserved-instances --headers
Type ReservedInstancesId AvailabilityZone InstanceType ProductDescription
Duration FixedPrice UsagePrice InstanceCount Start State Currency Instan
ceTenancy OfferingType
RESERVEDINSTANCES f127bd27-9f30-41d3-bf45-9af45example sa-east-1a m1.large
Linux/UNIX 10m 1.0 0.0 1 2012-08-22T21:41:51+0000 active USD default Medium
Utilization
RESERVEDINSTANCES 1ba8e2e3-d20d-44ec-b202-fcb6aexample sa-east-1b m1.small
Linux/UNIX 10m 1.2 0.032 3 2012-08-21T14:02:00+0000 retired USD default
Medium Utilization
RESERVEDINSTANCES 4357912c-6f69-4966-a407-6f0cbexample sa-east-1b m1.small
Linux/UNIX 10m 1.2 0.032 3 2012-08-21T14:02:00+0000 active USD default
Medium Utilization
RESERVEDINSTANCES 4357912c-d032-4a97-9b49-5eb3aexample sa-east-1b m1.small
Linux/UNIX 10m 1.2 0.032 1 2012-08-21T14:02:00+0000 retired USD default
Medium Utilization
...
```

リザーブドインスタンスマーケットプレイス に出品するリザーブドインスタンスのリザーブドインスタンス ID を選択します。

2. 出品するリザーブドインスタンスのリザーブドインスタンス ID を指定し、`ec2-create-reserved-instances-listing` を呼び出します。以下の必須パラメータを指定する必要があります。

- リザーブドインスタンス ID
- インスタンス数
- MONTH:PRICE

コマンドは、次の例のようになります。

```
PROMPT> ec2-create-reserved-instances-listing --reserved-instance b847fa93-
c736-4eae-bca1-3147example --instance-count 1 05:01.20 04:01.00 01:00.75 -
-headers
```

次のような出力が Amazon EC2 から返されます。

```
PROMPT> LISTING 2a0ff720-f62e-4824-8ed1-7dd0aexample b847fa93-c736-4eae-bca1-
e3147example Wed Aug 29 13:59:11 PDT 2012 Wed Aug 29 13:59:11 PDT 2012 active
active
INSTANCE-COUNT available 1
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 0
```

```
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.0
PRICE-SCHEDULE 3 $1.0
PRICE-SCHEDULE 2 $1.0
PRICE-SCHEDULE 1 $0.75
```

3. リザーブドインスタンスの出品の詳細を表示するには、出品 ID `ec2-describe-reserved-instances-listings` を使用して `095c0e18-c9e6-4692-97e5-653e0example` を実行します。

```
PROMPT> ec2-describe-reserved-instances-listings 095c0e18-c9e6-4692-97e5-653e0example
```

次のような出力が Amazon EC2 から返されます。

```
PROMPT> ec2-describe-reserved-instances-listings 095c0e18-c9e6-4692-97e5-653e0example
Type ReservedInstancesListingId ReservedInstancesId CreateDate UpdateDate
Status StatusMessage
LISTING 095c0e18-c9e6-4692-97e5-653e0example b847fa93-c736-4eae-bca1-e3147example Tue Aug 28 18:21:07 PDT 2012 Tue Aug 28 18:21:07 PDT 2012 active
active
INSTANCE-COUNT available 1
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.2
PRICE-SCHEDULE 3 $1.2
PRICE-SCHEDULE 2 $1.2
PRICE-SCHEDULE 1 $1.2
```

## Amazon EC2 API

リザーブドインスタンスマーケットプレイスでリザーブドインスタンスを出品するには

1. `DescribeReservedInstances` を呼び出して、リザーブドインスタンスのリストを取得します。

```
https://ec2.amazonaws.com/
?Action=DescribeReservedInstances
&AUTHPARAMS
```

以下に、応答の例を示します。

```
<DescribeReservedInstancesResponse xmlns='http://ec2.amazonaws.com/doc/2012-08-15/'>
  <requestId>ebe3410a-8f37-441d-ae11-2e78eexample</requestId>
  <reservedInstancesSet>
    <item>
      <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</reservedInstancesId>
```

```
<instanceType>m1.large</instanceType>
<availabilityZone>us-east-1a</availabilityZone>
<start>2012-08-07T15:19:31.071Z</start>
<duration>31536000</duration>
<fixedPrice>276.0</fixedPrice>
<usagePrice>0.156</usagePrice>
<instanceCount>5</instanceCount>
<productDescription>Linux/UNIX</productDescription>
<state>active</state>
<instanceTenancy>default</instanceTenancy>
<currencyCode>USD</currencyCode>
<offeringType>Light Utilization</offeringType>
<recurringCharges/>
</item>
</reservedInstancesSet>
</DescribeReservedInstancesResponse>
```

リザーブドインスタンスマーケットプレイス に出品するリザーブドインスタンスのリザーブドインスタンス ID を書き留めておきます。

- リザーブドインスタンス ID *f127bd27-cee4-443a-a76b-a5af9example* から 3 つのリザーブドインスタンスの出品を作成し、次のように価格体系を指定します。

期間 ( 残余月数 )	11	10	9	8	7	6	5	4	3	2	1
期間ごとに指定した価格	2.5			2.0			1.5		0.7		0.1
価格	2.5	2.5	2.5	2.0	2.0	2.0	1.5	1.5	0.7	0.7	0.1

呼び出しは、次の例のようになります。

```
https://ec2.amazonaws.com/?Action=CreateReservedInstancesListing
&ClientToken=myIdempToken1
&ReservedInstancesId=f127bd27-cee4-443a-a76b-a5af9example
&InstanceCount=3
&PriceSchedules.0.Price=2.5&PriceSchedules.0.Term=11
&PriceSchedules.1.Price=2.0&PriceSchedules.1.Term=8
&PriceSchedules.2.Price=1.5&PriceSchedules.2.Term=5
&PriceSchedules.3.Price=0.7&PriceSchedules.3.Term=3
&PriceSchedules.4.Price=0.1&PriceSchedules.4.Term=1
&AUTHPARAMS
```

以下に、応答の例を示します。

```
<CreateReservedInstancesListingResponse>
<requestId>a42481af-335a-4e9e-b291-bd18dexample</requestId>
<reservedInstancesListingsSet>
  <item>
    <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-9e57dexample</reservedInstancesListingId>
    <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</reservedInstancesId>
    <createDate>2012-08-30T17:11:09.449Z</createDate>
```

```
<updateDate>2012-08-30T17:11:09.468Z</updateDate>
<status>active</status>
<statusMessage>active</statusMessage>
<instanceCounts>
  <item>
    <state>Available</state>
    <instanceCount>3</instanceCount>
  </item>
  <item>
    <state>Sold</state>
    <instanceCount>0</instanceCount>
  </item>
  <item>
    <state>Cancelled</state>
    <instanceCount>0</instanceCount>
  </item>
  <item>
    <state>Pending</state>
    <instanceCount>0</instanceCount>
  </item>
</instanceCounts>
<priceSchedules>
  <item>
    <term>11</term>
    <price>2.5</price>
    <currencyCode>USD</currencyCode>
    <active>true</active>
  </item>
  <item>
    <term>10</term>
    <price>2.5</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
  <item>
    <term>9</term>
    <price>2.5</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
  <item>
    <term>8</term>
    <price>2.00</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
  <item>
    <term>7</term>
    <price>2.0</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
  <item>
    <term>6</term>
    <price>2.0</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
</priceSchedules>
```

```
<item>
  <term>5</term>
  <price>1.5</price>
  <currencyCode>USD</currencyCode>
  <active>>false</active>
</item>
<item>
  <term>4</term>
  <price>1.5</price>
  <currencyCode>USD</currencyCode>
  <active>>false</active>
</item>
<item>
  <term>3</term>
  <price>0.7</price>
  <currencyCode>USD</currencyCode>
  <active>>false</active>
</item>
<item>
  <term>2</term>
  <price>0.7</price>
  <currencyCode>USD</currencyCode>
  <active>>false</active>
</item>
<item>
  <term>1</term>
  <price>0.1</price>
  <currencyCode>USD</currencyCode>
  <active>>false</active>
</item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</CreateReservedInstancesListingResponse>
```

- リザーブドインスタンスの出品の詳細を表示するには、DescribeReservedInstancesListings を実行します。

コマンドは、次の例のようになります。

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstancesListings
&AUTHPARAMS
```

以下に、応答の例を示します。

```
<DescribeReservedInstancesListingsResponse>
  <requestId>cec5c904-8f3a-4de5-8f5a-ff7f9example</requestId>
  <reservedInstancesListingsSet>
    <item>
      <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
      <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re
servedInstancesId>
      <createDate>2012-08-30T17:11:09.449Z</createDate>
```

```
<updateDate>2012-08-30T17:11:09.468Z</updateDate>
<status>active</status>
<statusMessage>active</statusMessage>
<instanceCounts>
  <item>
    <state>Available</state>
    <instanceCount>3</instanceCount>
  </item>
  <item>
    <state>Sold</state>
    <instanceCount>0</instanceCount>
  </item>
  <item>
    <state>Cancelled</state>
    <instanceCount>0</instanceCount>
  </item>
  <item>
    <state>Pending</state>
    <instanceCount>0</instanceCount>
  </item>
</instanceCounts>
<priceSchedules>
  <item>
    <term>11</term>
    <price>2.5</price>
    <currencyCode>USD</currencyCode>
    <active>true</active>
  </item>
  <item>
    <term>10</term>
    <price>2.5</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
  <item>
    <term>9</term>
    <price>2.5</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
  <item>
    <term>8</term>
    <price>2.0</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
  <item>
    <term>7</term>
    <price>2.0</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
  <item>
    <term>6</term>
    <price>2.0</price>
    <currencyCode>USD</currencyCode>
    <active>false</active>
  </item>
</priceSchedules>
```

```
<item>
  <term>5</term>
  <price>1.5</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>4</term>
  <price>1.5</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>3</term>
  <price>0.7</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>2</term>
  <price>0.7</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>1</term>
  <price>0.1</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

## 出品内容のキャンセル、変更を行います。

リザーブドインスタンスマーケットプレイスにリザーブドインスタンスを出品した後、[AWS Management Console \(p. 276\)](#)、[Amazon EC2 CLI \(p. 277\)](#)、または [Amazon EC2 API \(p. 279\)](#) を使用してその出品を管理することができます。出品の詳細を表示できます。購入されていないインスタンスの出品については、キャンセルすることもできます。

現時点では、出品内容を直接変更することはできません。ただし、最初に出品をキャンセルしてから、新しいパラメータで別の出品を作成することはできます。

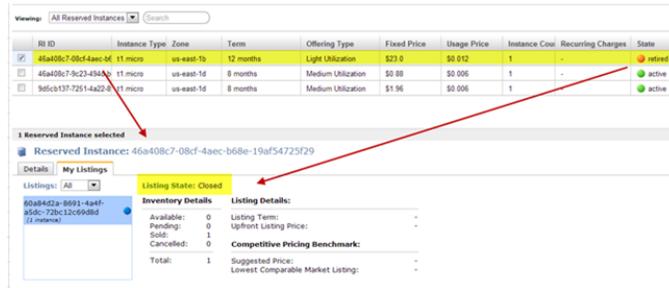
このセクションでは、リザーブドインスタンスマーケットプレイスの出品を、コンソール、CLI、および API を使用して表示、キャンセル、変更するタスクの実施方法について説明します。

### AWS Management Console

出品したインスタンスを表示するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. [Navigation] ペインで [Reserved Instances] をクリックします。

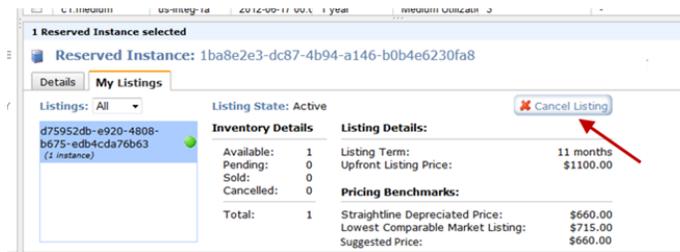
3. 自分のリザーブドインスタンスを右クリックします。
4. [My Listings] タブをクリックします。選択したリザーブドインスタンスの詳細が表示されます。



### 出品をキャンセルするには

出品は、Active 状態であればいつでもキャンセルできます。既にマッチングされていたり、販売処理が行われている出品はキャンセルできません。出品したインスタンスの一部がマッチングされている場合にその出品をキャンセルすると、マッチングされていない残りのインスタンスが出品から削除されます。

1. Amazon EC2 コンソールの [Reserved Instances] ページに移動し、リザーブドインスタンスを右クリックします。
2. [My Listings] タブで、[Cancel Listing] をクリックします。



### 出品を変更するには

現時点では、出品内容を直接変更することはできません。ただし、最初に出品をキャンセルしてから、新しいパラメータで別の出品を作成することはできます。

1. アクティブなリザーブドインスタンスの出品をキャンセルします。詳細については、前述の手順を参照してください。
2. 新しい出品を作成します。詳細については、「[リザーブドインスタンスの出品 \(p. 267\)](#)」を参照してください。

### Amazon EC2 CLI

#### 出品したインスタンスを表示するには

- `ec2-describe-reserved-instances-listing` を実行して、出品の詳細を取得します。

```
PROMPT> ec2-describe-reserved-instances-listings
```

次のような出力が Amazon EC2 から返されます。

```
PROMPT> ec2-describe-reserved-instances-listings
Type ReservedInstancesListingId ReservedInstancesId CreateDate UpdateDate
Status StatusMessage
LISTING 615d8a10-8224-4c19-ba7d-b9aa0example 1ba8e2e3-d20d-44ec-b202-
fcb6aexample Wed Aug 22 09:02:58 PDT 2012 Wed Aug 22 14:24:26 PDT 2012 can
celled cancelled
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 1
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 10 $1.2
PRICE-SCHEDULE 9 $1.08
PRICE-SCHEDULE 8 $0.96
PRICE-SCHEDULE 7 $0.84
PRICE-SCHEDULE 6 $0.72
PRICE-SCHEDULE 5 $0.6
PRICE-SCHEDULE 4 $0.48
PRICE-SCHEDULE 3 $0.36
PRICE-SCHEDULE 2 $0.24
PRICE-SCHEDULE 1 $0.12
LISTING d5fa5166-83c3-40e4-abb2-b7298example 1ba8e2e3-d20d-44ec-b202-
fcb6aexample Wed Aug 22 14:31:55 PDT 2012 Wed Aug 22 14:42:40 PDT 2012 closed
closed
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 1
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 10 $0.9
PRICE-SCHEDULE 9 $0.81
PRICE-SCHEDULE 8 $0.72
PRICE-SCHEDULE 7 $0.63
PRICE-SCHEDULE 6 $0.54
PRICE-SCHEDULE 5 $0.45
PRICE-SCHEDULE 4 $0.36
PRICE-SCHEDULE 3 $0.27
PRICE-SCHEDULE 2 $0.18
PRICE-SCHEDULE 1 $0.09
....
LISTING 095c0e18-c9e6-4692-97e5-653e0example b847fa93-c736-4eae-bca1-
e3147example Tue Aug 28 18:21:07 PDT 2012 Tue Aug 28 18:21:07 PDT 2012 active
active
INSTANCE-COUNT available 1
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 0
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.2
PRICE-SCHEDULE 3 $1.2
PRICE-SCHEDULE 2 $1.2
PRICE-SCHEDULE 1 $1.2
```

## 出品をキャンセルするには

出品は、*Active* 状態であればいつでもキャンセルできます。既にマッチングされていたり、販売処理が行われている出品はキャンセルできません。

- `ec2-cancel-reserved-instances-listing` を実行して出品をキャンセルします。

```
PROMPT> ec2-cancel-reserved-instances-listing 095c0e18-c9e6-4692-97e5-653e0example
```

次のような出力が Amazon EC2 から返されます。

```
PROMPT> ec2-cancel-reserved-instances-listing
Type ReservedInstancesListingId ReservedInstancesId CreateDate UpdateDate
Status StatusMessage
LISTING 095c0e18-c9e6-4692-97e5-653e0example b847fa93-c736-4eae-bca1-
e3147example Tue Aug 28 18:21:07 PDT 2012 Tue Aug 28 18:21:07 PDT 2012 can
celled cancelled
INSTANCE-COUNT available 0
INSTANCE-COUNT sold 0
INSTANCE-COUNT cancelled 1
INSTANCE-COUNT pending 0
PRICE-SCHEDULE 5 $1.2
PRICE-SCHEDULE 4 $1.2
PRICE-SCHEDULE 3 $1.2
PRICE-SCHEDULE 2 $1.2
PRICE-SCHEDULE 1 $1.2
```

## 出品を変更するには

現時点では、出品内容を直接変更することはできません。ただし、最初に出品をキャンセルしてから、新しいパラメータで別の出品を作成することはできます。

1. アクティブなリザーブドインスタンスの出品をキャンセルします。詳細については、前述の手順を参照してください。
2. 新しい出品を作成します。詳細については、「[リザーブドインスタンスの出品 \(p. 267\)](#)」を参照してください。

## Amazon EC2 API

### 出品したインスタンスを表示するには

- `DescribeReservedInstancesListings` を呼び出して出品の詳細を取得します。

コマンドは、次の例のようになります。

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstancesListings
&AUTHPARAMS
```

以下に、応答の例を示します。

```
<DescribeReservedInstancesListingsResponse>
  <requestId>cec5c904-8f3a-4de5-8f5a-ff7f9example</requestId>
```

```
<reservedInstancesListingsSet>
  <item>
    <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
    <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re
servedInstancesId>
    <createDate>2012-08-30T17:11:09.449Z</createDate>
    <updateDate>2012-08-30T21:00:42.300Z</updateDate>
    <status>active</status>
    <statusMessage>active</statusMessage>
    <instanceCounts>
      <item>
        <state>Available</state>
        <instanceCount>2</instanceCount>
      </item>
      <item>
        <state>Sold</state>
        <instanceCount>1</instanceCount>
      </item>
      <item>
        <state>Cancelled</state>
        <instanceCount>0</instanceCount>
      </item>
      <item>
        <state>Pending</state>
        <instanceCount>0</instanceCount>
      </item>
    </instanceCounts>
    <priceSchedules>
      <item>
        <term>11</term>
        <price>2.5</price>
        <currencyCode>USD</currencyCode>
        <active>true</active>
      </item>
      <item>
        <term>10</term>
        <price>2.5</price>
        <currencyCode>USD</currencyCode>
        <active>false</active>
      </item>
      <item>
        <term>9</term>
        <price>2.5</price>
        <currencyCode>USD</currencyCode>
        <active>false</active>
      </item>
      <item>
        <term>8</term>
        <price>2.0</price>
        <currencyCode>USD</currencyCode>
        <active>false</active>
      </item>
      <item>
        <term>7</term>
        <price>2.0</price>
        <currencyCode>USD</currencyCode>
        <active>false</active>
      </item>
    </priceSchedules>
  </item>
</reservedInstancesListingsSet>
```

```
</item>
<item>
  <term>6</term>
  <price>2.0</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>5</term>
  <price>1.5</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>4</term>
  <price>1.5</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>3</term>
  <price>0.7</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>2</term>
  <price>0.7</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
<item>
  <term>1</term>
  <price>0.1</price>
  <currencyCode>USD</currencyCode>
  <active>false</active>
</item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

### 出品をキャンセルするには

出品は、Active状態であればいつでもキャンセルできます。既にマッチングされていたり、販売処理が行われている出品はキャンセルできません。

- CancelReservedInstancesListing を実行して出品されている `5ec28771-05ff-4b9b-aa31-9e57dexample` をキャンセルし、リザーブドインスタンスマーケットプレイス から削除します。

コマンドは、次の例のようになります。

```
https://ec2.amazonaws.com/?Action=CancelReservedInstancesListing
&ReservedInstancesListingId.0=5ec28771-05ff-4b9b-aa31-9e57dexample
&AUTHPARAMS
```

以下に、応答の例を示します。

```
<CancelReservedInstancesListingResponse>
  <requestId>bec2cf62-98ef-434a-8a15-886fcexample</requestId>
  <reservedInstancesListingsSet>
    <item>
      <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
      <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re
servedInstancesId>
      <createDate>2012-08-30T17:11:09.449Z</createDate>
      <updateDate>2012-08-31T14:12:23.468Z</updateDate>
      <status>cancelled</status>
      <statusMessage>cancelled</statusMessage>
      <instanceCounts>
        <item>
          <state>Available</state>
          <instanceCount>0</instanceCount>
        </item>
        <item>
          <state>Sold</state>
          <instanceCount>1</instanceCount>
        </item>
        <item>
          <state>Cancelled</state>
          <instanceCount>2</instanceCount>
        </item>
        <item>
          <state>Pending</state>
          <instanceCount>0</instanceCount>
        </item>
      </instanceCounts>
      <priceSchedules>
        <item>
          <term>11</term>
          <price>2.5</price>
          <currencyCode>USD</currencyCode>
          <active>false</active>
        </item>
        <item>
          <term>10</term>
          <price>2.5</price>
          <currencyCode>USD</currencyCode>
          <active>false</active>
        </item>
        <item>
          <term>9</term>
          <price>2.5</price>
          <currencyCode>USD</currencyCode>
          <active>false</active>
        </item>
        <item>
          <term>8</term>
```

```
        <price>2.0</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>7</term>
        <price>2.0</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>6</term>
        <price>2.0</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>5</term>
        <price>1.5</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>4</term>
        <price>1.5</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>3</term>
        <price>0.7</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>2</term>
        <price>0.7</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>1</term>
        <price>0.1</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</CancelReservedInstancesListingResponse>
```

キャンセルリクエストに対するレスポンスで、出品がキャンセルされたことが分かります。キャンセルした出品を確認する別の方法は、DescribeReservedInstancesListings を呼び出すことです。リクエストは、次の例のようになります。

```
http://ec2.amazonaws.com/?Action=DescribeReservedInstancesListings
&AUTHPARAMS
```

レスポンスは、次の例のようになります。

```
<DescribeReservedInstancesListingsResponse>
  <requestId>bec2cf62-98ef-434a-8a15-367c0example</requestId>
  <reservedInstancesListingsSet>
    <item>
      <reservedInstancesListingId>5ec28771-05ff-4b9b-aa31-
9e57dexample</reservedInstancesListingId>
      <reservedInstancesId>f127bd27-cee4-443a-a76b-a5af9example</re
servedInstancesId>
      <createDate>2012-08-30T17:11:09.449Z</createDate>
      <updateDate>2012-08-31T14:12:23.468Z</updateDate>
      <status>cancelled</status>
      <statusMessage>cancelled</statusMessage>
      <instanceCounts>
        <item>
          <state>Available</state>
          <instanceCount>0</instanceCount>
        </item>
        <item>
          <state>Sold</state>
          <instanceCount>1</instanceCount>
        </item>
        <item>
          <state>Cancelled</state>
          <instanceCount>2</instanceCount>
        </item>
        <item>
          <state>Pending</state>
          <instanceCount>0</instanceCount>
        </item>
      </instanceCounts>
      <priceSchedules>
        <item>
          <term>11</term>
          <price>2.5</price>
          <currencyCode>USD</currencyCode>
          <active>false</active>
        </item>
        <item>
          <term>10</term>
          <price>2.5</price>
          <currencyCode>USD</currencyCode>
          <active>false</active>
        </item>
        <item>
          <term>9</term>
          <price>2.5</price>
          <currencyCode>USD</currencyCode>
          <active>false</active>
        </item>
        <item>
          <term>8</term>
          <price>2.0</price>
        </item>
      </priceSchedules>
    </item>
  </reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

```
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>7</term>
        <price>2.0</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>6</term>
        <price>2.0</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>5</term>
        <price>1.5</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>4</term>
        <price>1.5</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>3</term>
        <price>0.7</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>2</term>
        <price>0.7</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
    <item>
        <term>1</term>
        <price>0.1</price>
        <currencyCode>USD</currencyCode>
        <active>>false</active>
    </item>
</priceSchedules>
<tagSet/>
<clientToken>listRI1</clientToken>
</item>
</reservedInstancesListingsSet>
</DescribeReservedInstancesListingsResponse>
```

## 出品を変更するには

現時点では、出品内容を直接変更することはできません。ただし、最初に出品をキャンセルしてから、新しいパラメータで別の出品を作成することはできます。

1. アクティブなリザーブドインスタンスの出品をキャンセルします。詳細については、前述の手順を参照してください。
2. 新しい出品を作成します。詳細については、「[リザーブドインスタンスの出品 \(p. 267\)](#)」を参照してください。

## リザーブドインスタンスが売却された後

### Topics

- [リザーブドインスタンスの出品状態 \(p. 286\)](#)
- [出品のライフサイクル \(p. 286\)](#)
- [支払いを受け取る \(p. 287\)](#)
- [通知 \(p. 287\)](#)

リザーブドインスタンスが売却されると、AWS から E メール の通知が送信されます。何らかのアクティビティ ( 出品の作成、出品の販売、または AWS から口座への送金など ) がある日は、その日のすべてのアクティビティを記載した E メール の通知が送信されます。詳細については、「[通知 \(p. 287\)](#)」を参照してください。

Amazon EC2 コンソールの [Reserved Instance] ページで選択したリザーブドインスタンスの [My Listings] タブで、リザーブドインスタンスの出品の状態を追跡できます。タブには、[Listing State] の他、期間、出品価格、および出品されているインスタンスの中で使用可能、売却済み、およびキャンセルされたものがいくつかあるかの詳細といった情報が示されます。また、`ec2-describe-reserved-instances-listings` CLI コマンドまたは `DescribeReservedInstancesListings` API 呼び出しも使用でき、適切なフィルタを使用することにより、リザーブドインスタンスの出品に関する情報を取得できます。

## リザーブドインスタンスの出品状態

[Listing State] には、リザーブドインスタンスの出品の現在の状態が表示されます。

- [Active] – 購入できます。
- [Cancelled] – 出品がキャンセルされ、マーケットプレイスでの購入ができません。
- [Closed] – 出品されていません。リザーブドインスタンスは、出品が完了したために [Closed] になっている可能性があります。

[Listing State] に表示される情報は、リザーブドインスタンスマーケットプレイス へのお客様の出品の状態に関するものです。これは、[Reserved Instance] ページの [State] 列に表示される状態情報とは異なります。この [State] 情報は、お客様のリザーブドインスタンスに関するものです。詳細については、「[リザーブドインスタンスの状態 \(p. 246\)](#)」を参照してください。

## 出品のライフサイクル

出品を作成したので、出品したものが売れたときに何が起きるかを見てみましょう。

出品したすべてのインスタンスがマッチングされて売れると、[My Listings] タブに表示される [Total instance count] が [Sold] の下に表示された数と同じになり、出品で残っている [Available] インスタンスがなくなり、[Status] が [closed] になります。

出品の一部だけが売れた場合、AWS は出品されているリザーブドインスタンスを取り下げ、残りのリザーブドインスタンスと同数のリザーブドインスタンスを作成します。したがって、リザーブドインスタンス ID とその ID の出品は、インスタンス数が少なくなっていますがアクティブのままです。

この出品内の売却はこの後、同様にして行われます。出品内のすべてのリザーブドインスタンスが売れると、この出品は `[closed]` とマークされます。

例えば、インスタンス数が 5 のリザーブドインスタンス ID `5ec28771-05ff-4b9b-aa31-9e57dexample` の出品を作成したとします。

Amazon EC2 コンソールの [Reserved Instance] ページの [My Listings] タブに、次のように出品が表示されます。

*リザーブドインスタンスの出品 ID 5ec28771-05ff-4b9b-aa31-9e57dexample*

- Total instance count = 5
- Sold = 0
- Available = 5
- Status = active

購入者がインスタンスを 2 つ購入したとします。まだ 3 つのインスタンスが販売中のままです。一部分が売れたため、AWS では引き続き販売中の残りの 3 つに相当する、インスタンスが 3 つある新しいリザーブドインスタンスが作成されます。

お客様の出品は、[My Listings] タブで次のように表示されます。

*リザーブドインスタンスの出品 ID 5ec28771-05ff-4b9b-aa31-9e57dexample*

- Total instance count = 5
- Sold = 2
- Available = 3
- Status = active

出品をキャンセルする場合、その一部が既に売れているとき、売却済みの部分についてのキャンセルは無効です。まだ売れていない部分のみが リザーブドインスタンスマーケットプレイス からなくなります。

## 支払いを受け取る

AWS がリザーブドインスタンスの購入者からの支払い金を受領するとすぐに、AWS から販売者の E メールアドレス (売却されたリザーブドインスタンスの所有者として登録されているアカウントに関連付けられた E メールアドレス) にメッセージが送信されます。

AWS は、リザーブドインスタンスマーケットプレイスの登録時に指定した銀行口座に、自動決済機関 (ACH) による電子送金を行います。通常、この送金は、リザーブドインスタンスの照会後 1~3 日の間に行われます。この支払いの状態を確認するには、リザーブドインスタンス支払いレポートを表示します。支払いは、1 日に 1 回行われます。AWS が銀行からの検証結果を受領するまで、支払い金を受け取れないことに注意してください。これには最大 2 週間かかります。

お客様が販売したリザーブドインスタンスは、お客様が実行した `DescribeReservedInstances` 呼び出しの結果に引き続き表示されます。

## 通知

リザーブドインスタンスマーケットプレイスの販売者は、アカウントに関係する リザーブドインスタンスマーケットプレイス アクティビティの抜粋版を E メールで受け取ります。毎日 1 通、抜粋版の E

メールが送信されます。このような E メールは、以下の 1 つまたは複数のアクティビティの組み合わせが発生した日にのみ送信されます。

- リザーブドインスタンスマーケットプレイス で新しい出品を作成した。
- 出品したリザーブドインスタンスが 1 つ以上売れた。
- リザーブドインスタンスマーケットプレイス で一部またはすべての出品が売れ、AWS から銀行口座に支払いが実行された。

抜粋版の E メールは、次のようになります。

Dear John Doe,

Your account's Reserved Instance Marketplace activity for 2012/08/27 is shown below:

**New Listings**

Platform	Instance Type	Offering Type	Term	Availability Zone	Price	Count	Total Potential Earnings
Linux/UNIX	m1.large	Medium Utilization	2 months	us-east-1a	\$24.44	2	\$99.11

**Recent Sales**

Platform	Instance Type	Offering Type	Term	Availability Zone	Price	Count	Total Earnings
Linux/UNIX (Amazon VPC)	m1.large	Heavy Utilization	35 months	us-east-1a	\$1,480.50	1	\$1,317.04
Linux/UNIX (Amazon VPC)	m1.large	Medium Utilization	35 months	us-east-1a	\$1,456.00	1	\$1,426.50

**Disbursements**

Zip Code	Country	Disbursement Amount	Status
12085	United States	\$1,056.11	Success
	<b>Total</b>	<b>\$1,056.11</b>	

We initiated a transfer to your checking account (ending with 2634) of \$1,056.11 on 2012/07/19. Funds usually arrive within 3 - 5 banking days, but times vary by bank.

If you have questions about your Reserved Instance Marketplace activity above, please visit the Reserved Instance Marketplace web page at <http://aws.amazon.com/reserved-instance-marketplace>.

Sincerely,  
The Amazon EC2 Team

## リザーブドインスタンスの要件チェックリスト

Amazon EC2 リザーブドインスタンスと リザーブドインスタンスマーケットプレイスは、お客様がビジネスを運営する上で強力的かつ低コストの戦略として利用できます。しかし、リザーブドインスタンスや リザーブドインスタンスマーケットプレイスを使用する前に、お客様が購入および販売の要件を満たしていることをご確認ください。また、リザーブドインスタンスおよびリザーブドインスタンスマーケットプレイスの一定の要素について、詳細と制限を理解する必要があります。販売者としての登録、銀行、AWS 無料利用枠、キャンセルしたインスタンスの扱いなどです。このトピックは、リザーブドインスタンスを購入および販売するとき、および リザーブドインスタンスマーケットプレイス で売買を行うときのチェックリストとして使用してください。

### リザーブドインスタンス

- AWS アカウント – リザーブドインスタンスを購入するには、AWS アカウントが必要です。AWS アカウントをお持ちでない場合は、Amazon EC2 アカウントおよび認証情報にサインアップする方法を説明している「[Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#)」の指示を読み、実行する必要があります。
- AWS の無料利用枠 – AWS の無料利用枠は、新しい AWS アカウントで使用できます。(AWS の) 無料利用枠を使用して Amazon EC2 インスタンスを実行している場合、リザーブドインスタンスを購入すると、そのリザーブドインスタンスは標準の料金ガイドラインに基づいて課金されます。無料範囲、適用されるサービス、および使用量については、「[AWS 無料利用枠](#)」を参照してください。

### リザーブドインスタンスを購入する

- 重度使用の利用料 – 軽度使用 および 中度使用 リザーブドインスタンスを使用する場合は、1 回だけの前払い料金を支払い、その後はインスタンスを使用したときにのみ時間単位の料金を支払います。重度使用 リザーブドインスタンスを利用する場合は、1 回だけの低額の前払い料金を支払い、リザーブドインスタンスの期間内におけるすべての時間について、インスタンスを使用するかどうかに関係なく時間単位の料金を支払う契約を結びます。詳細については、「[使用計画に基づいてリザーブドインスタンスを選択する \(p. 225\)](#)」を参照してください。

- 購入に対する範囲による割引 – リザーブインスタンスの価格範囲割引は、AWS からの購入にのみ適用されます。この種の割引は、サードパーティのリザーブインスタンスの購入には適用されません。詳細については、「[リザーブインスタンス料金範囲を理解する \(p. 226\)](#)」を参照してください。
- 購入のキャンセル – 購入を承認する前に、購入しようとしているリザーブインスタンスの詳細を見て、すべてのパラメータが正しいことを確認してください。リザーブインスタンスを購入した後では (リザーブインスタンスマーケットプレイスでサードパーティの販売者から、またはAWSのいずれの場合も)、購入をキャンセルすることはできません。ただし、ニーズが変化したときは、リザーブインスタンスを売却できます。詳細については、[リザーブインスタンスを販売する \(p. 266\)](#)を参照してください。

#### リザーブインスタンスの販売と リザーブインスタンスマーケットプレイス

- 販売者の要件 – リザーブインスタンスマーケットプレイス で販売者になるには、販売者として登録する必要があります。詳細については、「[リザーブインスタンスを販売する \(p. 266\)](#)」を参照してください。
- 銀行の要件 – AWS では、リザーブインスタンスの売却時に集金された金額をお支払いするために、お客様の銀行情報が必要です。住所が米国内の銀行でなければなりません。詳細については、「[銀行 \(p. 263\)](#)」を参照してください。
- 税金の要件 – リザーブインスタンスの取引数が 200 件以上ある販売者や 20,000 USD 以上の出品を予定している販売者は、税務上の理由から、ビジネスに関する追加情報を提供する必要があります。詳細については、「[税金情報 \(p. 264\)](#)」を参照してください。
- 最低販売価格 – リザーブインスタンスマーケットプレイス で許容される最低販売価格は、1.01 USD です。
- リザーブインスタンスを販売できる時期 – リザーブインスタンスを販売できるのは、AWS が前払い料金を受け取り、リザーブインスタンスがアクティブになってから (所有してから) 少なくとも 30 日が経過してからです。さらに、出品するリザーブインスタンスの残りの有効期間が 1 か月以上あることが必要です。
- 出品の変更 – 現在、リザーブインスタンスマーケットプレイス で出品を直接変更することはできません。ただし、最初に出品をキャンセルしてから、新しいパラメータで別の出品を作成することはできます。詳細については、[出品内容のキャンセル、変更を行います。 \(p. 276\)](#)を参照してください。出品する前にリザーブインスタンスを変更することもできます。詳細については、「[リザーブインスタンスの変更 \(p. 251\)](#)」を参照してください。
- 割引されたリザーブインスタンスの販売 – 料金範囲に基づく割引価格で購入された Amazon EC2 リザーブインスタンスは、リザーブインスタンスマーケットプレイス で販売することはできません。詳細については、[リザーブインスタンスマーケットプレイス \(p. 233\)](#)を参照してください。
- サービス料 – マーケットプレイスで販売するリザーブインスタンスごとに、前払い価格の総額の 12% がサービス料として AWS から課金されます。(前払い価格は、販売者が自分の販売するリザーブインスタンスに課金する価格です。)
- 他の AWS リザーブインスタンス – リザーブインスタンスマーケットプレイスで販売できるのは、Amazon EC2 リザーブインスタンスだけです。Amazon Relational Database Service ( Amazon RDS ) や Amazon ElastiCache リザーブインスタンスなど、その他の AWS リザーブインスタンスは、リザーブインスタンスマーケットプレイスで販売できません。

## インスタンスメタデータとユーザーデータ

### Abstract

インスタンスメタデータ、動的データ、およびユーザーデータにアクセスして、実行中のインスタンスを管理します。

インスタンスメタデータは、インスタンスに関するデータで、実行中のインスタンスを設定または管理するために使用します。インスタンスメタデータはいくつかのカテゴリに分けられます。詳細については、「[インスタンスメタデータのカテゴリ \(p. 297\)](#)」を参照してください。

EC2 インスタンスには、インスタンスの起動時に生成されるインスタンスアイデンティティドキュメントなどの動的データも含まれます。詳細については、「[動的データのカテゴリ \(p. 300\)](#)」を参照してください。

また、インスタンスの起動時に提供したユーザーデータにアクセスすることもできます。例えば、インスタンスを設定するためにパラメータを指定したり、単純なスクリプトをアタッチしたりできます。また、このデータを使用すると、より一般的な AMI を構築することもでき、この AMI は、起動時に提供される設定ファイルにより、変更が可能になります。例えば、さまざまな小規模ビジネスを対象としたウェブサーバーを実行する場合に、すべてのサーバーで同じ AMI を使用し、起動時にユーザーデータで指定した Amazon S3 バケットからコンテンツを取得できます。新しい顧客をいつでも簡単に追加できます。その顧客のバケットを作成し、顧客のコンテンツを追加して、AMI を起動すればよいのです。複数のインスタンスを同時に起動する場合、ユーザーデータはその予約においてすべてのインスタンスで使用可能です。

実行中のインスタンス内からインスタンスメタデータおよびユーザーデータにアクセスできるので、Amazon EC2 コンソールまたは CLI ツールを使用する必要はありません。これは、インスタンス内から実行するスクリプトを記述しているときに便利です。例えば、実行中のインスタンス内からインスタンスのローカル IP アドレスにアクセスして、外部アプリケーションへの接続を管理することができます。



#### Important

インスタンスメタデータおよびユーザーデータにはそのインスタンス自体内からのみアクセスできるものの、データは暗号化手法によって保護されていません。インスタンスにアクセスできるユーザーなら誰でもそのメタデータを参照できます。したがって、重要なデータ（長期間使用する暗号化キーなど）については、適切な安全策によって保護する必要があります。パスワードなどの機密データは、ユーザーデータとして保存しないでください。

インスタンスの起動時にユーザーデータを追加する方法の詳細については、「[インスタンスの起動 \(p. 325\)](#)」を参照してください。Amazon EBS-backed インスタンスの停止中に、これらのインスタンス上でユーザーデータを追加または変更することができます。停止したインスタンスへのユーザーデータの追加の詳細については、「[停止されているインスタンスの変更 \(p. 348\)](#)」を参照してください。

ユーザーデータを追加するときは、次の点に留意してください。

- ユーザーデータは不透明なデータとして取り扱われ、指定したデータがそのまま返されます。このデータを解釈できるかどうかは、インスタンスによって異なります。
- ユーザーデータのサイズの上限は 16 KB です。この制限は、base64 エンコード形式でなく、raw 形式のデータに適用されます。
- ユーザーデータは、API に送信する前に base64 でエンコードする必要があります。base64 エンコードは API コマンドラインツールで実行できます。このデータは、インスタンスに提示される前にデコードされます。base64 エンコードの詳細については、<http://tools.ietf.org/html/rfc4648> を参照してください。

#### Topics

- [インスタンスメタデータの取得 \(p. 291\)](#)
- [ユーザーデータの取得 \(p. 293\)](#)
- [動的データの取得 \(p. 294\)](#)
- [例: AMI 起動インデックス値 \(p. 294\)](#)
- [インスタンスメタデータのカテゴリ \(p. 297\)](#)

## インスタンスメタデータの取得

実行中のインスタンス内からインスタンスメタデータのすべてのカテゴリを表示するには、次の URI を使用します。

```
http://169.254.169.254/latest/meta-data/
```

インスタンスメタデータおよびユーザーデータの取得に使用する HTTP リクエストに対しては課金されません。

Linux インスタンスでは、cURL などのツールを使用できます。インスタンスでサポートされている場合は、次のような GET コマンドを使用することができます。

```
$ GET http://169.254.169.254/latest/meta-data/
```

```
$ curl http://169.254.169.254/latest/meta-data/
```

また、Instance Metadata Query ツールをダウンロードすることもできます。これにより、URI 全体またはカテゴリ名を入力する必要なく、インスタンスメタデータに対してクエリを実行できます。

<http://aws.amazon.com/code/1825>

Windows インスタンスでは、GNU Wget や cURL などのツールをインストールして、コマンドラインでインスタンスメタデータを取得するか、または URI をブラウザにコピーして貼り付けることができます。サードパーティツールをインストールしない場合は、PowerShell cmdlet を使用して URI を取得できます。例えば、バージョン 3.0 以降の PowerShell を実行している場合、次の cmdlet を使用します。

```
C:\> invoke-restmethod -uri http://169.254.169.254/latest/meta-data/
```



### Important

Windows インスタンスにサードパーティツールをインストールする場合は、HTTP の呼び出し方法および出力形式がここに記載されているものとは異なることがあるので、必ず付属のドキュメントをよく読んでください。

すべてのメタデータがテキスト (コンテンツタイプ text/plain) として返されます。特定のメタデータリソースに対するリクエストは、適切な値または 404 - Not Found HTTP エラーコード (リソースを使用できない場合) を返します。

一般的なメタデータリソースに対するリクエスト (/ で終わる URI) は、使用可能なリソースのリストまたは 404 - Not Found HTTP エラーコード (使用可能なリソースがない場合) を返します。リスト項目は個別の行に表示され、各行の末尾には改行記号 (ASCII 10) が付いています。

## インスタンスメタデータの取得の例

Linux インスタンスでのリクエストとレスポンスの例を次に示します。

次の例では、使用できるインスタンスメタデータのバージョンを取得しています。これらのバージョンは、Amazon EC2 API バージョンと必ずしも関連しているとは限りません。以前のバージョンに存在する構造および情報に依存するスクリプトがある場合は、以前のバージョンを使用することができます。

```
$ curl http://169.254.169.254/  
1.0  
2007-01-19  
2007-03-01  
2007-08-29  
2007-10-10  
2007-12-15  
2008-02-01  
2008-09-01  
2009-04-04  
2011-01-01  
2011-05-01  
2012-01-12  
latest
```

次の例では、上位レベルのメタデータ項目を取得しています。一部の項目は、VPC 内のインスタンスでのみ使用可能です。これらの項目の詳細については、「[インスタンスメタデータのカテゴリ \(p.297\)](#)」を参照してください。

```
$ curl http://169.254.169.254/latest/  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
hostname  
instance-action  
instance-id  
instance-type  
kernel-id  
local-hostname  
local-ipv4  
mac  
network/  
placement/  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups
```

次の例では、前の例からメタデータ項目の一部の値を取得しています。

```
$ curl http://169.254.169.254/latest/ami-id  
ami-2bb65342
```

```
$ curl http://169.254.169.254/latest/reservation-id  
r-fea54097
```

```
$ curl http://169.254.169.254/latest/hostname  
ec2-67-202-51-223.compute-1.amazonaws.com
```

次の例では、使用できるパブリックキーのリストを取得しています。

```
$ curl http://169.254.169.254/latest/public-keys/  
0=my-public-key
```

次の例は、パブリックキー 0 の形式を示しています。

```
$ curl http://169.254.169.254/latest/public-keys/0  
openssh-key
```

次の例では、パブリックキー 0 を取得しています ( OpenSSH キー形式 )。

```
$ curl http://169.254.169.254/latest/public-keys/0/openssh-key  
ssh-rsa MIICiTCcAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC  
VVMxCzAJBgNVBAGTAldBMRAdG9YDVQQHEwTZWZ0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAStC0lBTsBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q2lsYWVWbWxHZA  
BgkqhkiG9w0BCQEWEG5vb25lQGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAdG9YD  
VQQHEwTZWZ0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC0lBTsBDb25z  
b2x1MRIwEAYDVQQDEw1UZXR0Q2lsYWVWbWxHZAAdBgkqhkiG9w0BCQEWEG5vb25l  
QGftYXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWI  
WJ21uUSfwfEvySWtC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9  
T rDHudUZg3qX4waLG5M43q7Wgc /MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE  
Ibb3OhjZnzcvcQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJiLJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvJx79LjSTb  
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE my-public-key
```

次の例は、EC2-Classic プラットフォームの NAT インスタンス上の特定のネットワークインターフェイス ( MAC アドレスで示される ) で使用可能な情報を示しています。

```
$ curl http://169.254.169.254/latest/network/interfaces/mac/02:29:96:8f:6a:2d/  
device-number  
local-hostname  
local-ipv4s  
mac  
owner-id  
public-hostname  
public-ipv4s
```

次の例では、VPC 内に起動されたインスタンスのサブネット ID を取得しています。

```
$ curl http://169.254.169.254/latest/network/inter  
faces/mac/02:29:96:8f:6a:2d/subnet-id  
subnet-be9b61d7
```

## ユーザーデータの取得

ユーザーデータを取得するには、次の URI を使用します。

```
http://169.254.169.254/latest/user-data
```

ユーザーデータのリクエストは、データをそのままの状態で見返します ( コンテンツタイプ application/octetstream )。

次の例では、カンマで区切られたユーザーデータを返しています。

```
$ curl http://169.254.169.254/latest/user-data
1234,john,reboot,true | 4512,richard, | 173,,,
```

次の例では、行で区切られたユーザーデータを返しています。

```
$ curl http://169.254.169.254/latest/user-data
[general]
instances: 4

[instance-0]
s3-bucket: <user_name>

[instance-1]
reboot-on-error: yes
```

## 動的データの取得

実行中のインスタンス内から動的データを取得するには、次の URI を使用します。

```
http://169.254.169.254/latest/dynamic/
```

この例では、高レベルのインスタンスアイデンティティカテゴリを取得する方法を表示しています。

```
curl http://169.254.169.254//latest/dynamic/instance-identity/
pkcs7
signature
document
```

## 例: AMI 起動インデックス値

この例は、ユーザーデータおよびインスタンスメタデータの両方を使用してインスタンスを設定する方法を示しています。

この例では、Alice がお気に入りの Linux データベース AMI の 4 つのインスタンスを起動したいと考えています。そのうち最初の 1 つはマスターとして、残りの 3 つはレプリカとして動作します。これらのインスタンスを起動するときに、各レプリカントのレプリケーション戦略に関するユーザーデータを追加したいと考えています。このデータはすべての 4 つのインスタンスで使用可能となるので、どの部分が各インスタンスに該当するかをそれぞれが認識できるように、ユーザーデータを構築する必要があります。この構築は、各インスタンスに対して一意となる `ami-launch-index` インスタンスメタデータ値を使用して行うことができます。

Alice が構築したユーザーデータを次に示します。

```
replicate-every=1min | replicate-every=5min | replicate-every=10min
```

`replicate-every=1min` データは最初のレプリカントの設定を定義し、`replicate-every=5min` は 2 番目のレプリカントの設定を定義するというように、それぞれが定義を行います。Alice は、個別のインスタンスのデータをパイプシンボル ( | ) で区切って、このデータを ASCII 文字列として指定することにしました。

Alice は次のようにユーザーデータを指定して、4 つのインスタンスを起動します。

```
PROMPT> ec2-run-instances ami-2bb65342 -n 4 -d "replicate-every=1min | replicate-every=5min | replicate-every=10min"
```

```
RESERVATION      r-fea54097          598916040194    default
INSTANCE i-10a64379  ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
INSTANCE i-10a64380  ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
INSTANCE i-10a64381  ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
INSTANCE i-10a64382  ami-2bb65342 pending 0 m1.small 2010-03-19T13:59:03+0000
us-east-1a aki-94c527fd ari-96c527ff monitoring-disabled ebs
```

起動したすべてのインスタンスに、ユーザーデータのコピーと次に示す一般的なメタデータが含まれています。

- AMI ID: ami-2bb65342
- 予約 ID: r-fea54097
- パブリックキー: none
- セキュリティグループ名: default
- インスタンスタイプ: m1.small

ただし、各インスタンスには所定の一意的なメタデータが含まれます。

#### インスタンス 1

メタデータ	値
instance-id	i-10a64379
ami-launch-index	0
public-hostname	ec2-67-202-51-223.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-hostname	ip-10-251-50-35.ec2.internal
local-ipv4	10.251.50.35

#### インスタンス 2

メタデータ	値
instance-id	i-10a64380
ami-launch-index	1
public-hostname	ec2-67-202-51-224.compute-1.amazonaws.com
public-ipv4	67.202.51.224
local-hostname	ip-10-251-50-36.ec2.internal
local-ipv4	10.251.50.36

### インスタンス 3

メタデータ	値
instance-id	i-10a64381
ami-launch-index	2
public-hostname	ec2-67-202-51-225.compute-1.amazonaws.com
public-ipv4	67.202.51.225
local-hostname	ip-10-251-50-37.ec2.internal
local-ipv4	10.251.50.37

### インスタンス 4

メタデータ	値
instance-id	i-10a64382
ami-launch-index	3
public-hostname	ec2-67-202-51-226.compute-1.amazonaws.com
public-ipv4	67.202.51.226
local-hostname	ip-10-251-50-38.ec2.internal
local-ipv4	10.251.50.38

Alice は `ami-launch-index` 値を使用して、ユーザーデータのどの部分が特定のインスタンスに該当するかを判断できます。

1. そのインスタンスの 1 つに接続し、`ami-launch-index` を取得して、それがレプリカントの 1 つであることを確認します。

```
$ curl http://169.254.169.254/latest/ami-launch-index
2
```

2. `ami-launch-index` を変数として保存します。

```
$ ami_launch_index=`curl http://169.254.169.254/latest/ami-launch-index`
```

3. ユーザーデータを変数として保存します。

```
$ user_data=`curl http://169.254.169.254/latest/user-data/`
```

4. 最後に、Alice は Linux `cut` コマンドを実行して、そのインスタンスに該当するユーザーデータの部分を抽出します。

```
$ echo $user_data | cut -d"|" -f"$ami_launch_index"
replicate-every=5min
```

## インスタンスメタデータのカテゴリ

次の表は、インスタンスメタデータのカテゴリをまとめたものです。

データ	説明	導入されたバージョン
ami-id	インスタンスの起動に使用される AMI ID。	1.0
ami-launch-index	同時に複数のインスタンスを起動した場合、この値はインスタンスが起動された順序を示します。最初に起動されたインスタンスの値は 0 です。	1.0
ami-manifest-path	Amazon S3 での AMI のマニフェストファイルのパス。Amazon EBS-Backed AMI を使用してインスタンスを起動した場合、返される結果は unknown です。	1.0
ancestor-ami-ids	この AMI を作成するために再バンドルされたあらゆるインスタンスの AMI ID。この値は、AMI マニフェストファイルが ancestor-amis キーを含む場合にのみ存在します。	2007-10-10
block-device-mapping/ami	root/boot ファイルシステムを含む仮想デバイス。	2007-12-15
block-device-mapping/ebs N	Amazon EBS ボリュームに関連付けられた仮想デバイス (存在する場合)。この値は、起動時に存在する場合のみ、メタデータで使用可能です。N は、Amazon EBS ボリュームのインデックス (ebs1 や ebs2 など) を示します。	2007-12-15
block-device-mapping/eph emeral N	エフェメラルデバイスに関連付けられた仮想デバイス (存在する場合)。N は、エフェメラルボリュームのインデックスを示します。	2007-12-15
block-device-mapping/root	ルートデバイスに関連付けられた仮想デバイスまたはパーティション、あるいは仮想デバイス上のパーティション。ルート (/ または C:) ファイルシステムは、所定のインスタンスに関連付けられています。	2007-12-15
block-device-mapping/swap	swap に関連付けられた仮想デバイス。存在しない場合もあります。	2007-12-15
hostname	インスタンスのプライベートホスト名。複数のネットワークインターフェイスが存在する場合、これは eth0 デバイス (デバイス番号が 0 のデバイス) を示します。	1.0

データ	説明	導入されたバージョン
iam/info	インスタンスの LastUpdated、InstanceProfileArn、InstanceProfileId など、インスタンスプロファイルの最終更新日時に関する情報を返します。	2012-06-01
iam/security-credentials/ role-name	<i>role-name</i> は、インスタンスに関連付けられた IAM ロールの名前です。IAM ロールに関連付けられた一時的なセキュリティ認証情報 ( AccessKeyId、SecretAccessKey、SessionToken、および Expiration ) を返します。	2012-06-01
instance-action	バンドルの準備のために再起動する必要があることをインスタンスに伝えます。有効な値: none   shutdown   bundle-pending。	2008-09-01
instance-id	このインスタンスの ID。	1.0
instance-type	インスタンスの種類。詳細については、「 <a href="#">インスタンスタイプ (p. 109)</a> 」を参照してください。	2007-08-29
kernel-id	このインスタンスで起動したカーネルの ID ( ある場合 )。	2008-02-01
local-hostname	インスタンスのプライベート DNS ホスト名。複数のネットワークインターフェイスが存在する場合、これは eth0 デバイス ( デバイス番号が 0 のデバイス ) を示します。	2007-01-19
local-ipv4	インスタンスのプライベート IP アドレス。複数のネットワークインターフェイスが存在する場合、これは eth0 デバイス ( デバイス番号が 0 のデバイス ) を示します。	1.0
mac	インスタンスのメディアアクセスコントロール ( MAC ) アドレス。複数のネットワークインターフェイスが存在する場合、これは eth0 デバイス ( デバイス番号が 0 のデバイス ) を示します。	2011-01-01
network/interfaces/macs/ mac/device-number	そのインターフェイスに関連付けられたデバイス番号。各インターフェイスは一意的なデバイス番号を持つ必要があります。デバイス番号は、インスタンスのデバイス名への手掛かりとなります。例えば、eth2 デバイスの device-number は 2 です。	2011-01-01
network/interfaces/macs/ mac/ipv4-associations/public-ip	各 public-ip アドレスに関連付けられ、そのインターフェイスに割り当てられたプライベート IPv4 アドレス。	2011-01-01

データ	説明	導入されたバージョン
network/interfaces/mac/mac/local-hostname	インターフェイスのローカルホスト名。	2011-01-01
network/interfaces/mac/mac/local-ipv4s	インターフェイスに関連付けられたプライベート IP アドレス。	2011-01-01
network/interfaces/mac/mac/mac	インスタンスの MAC アドレス。	2011-01-01
network/interfaces/mac/mac/owner-id	ネットワークインターフェイスの所有者の ID。複数インターフェイスの環境では、インターフェイスは Elastic Load Balancing などのサードパーティによってアタッチできます。インターフェイス上のトラフィックは、常にインターフェイス所有者に対して課金されます。	2011-01-01
network/interfaces/mac/mac/public-hostname	インターフェイスのパブリック DNS。インスタンスが VPC 内に存在する場合、このカテゴリは、 <code>enableDnsHostnames</code> 属性が <code>true</code> に設定されている場合にのみ返されます。詳細については、「 <a href="#">Using DNS with Your VPC</a> 」を参照してください。	2011-01-01
network/interfaces/mac/mac/public-ipv4s	インターフェイスに関連付けられた Elastic IP アドレス。インスタンスには複数の IP アドレスが存在する場合があります。	2011-01-01
network/interfaces/mac/mac/security-groups	ネットワークインターフェイスが属するセキュリティグループ。VPC 内に起動されたインスタンスに対してのみ返されます。	2011-01-01
network/interfaces/mac/mac/security-group-ids	ネットワークインターフェイスが属するセキュリティグループの ID。VPC 内に起動されたインスタンスに対してのみ返されます。EC2-VPC プラットフォームのセキュリティグループの詳細については、「 <a href="#">Security Groups for Your VPC</a> 」を参照してください。	2011-01-01
network/interfaces/mac/mac/subnet-id	インターフェイスが存在するサブネットの ID。VPC 内に起動されたインスタンスに対してのみ返されます。	2011-01-01
network/interfaces/mac/mac/subnet-ipv4-cidr-block	インターフェイスが存在するサブネットの CIDR ブロック。VPC 内に起動されたインスタンスに対してのみ返されます。	2011-01-01
network/interfaces/mac/mac/vpc-id	インターフェイスが存在する VPC の ID。VPC 内に起動されたインスタンスに対してのみ返されます。	2011-01-01

データ	説明	導入されたバージョン
<code>network/interfaces/mac/mac/vpc-ipv4-cidr-block</code>	インターフェイスが存在する VPC の CIDR ブロック。VPC 内に起動されたインスタンスに対してのみ返されます。	2011-01-01
<code>placement/availability-zone</code>	インスタンスが起動した利用可能ゾーン。	2008-02-01
<code>product-codes</code>	インスタンスに関連付けられた製品コード (ある場合)。	2007-03-01
<code>public-hostname</code>	インスタンスのパブリック DNS。インスタンスが VPC 内に存在する場合、このカテゴリは、 <code>enableDnsHostnames</code> 属性が <code>true</code> に設定されている場合にのみ返されます。詳細については、「 <a href="#">Using DNS with Your VPC</a> 」を参照してください。	2007-01-19
<code>public-ipv4</code>	パブリック IP アドレス。インスタンスに Elastic IP アドレスが関連付けられている場合、返される値は Elastic IP アドレスです。	2007-01-19
<code>public-keys/0/openssh-key</code>	パブリックキー。インスタンスの起動時に指定された場合にのみ返されます。	1.0
<code>ramdisk-id</code>	起動時に指定された RAM ディスクの ID (該当する場合)。	2007-10-10
<code>reservation-id</code>	予約の ID。	1.0
<code>security-groups</code>	<p>インスタンスに適用されるセキュリティグループの名前。</p> <p> <b>Note</b></p> <p>VPC 内に起動されたインスタンスのみが、起動後にセキュリティグループを変更できます。これらの変更は、この場所と <code>network/interfaces/mac/<del>mac</del>security-groups</code> に反映されます。</p>	1.0

## 動的データのカテゴリ

次の表は、動的データのカテゴリをまとめたものです。

データ	説明	導入されたバージョン
fws/instance-monitoring	顧客が CloudWatch で詳細な 1 分間隔のモニタリングを有効にしているかどうかを示す値。有効な値: enabled   disabled	2009-04-04
instance-identity/document	インスタンス ID、プライベート IP アドレスなど、インスタンスの属性を含む JSON。	2009-04-04
instance-identity/pkcs7	署名に対してドキュメントの真正性およびコンテンツを確認するために使用されます。	2009-04-04
instance-identity/signature	オリジンおよび権限を確認するために使用できるデータ。	2009-04-04

## インスタンスのインポートとエクスポート

### Abstract

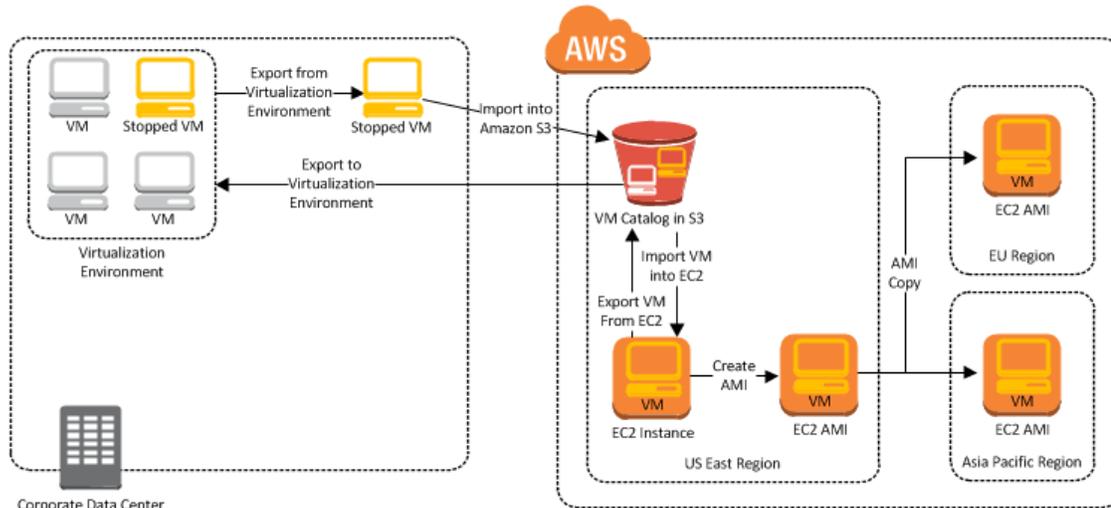
Citrix Xen、Microsoft Hyper-V、VMware vSphere などの仮想化プラットフォームから仮想マシンをインポートし、Amazon EC2 でその仮想マシンを起動します。

アマゾン ウェブ サービス (AWS) VM Import/Export ツールを使用して、仮想マシン (VM) イメージをローカル環境から AWS にインポートし、使用可能な状態の Amazon EC2 インスタンスに変換することができます。その VM イメージは、後でローカル環境にエクスポートすることができます。VM Import/Export を使用すると、仮想マシンがすぐに使えるインスタンスとして Amazon Elastic Compute Cloud (Amazon EC2) に移行されるため、IT セキュリティ、構成管理、およびコンプライアンス要件に合わせて作成した仮想マシンへの投資が無駄になりません。VM Import/Export は、Citrix Xen、Microsoft Hyper-V、または VMware vSphere の仮想環境と互換性があります。

VM Import/Export は、アプリケーションおよびワークロードの移行、VM イメージカタログのコピー、または VM イメージの障害復旧リポジトリの作成に使用できます。

- 既存のアプリケーションおよびワークロードを Amazon EC2 に移行する — VM ベースのアプリケーションおよびワークロードを Amazon EC2 に移行して、ソフトウェアと構成の設定を維持することができます。アプリケーションとワークロードを Amazon EC2 インスタンスにインポートした後、Amazon マシンイメージ (AMI) を作成し、同じイメージの複数のコピーを実行できます。スナップショットを作成し、それを使用してデータをバックアップできます。AMI とスナップショットコピーを使用して、アプリケーションとワークロードを世界中にレプリケーションできます。AMI のコピーの詳細については、[AMI のコピー \(p. 90\)](#) を参照してください。
- VM イメージカタログを Amazon EC2 にコピーする — 既存の VM イメージカタログを Amazon EC2 にコピーできます。承認された VM イメージのカタログを維持する場合、Amazon EC2 にイメージカタログをコピーして、インポートした VM イメージから Amazon EC2 インスタンスを作成することができます。ウイルス対策ソフトウェアや侵入検知システムなどのインストール済みの製品を含む既存のソフトウェアは、すべて VM イメージと一緒にインポートすることができます。作成された Amazon EC2 インスタンスを使用して、Amazon EC2 AMI を作成することができます。この AMI は Amazon EC2 内でイメージカタログとして使用できます。
- VM イメージの障害復旧リポジトリを作成する — バックアップと障害復旧のために、ローカル VM イメージを Amazon EC2 にインポートすることができます。インポートしたイメージは Amazon Elastic Block Store (Amazon EBS)-backed AMI として保存すれば、必要ときに Amazon EC2 でいつでも起動できます。ローカル環境で障害が発生した場合、インスタンスをすばやく起動してビジネスを継続させると同時に、そのインスタンスをエクスポートしてローカルのインフラストラクチャを再構築することができます。

次の図は、オンプレミス仮想化環境から AWS へ VM をエクスポートするプロセスを示しています。



## Topics

- [前提条件 \(p. 302\)](#)
- [Amazon EC2 に VM をインポートする \(p. 305\)](#)
- [Amazon EC2 インスタンスをエクスポートする \(p. 314\)](#)
- [VM Import/Export のトラブルシューティング \(p. 316\)](#)

## 前提条件

### Abstract

仮想マシンのインポートを開始する前に基礎と制限事項を確認します。

仮想化環境からのインスタンスのエクスポート、または Amazon EC2 から VM のインポートおよびエクスポートのプロセスを開始する前に、AWS でサポートされているオペレーティングシステムとイメージ形式を認識し、エクスポートするインスタンスおよびボリュームに関する制限事項を理解する必要があります。

また、コマンドラインインターフェイス (CLI) または API を使用してインスタンスをエクスポートする場合は、事前に Amazon EC2 コマンドラインツールをダウンロードしてインストールする必要があります。詳細については、[Setting Up the Amazon EC2 Tools](#) を参照してください。

オペレーティングシステム — 次のオペレーティングシステムは、Amazon EC2 とのインポートおよびエクスポートが可能です。

### Windows ( 32 ビットと 64 ビット )

- Microsoft Windows Server 2003 ( Standard、Datacenter、Enterprise ) ( Service Pack 1 適用済み ) 以降。
- Microsoft Windows Server 2003 R2 ( Standard、Datacenter、Enterprise ) 。
- Microsoft Windows Server 2008 ( Standard、Datacenter、Enterprise ) 。
- Microsoft Windows Server 2008 R2 ( Standard、Datacenter、Enterprise ) 。
- Microsoft Windows Server 2012 ( Standard、Datacenter、 ) 。

### Linux/Unix ( 64 ビット )

- Red Hat Enterprise Linux ( RHEL ) 5.1-6.5



#### Note

VM Import は、RHEL インスタンスのライセンスポータビリティをサポートしています。既存の RHEL ライセンスは、関連付けられた RHEL インスタンスと一緒にインポートされます。Red Hat Cloud Access の適格性の詳細については、Red Hat ウェブサイトの[適格性](#)を参照してください。

- CentOS 5.1-6.5
- Ubuntu 12.04, 12.10, 13.04, 13.10
- Debian 6.0.0-6.0.8, 7.0.0-7.2.0

イメージ形式を Amazon EC2 にインポートする—AWS は、次のイメージ形式について、Amazon EC2 へのボリュームとインスタンスの両方のインポートをサポートしています。

- RAW 形式。ボリュームとインスタンスのインポート用です。
- Virtual Hard Disk ( VHD ) イメージ形式。Microsoft Hyper-V および Citrix Xen 仮想化製品と互換性があります。
- ESX Virtual Machine Disk ( VMDK ) イメージ形式。VMware ESX および VMware vSphere 仮想化製品と互換性があります。

イメージ形式を Amazon EC2 からエクスポートする—AWS は、次のイメージ形式について、Amazon EC2 からのボリュームとインスタンスの両方のエクスポートをサポートしています。

- ストリームに最適化された ESX Virtual Machine Disk ( VMDK ) イメージ形式。VMware ESX および VMware vSphere バージョン 4 および 5 仮想化製品と互換性があります。
- Open Virtual Appliance ( OVA ) イメージ形式。VMware vSphere バージョン 4 および 5 と互換性があります。
- Virtual Hard Disk ( VHD ) イメージ形式。Citrix Xen および Microsoft Hyper-V 仮想化製品と互換性があります。

VM をインポートできるインスタンスタイプ—AWS では、Windows インスタンスを任意のインスタンスタイプにインポートできます。Linux インスタンスは、次のインスタンスタイプにインポートできます。

- c3.large
- c3.xlarge
- c3.2xlarge
- c3.4xlarge
- cc1.4xlarge
- cc2.8xlarge
- cg1.4xlarge
- cr1.8xlarge
- hi1.4xlarge
- hs1.8xlarge
- i2.xlarge
- i2.2xlarge
- i2.4xlarge
- m3.medium

- m3.large
- m3.xlarge
- m3.2xlarge

Amazon EC2 への VM のインポートに関する既知の制限事項— インスタンスとボリュームのインポートには、次の制限事項があります。

- リージョンあたり最大 5 つのインポートタスクを同時に実行できます。
- インスタンスがインポートされると、Hardware Virtual Machine ( HVM ) 仮想化を使用する EC2 インスタンスが作成されます。準仮想化 ( PV ) を使用するインスタンスを VM Import によって作成することは、サポートされていません。Linux PVHVM ドライバは、インポートされたインスタンス内でサポートされます。
- インポートされた Red Hat Enterprise Linux ( RHEL ) インスタンスは Cloud Access ( BYOL ) ライセンスを使用する必要があります。
- インポートされた Linux インスタンスは 64 ビットイメージを使用する必要があります。32 ビット Linux イメージのインポートは、サポートされていません。
- 通常、ディスクイメージの圧縮されたバージョンをインポートします。展開したイメージを 1 TB より大きくすることはできません。
- VM で 1 つのディスクしか使用されていないことを確認します。VM に複数のディスクがある場合はインポートできません。( Linux VM の場合、/boot および / はさまざまな場所に配置できますが、同じディスク上になければなりません )。

ブートボリュームだけを含めて VM をインポートし、追加のディスクは `ec2-import-volume` コマンドを使用してインポートすることをお勧めします。ImportInstance タスクが完了したら、`ec2-attach-volume` コマンドを使用して、追加のボリュームをインスタンスと関連付けます。

- VHD イメージには、動的な VHD イメージ形式を使用する必要があります。固定または差分 VHD イメージ形式はサポートされていません。
- Microsoft Windows を実行する Amazon EC2 にインポートする VM には、ドライバやその他のソフトウェアをインストールするために利用できるディスク容量が少なくとも 250MB あることを確認します。
- インポートしたインスタンスは、ホストコンピュータに物理的にアタッチされているディスクから一時ディスクストレージである Amazon EC2 インスタンスストアに自動的にアクセスできるようになります。インポート中にこれを無効にすることはできません。インスタンスストレージの詳細については、[Amazon EC2 インスタンスストア \(p. 631\)](#) を参照してください。
- タスクは開始日から 7 日間以内に完了する必要があります。
- 現在、複数のネットワークインターフェイスはサポートされていません。変換とインポートが完了すると、インスタンスには 1 つの仮想 NIC が与えられ、DHCP を使用してアドレスが割り当てられます。
- インターネットプロトコルバージョン 6 ( IPv6 ) の IP アドレスはサポートされていません。
- vCenter 4.0 または vSphere 4.0 を使用している場合、アタッチされている CD-ROM イメージまたは ISO を仮想マシンから削除してください。
- Amazon VM Import は、c3 および i2 インスタンスタイプにシングルルート I/O 仮想化 ( SR-IOV ) ドライバをインストールしません。これらのドライバは、より優れたパフォーマンス ( パケット毎秒 )、レイテンシーとストレスの低減を可能にする拡張ネットワーキングを使用しない場合は不要です。VM をインポートした後に c3 または i2 インスタンスタイプで拡張ネットワーキングを有効にする場合は、[VPC 内の Linux インスタンスでの拡張ネットワーキングの有効化 \(p. 559\)](#) を参照してください。

Amazon EC2 からの VM のエクスポートに関する既知の制限事項— インスタンスとボリュームのエクスポートには、次の制限事項があります。

- Amazon Elastic Block Store ( Amazon EBS ) データボリュームをエクスポートすることはできません。
- 複数の仮想ディスクのあるインスタンスはエクスポートできません。

- 複数のネットワークインターフェイスのあるインスタンスはエクスポートできません。
- インスタンスが事前に AWS にインポートされていない場合、そのインスタンスをエクスポートすることはできません。

## Amazon EC2 に VM をインポートする

### Abstract

Citrix Xen、Microsoft Hyper-V、または VMware vSphere 仮想化環境からインスタンスをインポートします。

Amazon EC2 でインスタンスを起動する方法は 2 つあります。Amazon マシンイメージ (AMI) からインスタンスを起動する方法と、Citrix Xen、Microsoft Hyper-V、または VMware vSphere のような仮想化環境からインポートした仮想マシン (VM) からインスタンスを起動する方法です。このセクションでは、VM をインポートし、Amazon EC2 インスタンスとして起動する方法について説明します。AMI から Amazon EC2 インスタンスを起動する方法の詳細は、[インスタンスの起動 \(p. 324\)](#) を参照してください。

VM を Amazon EC2 でインスタンスとして使用するには、最初に仮想化環境からエクスポートし、次にコマンドラインインターフェイス (CLI) または API ツールを使用して Amazon EC2 にインポートします。

CLI と API のどちらを使用する場合も、Amazon EC2 に VM またはボリュームをインポートするのと同じ手順に従います。次に CLI を使用するプロセスを示します。

1. CLI をインストールします。詳細については、「[ステップ 1: Amazon EC2 CLI をインストールする \(p. 305\)](#)」を参照してください。
2. Amazon EC2 にインポートする VM を準備します。詳細については、「[ステップ 2: VM を準備する \(p. 306\)](#)」を参照してください。
3. VM を仮想化環境からエクスポートします。詳細については、「[ステップ 3: VM をその仮想環境からエクスポートする \(p. 307\)](#)」を参照してください。
4. VM を Amazon EC2 にインポートします。詳細については、[ステップ 4: Amazon EC2 に VM をインポートする \(p. 307\)](#) を参照してください。
5. Amazon EC2 でインスタンスを起動します。詳細については、「[ステップ 5: Amazon EC2 でインスタンスを起動する \(p. 314\)](#)」を参照してください。

## ステップ 1: Amazon EC2 CLI をインストールする

### Abstract

Amazon EC2 コマンドラインツールを使用して、Citrix、Microsoft Hyper-V、または VMware vSphere 仮想マシンをインポートします。

Citrix、Microsoft Hyper-V、または VMware vSphere 仮想マシンを Amazon EC2 にインポートしたり、Amazon EC2 からエクスポートするには、Amazon EC2 CLI をインストールする必要があります。まだ Amazon EC2 CLI をインストールしていない場合は、[Setting Up the Amazon EC2 Tools](#) を参照してください。

次の Amazon EC2 コマンドを使用して、VM のインポートやエクスポートを行います。

コマンド	説明
<code>ec2-import-instance</code>	指定したディスクイメージのメタデータを使用してインスタンスをインポートする新しいタスクが作成され、そのインスタンスが Amazon EC2 にインポートされます。

コマンド	説明
<code>ec2-import-volume</code>	指定したディスクイメージのメタデータを使用してボリュームをインポートする新しいタスクが作成され、そのボリュームが Amazon EC2 にインポートされます。
<code>ec2-resume-import</code>	インスタンスのインポートタスクまたはボリュームインポートタスクの ID と関連付けられたディスクイメージのアップロードを再開します。
<code>ec2-describe-conversion-tasks</code>	インポートタスクを列挙して説明します。
<code>ec2-cancel-conversion-task</code>	アクティブなインポートタスクをキャンセルします。インスタンスまたはボリュームのいずれのインポートタスクでもかまいません。
<code>ec2-delete-disk-image</code>	Amazon S3 バケットからのインポートのため、一部または完全にアップロードされたディスクイメージを削除します。
<code>ec2-create-image-export-task</code>	実行中または停止したインスタンスを Amazon S3 バケットにエクスポートします。
<code>ec2-cancel-export-task</code>	アクティブなエクスポートタスクをキャンセルします。
<code>ec2-describe-export-tasks</code>	エクスポートタスクを列挙して説明します。これには最後にキャンセルされたタスクと完了したタスクも含まれます。

これらのコマンドと他の Amazon EC2 コマンドについては、[Amazon Elastic Compute Cloud コマンドラインリファレンス](#) を参照してください。

## ステップ 2: VM を準備する

仮想化環境からエクスポートする前に VM を設定する場合は、次のガイドラインを使用してください。

- 前提条件を確認します。詳細については、「[前提条件 \(p. 302\)](#)」を参照してください。
- VM のウイルス対策ソフトウェアまたは侵入検出ソフトウェアを無効にします。インポートプロセスが完了したら、これらのサービスを再度有効にすることができます。
- あらゆる ( 仮想または物理 ) CD-ROM ドライブを切断します。
- ネットワークを、静的 IP アドレスではなく DHCP に設定します。静的プライベート IP アドレスを割り当てる必要がある場合、VPC サブネットには必ず予約されていないプライベート IP アドレスを使用します。Amazon Virtual Private Cloud ( Amazon VPC ) により、VPC サブネットに最初の 4 つのプライベート IP アドレスが予約されます。
- エクスポートする前に VM をシャットダウンします。

### Windows

- リモートアクセスのためのリモートデスクトップ ( RDP ) を有効にする
- ホストのファイアウォール ( Windows ファイアウォールなど ) を設定している場合、RDP へのアクセスが許可されていることを確認します。そうしないと、インポートが完了してもインスタンスにアクセスできなくなります。

- 管理者アカウントおよび他のすべてのユーザーアカウントが安全なパスワードを使用していることを確認します。すべてのアカウントにパスワードが必要です。パスワードがないとインポートは失敗します。
- Windows VM に .NET Framework 3.5 がインストールされていることを確認します。これは [Amazon Windows EC2Config Service](#) に必要です。
- Windows VM イメージでシステムの準備 ( Sysprep ) を実行しないでください。イメージをインポートしてから、Amazon EC2 Config サービスを使用して Sysprep を実行することをお勧めします。
- Windows VM の自動ログオンを無効にします。
- 保留中の Microsoft 更新プログラムがなく、再起動時にソフトウェアをインストールするようにコンピュータが設定されていないことを確認します。

#### Linux

- リモートアクセスの Secure Shell ( SSH ) を有効にします。
- ホストのファイアウォール ( Linux iptables など ) で SSH へのアクセスが許可されていることを確認します。そうしないと、インポートが完了してもインスタンスにアクセスできなくなります。
- インポート後に、非ルートユーザーはパブリックキーベースの SSH を使用してインスタンスにアクセスするように設定されていることを確認します。パスワードベースの SSH の利用と SSH を介したルートログインはどちらも可能ですが、お勧めしません。パブリックキーおよび非ルートユーザーの使用は、安全性を向上させるため、お勧めします。VM Import では、インポートプロセスの一環として `ec2-user` アカウントを設定しません。
- Linux VM でブートローダーとして GRUB ( GRUB レガシー ) または GRUB 2 が使用されていることを確認します。
- Linux VM で EXT2、EXT3、EXT4、Btrfs、JFS、XFS のいずれかのルートファイルシステムが使用されていることを確認します。

## ステップ 3: VM をその仮想環境からエクスポートする

VM をエクスポートする準備ができたら、それを仮想化環境からエクスポートできます。VM を仮想化環境からエクスポートする方法については、Citrix、Microsoft Hyper-V、または VMware vCenter の仮想化環境に関するドキュメントを参照してください。

Citrix: 詳細については、Citrix のウェブサイトで「[Export VMs as OVF/OVA](#)」を参照してください。

Microsoft Hyper-V: 詳細については、Microsoft のウェブサイトで「[Hyper-V - Export & Import](#)」を参照してください。

VMware: 詳細については、VMware のウェブサイトで「[Export an OVF Template](#)」を参照してください。

## ステップ 4: Amazon EC2 に VM をインポートする

VM を仮想化環境からエクスポートした後に、それを Amazon EC2 にインポートできます。VM のエクスポート元にかかわらず、インポートプロセスは同じです。

VM インスタンスに関する重要事項と、いくつかのセキュリティおよびストレージに関する推奨事項を次に示します。

- Amazon EC2 によって、DHCP IP アドレスがインスタンスに自動的に割り当てられます。インスタンスの実行が開始されると、`ec2-describe-instances` コマンドで DNS 名と IP アドレスを使用できるようになります。
- 各インスタンスにはイーサネットネットワークインターフェイスが 1 つだけあります。
- インポートタスクを作成するとき使用する Amazon Virtual Private Cloud ( Amazon VPC ) サブネットを指定するには、`ec2-import-instance` コマンドで `--subnet subnet_id` オプションを使用し

ます。指定しない場合、インスタンスはパブリック IP アドレスを使用します。セキュリティグループの制限を使用して、インスタンスへのアクセスを制御することをお勧めします。--subnet *subnet\_id* オプションの詳細については、[ec2-import-instance](#) を参照してください。

- Windows インスタンスのすべてのユーザーアカウントに強力なパスワードを設定することをお勧めします。Linux インスタンスでは SSH のパブリックキーを使用することをお勧めします。
- Windows インスタンスの場合は、仮想マシンを Amazon EC2 にインポートした後に [Amazon Windows EC2Config Service](#) をインストールすることをお勧めします。

### 仮想マシンをインポートするには

VM を Amazon EC2 にインポートできます。VM のインポートが中断された場合、`ec2-resume-import` コマンドを使用して、停止したところからインポートを再開できます。詳細については、「[アップロードを再開する \(p. 312\)](#)」を参照してください。

- `ec2-import-instance` を使用して、インスタンスをインポートする新しいタスクを作成します。コマンドの構文は次のとおりです。

```
ec2-import-instance DISK_IMAGE_FILENAME -t INSTANCETYPE -f FORMAT -a ARCHITECTURE-SYSTEM -b S3_BUCKET_NAME -o OWNER -w SECRETKEY -p PLATFORM_NAME
```

次のコマンドは、Windows Server 2008 SP2 ( 32 ビット ) VM をインポートするインポートインスタスタスクを作成します。



#### Note

この例では、VMDK 形式を使用します。VHD または RAW を使用することもできます。

```
ec2-import-instance ./WinSvr8-2-32-disk1.vmdk -f VMDK -t m1.small -a i386 -b myawsbucket -o AKIAIOSFODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY -p Windows
```

このリクエストでは、VMDK ファイルの `WinSvr8-2-32-disk1.vmdk` を使用してインポートタスクを作成します。リクエストボリュームのサイズを指定しない場合は、デフォルトサイズの 4 GB が使用されます。出力は次の例のようになります。

```
Requesting volume size: 25 GB
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
TaskType          IMPORTINSTANCE TaskId import-i-fhbx6hua ExpirationTime
2011-09-09T15:03:38+00:00 Status active StatusMessage Pending
InstanceID        i-6ced060c
DISKIMAGE         DiskImageFormat VMDK   DiskImageSize 5070303744
VolumeSize        25      AvailabilityZone us-east-1c    Approximate
BytesConverted    0       Status active StatusMessage Pending
Creating new manifest at testImport/9cba4345-b73e-4469-8106-
2756a9f5a077/Win_2008_R1_EE_64.vmdkmanifest.xml
Uploading the manifest file
Uploading 5070303744 bytes across 484 parts
0% |-----| 100%
   |=====|
Done
```

```
ec2-import-instance DISK_IMAGE_FILENAME -t INSTANCETYPE -f FORMAT -a ARCHITECTURE-SYSTEM -b S3_BUCKET_NAME -o OWNER -w SECRETKEY -p PLATFORM_NAME -o AWS_ACCESS_KEY -w AWS_SECRET_KEY --region REGION_ID
```

次の例では、64 ビット Linux 仮想マシンをインポートするインポートインスタンスタスクを作成します。

```
ec2-import-instance rhel6.4-64bit-disk.vhd -O AKIAIOSFODNN7EXAMPLE -W wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY -f vhd -p Linux -o AKIAI44QH8DHBEXAMPLE -w je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY -t m3.xlarge -a x86_64 -b myawsbucket --region us-east-1
```

このリクエストでは、VHD ファイル [rhel6.4-64bit-disk.vhd] を使用してインポートタスクを作成します。出力は次の例のようになります。

```
Requesting volume size: 8 GB
TaskType          IMPORTINSTANCE TaskId    import-i-ffnzq636      ExpirationTime
2013-12-12T22:55:18Z      Status      active  StatusMessage      Pending In
stanceID          i-a56ab6dd
DISKIMAGE         DiskImageFormat VHD      DiskImageSize      861055488
VolumeSize        8           AvailabilityZone    us-east-1d          Approximate
BytesConverted    0           Status             active  StatusMessage      Pending
Creating new manifest at myawsbucket/b73bae14-7ec5-4122-8958-
4234028e1d9f/rhel6.4-64bit-disk.vhdmanifest.xml
Uploading the manifest file
Uploading 861055488 bytes across 83 parts
0% |-----| 100%
   |=====|
Done
Average speed was 11.054 MBps
The disk image for import-i-ffnzq636 has been uploaded to Amazon S3 where
it is being converted into an EC2 instance. You may monitor the progress of
this task by running ec2-describe-conversion-tasks. When the task is com
pleted, you may use ec2-delete-disk-image to remove the image from S3.
```

## インポートタスクのステータスをチェックする

ec2-describe-conversion-tasks コマンドはインポートタスクのステータスを返します。次のようなステータス値があります。

- active – インスタンスまたはボリュームはまだインポート中です。
- cancelling – インスタンスまたはボリュームはまだキャンセル中です。
- cancelled – インスタンスまたはボリュームはキャンセルされました。
- completed – インスタンスまたはボリュームは使用できる状態です。

インポートしたインスタンスが停止状態にあります。ec2-start-instance を使用してそれを起動します。詳細については、[Amazon Elastic Compute Cloud コマンドラインリファレンスの ec2-start-instances](#) を参照してください。

## インポートタスクのステータスを確認するには

- タスクのステータスを返す `ec2-describe-conversion-task` を使用します。コマンドの構文は次のとおりです。

```
ec2-describe-conversion-tasks --region REGIONID TASKID
```

次の例で、インポートインスタンスタスクのステータスを確認できます。

```
ec2-describe-conversion-tasks --region us-east-1 import-i-ffvko9js
```

次のレスポンスは、IMPORTINSTANCE ステータスが `active` であり、893968896 バイト中の 73747456 バイトが変換されたことを示します。

```
TaskType      IMPORTINSTANCE TaskId import-i-ffvko9js      ExpirationTime
2011-06-07T13:30:50+00:00      Status active StatusMessage Pending
InstanceID    i-17912579
DISKIMAGE     DiskImageFormat VMDK   DiskImageSize 893968896 VolumeSize
12            AvailabilityZone us-east-1      ApproximateBytesConverted
73747456      Status active StatusMessage Pending
```

次のレスポンスは、IMPORTINSTANCE ステータスが `active` であり、7% の進行状況で、DISKIMAGE が完了したことを示します。

```
TaskType      IMPORTINSTANCE TaskId import-i-ffvko9js      ExpirationTime
2011-06-07T13:30:50+00:00      Status active StatusMessage Progress:
7% InstanceID    i-17912579
DISKIMAGE     DiskImageFormat VMDK   DiskImageSize 893968896 VolumeId
vol-9b59daf0  VolumeSize 12      AvailabilityZone us-
east-1        ApproximateBytesConverted 893968896 Status completed
```

次のレスポンスは、IMPORTINSTANCE ステータスが `completed` であることを示します。

```
TaskType      IMPORTINSTANCE TaskId import-i-ffvko9js      ExpirationTime
2011-06-07T13:30:50+00:00      Status completed InstanceID i-
17912579
DISKIMAGE     DiskImageFormat VMDK   DiskImageSize 893968896 VolumeId
vol-9b59daf0  VolumeSize 12      AvailabilityZone us-
east-1        ApproximateBytesConverted 893968896 Status completed
```



### Note

IMPORTINSTANCE ステータスは、最終ステータスを判断するために使用します。DISKIMAGE ステータスが `completed` になる前の一定期間、IMPORTINSTANCE ステータスが `completed` になります。

`ec2-stop-instance`、`ec2-start-instance`、`ec2-reboot-instance`、`ec2-terminate-instance` などのコマンドを使用して、インスタンスを管理できます。



#### Note

デフォルトでは、インスタンスを終了させたときに、Amazon EC2 は関連付けられた Amazon EBS ボリュームを削除しません。オプションで、`ec2-modify-instance-attribute` コマンドを使用してこの動作を変更できます。

## Amazon EBS にボリュームをインポートする

このセクションでは、データストレージを Amazon EBS にインポートし、それを既存の EC2 インスタンスのいずれかにアタッチする方法について説明します。Amazon EC2 は、RAW および VMDK ディスク形式をサポートします。



#### Important

Amazon EC2 セキュリティグループを利用して、インポートしたインスタンスへのネットワークアクセスを制限することをお勧めします。信頼できる EC2 インスタンスとリモートホストにのみ、RDP やその他のサービスポートへの接続を許可するようにセキュリティグループを設定します。セキュリティグループの詳細については、[Amazon EC2 セキュリティグループ \(p. 477\)](#) を参照してください。

仮想化環境から仮想マシンをエクスポートしたら、たった 1 つのステップでボリュームを Amazon EBS にインポートします。インポートタスクを作成し、ボリュームをアップロードします。

Amazon EBS にボリュームをインポートするには

1. `ec2-import-volume` を使用して、ボリュームを Amazon EBS にアップロードできるようにするタスクを作成します。コマンドの構文は次のとおりです。

```
$ ec2-import-volume DISK_IMAGE_FILENAME -f FORMAT -s SIZE-IN-GB -z AVAILABILITY_ZONE -b S3_BUCKET_NAME -o OWNER -w SECRETKEY
```

次の例では、ボリュームを d アベイラビリティゾーンの us-east-1 リージョンにインポートするためのインポートボリュームタスクを作成します。

```
ec2-import-volume Win_2008_R1_EE_64.vmdk -f vmdk -s 25 -z us-east-1d -b myawsbucket -o AKIAIOSFODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY --region us-east-1 -o AKIAI44QH8DHBEXAMPLE -w je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
```



#### Note

この例では、VMDK 形式を使用します。VHD または RAW を使用することもできます。

```
Requesting volume size: 25 GB
Disk image format: Stream-optimized VMDK
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
TaskType          IMPORTVOLUME      TaskId  import-vol-ffut5xv4      ExpirationTime
2011-09-09T15:22:30+00:00      Status  active  StatusMessage  Pending
DISKIMAGE          DiskImageFormat  VMDK    DiskImageSize  5070303744
VolumeSize         25              AvailabilityZone  us-east-1d      Approximate
BytesConverted      0
Creating new manifest at myawsbucket/0fd8f5-04d8-44ae-981f-
```

```
3c9f56d04520/Win_2008_R1_EE_64.vmdkmanifest.xml
Uploading the manifest file
Uploading 5070303744 bytes across 484 parts
0% |-----| 100%
   |=====|
Done
```

Amazon EC2 は、次のステップで使用するタスク ID を返します。この例では、ID は `import-vol-ffut5xv4` です。

2. `ec2-describe-conversion-tasks` を使用して、ボリュームのインポートが成功したことを確認します。

```
$ ec2-describe-conversion-tasks --region us-east-1 import-vol-ffut5xv4
TaskType      IMPORTVOLUME  TaskId import-vol-ffut5xv4  ExpirationTime
2011-09-09T15:22:30+00:00      Status completed
DISKIMAGE     DiskImageFormat VMDK    DiskImageSize  5070303744
VolumeId      vol-365a385c  VolumeSize  25    AvailabilityZone
us-east-1d    ApproximateBytesConverted  5070303744
```

この例のステータスは `completed` です。これは、インポートが成功したことを示します。

3. `ec2-attach-volume` を使用して、Amazon EBS ボリュームを既存の EC2 インスタンスのいずれかにアタッチします。次の例では、ボリューム `vol-2540994c` をデバイス `i-a149ec4a` の `/dev/sde` インスタンスにアタッチします。

```
$ ec2-attach-volume vol-2540994c -i i-a149ec4a -d /dev/sde
ATTACHMENT vol-2540994c i-a149ec4a /dev/sde attaching 2010-03-
23T15:43:46+00:00
```

## アップロードを再開する

接続性の問題によりアップロードが中断されることがあります。アップロードを再開すると、Amazon EC2 によって、停止したところからアップロードが自動的に開始されます。次の手順では、アップロードの進行状況を確認する方法とアップロードを再開する方法について説明します。

アップロードを再開するには

- `ec2-resume-import` でタスク ID を使用してアップロードを続行します。このコマンドは HTTP HEAD アクションを使用して、再開する位置を判断します。

```
ec2-resume-import DISK_IMAGE_FILENAME -t TASK_ID -o OWNER -w SECRETKEY
```

次の例では、インスタンスインポートタスクを再開します。

```
$ec2-resume-import Win_2008_R1_EE_64.vmdk -t import-i-ffni8aei -o AKIAIOS
FODNN7EXAMPLE -w wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

次に、インスタンスインポートタスクが完了したときの出力を示します。

```
Disk image size: 5070303744 bytes (4.72 GiB)
Disk image format: Stream-optimized VMDK
```

```
Converted volume size: 26843545600 bytes (25.00 GiB)
Requested EBS volume size: 26843545600 bytes (25.00 GiB)
Uploading 5070303744 bytes across 484 parts
0% |-----| 100%
   |=====|
Done
Average speed was 10.316 MBps
The disk image for import-i-ffni8aei has been uploaded to Amazon S3
where it is being converted into an EC2 instance. You may monitor the
progress of this task by running ec2-describe-conversion-tasks. When
the task is completed, you may use ec2-delete-disk-image to remove the
image from S3.
```

## アップロードをキャンセルする

`ec2-cancel-conversion-task` を使用して、アクティブなインポートタスクをキャンセルします。インスタンスまたはボリュームのいずれのアップロードタスクでもかまいません。このコマンドは、アップロードされたボリュームやインスタンスを含め、インポートのすべての生成物を削除します。

インポートが完了したか、最後のディスクイメージを転送中の場合、コマンドは失敗し、次のような例外が返されます。

```
Client.CancelConversionTask Error: Failed to cancel conversion task import-i-
fh95npoc
```

アップロードタスクをキャンセルするには

- `ec2-cancel-conversion-task` で削除するアップロードのタスク ID を使用します。次の例では、タスク ID `import-i-fh95npoc` に関連付けられたアップロードをキャンセルします。

```
PROMPT> ec2-cancel-conversion-task -region us-east-1 import-i-fh95npoc
```

キャンセルが成功した場合、出力は次のようになります。

```
CONVERSION-TASK import-i-fh95npoc
```

`ec2-describe-conversion-tasks` コマンドを使用して、キャンセルのステータスを確認できます。次に例を示します。

```
$ ./ec2-describe-conversion-tasks -region us-east-1 import-i-fh95npoc
TaskType      IMPORTINSTANCE TaskId      import-i-fh95npoc      ExpirationTime
2010-12-20T18:36:39+00:00      Status      cancelled      InstanceID      i-825063ef
DISKIMAGE     DiskImageFormat VMDK      DiskImageSize      2671981568
VolumeSize    40      AvailabilityZone      us-east-1c      ApproximateBytesCon
verted        0      Status      cancelled
```

この例では、ステータスは `cancelled` です。処理中だった場合、ステータスは `cancelling` になります。

## アップロード後にクリーンアップする

`ec2-delete-disk-image` を使用し、イメージファイルをアップロード後に削除できます。削除しない場合、Amazon S3 のストレージの料金が課金されます。

ディスクイメージを削除するには

- `ec2-delete-disk-image` で削除するディスクイメージのタスク ID を使用します。

次の例では、タスク ID `import-i-fh95npoc` に関連付けられているディスクイメージを削除します。

```
PROMPT> ec2-delete-disk-image -region us-east-1 import-i-fh95npoc
```

キャンセルが成功した場合、出力は次のようになります。

```
DELETE-TASK import-i-fh95npoc
```

## ステップ 5: Amazon EC2 でインスタンスを起動する

VM を AWS にインポートしたので、Amazon EC2 インスタンスとして起動できます。Amazon EC2 インスタンスの起動方法の詳細については、[インスタンスの起動 \(p. 324\)](#) を参照してください。

## Amazon EC2 インスタンスをエクスポートする

### Abstract

インスタンスをエクスポートして、Amazon EC2 インスタンスのコピーをオンサイトの仮想化環境にデプロイします。

### Topics

- [インスタンスのエクスポート \(p. 314\)](#)
- [インスタンスのエクスポートのキャンセルまたは停止 \(p. 315\)](#)

インスタンスを Amazon EC2 に既にインポートしてある場合は、コマンドラインツールを使用して、そのインスタンスを Citrix Xen、Microsoft Hyper-V、または VMware vSphere にエクスポートすることができます。前にインポートしたインスタンスのエクスポートは、EC2 インスタンスのコピーをオンサイト仮想化環境にデプロイする場合に便利です。

## インスタンスのエクスポート

Amazon EC2 CLI を使用してインスタンスをエクスポートできます。[ec2-create-instance-export-task](#) コマンドは、選択した仮想化形式で、インスタンスを正しくエクスポートするために必要なすべての情報 (インスタンス ID、エクスポートされたイメージを保持する Amazon S3 バケット名、エクスポートされたイメージ名、VMDK、OVA または VHD 形式など) を収集します。エクスポートされたファイルは、指定した Amazon S3 バケットに保存されます。



### Note

インスタンスをエクスポートする際に、エクスポートされた VM が保存されるバケットに対して、標準的な Amazon S3 料金が課金されます。さらに、Amazon EBS スナップショットの一

時的な使用を反映する小額の料金が請求書に記載される場合があります。Amazon S3 の料金の詳細については、[Amazon Simple Storage Service \(S3\) Pricing](#) を参照してください。

### インスタンスをエクスポートするには

1. エクスポートされたインスタンスを格納する Amazon S3 バケットを作成します。Amazon S3 バケットでは、[vm-import-export@amazon.com] アカウントへの [Upload/Delete] および [View Permissions] アクセスが付与される必要があります。詳細については、[Amazon Simple Storage Service コンソールユーザーガイドの Creating a Bucket](#) および [Editing Bucket Permissions](#) を参照してください。
2. コマンドプロンプトで、次のコマンドを入力します。

```
ec2-create-instance-export-task INSTANCE_ID -e TARGET_ENVIRONMENT -f  
DISK_IMAGE_FORMAT -c CONTAINER_FORMAT -b S3_BUCKET
```

各パラメータの意味は次のとおりです。

*INSTANCE\_ID* は、エクスポートするインスタンスの ID です。

*TARGET\_ENVIRONMENT* は、VMware、Citrix、または Microsoft です。

*DISK\_IMAGE\_FORMAT* は、VMware 向けの VMDK、または Microsoft Hyper-V および Citrix Xen 向けの VHD です。

*CONTAINER\_FORMAT* は、VMware にエクスポートする場合に、オプションで OVA に設定できる場合があります。

*S3\_BUCKET* は、インスタンスをエクスポートする Amazon S3 バケットの名前です。

3. インスタンスのエクスポートを監視するには、コマンドプロンプトで次のコマンドを入力します。

```
ec2-describe-export-tasks --region REGION_ID TASK_ID
```

各パラメータの意味は次のとおりです。

*REGION\_ID* は、インスタンスが存在するリージョンの ID です。

*TASK\_ID* は、エクスポートタスクの ID です。

## インスタンスのエクスポートのキャンセルまたは停止

Amazon EC2 CLI を使用して、完了ポイントまでインスタンスのエクスポートをキャンセルまたは停止できます。[ec2-cancel-export-task](#) コマンドによって、一部作成された Amazon S3 オブジェクトを含め、エクスポートのすべての生成物が削除されます。エクスポートタスクが完了したか、最終的なディスクイメージ変換プロセスにある場合、コマンドは失敗し、エラーが返されます。

### インスタンスのエクスポートをキャンセルまたは停止するには

- コマンドプロンプトで次のコマンドを入力します。ec2-cancel-export-task --region *REGION\_ID* *TASK\_ID*

各パラメータの意味は次のとおりです。

*REGION\_ID* は、エクスポートタスクが開始されたリージョンの ID です。

*TASK\_ID* は、キャンセルするエクスポートタスクの ID です。

## VM Import/Export のトラブルシューティング

### Abstract

EC2 インスタンスに仮想マシンをインポートする際に発生する問題のトラブルシューティングを行います。

VM をインポートまたはエクスポートするときのエラーの多くは、サポートされていない操作をしようとしたときに発生します。これらのエラーを回避するために、インポートやエクスポートを開始する前に、[前提条件 \(p. 302\)](#) を確認してください。

### AWS エラーコード: InvalidParameter、AWS エラーメッセージ: パラメータ disk-image-size=0 の形式が無効です。

使用したイメージ形式はサポートされていません。

#### 解決

サポートされているイメージ形式 (RAW、VHD、VMDK) の 1 つを使用して再試行してください。

### Client.UnsupportedOperation: このインスタンスに複数のボリュームがアタッチされています。余分なボリュームを削除してください。

VM に複数のディスクがアタッチされています。

#### 解決

余分なドライブをデタッチして、再試行します。他のボリュームのデータが必要な場合は、データをルートボリュームにコピーし、VM を再度エクスポートします。

### ClientError: フッターの不一致

固定または差分 VHD イメージをインポートしようとしたか、VHD イメージの作成でエラーが発生しました。

#### 解決

VM で動的な VHD が使用されることを確認します。VM を再度エクスポートし、Amazon EC2 へのインポートを再試行します。

### ClientError: 非圧縮データの長さが無効です。

VMDK ファイルが破損しています。

#### 解決

VMDK ファイルの修復または再作成を試みるか、インポートに別のファイルを使用します。

### エラー: バケット <MyBucketName> は <RegionName> リージョンではなく、<RegionName> にあります。

Amazon S3 バケットがインポートするインスタンスと同じリージョンに存在しません。

#### 解決

`--ignore-region-affinity` オプションを付加して、バケットのリージョンとインポートタスクが作成されたリージョンの一致を無視することもできます。または、Amazon Simple Storage Service コンソールを使用して Amazon S3 バケットを作成し、そのリージョンを VM をインポートするリージョンに設定することもできます。再度コマンドを実行し、作成した新しいバケットを指定します。

## エラー: ファイルにサポートされない圧縮アルゴリズム 0 が使用されています。

VMDK は OVF 形式ではなく OVA 形式を使用して作成されました。

解決

OVF 形式で VMDK を作成します。

## エラー: これは、動的な VHD イメージとは判断されません。イメージは、`0xc?>Ä` で始まります。

インポートしようとしている VM には固定または差分 VHD イメージ形式が使用されていますが、それはサポートされていません。VM Import では、動的な VHD イメージの使用のみがサポートされます。

解決

VHD イメージを動的な VHD に変換し、VM のインポートを再試行します。

## java.lang.OutOfMemoryError: Java のヒープスペース

Java を起動するために使用できる仮想メモリが不足しているか、インポートするイメージが大きすぎます。

解決

Java に増設メモリを割り当てると、増設メモリは JVM にのみ適用されますが、この設定が ( EC2 コマンドラインツールに対して明示的に ) 指定されている場合は、グローバル設定がオーバーライドされます。たとえば、"`set EC2_JVM_ARGS=-Xmx512m`" コマンドを使用して、512 MB の増設メモリを Java に割り当てることができます。

## Service.InternalError: 内部エラーが発生しました。ステータスコード: 500、AWS サービス: AmazonEC2

デフォルト VPC のないインスタンスをサブネットとアベイラビリティゾーンを指定せずにインポートしようとした。

解決

デフォルト VPC のないインスタンスをインポートする場合は、必ずサブネットとアベイラビリティゾーンを指定します。

## FirstBootFailure: This import request failed because the Windows instance failed to boot and establish network connectivity.

`ec2-import-instance` コマンドを使用して VM をインポートすると、インポートタスクは完了する前に停止し、失敗する可能性があります。失敗の原因を調査するために、`ec2-describe-conversion-tasks` コマンドを使用し、インスタンスの内容を表示できます。FirstBootFailure エラーメッセージが表示された場合、仮想ディスクイメージで次のステップのいずれかを実行できなかったことを意味します。

- 起動し、Windows を開始する。
- Amazon EC2 ネットワーキングとディスクドライバをインストールする。
- DHCP が設定されたネットワークインターフェイスを使用し、IP アドレスを取得する。
- Amazon EC2 Windows ボリュームライセンスを使用し、Windows をアクティベートする。

## 最初の起動失敗を回避するためのベストプラクティス

次のベストプラクティスにより、Windows の最初の起動失敗を回避できることがあります。

**アンチウイルスとアンチスパイウェアのソフトウェアとファイアウォールを無効にします。**

この種のソフトウェアは新しい Windows サービスまたはドライバのインストールを防いだり、未知のバイナリの実行を防いだりすることがあります。ソフトウェアとファイアウォールはインポート後に再有効化できます。

**オペレーティングシステムにハードニングを行いません。**

ハードニング (強化) とも呼ばれるセキュリティ設定により、Amazon EC2 ドライバの自動インストールが阻止されることがあります。インポートを阻止する可能性がある Windows 設定が数多く存在します。そのような設定はインポート後に再適用できます。

**複数のブート可能パーティションを無効にするか、削除します。**

仮想マシンが起動し、使用するブートパーティションの選択を要求する場合、インポートが失敗することがあります。

## 最初の起動失敗の考えられる理由

仮想ディスクイメージが起動できず、ネットワーク接続を確立できないのは、次の原因のいずれかによることが考えられます。

### Topics

- 仮想マシンで Windows のインストールが無効になっています。 (p. 318)
- TCP/IP ネットワーキングと DHCP が有効になっていません。 (p. 319)
- Windows が必要とするボリュームが仮想マシンにありません。 (p. 319)
- Windows を起動すると必ずシステムリカバリオプションが表示されます。 (p. 319)
- 仮想マシンが P2V (物理から仮想へ) 変換プロセスにより作成されました。 (p. 319)
- Windows のアクティベーションが失敗します。 (p. 320)
- ブート可能なパーティションが見つかりませんでした。 (p. 320)
- 対象のインスタンスで Linux がサポートされていません。 (p. 320)

**仮想マシンで Windows のインストールが無効になっています。**

仮想マシンを正常にインポートするには、Windows のインストールが有効になっている必要があります。

### 解決

EC2 インスタンスをシャットダウンする前にシステムの準備 (Sysprep) を実行しないでください。インスタンスのインポート後に、インスタンスから SYSPREP を実行し、それから AMI を作成できます。インポートにより単一のインスタンスが作成されるので、SYSPREP を実行する必要はありません。

インストールプロセスが完全に完了し、(ユーザーの介入なしで) Windows が起動してログインプロンプトが表示されることを確認します。

### TCP/IP ネットワーキングと DHCP が有効になっていません。

Amazon EC2 インスタンスに対して ( Amazon VPC の Amazon EC2 インスタンスも含む )、TCP/IP ネットワーキングと DHCP を有効にする必要があります。VPC 内では、インスタンスをインポートする前と後のいずれかでインスタンスの IP アドレスを定義する必要があります。インスタンスをエクスポートする前に静的 IP アドレスを設定しないでください。

#### 解決

TCP/IP ネットワーキングが有効になっていることを確認します。詳細については、Microsoft TechNet ウェブサイトにある「[Setting up TCP/IP \(Windows Server 2003\)](#)」または「[Configuring TCP/IP \(Windows Server 2008\)](#)」を参照してください。

DHCP が有効になっていることを確認します。詳細は、Microsoft TechNet ウェブサイトにある「[What is DHCP](#)」を参照してください。

### Windows が必要とするボリュームが仮想マシンにありません。

VM を Amazon EC2 にインポートすると、起動ディスクのみがインポートされます。仮想マシンをインポートする前に、他のすべてのディスクをデタッチし、Windows を起動できる状態にする必要があります。例えば、アクティブディレクトリは、多くの場合、D:\ ドライブにアクティブディレクトリデータベースを格納します。アクティブディレクトリデータベースがない、またはそれにアクセスできない場合、ドメインコントローラは起動できません。

#### 解決

エクスポート前に、Windows VM にアタッチされたセカンダリディスクとネットワークディスクをデタッチします。

アクティブディレクトリデータベースをセカンダリドライブまたはパーティションからプライマリ Windows パーティションに移動します。詳細は、Microsoft Support ウェブサイトで「["Directory Services cannot start" error message when you start your Windows-based or SBS-based domain controller](#)」を参照してください。

### Windows を起動すると必ずシステムリカバリオプションが表示されます。

さまざまな理由により、Windows を起動するとシステムリカバリオプションが表示されます。例えば、Windows が物理マシンから仮想化環境に入るとき ( P2V と呼ばれています ) などが該当します。

#### 解決

エクスポートし、インポートの準備をする前に、Windows が起動してログインプロンプトが表示されることを確認します。

物理マシンから入ってきた仮想化 Windows インスタンスをインポートしないでください。

### 仮想マシンが P2V ( 物理から仮想へ ) 変換プロセスにより作成されました。

P2V 変換は、物理マシンで Windows インストールプロセスを実行し、その Windows インストールのコピーを VM にインポートすることでディスクイメージを作成するときに行われます。P2V 変換の結果として作成された VM は、Amazon EC2 VM インポートではサポートされません。Amazon EC2 VM インポートでは、インポート元の VM 内にネイティブとしてインストールされた Windows イメージのみがサポートされます。

#### 解決

Windows を仮想化環境でインストールし、インストールしたソフトウェアをその新しい VM に移行します。

### Windows のアクティベーションが失敗します。

起動中に、Windows がハードウェアの変更を検出し、アクティベーションを試行します。インポートプロセス中に、Windows のライセンス許諾メカニズムをアマゾン ウェブ サービスが提供するボリュームライセンスに切り替える試みが行われます。しかしながら、Windows アクティベーションプロセスが失敗した場合、インポートも失敗します。

#### 解決

インポートする Windows のバージョンでボリュームライセンスがサポートされていることを確認します。Windows のベータ版またはプレビュー版ではサポートされていない場合があります。

### ブート可能なパーティションが見つかりませんでした。

仮想マシンのインポートプロセス中に、ブートパーティションが見つかりませんでした。

#### 解決

インポートするディスクにブートパーティションがないことを確認します。複数ディスクのインポートはサポートされていません。

### 対象のインスタンスで Linux がサポートされていません。

Linux のインポートは、特定のインスタンスタイプでしかサポートされていません。サポートされていないインスタンスタイプのインポートが試みられています。

#### 解決

サポートされているいずれかのインスタンスタイプを使用して再試行してください。

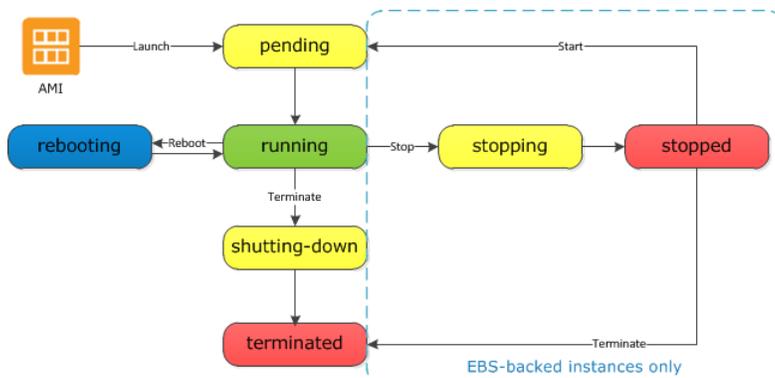
# インスタンスのライフサイクル

## Abstract

Amazon EC2 インスタンスのライフサイクルについて、起動から終了まで説明します。

このトピックでは、インスタンスの起動から終了まで、Amazon EC2 インスタンスのライフサイクルについて説明します。Amazon EC2 を操作してインスタンスを管理することで、インスタンス上でホストされているアプリケーションやホストに関して、お客様が最高のエクスペリエンスが得られるようにします。

次の図は、インスタンス状態の遷移を示しています。



## インスタンスの起動

インスタンスを起動すると、インスタンスは `pending` 状態に移行します。起動時に指定したインスタンスタイプによって、インスタンスのホストコンピュータのハードウェアが決定します。起動時に指定された Amazon マシンイメージ (AMI) を使って、インスタンスを再起動します。インスタンスの準備ができると、`running` 状態へ移行します。実行中のインスタンスに接続して、自分の前にあるコンピュータと同じように使用することができます。

インスタンスが起動するとすぐに、インスタンスの実行時間に応じて (インスタンスがアイドル状態で残っていて、それに接続していない場合でも) 課金されます。

詳細については、「[インスタンスの起動 \(p. 324\)](#)」および「[インスタンスへの接続 \(p. 334\)](#)」を参照してください。

## インスタンスの停止と起動

インスタンスのステータスチェックに失敗するか、インスタンスでアプリケーションが想定通りに動作しておらず、インスタンスのルートボリュームが Amazon EBS である場合、インスタンスの停止と起動を行い、問題が解決するか試してみることができます。

インスタンスを停止した場合、インスタンスは `stopping` 状態に移行してから、`stopped` 状態になります。停止後のインスタンスに対して時間単位の使用料金やデータ転送料金が課金されることはありませんが、Amazon EBS ボリュームのストレージについては課金されます。インスタンスが `stopped` 状態の間、インスタンスタイプなど、インスタンスの特定の属性を変更できます。

インスタンスを起動すると、`pending` 状態に移行し、新しいホストコンピュータに移動されます。このため、インスタンスの停止と起動を行うと、前のホストコンピュータ上のインスタンスストアボリューム上に存在していたすべてのデータが失われます。

インスタンスが EC2-Classic で実行されている場合、インスタンスは新しいプライベート IP アドレスを受け取ります。つまり、プライベート IP アドレスに関連付けられていた Elastic IP アドレスは、インスタンスとの関連付けが解除されたということです。インスタンスが EC2-VPC で実行されている場合、プライベート IP アドレスは保持されます。つまり、プライベート IP アドレスまたはネットワークインターフェイスに関連付けられていた Elastic IP アドレスは、インスタンスとの関連付けが継続されるということです。

インスタンスを `stopped` から `running` に移行するたびに、1 時間分の使用料が請求されます。1 時間以内に何度も停止から起動へ移行した場合でも、その都度 1 時間分の料金が請求されます。

詳細については、「[インスタンスの停止と起動 \(p. 346\)](#)」を参照してください。

## インスタンスの再起動

Amazon EC2 コンソール、Amazon EC2 CLI、および Amazon EC2 API を使って、インスタンスを再起動できます。インスタンスからオペレーティングシステムの再起動コマンドを実行する代わりに、Amazon EC2 を使ってインスタンスを再起動することをお勧めします。

インスタンスの再起動はオペレーティングシステムの再起動と同様です。インスタンスは同じホストコンピュータに残り、そのパブリック DNS 名、プライベート IP アドレス、およびその他のデータをインスタンスストアボリュームに維持します。通常、再起動が完了するまでに数分かかりますが、再起動に必要な時間は、インスタンスの設定によって異なります。

インスタンスを再起動しても、新しいインスタンスの課金時間は開始されません。

詳細については、「[インスタンスの再起動 \(p. 349\)](#)」を参照してください。

## インスタンスのリタイア

インスタンスをホストしている基盤のハードウェアで回復不可能な障害が検出されると、AWS によってインスタンスのリタイアが予定されます。予定されたリタイア日になると、インスタンスは AWS によって停止または終了されます。インスタンスのルートデバイスが Amazon EBS ボリュームである場合、インスタンスは停止されますが、その後いつでも再び起動できます。インスタンスのルートデバイスがインスタンスストアボリュームである場合、インスタンスは終了し、再び使用することはできません。

詳細については、「[インスタンスのリタイア \(p. 350\)](#)」を参照してください。

## インスタンスの終了

インスタンスが必要なくなったら、終了することができます。インスタンスのステータスが `shutting-down` または `terminated` に変わったら、そのインスタンスへの課金は停止します。

停止保護が有効な場合、コンソール、CLI、または API を使用してインスタンスを終了できないことにご注意ください。

インスタンスの終了後、インスタンスはしばらくの間コンソールに表示されたままですが、エントリは削除されます。CLI および API を使って、終了したインスタンスを記述することもできます。終了したインスタンスへの接続や復旧はできません。

Amazon EBS バックアップした各インスタンスは、`InstanceInitiatedShutdownBehavior` 属性をサポートしています。この属性は、インスタンス自体からシャットダウンを開始した場合 (Linux で `shutdown` コマンドを使用した場合など)、インスタンスを停止または終了するかを制御します。デフォルトの動作は、インスタンスの停止です。インスタンスの実行中または停止中に、この属性の設定を変更できます。

各 Amazon EBS ボリュームは `DeleteOnTermination` 属性をサポートします。この属性は、アタッチされたインスタンスを終了するときに、ボリュームの削除や保持を制御します。デフォルトでは、実行中のインスタンスにアタッチするボリュームは保持され、ルートボリュームなど、起動時にアタッチするボリュームは削除されます。

詳細については、「[インスタンスの終了 \(p. 352\)](#)」を参照してください。

## 再起動、停止、終了の違い

次の表に、インスタンスの再起動、停止、終了の主な違いをまとめました。

特徴	再起動	停止/起動	終了
ホストコンピュータ	インスタンスは、同じホストコンピュータで保持される	インスタンスは新しいホストコンピュータで実行される	なし
プライベート IP アドレスとパブリック IP アドレス	同一のまま保持される	EC2-Classic: インスタンスは新しいプライベート IP アドレスとパブリック IP アドレスを取得します  EC2-VPC: インスタンスはプライベート IP アドレスを保持します。インスタンスは、(停止/開始の際に変更されない Elastic IP アドレスを持っていない限り) 新しいパブリック IP アドレスを取得します。	なし

特徴	再起動	停止/起動	終了
Elastic IP アドレス	Elastic IP アドレスは、インスタンスに関連付けられたまま維持される	EC2-Classic: Elastic IP アドレスは、インスタンスの関連付けが解除される EC2-VPC: Elastic IP アドレスは、インスタンスに関連付けられたまま維持される	Elastic IP アドレスはインスタンスの関連付けが解除される
インスタンスストアボリューム	データは保持される	データは消去される	データは消去される
ルートデバイスボリューム	ボリュームは保持される	ボリュームは保持される	ボリュームはデフォルトで削除される

## インスタンスの起動

### Abstract

インスタンスを起動します。これは AWS クラウドの仮想サーバーです。

インスタンスとは AWS クラウドにある仮想サーバーです。Amazon マシンイメージ (AMI) からインスタンスを起動します。AMI はインスタンスに対して、オペレーティングシステム、アプリケーションサーバ、およびアプリケーションを提供します。

AWS にサインアップすると、[AWS 無料利用枠](#)を使って、Amazon EC2 を無料で開始することができます。無料利用枠を利用することも、12 か月無料でマイクロインスタンスを使用することもできます。無料利用枠に含まれないインスタンスを起動する場合は、そのインスタンスの通常の Amazon EC2 使用料がかかります。詳細については、[Amazon EC2 料金表](#)を参照してください。

次の方法を使用してインスタンスを起動できます。

メソッド	ドキュメント
選択する AMI とともに Amazon EC2 コンソールを使用する	<a href="#">インスタンスの起動 (p. 325)</a>
作成した Amazon EBS スナップショットとともに Amazon EC2 コンソールを使用する	<a href="#">バックアップからインスタンスを起動する (p. 331)</a>
AWS Marketplace から購入した AMI とともに Amazon EC2 を使用する	<a href="#">AWS Marketplace インスタンスの起動 (p. 332)</a>
選択する AMI とともに AWS CLI を使用する	<a href="#">Using Amazon EC2 through the AWS CLI</a>
選択した AMI とともに Amazon EC2 CLI を使用する	<a href="#">Launching an Instance Using the Amazon EC2 CLI</a>
選択する AMI とともに AWS Tools for Windows PowerShell を使用する	<a href="#">Amazon EC2 from the AWS Tools for Windows PowerShell</a>

インスタンスを起動した後、インスタンスに接続して使用できます。最初、インスタンスの状態は `pending` です。インスタンスの状態が `running` の場合、インスタンスは起動を開始します。インスタンスに接続するまで、少し時間がかかることがあります。インスタンスは、パブリック DNS 名を受信

します。この名前はインスタンスをインターネットに接続する場合に使用できます。インスタンスはプライベート DNS 名も受信します。同じ Amazon EC2 ネットワーク ( EC2-Classic または EC2-VPC ) 内の他のインスタンスはこの名前を使って、インスタンスと接続することができます。インスタンスへの接続の詳細については、[インスタンスへの接続 \(p. 334\)](#) を参照してください。

インスタンスを使い終わったら、必ずインスタンスを終了してください。詳細については、「[インスタンスの終了 \(p. 352\)](#)」を参照してください。

## インスタンスの起動

### Abstract

Amazon EC2 コンソールを使用して AMI からインスタンスを起動します。

インスタンスを起動する前に、セットアップが終了していることを確認してください。詳細については、「[Amazon EC2 でのセットアップ \(p. 21\)](#)」を参照してください。



### Important

**AWS 無料利用枠**に含まれないインスタンスを起動すると、インスタンスを起動して実行すると、アイドル状態であっても、料金が発生します。

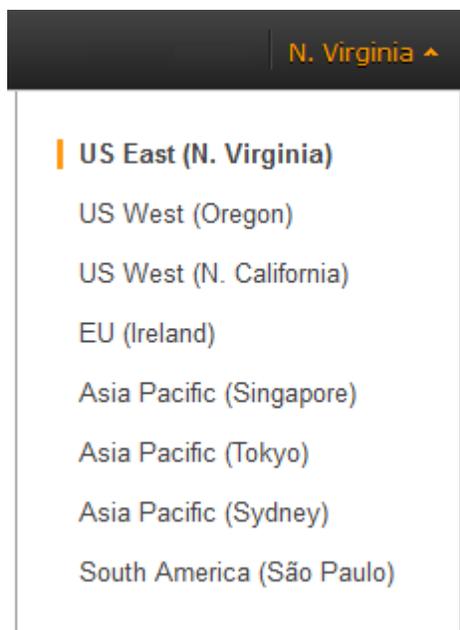
### Topics

- [AMI からのインスタンスの起動 \(p. 325\)](#)
- [既存のインスタンスをテンプレートとして使用したインスタンスを起動 \(p. 330\)](#)

## AMI からのインスタンスの起動

インスタンスを起動するには

1. Amazon EC2 コンソールを開きます。
2. 画面の上のナビゲーションバーで、現在のリージョンが表示されます。リージョンの名前をクリックして、インスタンスのリージョンを選択します。この選択は、リージョン間で共有できるのが一部の Amazon EC2 リソースのみであるため、重要です。ニーズに合ったリージョンを選択してください。詳細については、「[リソースの場所 \(p. 658\)](#)」を参照してください。



3. Amazon EC2 コンソールダッシュボードで、[Launch Instance] をクリックします。
4. [Choose an Amazon Machine Image (AMI)] ページには、Amazon マシンイメージ (AMI) と呼ばれる基本設定リストが表示されます。AMIには、新しいインスタンスを作成するために必要なすべての情報が含まれています。例えば、あるAMIにはウェブサーバーとして動作するためのすべてのソフトウェア (Linux、Apache、ウェブサイトなど) が格納されます。

使用するAMIを選択し、[Select] をクリックします。初めに、左のペインのカテゴリを使って、使用するAMIの種類を選択します。

- [Quick Start] カテゴリには、作業をすぐに開始できるようにするための一般的なAMIが表示されます。わかりやすくするために、無料利用枠内で利用できるAMIには [Free tier eligible] のマークが表示されます。
- [My AMIs] カテゴリには、お客様が所有しているプライベートAMI、またはお客様と共有されているプライベートAMIが表示されます。
- [AWS Marketplace] は、インスタンスの起動に使用できるAMIを含めて、AWSで実行されるソフトウェアを購入できるオンラインストアです。AWS Marketplaceからのインスタンスの起動の詳細については、[AWS Marketplace インスタンスの起動 \(p. 332\)](#) を参照してください。
- [Community AMIs] オプションでは、AWSコミュニティメンバーが他のメンバーに対して利用できるようにしたパブリックAMIが表示されます。

利用できるAMIの一覧をフィルタするには、左側のフィルタオプションを使用します。例えば、Amazon EBS-Backed コミュニティAMIを表示するには、[Community AMIs] カテゴリを選択し、[Operating system] の [Amazon Linux] チェックボックスをオンにして、[Root device type] の [EBS] を選択します。



#### Note

AMIを選択するときは、そのAMIが Instance store-Backed か Amazon EBS-Backed かの違いが重要になります。詳細については、[ルートデバイスのストレージ \(p. 57\)](#) を参照してください。

5. [Choose an Instance Type] ページで、起動するインスタンスのハードウェア設定とサイズを選択します。インスタンスタイプが大きくなると、CPU およびメモリも増えます。インスタンスファミリーの詳細については、[インスタンスタイプ \(p. 109\)](#) を参照してください。

無料利用枠内に収まるようにするには、[t1.micro] インスタンスを選択して、[Next: Configure Instance Details] をクリックします。



#### Note

AWS のご利用が初めてで、インスタンスをすばやく簡単にセットアップしたい場合は、この時点で [Review and Launch] をクリックしてデフォルトの設定を受け入れ、インスタンスを起動できます。詳細については、[Amazon EC2 Linux インスタンスの使用開始 \(p. 27\)](#) を参照してください。

6. [Configure Instance Details] ページで、必要に応じて次の設定を変更し (すべての設定を表示するには [Advanced Details] を展開)、[Next: Add Storage] をクリックします。

- [Number of instances]: 起動するインスタンスの数を入力します。
- [Purchasing option]: [Request Spot Instances] を選択してスポットインスタンスを起動します。詳細については、[スポットインスタンス \(p. 134\)](#) を参照してください。
- [Network]: アカウントは、EC2-Classic および EC2-VPC プラットフォーム、または EC2-VPC のみをサポートする場合があります。詳細については、[サポートされているプラットフォーム \(p. 527\)](#) を参照してください。EC2-Classic に起動するには、[Launch into EC2-Classic] を選択します。アカウントが EC2-VPC のみをサポートしている場合は、インスタンスをデフォルトの VPC に起動できます。この VPC は既に作成されています。アカウントがサポートしているプラットフォームとは関係なく、お客様が作成した VPC (非デフォルト VPC) にインスタンスを起動することもできます。

#### EC2-Classic に起動する場合

- [Availability Zone]: 使用するアベイラビリティゾーンを選択します。AWS で自動的にアベイラビリティゾーンを選択するには、[No preference] を選択します。

#### VPC に起動する場合

- [Network]: VPC を選択します。新しい VPC を作成するには、[Create new VPC] をクリックして Amazon VPC コンソールに移動します。終了したら、ウィザードに戻り、[Refresh] をクリックして、一覧に VPC を読み込みます。
- [Subnet]: インスタンス起動するサブネットを選択します。アカウントが EC2-VPC のみの場合は、[No preference] を選択して、AWS で任意のアベイラビリティゾーンのデフォルトサブネットを自動的に選択できます。新しいサブネットを作成するには、[Create new subnet] をクリックして Amazon VPC コンソールに移動します。終了したらウィザードに戻り、[Refresh] をクリックして一覧にサブネットを読み込みます。
- [Public IP]: インスタンスがパブリック IP アドレスを受け取るように要求するには、このチェックボックスをオンにします。パブリック IP アドレス設定の詳細については、[Amazon EC2 インスタンスの IP アドレッシング \(p. 529\)](#) を参照してください。
- [IAM Role]: 該当する場合は、インスタンスに関連付ける AWS Identity and Access Management (IAM) ロールを選択します。詳細については、[Amazon EC2 の IAM ロール \(p. 516\)](#) を参照してください。
- [Shutdown behavior]: シャットダウン時にインスタンスを停止するか終了するかを選択します。詳細については、[インスタンスによって起動されたシャットダウン動作の変更 \(p. 355\)](#) を参照してください。
- [Enable termination protection]: 偶発的な終了を防ぐには、このチェックボックスをオンにします。詳細については、「[インスタンスの終了保護の有効化 \(p. 354\)](#)」を参照してください。
- [Monitoring]: Amazon CloudWatch を使用したインスタンスの詳細モニタリングを有効にするには、このチェックボックスをオンにします。追加の変更が適用されます。詳細については、「[CloudWatch によるインスタンスのモニタリング \(p. 395\)](#)」を参照してください。

- [EBS-Optimized instance]: Amazon EBS 最適化インスタンスは、最適化された設定スタックを使用し、Amazon EBS I/O に対して追加の専用の容量を提供します。インスタンスタイプがこの機能をサポートしている場合は、このチェックボックスをオンにして機能を有効にします。追加の変更が適用されます。詳細については、「[Amazon EBS 最適化インスタンス \(p. 127\)](#)」を参照してください。
  - [Tenancy]: インスタンスを VPC に起動する場合は、[Dedicated tenancy] を選択して、分離された専用のハードウェアでインスタンスを実行できます。追加の変更が適用されます。詳細については、*Dedicated Instances* の [Amazon Virtual Private Cloud ユーザーガイド](#) を参照してください。
  - [Network interfaces]: インスタンスを VPC に起動し、サブネットに対して [No Preference] を選択しなかった場合、ウィザードで最大 2 つのネットワークインターフェイスを指定できます。選択したインターフェイスに対して複数の IP アドレスを割り当てるには、[Add IP] をクリックします。ネットワークインターフェイスの詳細については、[Elastic Network Interface \( ENI \) \( p. 545 \)](#) を参照してください。上記の [Public IP] チェックボックスをオンにした場合、デバイスインデックスが eth0 である新しいネットワークインターフェイスに、1 つのみパブリック IP アドレスを割り当てることができます。詳細については、[パブリック ID アドレスの割り当て \( p. 533 \)](#) を参照してください。
  - [Kernel ID]: 特定のカーネルを使用する場合を除き、[Use default] を選択します。
  - [RAM disk ID]: 特定の RAM ディスクを使用する場合を除き、[Use default] を選択します。カーネルを選択した場合は、サポートするドライバとともに特定の RAM ディスクを選択しなければならない可能性があります。
  - [Placement group]: プレイACEMENTグループは、クラスターインスタンスの論理グループです。既存のプレイACEMENTグループを選択するか、新しいグループを作成します。このオプションは、プレイACEMENTグループをサポートするインスタンスタイプを選択した場合にのみ使用できます。詳細については、[プレイACEMENTグループ \( p. 128 \)](#) を参照してください。
  - [User data]: 起動時にインスタンスを設定するユーザーデータ、または設定スクリプトを実行するユーザーデータを指定できます。ファイルを添付するには、[As file] オプションを選択し、添付するファイルを参照します。
7. [Add Storage] ページで、AMI によって指定されるボリューム ( ルートデバイスボリュームなど ) 以外に、インスタンスにアタッチするボリュームを指定します。次のオプションを変更できます。終了したら、[Next: Tag Instance] をクリックします。
- [Type]: インスタンスと関連付けるインスタンスストアまたは Amazon EBS ボリュームを選択します。一覧で利用できるボリュームの種類は、選択したインスタンスタイプに応じて異なります。詳細については、[Amazon EC2 インスタンスストア \( p. 631 \)](#) および [Amazon EBS ボリューム \( p. 567 \)](#) を参照してください。
  - [Device]: ボリュームで利用できるデバイス名の一覧から選択します。
  - [Snapshot]: ボリュームを復元するスナップショットの名前または ID を入力します。[Snapshot] フィールドにテキストを入力して、パブリックスナップショットを検索することもできます。スナップショットの説明では大文字と小文字が区別されます。
  - [Size]: Amazon EBS-Backed ボリュームについては、ストレージサイズを指定できます。無料利用枠の対象となる AMI とインスタンスを選択した場合でも、無料利用枠内に収まるようにするには、合計ストレージを 30 GB 以下に維持する必要があります。



#### Note

この時点でルートボリューム ( またはスナップショットから作成したその他のボリューム ) のサイズを増やした場合は、追加のスペースを使用するためにそのボリュームのファイルシステムを拡張する必要があります。インスタンスの起動後、ファイルシステムを拡張する詳細については、[ボリュームのストレージ領域を拡張する \( p. 593 \)](#) を参照してください。

- [Volume Type]: Amazon EBS ボリュームについては、スタンダード IOPS ボリューム、またはプロビジョンド IOPS ボリュームを選択します。詳細については、[Amazon EBS ボリュームの種類 \(p. 569\)](#) を参照してください。
  - [IOPS]: プロビジョンド IOPS ボリュームを選択した場合は、ボリュームがサポートできる I/O オペレーション/秒 ( IOPS ) を入力できます。
  - [Delete on Termination]: Amazon EBS ボリュームについては、インスタンスが終了したときにボリュームを削除するには、このチェックボックスをオンにします。詳細については、[インスタンスの終了で Amazon EBS ボリュームを保持する \(p. 355\)](#) を参照してください。
8. [Tag Instance] ページで、キーと値の組み合わせを指定することによって、インスタンスのタグを指定します。[Create Tag] をクリックして、複数のタグをリソースに追加します。完了したら、[Next: Configure Security Group] をクリックします。

タグの詳細については、[Amazon EC2 リソースにタグを付ける \(p. 662\)](#) を参照してください。

9. [Configure Security Group] ページで、ウィザードは自動的に launch-wizard-x セキュリティグループを定義して、インスタンスに接続できるようにします。

セキュリティグループとは、インスタンスのファイアウォールルールを定義するものです。このルールでは、どの着信ネットワークトラフィックをインスタンスに配信するかを指定します。他のすべてのトラフィックは遮断されます。launch-wizard-x セキュリティグループは、Linux インスタンスの場合は SSH ( ポート 22 )、Windows インスタンスの場合は RDP ( ポート 3389 ) で自動的にトラフィックを許可します。



#### Caution

launch-wizard-x セキュリティグループは、すべての IP アドレス ( 0.0.0.0/0 ) が指定されたポート経由でインスタンスにアクセスできるようにします。これは、この短期間の演習では許容されますが、本稼働環境では安全ではありません。特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを限定してください。

ニーズに合わせて launch-wizard-x セキュリティグループのデフォルトのルールを変更できます。例えば、インスタンスをウェブサーバーとして使用する場合は、ポート 80 ( HTTP ) および 443 ( HTTPS ) を開いて、インターネットトラフィックがインスタンスに到達するのを許可します。

[Source] リストから [My IP] を選択し、ウィザードでコンピュータのパブリック IP アドレスを追加します。ただし、ISP 経由で、またはファイアウォールの内側から静的な IP アドレスなしで接続している場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。さらにルールをグループに追加するには、[Add Rule] をクリックします。

また、[Select an existing security group] オプションを選択して、既存のいずれかのセキュリティグループ ( セットアップ中に作成したものなど ) を使用することもできます。既存のグループのルールは編集できませんが、そのグループの [Copy to new] リンクをクリックすると、ルールを新しいグループにコピーすることができます。

セキュリティグループの詳細については、[Amazon EC2 セキュリティグループ \(p. 477\)](#) を参照してください。

終了したら、[Review and Launch] をクリックします。

10. [Review Instance Launch] ページで、インスタンスの詳細をチェックし、適切な [Edit] リンクをクリックして必要な変更を加えます。

準備ができたら、[Launch] をクリックします。

11. [Select an existing key pair or create a new key pair] ダイアログボックスで、既存のキーペアを選択するか、新しいキーペアを作成できます。例えば、[Choose an existing key pair] を選択し、セットアップ中に作成したキーペアを選択します。

インスタンスを起動するには、確認のチェックボックスをオンにし、[Launch Instances] をクリックします。

キーペアの作成の詳細については、[Amazon EC2 のキーペア \(p. 470\)](#) を参照してください。



#### Important

[Proceed without key pair] オプションを選択することをお勧めします。キーペアなしでインスタンスを起動すると、インスタンスに接続できなくなります。このオプションは、独自のAMIを作成するときに、インスタンスに接続する必要がない場合にのみ使用します。

12. インスタンスの状態が「running」にならずに、すぐに「terminated」になってしまう場合は、インスタンスが起動しなかった理由について確認できます。詳細については、「[インスタンスがすぐに終了する場合の対処方法 \(p. 685\)](#)」を参照してください。

## 既存のインスタンスをテンプレートとして使用したインスタンスを起動

Amazon EC2 コンソールには、現在のインスタンスを他のインスタンスを起動するテンプレートとして使用できるようにする、[Launch More Like This] ウィザードオプションが用意されています。このオプションでは、Amazon EC2 起動ウィザードで、選択されたインスタンスから自動的に特定の設定が入力されます。



#### Note

[Launch More Like This] ウィザードオプションでは、選択されたインスタンスは複製されません。一部の設定が複製されるのみです。インスタンスのコピーを作成するには、最初にインスタンスからAMIを作成して、AMIからさらに多くのインスタンスを起動します。

次の設定詳細は、選択されたインスタンスから起動ウィザードにコピーされます。

- AMI ID
- インスタンスタイプ
- アベイラビリティゾーン、または選択されたインスタンスがある VPC とサブネット
- 該当する場合は、インスタンスに関連付けられたタグ
- 該当する場合は、カーネル ID および RAM ディスク ID
- 該当する場合は、インスタンスに関連付けられた IAM ロール
- インスタンスに関連付けられたセキュリティグループ
- VPC (共有または専用) に起動する場合は、テナンシー設定
- Amazon EBS 最適化設定 (true または false)

次の設定の詳細は選択されたインスタンスからコピーされず、代わりにウィザードがデフォルトの設定または動作を適用します。

- ストレージ: デフォルトのストレージ設定は AMI およびインスタンスタイプによって決まります。
- パブリック IP アドレス: インスタンスにパブリック IP アドレスを割り当てるオプションは、デフォルトのサブネットに起動するときにデフォルトで有効になります。
- 終了保護: デフォルトで無効になります。
- シャットダウン動作: デフォルトで「stop」に設定されます。
- ユーザーデータ: デフォルトではありません。

現在のインスタンスをテンプレートとして使用するには

1. [インスタンス] ページで、使用するインスタンスを選択します。
2. [Actions] をクリックし、[Launch More Like This] を選択します。
3. [Review Instance Launch] ページで起動ウィザードが開きます。インスタンスの詳細をチェックし、適切な [Edit] リンクをクリックして、必要な変更を行うことができます。

準備ができたなら、[Launch] をクリックしてキーペアを選択し、インスタンスを起動します。

## バックアップからインスタンスを起動する

### Abstract

Amazon EBS スナップショットからインスタンスを起動します。

Amazon EBS でバックアップしたインスタンスを使うと、スナップショットを作成することで、インスタンスのルートデバイスボリュームをバックアップすることができます。インスタンスのルートデバイスボリュームのスナップショットがある場合、そのインスタンスを終了して、後でスナップショットから新しいインスタンスを起動できます。インスタンスの起動元のオリジナルの AMI がないけれども、同じイメージを使ってインスタンスを起動する必要がある場合に、これは便利です。



### Important

この時点で、スナップショットから Windows AMI を作成できますが、AMI からインスタンスを起動することはできません。

インスタンスのルートボリュームから AMI を作成するには、次の手順に従います。好みに応じて、[ec2-register](#) コマンドを使用することもできます。

ルートボリュームから AMI を作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Elastic Block Store] の [Snapshots] をクリックします。
3. Create Snapshot をクリックします。
4. [Volumes] フィールドで、ルートボリュームの名前または ID の入力を開始し、オプションのリストから選択します。
5. 先ほど作成したスナップショットを選択し、[Actions] リストから [Create Image] を選択します。
6. [Create Image from EBS Snapshot] ダイアログボックスで、AMI を作成するためのフィールドに入力し、[Create] をクリックします。必ず以下の操作を実行してください。
  - [Architecture] リスト ( 32 ビットの場合は [i386]、64 ビットの場合は [x86\_64] ) からアーキテクチャを選択します。
  - [Kernel ID] リストから AKI を選択します。デフォルトの AKI を選択する場合、あるいは AKI を選択しない場合も、インスタンスの起動時に必ず AKI を指定する必要があります。また、デフォルトの AKI がインスタンスと互換性がないために、インスタンスのヘルスチェックが失敗する場合があります。
7. ナビゲーションペインの [AMIs] を選択します。
8. 作成した AMI を選択し、[Launch] をクリックします。ウィザードにしたがって、インスタンスを起動します。ウィザードの各ステップの設定方法については、[インスタンスの起動 \(p. 325\)](#) を参照してください。

# AWS Marketplace インスタンスの起動

## Abstract

AWS Marketplace 製品を受信登録し、Amazon EC2 起動ウィザードを使用して AMI からインスタンスを起動します。

AWS Marketplace 製品を受信登録し、Amazon EC2 起動ウィザードを使用して、この製品の AMI からインスタンスを起動できます。有料の AMI の詳細については、[有料 AMI \(p. 73\)](#) を参照してください。起動後に受信登録をキャンセルするには、初めに受信登録から、実行されているすべてのインスタンスを終了する必要があります。詳細については、「[AWS Marketplace 受信登録を管理する \(p. 77\)](#)」を参照してください。

起動ウィザードを使用して AWS Marketplace からインスタンスを起動するには

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. Amazon EC2 ダッシュボードで、[Launch Instance] をクリックします。
3. [Choose an Amazon Machine Image (AMI)] ページで、左の [AWS Marketplace] カテゴリを選択します。カテゴリを参照するか、検索機能を使用して適切な AMI を見つけます。[Search] をクリックして製品を選択します。
4. ダイアログに、選択した製品の概要が表示されます。価格情報と、ベンダーが提供したその他の情報を表示できます。準備が完了したら、[Continue] をクリックします。



### Note

AMI でインスタンスを起動するまで、製品の使用料は発生しません。ウィザードの次のページでは、インスタンスタイプの選択が求められるため、サポートされているインスタンスタイプの料金をメモしておいてください。

5. [Choose an Instance Type] ページで、起動するインスタンスのハードウェア設定とサイズを選択します。終了したら、[Next: Configure Instance Details] をクリックします。
6. ウィザードの次のページでは、インスタンスの設定、ストレージの追加、およびタグの追加を行うことができます。設定できるさまざまなオプションの詳細については、[インスタンスの起動 \(p. 325\)](#) を参照してください。[Configure Security Group] ページが表示されるまで、[Next] をクリックします。

製品に関するベンダーの仕様にしたがって、新しいセキュリティグループが作成されます。セキュリティグループには、SSH (ポート 22) または RDP (ポート 3389) ですべての IP アドレス (0.0.0.0/0) を許可するルールが含まれる場合があります。これらのルールを調整して、特定の IP アドレスまたはアドレスの範囲のみが、これらの特定のポート経由でインスタンスにアクセスできるようにすることをお勧めします。

準備ができたら、[Review and Launch] をクリックします。

7. [Review Instance Launch] ページで、インスタンスを起動しようとしている AMI の詳細と、ウィザードでセットアップするその他の設定の詳細をチェックします。準備ができたら、[Launch] をクリックしてキーペアを選択または作成し、インスタンスを起動します。
8. 受信登録した製品によっては、インスタンスの起動には数分以上かかります。インスタンスが起動する前に、まず製品に登録されます。クレジットカードの詳細に問題がある場合は、アカウントの詳細を更新するように求められます。起動確認のページが表示されたら、[View Instances] をクリックして [インスタンス] ページに移動します。



#### Note

インスタンスが実行されている限り、アイドル状態であっても、受信登録費用が発生します。インスタンスが停止している場合でも、ストレージに対して課金されることがあります。

9. インスタンスの状態が [running] の場合、そのインスタンスに接続することができます。そのためには、一覧でインスタンスを選択し、[Connect] をクリックします。ダイアログの指示にしたがいます。インスタンスへの接続の詳細については、[インスタンスへの接続 \(p. 334\)](#) を参照してください。



#### Important

インスタンスにログインするには、特定のユーザー名を使用しなければならない場合があります。そのため、ベンダーの使用手順を慎重に確認してください。受信登録の詳細へのアクセスについては、[AWS Marketplace 受信登録を管理する \(p. 77\)](#) を参照してください。

## API と CLI を使用した AWS Marketplace AMI インスタンスの起動

API またはコマンドラインツールを使用して、AWS Marketplace 製品からインスタンスを起動するには、まず製品に登録していることを確認します。次の方法を使用して、製品の AMI ID でインスタンスを起動できます。

メソッド	ドキュメント
AWS CLI	<a href="#">run-instances</a> コマンドを使用するか、詳細について「 <a href="#">インスタンスの起動</a> 」を参照してください。
Amazon EC2 CLI	<a href="#">ec2-run-instances</a> コマンドを使用するか、詳細について <a href="#">Amazon EC2 CLI を使用したインスタンスの起動</a> を参照してください。
AWS Tools for Windows PowerShell	<a href="#">New-EC2Instance</a> コマンドを使用するか、詳細について <a href="#">Windows PowerShell を使用した Amazon EC2 インスタンスの起動</a> を参照してください。
クエリ API	<a href="#">RunInstances</a> リクエストを使用します。

# インスタンスへの接続

## Abstract

インスタンスに接続し、ローカルコンピュータとの間でデータを転送します。

このセクションでは、起動したインスタンスに接続する方法と、ローカルコンピュータとインスタンスの間でファイルを転送する方法について説明します。

コンピュータ	ハードウェア	トピック
Linux/Unix	Linux/Unix	<a href="#">SSH を使用した Linux/Unix インスタンスへの接続 (p. 334)</a>
Windows	Linux/Unix	<a href="#">PuTTY を使用した Windows から Linux/Unix インスタンスへの接続 (p. 338)</a>
すべて	Linux/Unix	<a href="#">Mindterm を使ったウェブブラウザからインスタンスへの接続 (p. 343)</a>
All	Windows	<a href="#">RDP を使用した Windows インスタンスへの接続 (p. 344)</a>

インスタンスに接続した後で、[チュートリアル: LAMP ウェブサーバーのインストール \(p. 39\)](#) や [チュートリアル: Amazon EC2 を使った WordPress ブログのホスティング \(p. 46\)](#) など、いずれかのチュートリアルを試すことができます。

## SSH を使用した Linux/Unix インスタンスへの接続

### Abstract

SSH クライアントを使用して Linux/Unix インスタンスに接続します。

インスタンスを起動したら、これに接続し、普通のコンピュータと同じように使用できます。

インスタンスの接続でエラーが発生した場合は、「[Troubleshooting Connecting to Your Instance](#)」を参照してください。

次の手順では、SSH クライアントを使って、インスタンスに接続する方法について説明します。

#### Topics

- [前提条件 \(p. 334\)](#)
- [Linux/Unix インスタンスへの接続 \(p. 335\)](#)
- [SCP による Linux/Unix から Linux/Unix インスタンスへのファイルの転送 \(p. 336\)](#)

## 前提条件

- SSH クライアントのインストール  
Linux コンピュータではほとんどの場合、デフォルトで SSH クライアントが含まれています。SSH クライアントがあるかどうかを確認するには、コマンドラインで ssh と入力します。使用しているコンピュータでコマンドが認識されない場合は、OpenSSH プロジェクトから、SSH ツールの完全なスイートの無料実装が提供されています。詳細については、<http://www.openssh.org> を参照してください。
- Amazon EC2 CLI ツールをインストールする

( オプション ) サードパーティのパブリック AMI を使用する場合は、`ec2-get-console-output` コマンドを使用してフィンガープリントを確認します。

- インスタンスの ID を取得する  
Amazon EC2 コンソールを使用して、インスタンスの ID を取得できます ( [Instance ID] 列を確認します )。もしくは、`ec2-describe-instances` コマンドを使用できます。
- インスタンスのパブリック DNS 名を取得する  
Amazon EC2 コンソールを使用して、インスタンスのパブリック DNS を取得できます ( [Public DNS] 列を確認します。この列が非表示の場合は、[Show/Hide] アイコンをクリックして [Public DNS] を選択します )。もしくは、`ec2-describe-instances` コマンドを使用できます。
- プライベートキーを見つける  
インスタンスの起動時に指定したキーペアの `.pem` ファイルの完全修飾パスが必要です。
- IP アドレスからインスタンスへのインバウンド SSH トラフィックを有効にする  
インスタンスに関連付けられているセキュリティグループで、IP アドレスからの受信 SSH トラフィックが許可されることを確認します。詳細については、「[Authorizing Network Access to Your Instances](#)」を参照してください。

## Linux/Unix インスタンスへの接続

SSH を使用してインスタンスに接続するには

1. ( オプション ) サードパーティ製のパブリック AMI を起動している場合、インスタンス上ではなく、ローカルシステム上で `ec2-get-console-output` コマンドを実行し、SSH HOST KEY FINGERPRINTS セッションを割り当てます。インスタンスのフィンガープリントと比較できるように、フィンガープリント ( `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f` など ) を控えておきます。

```
ec2-get-console-output instance_id
```

2. コマンドラインシェルで、インスタンスを起動した時に作成した秘密キーファイルの場所にディレクトリを変更します。
3. `chmod` コマンドを使って、秘密キーファイルが公開されていないことを確認します。例えば、秘密キーファイルが `my-key-pair.pem` という名前の場合、次のコマンドを使用します。

```
chmod 400 my-key-pair.pem
```

4. `ssh` コマンドを使って、インスタンスに接続します。秘密キー ( `.pem` ) ファイルと `user_name@public_dns_name` を指定します。Amazon Linux の場合は、デフォルトのユーザー名は `ec2-user` です。RHEL5 の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。Ubuntu の場合は、ユーザー名は `ubuntu` です。SUSE Linux の場合は、ユーザー名は `root` です。それ以外の場合は、AMI プロバイダーに確認してください。

```
ssh -i my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

以下のようなレスポンスが表示されます。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is
```

```
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.  
Are you sure you want to continue connecting (yes/no)?
```

- (オプション)パブリック AMI を起動している場合、セキュリティアラートのフィンガープリントが、ステップ1で入手したフィンガープリントと一致していることを確認します。これらのフィンガープリントが一致しない場合、「中間者 (MITM)」攻撃を受けている可能性があります。一致した場合は、次の手順に進んでください。
- yes と入力します。

以下のようなレスポンスが表示されます。

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)  
to the list of known hosts.
```

## SCP による Linux/Unix から Linux/Unix インスタンスへのファイルの転送

ローカルコンピュータと Linux/Unix インスタンスの間でファイルを転送する方法の1つとして、Secure Copy (SCP) を使用します。このセクションでは、SCP でファイルを転送する方法について説明します。この手順は、SSH を使用してインスタンスに接続する手順とよく似ています。

### 前提条件

- SCP クライアントのインストール  
ほとんどの Linux、Unix、および Apple コンピュータには、デフォルトで SCP クライアントが含まれています。含まれていない場合は、OpenSSH プロジェクトから、SSH ツールの完全なスイートの無料実装が提供されており、これに SCP クライアントが含まれます。詳細については、<http://www.openssh.org> を参照してください。
- インスタンスの ID を取得する  
Amazon EC2 コンソールを使用して、インスタンスの ID を取得できます ([Instance ID] 列を確認します)。もしくは、`ec2-describe-instances` コマンドを使用できます。
- インスタンスのパブリック DNS 名を取得する  
Amazon EC2 コンソールを使用して、インスタンスのパブリック DNS を取得できます ([Public DNS] 列を確認します。この列が非表示の場合は、[Show/Hide] アイコンをクリックして [Public DNS] を選択します)。もしくは、`ec2-describe-instances` コマンドを使用できます。
- プライベートキーを見つける  
インスタンスの起動時に指定したキーペアの `.pem` ファイルの完全修飾パスが必要です。
- IP アドレスからインスタンスへのインバウンド SSH トラフィックを有効にする  
インスタンスに関連付けられているセキュリティグループで、IP アドレスからの受信 SSH トラフィックが許可されることを確認します。詳細については、「[Authorizing Network Access to Your Instances](#)」を参照してください。

SCP を使用してファイルを転送するステップを次に示します。既に SSH でインスタンスに接続し、フィンガープリントの確認が完了している場合は、SCP コマンドを実行するステップ (ステップ 4) から開始できます。

SCP を使用してファイルを転送するには

1. (オプション) サードパーティ製のパブリック AMI を起動している場合、インスタンス上ではなく、ローカルシステム上で `ec2-get-console-output` コマンドを実行し、SSH HOST KEY FINGERPRINTS セッションを割り当てます。インスタンスのフィンガープリントと比較できるように、フィンガープリント ( `1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f` など ) を控えておきます。

```
ec2-get-console-output instance_id
```

2. コマンドラインシェルで、インスタンスを起動した時に指定した秘密キーファイルの場所にディレクトリを変更します。
3. `chmod` コマンドを使って、秘密キーファイルが公開されていないことを確認します。例えば、秘密キーファイルが `my-key-pair.pem` という名前の場合、次のコマンドを使用します。

```
chmod 400 my-key-pair.pem
```

4. インスタンスのパブリック DNS 名を使って、インスタンスにファイルを転送します。例えば、秘密キーファイルの名前が `my-key-pair`、転送するファイルが `SampleFile.txt`、インスタンスのパブリック DNS の名前が `ec2-198-51-100-1.compute-1.amazonaws.com` の場合、次のコマンドを使って、ファイルを `ec2-user` ホームディレクトリにコピーします。

```
scp -i my-key-pair.pem SampleFile.txt ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~
```



#### Tip

Amazon Linux の場合は、デフォルトのユーザー名は `ec2-user` です。RHEL5 の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。Ubuntu の場合は、ユーザー名は `ubuntu` です。SUSE Linux の場合は、ユーザー名は `root` です。それ以外の場合は、AMI プロバイダーに確認してください。

以下のようなレスポンスが表示されます。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

5. (オプション) パブリック AMI を起動している場合、セキュリティアラートのフィンガープリントが、ステップ1で入手したフィンガープリントと一致していることを確認します。これらのフィンガープリントが一致しない場合、「中間者 (MITM)」攻撃を受けている可能性があります。一致した場合は、次の手順に進んでください。
6. `yes` と入力します。

以下のようなレスポンスが表示されます。

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
```

```
Sending file modes: C0644 20 SampleFile.txt
Sink: C0644 20 SampleFile.txt
SampleFile.txt                                100%   20    0.0KB/s   00:00
```

逆の方向 ( Amazon EC2 インスタンスからローカルコンピュータに ) にファイルを転送する場合は、ホストパラメータの順番を逆にするだけです。例えば、SampleFile.txt ファイルを EC2 インスタンスからローカルコンピュータのホームディレクトリに SampleFile2.txt として転送するには、ローカルコンピュータで次のコマンドを実行します。

```
scp -i my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com:~/SampleFile.txt ~/SampleFile2.txt
```

## PuTTY を使用した Windows から Linux/Unix インスタンスへの接続

### Abstract

Windows 向けの無料の SSH クライアントである PuTTY を使用して、Linux/Unix インスタンスに接続します。

インスタンスを起動したら、これに接続し、普通のコンピュータと同じように使用できます。

インスタンスの接続でエラーが発生した場合は、「[Troubleshooting Connecting to Your Instance](#)」を参照してください。

次の手順では、Windows 用の無料の SSH クライアントである PuTTY を使用して、インスタンスに接続する方法について説明します。

### Topics

- [前提条件 \(p. 338\)](#)
- [PuTTYgen を使用した秘密キーの変換 \(p. 339\)](#)
- [PuTTY セッションの開始 \(p. 339\)](#)
- [PuTTY Secure Copy Client を使った、インスタンスへのファイルの転送 \(p. 341\)](#)
- [WinSCP を使った、インスタンスへのファイルの転送 \(p. 342\)](#)

## 前提条件

- PuTTY のインストール  
[PuTTY のダウンロードページ](#)から、PuTTY をダウンロードしてインストールします。必ずスイート全体をインストールしてください。
- インスタンスの ID を取得する  
Amazon EC2 コンソールを使用して、インスタンスの ID を取得できます ( [Instance ID] 列を確認します )。もしくは、[ec2-describe-instances](#) コマンドを使用できます。
- インスタンスのパブリック DNS 名を取得する  
Amazon EC2 コンソールを使用して、インスタンスのパブリック DNS を取得できます ( [Public DNS] 列を確認します。この列が非表示の場合は、[Show/Hide] アイコンをクリックして [Public DNS] を選択します )。もしくは、[ec2-describe-instances](#) コマンドを使用できます。
- プライベートキーを見つける  
インスタンスの起動時に指定したキーペアの .pem ファイルの完全修飾パスが必要です。
- IP アドレスからインスタンスへのインバウンド SSH トラフィックを有効にする

インスタンスに関連付けられているセキュリティグループで、IP アドレスからの受信 SSH トラフィックが許可されることを確認します。詳細については、「[Authorizing Network Access to Your Instances](#)」を参照してください。

## PuTTYgen を使用した秘密キーの変換

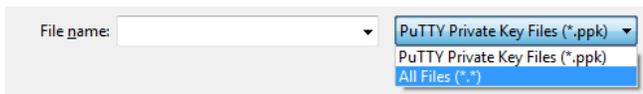
PuTTY は、Amazon EC2 によって生成される秘密キー形式 (.pem) をネイティブにサポートしていません。PuTTY には、キーを必要な PuTTY 形式 (.ppk) に変換できる PuTTYgen と呼ばれるツールが含まれています。PuTTY を使用してインスタンスへの接続を試みる前に、秘密キーをこの形式 (.ppk) に変換する必要があります。

### プライベートキーの変換方法

1. PuTTYgen を開始します (例:[Start] メニューで All Programs > PuTTY > PuTTYgen をクリック)。
2. [Type of key to generate] の下で、[SSH-2 RSA] を選択します。



3. [Load] をクリックします。デフォルトでは、PuTTYgen には拡張子 .ppk を持つファイルだけが表示されます。.pem ファイルの場所を特定するには、すべてのタイプのファイルを表示するオプションを選択します。



4. .pem ファイルを選択し、[Open] をクリックします。[OK] をクリックして、確認ダイアログボックスを閉じます。
5. [Save private key] をクリックして、PuTTY が使用できる形式でキーを保存します。PuTTYgen に、パスフレーズなしでキーを保存することに関する警告が表示されます。[Yes] をクリックします。



### Note

秘密キーのパスフレーズには、保護を強化する働きがあります。秘密キーが他者に見つかったとしても、パスフレーズがなければキーは使用不可能です。パスフレーズ使用における欠点は、インスタンスにログオンする、またはファイルをインスタンスにコピーするのに人間の介入が必要となるため、自動化が難しくなることです。

6. キーペアに使用した名前と同じ名前をキーに指定します (例: my-key-pair)。PuTTY は自動的にファイル拡張子 .ppk を加えます。

プライベートキーが PuTTY で使用するための正しい形式となりました。これで、PuTTY の SSH クライアントを使用してインスタンスに接続することができます。

## PuTTY セッションの開始

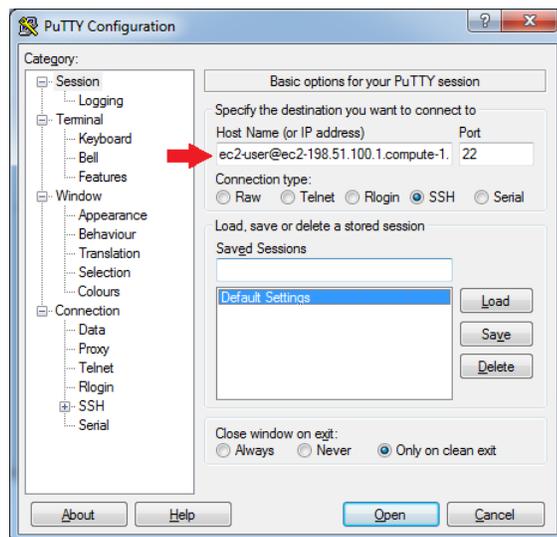
### PuTTY セッションの開始方法

1. (オプション) サードパーティ製のパブリック AMI を起動している場合、インスタンス上ではなく、ローカルシステム上で `ec2-get-console-output` コマンドを実行し、SSH HOST KEY

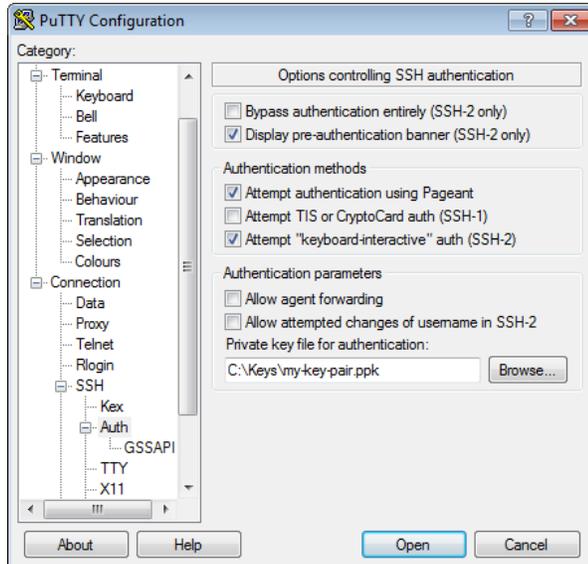
FINGERPRINTS セッションを割り当てます。インスタンスのフィンガープリントと比較できるように、フィンガープリント ( 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f など ) を控えておきます。

```
C:\> ec2-get-console-output instance_id
```

2. PuTTY を開始します ( [Start] メニューで All Programs > PuTTY > PuTTY をクリックします ) 。
3. [Category] ペインで [Session] を選択し、次のフィールドに入力します。
  - a. [Host Name] フィールドで、*user\_name@public\_dns\_name* を入力します。AMI に対して、必ず適切なユーザー名を指定してください。以下に例を示します。
    - Amazon Linux AMI の場合は、ユーザー名は *ec2-user* です。
    - RHEL5 AMI の場合は、ほとんどの場合ユーザー名は *root* ですが、*ec2-user* の場合もあります。
    - Ubuntu AMI の場合は、ユーザー名は *ubuntu* です。
    - それ以外の場合は、AMI プロバイダーに確認してください。
  - b. [Connection type] で [SSH] を選択します。
  - c. [Port] が 22 であることを確認します。



4. [Category] ペインで、[Connection]、[SSH] の順に展開し、[Auth] を選択します。次のように入力します。
  - a. [Browse] をクリックします。
  - b. キーペア用に生成した .ppk ファイルを選択し、[Open] をクリックします。
  - c. ( オプション ) 後でこのセッションを再度開始する場合、将来使用できるようにセッション情報を保存できます。[Category] ツリーで [Session] を選択し、[Saved Sessions] にセッションの名前を入力して、[Save] をクリックします。
  - d. [Open] をクリックして、PuTTY セッションを開始します。



5. このインスタンスに接続するのが初めての場合は、接続先のホストを信頼するかどうかを尋ねるセキュリティアラートダイアログボックスが表示されます。
6. (オプション) パブリック AMI を起動している場合、セキュリティアラートのフィンガープリントが、ステップ 1 で入手したフィンガープリントと一致していることを確認します。これらのフィンガープリントが一致しない場合、「中間者 (MITM)」攻撃を受けている可能性があります。一致した場合は、次の手順に進んでください。
7. [Yes] をクリックします。ウィンドウが開き、インスタンスに接続した状態になります。



#### Note

秘密キーを PuTTY 形式に変換するときにパスフレーズを指定した場合は、インスタンスへのログイン時にそのパスフレーズを指定する必要があります。

## PuTTY Secure Copy Client を使った、インスタンスへのファイルの転送

PuTTY Secure Copy Client (PSCP) は、Windows コンピュータと Linux/Unix インスタンスの間でファイルを転送できるようにするコマンドラインツールです。グラフィカルユーザーインターフェイス (GUI) を使用する場合は、WinSCP という名前のオープンソース GUI ツールを使用できます。詳細については、「[WinSCP を使った、インスタンスへのファイルの転送 \(p. 342\)](#)」を参照してください。

PSCP を使用するには、[PuTTYgen を使用した秘密キーの変換 \(p. 339\)](#) で生成したプライベートキーが必要です。また、Linux/Unix インスタンスのパブリック DNS アドレスも必要です。

次の例では、ファイル `sample_file.txt` を Windows コンピュータから Linux/Unix インスタンス上の `/usr/local` ディレクトリに転送します。

```
C:\> pscp -i C:\Keys\my-key-pair.ppk C:\sample_file.txt user_name@pub
lic_dns:/usr/local/sample_file.txt
```

## WinSCP を使った、インスタンスへのファイルの転送

WinSCP は Windows 用の GUI ベースのファイルマネージャで、SFTP、SCP、FTP、および FTPS プロトコルを使って、ファイルをリモートコンピュータにアップロードおよび転送することができます。WinSCP を使用すると、Windows マシンから Linux インスタンスにファイルをドラッグアンドドロップしたり、2つのシステム間でディレクトリ構造全体を同期させることができます。

WinSCP を使用するには、[PuTTYgen を使用した秘密キーの変換 \(p. 339\)](#) で生成した秘密キーが必要です。また、Linux/Unix インスタンスのパブリック DNS アドレスも必要です。

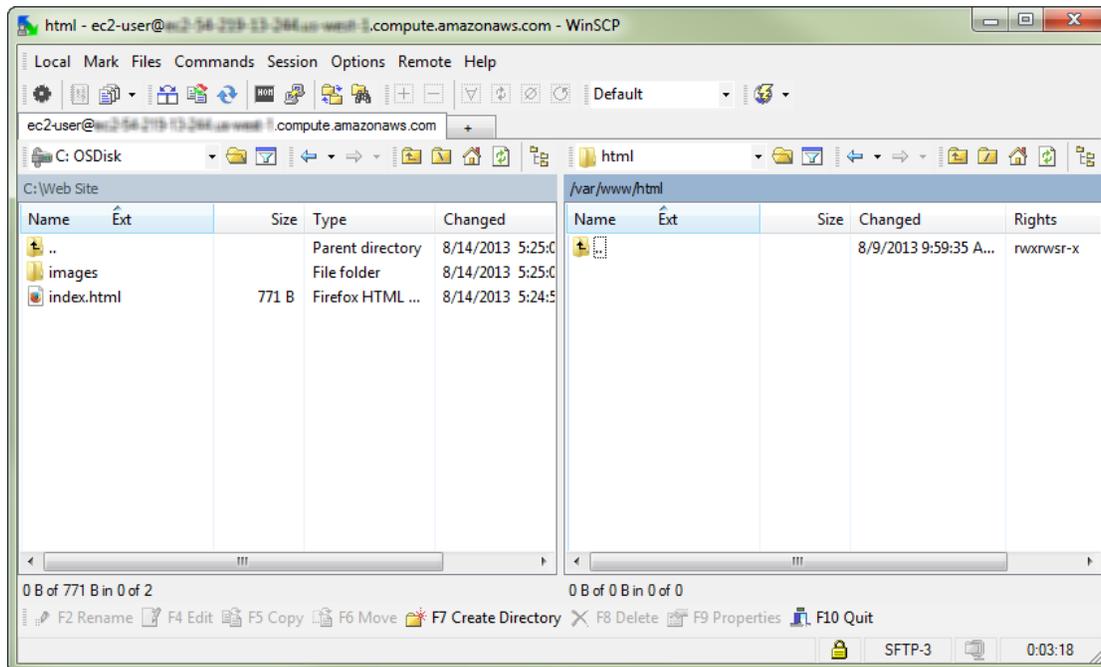
1. <http://winscp.net/eng/download.php> から WinSCP をダウンロードしてインストールします。ほとんどの場合、デフォルトのインストールオプションでかまいません。
2. WinSCP を起動します。
3. [WinSCP login] 画面で、インスタンスのパブリック DNS アドレスを [Host name] に入力します。
4. [Username] については、AMI のデフォルトユーザー名を入力します。Amazon Linux AMI の場合、ユーザー名は、`ec2-user` です。Red Hat AMI の場合、ユーザー名は `root` で、Ubuntu AMI の場合、ユーザー名は `ubuntu` です。
5. [Private key] に対して、秘密キーのパスを入力するか、[...] ボタンをクリックして、ファイルをブラウズします。



### Note

WinSCP は PuTTY 秘密キーファイル ( `.ppk` ) ファイルを必要とします。PuTTYgen を使って、`.pem` セキュリティキーファイルを `.ppk` フォーマットに変換することができます。詳細については、「[PuTTYgen を使用した秘密キーの変換 \(p. 339\)](#)」を参照してください。

6. ( オプション ) 左側のパネルで [Directories] をクリックし、[Remote directory] に対してファイルを追加するディレクトリのパスを入力します。
7. [Login] をクリックして接続し、[Yes] をクリックしてホストのフィンガープリントをホストのキャッシュに追加します。



8. 接続確立後、接続ウィンドウには Linux インスタンスが右側、ローカルマシンが左側に表示されます。ローカルマシンからリモートファイルシステムへ、ファイルを直接ドラッグアンドドロップすることができます。WinSCP の詳細については、<http://winscp.net/eng/docs/start> のドキュメントを参照してください。

## Mindterm を使ったウェブブラウザからインスタンスへの接続

### Abstract

MindTerm を使って、ウェブブラウザからインスタンスに接続します。

インスタンスを起動したら、これに接続し、普通のコンピュータと同じように使用できます。

インスタンスの接続でエラーが発生した場合は、「[Troubleshooting Connecting to Your Instance](#)」を参照してください。

次の手順では、Amazon EC2 コンソールを介して、Mindterm を使ってインスタンスに接続する方法について説明します。

### 前提条件

- Install Java  
ご使用の Linux コンピュータには通常、Java が組み込まれています。Java が組み込まれていない場合、[ウェブブラウザで Java を有効にするにはどうすればいいですか?](#) Windows または Mac クライアント上で、管理者認証情報を使用してブラウザを実行する必要があります。Linux の場合、`root` としてログインしていない場合は、追加のステップが必要になる場合があります。
- ご使用のブラウザで Java を有効にする  
手順については、[http://java.com/en/download/help/enable\\_browser.xml](http://java.com/en/download/help/enable_browser.xml) を参照してください。
- プライベートキーを見つける  
インスタンスの起動時に指定したキーペアの `.pem` ファイルの完全修飾パスが必要です。
- IP アドレスからインスタンスへのインバウンド SSH トラフィックを有効にする  
インスタンスに関連付けられているセキュリティグループで、IP アドレスからの受信 SSH トラフィックが許可されることを確認します。詳細については、「[Authorizing Network Access to Your Instances](#)」を参照してください。

### MindTerm の起動

MindTerm を使用して、ウェブブラウザによりインスタンスに接続するには

1. Amazon EC2 コンソール内で、ナビゲーションペインの [Instances] をクリックします。
2. インスタンスを選択し、[Connect] をクリックします。
3. [A Java SSH client directly from my browser (Java required)] をクリックします。
4. Amazon EC2 は、インスタンスのパブリック DNS 名を自動的に検出し、[Public DNS] に自動的に名前を入力します。また、インスタンスの起動時に指定したキーペアの名前も検出されます。次のように入力し、[Launch SSH Client] をクリックします。
  - a. [User name] に、インスタンスにログインするためのユーザー名を入力します。



#### Tip

Amazon Linux の場合は、デフォルトのユーザー名は `ec2-user` です。RHEL5 の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。Ubuntu の場合は、ユーザー名は `ubuntu` です。SUSE Linux の場合は、ユーザー名は `root` です。それ以外の場合は、AMI プロバイダーに確認してください。

- b. Private key path に、秘密キー ( `.pem` ) ファイルの完全修飾パスを入力します。
  - c. ( オプション ) [Store in browser cache] をクリックして、ブラウザキャッシュにプライベートキーの場所を保存します。これにより、Amazon EC2 はユーザーがブラウザのキャッシュをクリアするまで、以降のブラウザセッションで秘密キーの場所を検出します。
5. 必要に応じて、証明書を信頼するかどうかを指定する画面で [Yes] をクリックします。
  6. [Run] をクリックして MindTerm クライアントを実行します。
  7. ライセンス契約書に同意する場合は、[Accept] をクリックします。
  8. MindTerm を実行するのが初めての場合、ホームディレクトリなどの設定を確認するための一連のダイアログボックスが表示されます。これらの設定を確認します。ウィンドウが開き、インスタンスに接続した状態になります。

## RDP を使用した Windows インスタンスへの接続

### Abstract

RDP クライアントを使用して、Windows インスタンスに接続します。

インスタンスを起動したら、これに接続し、普通のコンピュータと同じように使用できます。

インスタンスに接続しようとするエラーが表示される場合は、「*Amazon Elastic Compute Cloud Microsoft Windows Guide*」の「[Windows インスタンスのトラブルシューティング](#)」を参照してください。

次の手順では、RDP クライアントを使ってインスタンスに接続する方法を説明します。

### Topics

- [前提条件 \(p. 344\)](#)
- [Windows インスタンスへの接続 \(p. 345\)](#)
- [Windows から Windows Server インスタンスへのファイルの転送 \(p. 346\)](#)

## 前提条件

- RDP クライアントのインストール  
Windows コンピュータにはデフォルトで、RDP クライアントが搭載されています。RDP クライアントを確認するには、コマンドプロンプトウィンドウで `mstsc` と入力します。お使いのコンピュータがこのコマンドを認識しない場合は、[Microsoft Windows ホームページ](#)を参照し、「リモートデスクトップ接続」を検索してダウンロードしてください。Mac OS X の場合は、[Microsoft のリモートデスクトップクライアント](#)を使います。Linux/Unix の場合は、`rdesktop` を使用できます。
- インスタンスの ID を取得する  
Amazon EC2 コンソールを使用して、インスタンスの ID を取得できます ( [Instance ID] 列を確認します )。もしくは、`ec2-describe-instances` コマンドを使用できます。
- インスタンスのパブリック DNS 名を取得する

Amazon EC2 コンソールを使用して、インスタンスのパブリック DNS を取得できます ([Public DNS] 列を確認します。この列が非表示の場合は、[Show/Hide] アイコンをクリックして [Public DNS] を選択します)。もしくは、[ec2-describe-instances](#) コマンドを使用できます。

- プライベートキーを見つける  
インスタンスの起動時に指定したキーペアの .pem ファイルの完全修飾パスが必要です。
- IP アドレスからインスタンスへのインバウンド RDP トラフィックを有効にする  
インスタンスに関連付けられているセキュリティグループで、IP アドレスからの受信 RDP トラフィックが許可されることを確認します。詳細については、「[Authorizing Network Access to Your Instances](#)」を参照してください。
- Internet Explorer で最良の結果を得るには、最新バージョンを実行してください。

## Windows インスタンスへの接続

Windows インスタンスに接続するには、初期の管理者パスワードを取得する必要があります。次に、リモートデスクトップを使用してインスタンスに接続するときに、このパスワードを指定します。



### Note

Windows インスタンスでは、一度に 2 つの同時リモート接続しか許可されていません。3 つめの接続を行おうとすると、エラーが発生します。詳細については、「[1 つの接続で実行可能な同時リモート接続数を構成する](#)」を参照してください。

Windows インスタンスに接続するには

1. Amazon EC2 コンソールでは、インスタンスを選択し、[Connect] をクリックします。
2. [Connect To Your Instance] ダイアログボックスで、[Get Password] をクリックします (インスタンスが起動してからパスワードが使用できるようになるまで数分かかります)。
3. [Browse] をクリックし、インスタンスの起動時に作成したプライベートキーファイルを探します。ファイルを選択して [Open] をクリックすると、ファイルの内容がすべて [Contents] ボックスにコピーされます。
4. [Decrypt Password] をクリックします。コンソールは、インスタンスのデフォルトの管理者パスワードを [Connect To Your Instance] ダイアログボックスに表示し、以前の実際のパスワードが [Get Password] へのリンクに置き換えられます。
5. デフォルトの管理者パスワードを記録するか、クリップボードにコピーします。このパスワードはインスタンスに接続するのに必要です。
6. [Download Remote Desktop File] をクリックします。ブラウザによって .rdp ファイルを開くか、保存するよう求められます。どちらでもかまいません。終了したら、[Close] をクリックして [Connect To Your Instance] ダイアログボックスを閉じます。
7. .rdp ファイルを開いた場合は、[Remote Desktop Connection] ダイアログボックスが表示されます。.rdp ファイルを保存した場合は、ダウンロードしたディレクトリまで移動して .rdp ファイルをダブルクリックすると、このダイアログボックスが表示されます。リモート接続の発行元が不明であるという警告が表示されることがあります。[Connect] をクリックしてインスタンスに接続します。セキュリティ証明書を認証できなかったという警告が表示されることがあります。[Yes] をクリックして続行します。
8. メッセージが表示されたら、以前記録またはコピーしたデフォルトの [Administrator] アカウントとデフォルトの管理者パスワードを使って、インスタンスにログインします。

接続したら、次の操作を実行することをお勧めします。

- 管理者パスワードをデフォルト以外の値に変更します。パスワードの変更は、そのインスタンスにログインした状態で行います。これは通常の Windows Server と同じです。

- 管理者権限を持つユーザーアカウントをもう1つインスタンスに作成します。これは管理者パスワードを忘れた場合や、管理者アカウントで問題が発生した場合の安全策です。

## Windows から Windows Server インスタンスへのファイルの転送

インスタンスの使い方は、通常の Windows Server と同じです。例えば、Windows リモートデスクトップのローカルファイル共有機能を使用して、Amazon EC2 Windows インスタンスとローカルの Windows コンピュータの間でファイルを転送できます。Windows リモートデスクトップ接続ソフトウェアでこのオプションを有効にすると、Amazon EC2 Windows インスタンスからローカルファイルにアクセスできます。例えば、ハードディスクドライブ、DVD ドライブ、ポータブルメディアドライブ、およびマップされたネットワークドライブ上のローカルファイルにアクセスすることができます。この機能については、[Microsoft サポートウェブサイト](#)または MSDN ブログウェブサイトの [The most useful feature of Remote Desktop I never knew about](#) を参照してください。

## インスタンスの停止と起動

### Abstract

ルートデバイスとして Amazon EBS ボリュームを使用しているすべてのインスタンスを停止して起動します。

インスタンスにルートデバイスとして Amazon EBS ボリュームがある場合、そのインスタンスを停止して再起動できます。インスタンスにはそのインスタンス ID が保持されますが、概要セクションで述べられているように変更することはできません。

ユーザーがインスタンスを停止すると、インスタンスはシャットダウンされます。停止されているインスタンスの毎時使用量またはデータ転送料金に対して課金しませんが、Amazon EBS ボリュームのストレージに対しては課金します。この切り替えを1時間以内に複数回行う場合でも、停止されているインスタンスを起動するたびに、全体のインスタンス時間に対して課金します。Amazon EBS-Backed インスタンスを停止し、`stopping` 状態に「`stuck`」が表示されている場合、インスタンスを強制終了できます。詳細については、[インスタンスの停止に関するトラブルシューティング \(p. 691\)](#) を参照してください。

インスタンスが停止している間、他のボリュームと同様にそのルートボリュームを扱い、変更することができます (ファイルシステムの問題を修復したり、ソフトウェアを更新したりするなど)。停止しているインスタンスからボリュームを接続解除し、それを実行中のインスタンスに接続して、変更を行い、実行中のインスタンスから接続解除して、停止しているインスタンスに再接続します。インスタンスのブロックデバイスマッピングにルートデバイスとして指定されたストレージデバイス名を使用して、ボリュームを接続解除していることを確認します。

インスタンスが必要なくなったら、終了することができます。インスタンスの状態が `shutting-down` または `terminated` に変わったら、そのインスタンスへの課金は停止します。詳細については、「[インスタンスの終了 \(p. 352\)](#)」を参照してください。

## 概要

停止できるのは Amazon EBS-Backed インスタンスだけです。インスタンスのルートデバイスタイプを確認するには、インスタンスを記述し、そのルートボリュームのデバイスタイプが `ebs` (Amazon EBS-Backed インスタンス) が `instance store` (Instance store-Backed インスタンス) かをチェックします。

実行中のインスタンスを停止すると、次の処理が実行されます。

- インスタンスは正常なシャットダウンを実行し、実行を停止します (ステータスは `stopping`、次に `stopped` に変わります)。
- Amazon EBS ボリュームはインスタンスに接続されたままとなり、そのデータは保持されます。
- ホストコンピュータの RAM またはホストコンピュータのインスタンスストアボリュームに保存されたデータはなくなります。
- EC2-Classic: インスタンスを停止すると、インスタンスのパブリックおよびプライベート IP アドレスを解放し、インスタンスを再起動すると、新しい IP アドレスを割り当てます。

EC2-VPC: 停止および再起動される際に、インスタンスはプライベート IP アドレスを保持します。パブリック IP アドレスを解放し、再起動されると新しいアドレスを割り当てます。

- EC2-Classic: インスタンスに関連付けられている Elastic IP アドレスの関連付けを解除します。インスタンスに関連付けられていない Elastic IP アドレスに対して課金されます。インスタンスを再起動すると、インスタンスと Elastic IP アドレスを関連付ける必要があります。これを自動的に行うことはありません。

EC2-VPC: インスタンスには関連付けられた Elastic IP アドレスが保持されます。停止されているインスタンスに関連付けられた Elastic IP アドレスに対して課金されます。

- Windows インスタンスを停止および再起動すると、デフォルトでインスタンスホスト名を、新しい IP アドレスに合わせて変更し、再起動を開始します。デフォルトでは、接続されている Amazon EBS ボリュームのドライブ文字も変更します。これらのデフォルトおよび変更方法については、『[Amazon Elastic Compute Cloud Microsoft Windows Guide](#)』の「[Configuring a Windows Instance Using the EC2Config Service](#)」を参照してください。
- インスタンスをロードバランサーに登録した場合、インスタンスを停止して再起動すると、ロードバランサーが、そのインスタンスに対してトラフィックのルート割り当てを行えなくなる可能性があります。インスタンスを停止したら、ロードバランサーにおけるそのインスタンスの登録は一度解除し、インスタンスの起動後に再度登録する必要があります。詳細については、『[Elastic Load Balancing 開発者ガイド](#)』の「[De-Registering and Registering Amazon EC2 Instances](#)」を参照してください。

以下のインスタンスの属性は停止されると、変更できます。

- インスタンスタイプ
- ユーザーデータ
- Kernel
- RAM ディスク

インスタンスの実行中にこれらの属性を変更しようとする、Amazon EC2 が `IncorrectInstanceState` エラーを返します。

## インスタンスの停止と起動

コンソールまたはコマンドラインを使用して、Amazon EBS-Backed インスタンスを起動および停止できます。

コンソールを使用して Amazon EBS-Backed インスタンスを停止および起動するには

1. ナビゲーションペインで [Instances] をクリックし、インスタンスを変更します。
2. [EC2-Classic] インスタンスに関連付けられた Elastic IP アドレスがある場合は、詳細ペインに表示された Elastic IP アドレスとインスタンス ID を書き留めます。
3. [Actions] をクリックして、[Stop] をクリックします。[Stop] が無効になっている場合は、インスタンスが既に停止しているか、またはルートボリュームがインスタンスストアボリュームです。
4. 確認ダイアログボックスで [Yes, Stop] をクリックします。インスタンスが停止するまで、数分かかる場合があります。

[EC2-Classic] インスタンスの状態が `stopped` になると、詳細ペインの [Elastic IP]、[Public DNS]、[Private DNS]、および [Private IPs] の各フィールドは空白になり、古い値がインスタンスと関連付けられなくなったことを示します。

5. インスタンスが停止されている間、特定のインスタンス属性を変更できません。詳細については、「[停止されているインスタンスの変更 \(p. 348\)](#)」を参照してください。
6. 停止されているインスタンスを再起動するには、インスタンスを選択して、[Actions] をクリックし、次に [Start] をクリックします。
7. 確認ダイアログボックスで、[Yes, Start] をクリックします。インスタンスが `running` 状態になるまで、数分かかる場合があります。

[EC2-Classic] インスタンスの状態が `running` になると、詳細ペインの [Public DNS]、[Private DNS]、[Private IPs] の各フィールドには、インスタンスに割り当てた新しい値が表示されます。

8. [EC2-Classic] インスタンスに関連する Elastic IP アドレスがある場合は、次のようにして再び関連付ける必要があります。
  - a. ナビゲーションペインで、[Elastic IPs] をクリックします。
  - b. インスタンスを停止する前に書き留めた Elastic IP アドレスを選択します。
  - c. [Associate Address] をクリックします。
  - d. インスタンスを停止する前に書き留めたインスタンス ID を選択し、[Associate] をクリックします。

コマンドラインを使用して Amazon EBS-Backed インスタンスを停止および起動するには

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

- `stop-instances` および `start-instances` ( AWS CLI )
- `ec2-stop-instances` および `ec2-start-instances` ( Amazon EC2 CLI )
- `Stop-EC2Instance` および `Start-EC2Instance` ( AWS Tools for Windows PowerShell )

## 停止されているインスタンスの変更

AWS Management Console または コマンドラインインターフェイスを使用して、インスタンスタイプおよびユーザーデータ属性を変更できます。AWS Management Console を使用して、カーネルまたは RAM ディスクの属性を変更することはできません。

コンソールを使用して停止しているインスタンスのインスタンスタイプを変更するには

1. ナビゲーションペインの [Instances] をクリックします。
2. 停止されているインスタンスを選択して、[Actions] をクリックし、次に [Change Instance Type] をクリックします。
3. [Change Instance Type] ダイアログボックスの [Instance Type] ドロップダウンリストで、必要なインスタンスのタイプを選択し、[Apply] をクリックします。

詳細については、[インスタンスのサイズ変更 \(p. 131\)](#) を参照してください。

コンソールを使用して停止しているインスタンスのユーザーデータを変更するには

1. ナビゲーションペインの [Instances] をクリックします。
2. 停止されているインスタンスを選択して、[Actions] をクリックし、次に [View/Change User Data] をクリックします。

3. [View/Change User Data] ダイアログボックスで、ユーザーデータを更新し、[Save] をクリックします。インスタンスの実行中はユーザーデータを変更できないことに注意してください。表示することはできません。

コマンドラインを使用してインスタンス属性を変更するには

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

- [modify-instance-attribute](#) ( AWS CLI )
- [ec2-modify-instance-attribute](#) ( Amazon EC2 CLI )
- [Edit-EC2InstanceAttribute](#) ( AWS Tools for Windows PowerShell )

## インスタンスの再起動

### Abstract

Amazon EC2 を使用してインスタンスを再起動し、必要なメンテナンスを行います。

インスタンスの再起動は、オペレーティングシステムの再起動と同等です。ほとんどの場合、インスタンスの再起動には数分しかかかりません。インスタンスを再起動すると、インスタンスは同じホスト上で保持されるため、インスタンスのパブリックドメイン名、プライベート IP アドレス、およびインスタンスストアボリューム上のすべてのデータは保持されます。

インスタンスを再起動しても、インスタンスの停止と再起動とは異なり、新しいインスタンスの課金時間は開始されません。

再起動を必要とする更新の適用など、必要なメンテナンスのために、インスタンスの再起動を予定する場合があります。ユーザーが操作する必要はありません。予定されている時間帯に自動的に行われる再起動まで待つことをお勧めします。詳細については、「[インスタンスのイベントのモニタリング \(p.392\)](#)」を参照してください。

インスタンスからオペレーティングシステムの再起動コマンドを実行する代わりに、Amazon EC2 を使ってインスタンスを再起動することをお勧めします。

コンソールを使用してインスタンスを再起動するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. インスタンスを選択し、[Actions] をクリックして、[Reboot] をクリックします。
4. 確認プロンプトが表示されたら、[Yes, Reboot] をクリックします。

コマンドラインを使用してインスタンスを再起動するには

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

- [reboot-instances](#) ( AWS CLI )
- [ec2-reboot-instances](#) ( Amazon EC2 CLI )
- [Restart-EC2Instance](#) ( AWS Tools for Windows PowerShell )

# インスタンスのリタイア

## Abstract

リタイアが予定されているインスタンスと、リタイアを管理するために行うアクションを特定します。

インスタンスをホストしている基盤のハードウェアで回復不可能な障害が検出されると、AWS によってインスタンスのリタイアが予定されます。予定されたリタイア日になると、インスタンスは AWS によって停止または終了されます。インスタンスのルートデバイスが Amazon EBS ボリュームである場合、インスタンスは停止されますが、その後いつでも再び起動できます。停止したインスタンスを開始すると、新しいハードウェアに移行されます。インスタンスのルートデバイスがインスタンスストアボリュームである場合、インスタンスは終了し、再び使用することはできません。

### Topics

- [リタイアが予定されているインスタンスの特定 \(p. 350\)](#)
- [リタイアが予定されているインスタンスの操作 \(p. 351\)](#)

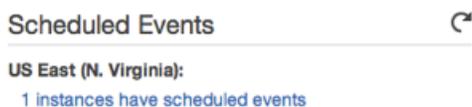
インスタンスイベントのタイプの詳細については、「[インスタンスのイベントのモニタリング \(p. 392\)](#)」を参照してください。

## リタイアが予定されているインスタンスの特定

インスタンスのリタイアが予定された場合、イベントの前に、当該のインスタンス ID とリタイア日を記載したメールが送信されます。このメールは、アカウントに関連付けられているアドレスに送信されます。これは、AWS Management Console へのログインに使用するメールアドレスと同じです。定期的を確認しないメールアカウントを使用している場合は、Amazon EC2 コンソールまたはコマンドラインを使用して、いずれかのインスタンスにリタイアが予定されているかどうかを判断できます。

コンソールを使用してリタイアが予定されているインスタンスを特定するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [EC2 Dashboard] をクリックします。[Scheduled Events] に、Amazon EC2 インスタンスおよびボリュームに関連付けられたイベントが表示されます。



3. インスタンスに予定されたイベントが表示されている場合は、リージョン名の下リンクをクリックして [Events] ページにアクセスします。
4. [Events] ページには、すべてのリソースとそれに関連付けられたイベントが一覧表示されます。リタイアが予定されているインスタンスを表示するには、1 つ目のフィルタリストから [Instance resources] を選択し、2 つ目のフィルタリストから [Instance retirement] を選択します。
5. フィルタの結果にインスタンスのリタイアが予定されていることが表示されたら、当該のインスタンスを選択し、詳細ペインの [Start time] フィールドの日時を書き留めます。これがインスタンスのリタイア日です。

コマンドラインを使用してリタイアが予定されているインスタンスを特定するには

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

- [describe-instance-status](#) ( AWS CLI )
- [ec2-describe-instance-status](#) ( Amazon EC2 CLI )
- [Get-EC2InstanceStatus](#) ( AWS Tools for Windows PowerShell )

## リタイアが予定されているインスタンスの操作

インスタンスのリタイアが予定されているときに実行できるアクションはいくつかあります。実行するアクションは、インスタンスのルートデバイスが Amazon EBS ボリュームであるかインスタンスストアボリュームであるかによって異なります。インスタンスのルートデバイスタイプが不明な場合は、Amazon EC2 コンソールまたはコマンドラインを使用して調べることができます。

### インスタンスのルートデバイスタイプの判別

コンソールを使用してインスタンスのルートデバイスタイプを判別するには

1. ナビゲーションペインの [Events] をクリックします。前述の手順「[リタイアが予定されているインスタンスの特定 \(p. 350\)](#)」で説明したように、フィルタリストを使用してリタイアが予定されているインスタンスを特定します。
2. [Resource ID] 列でインスタンス ID をクリックすると、[Instances] ページに移動します。
3. インスタンスを選択し、[Description] タブで [Root device type] フィールドを探します。この値が `ebs` の場合、インスタンスは EBS-Backed です。この値が `instance-store` の場合、インスタンスは、Instance Store-Backed です。

コマンドラインを使用してインスタンスのルートデバイスタイプを判別するには

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

- [describe-instances](#) ( AWS CLI )
- [ec2-describe-instances](#) ( Amazon EC2 CLI )
- [Get-EC2Instance](#) ( AWS Tools for Windows PowerShell )

### リタイアが予定されているインスタンスの管理

リタイアが予定されているインスタンスのデータを維持するには、以下に挙げるいずれかのアクションを実行します。予期しないダウンタイムやデータ消失を防ぐために、インスタンスのリタイア日より前にこのアクションを実行することが重要です。



#### Warning

Instance Store-Backed インスタンスの場合、リタイア日を過ぎるとインスタンスが終了し、インスタンスやインスタンスに格納されていたデータを復元できなくなります。インスタンスストアボリュームのデータは、インスタンスのルートデバイスにかかわらず、EBS-Backed インスタンスにアタッチされている場合でも、インスタンスがリタイアされると失われます。

インスタンスのルートデバイスタイプ	インスタンスのプラットフォーム	アクション
EBS	すべて	予定されるリタイア日を待ちます-その日になるとインスタンスが停止します-または、リタイア日の前に自らインスタンスを停止します。インスタンスはいつでも再び開始することができます。インスタンスの停止と開始、インスタンスを停止したときに予想される影響（インスタンスに関連付けられたパブリックIPアドレス、プライベートIPアドレス、およびElastic IPアドレスへの影響など）の詳細については、「 <a href="#">インスタンスの停止と起動 (p. 346)</a> 」を参照してください。
EBS	Linux/Unix	インスタンスから EBS-Backed AMI を作成し、代替インスタンスを起動します。詳細については、「 <a href="#">Amazon EBS-Backed Linux AMI の作成 (p. 77)</a> 」を参照してください。
EBS	Windows	インスタンスから EBS-Backed AMI を作成し、代替インスタンスを起動します。詳細については、「 <a href="#">Amazon EBS-Backed Windows AMI の作成</a> 」を参照してください。
インスタンスストア	Linux/Unix	AMI ツールを使用してインスタンスから Instance-Store Backed AMI を作成し、代替インスタンスを起動します。詳細については、「 <a href="#">Instance Store-Backed Linux AMI の作成 (p. 81)</a> 」を参照してください。
インスタンスストア	Windows	インスタンスをバンドルし、バンドル中に作成したマニフェストから Instance Store-Backed AMI を作成します。新しいAMI から代替インスタンスを起動できます。詳細については、「 <a href="#">Instance Store-Backed Windows AMI の作成</a> 」を参照してください。
インスタンスストア	Linux/Unix	データを EBS ポリリュームに転送し、ポリリュームのスナップショットを作成し、スナップショットから AMI を作成することによって、EBS-Backed インスタンスをインスタンスに変換します。新しい AMI から代替インスタンスを起動できます。詳細については、「 <a href="#">Instance Store-Backed AMI を Amazon EBS-Backed AMI に変換する (p. 87)</a> 」を参照してください。

## インスタンスの終了

### Abstract

不要になったインスタンスを終了し、そのインスタンスに対する課金が発生しないようにします。

インスタンスが必要なくなったら、終了することができます。インスタンスの状態が `shutting-down` または `terminated` に変わったら、そのインスタンスへの課金は停止します。

インスタンスを停止した後に、接続または再起動することはできません。ただし、同じAMIから別のインスタンスを起動することができます。インスタンスを停止および再起動する場合は、「[インスタンスの停止と起動 \(p. 346\)](#)」を参照してください。

インスタンスが通常より長く `shutting-down` 状態になっている場合、最終的に Amazon EC2 サービス内の自動プロセスによってクリーンアップ（終了）されます。詳細については、「[インスタンスの終了 \(シャットダウン\) のトラブルシューティング \(p. 693\)](#)」を参照してください。

## Topics

- [インスタンスの終了 \(p. 353\)](#)
- [インスタンスを終了する \(p. 353\)](#)
- [インスタンスの終了保護の有効化 \(p. 354\)](#)
- [インスタンスによって起動されたシャットダウン動作の変更 \(p. 355\)](#)
- [インスタンスの終了で Amazon EBS ボリュームを保持する \(p. 355\)](#)

## インスタンスの終了

インスタンスの終了後、インスタンスはしばらくの間コンソールに表示されたままですが、エントリは削除されます。

インスタンスが終了すると、そのインスタンスに関連付けられたすべてのインスタンスストアボリュームのデータが削除されます。

デフォルトでは、インスタンスの起動時に接続したすべての Amazon EBS ボリュームが、インスタンスの終了時に自動的に削除されます。ただし、デフォルトでは、実行中のインスタンスにアタッチしたボリュームは、インスタンスの終了後も維持されます。この動作はボリュームの `DeleteOnTermination` 属性によって制御されますが、変更できます。詳細については、[インスタンスの終了で Amazon EBS ボリュームを保持する \(p. 355\)](#) を参照してください。

AWS Management Console、CLI、および API を使用している他のユーザーによって、誤ってインスタンスを終了されないようにできます。この機能は、Amazon EC2 instance store-backed インスタンスと Amazon EBS-backed インスタンスの両方で使用できます。各インスタンスには、デフォルト値の `false` である `DisableApiTermination` 属性があります (インスタンスは Amazon EC2 によって終了される場合があります)。インスタンスの実行中または停止中に、このインスタンス属性を変更できます (Amazon EBS-backed インスタンスの場合)。詳細については、「[インスタンスの終了保護の有効化 \(p. 354\)](#)」を参照してください。

システムをシャットダウンするオペレーティングシステムコマンドを使用して、インスタンスからシャットダウンが開始されたときに、インスタンスを停止または終了するかどうかを制御できます。詳細については、「[インスタンスによって起動されたシャットダウン動作の変更 \(p. 355\)](#)」を参照してください。

インスタンスの終了時にスクリプトを実行した場合、シャットダウンスクリプトが実行されることを保証する方法がないため、異常な終了が発生する場合があります。Amazon EC2 はインスタンスを正常にシャットダウンして、システムシャットダウンスクリプトが実行されるように試みますが、特定のイベント (ハードウェア障害など) ではシステムシャットダウンスクリプトが実行されないことがあります。

## インスタンスを終了する

インスタンスは AWS Management Console またはコマンドラインを使用して終了できます。

コンソールを使用してインスタンスを終了するには

1. インスタンスを終了する前に、終了時に Amazon EBS ボリュームが削除され、必要なデータをすべてインスタンスストアボリュームから Amazon EBS または Amazon S3 にコピーしていることを確認して、データが失われないことを確認します。
2. Amazon EC2 コンソールを開きます。
3. ナビゲーションペインの [Instances] をクリックします。
4. インスタンスを選択して、[Actions] をクリックし、次に [Terminate] をクリックします。
5. 確認を求められたら、[Yes, Terminate] をクリックします。

コマンドラインを使用してインスタンスを終了するには

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

- `terminate-instances` ( AWS CLI )
- `ec2-terminate-instances` ( Amazon EC2 CLI )
- `Stop-EC2Instance` ( AWS Tools for Windows PowerShell )

## インスタンスの終了保護の有効化

デフォルトでは、Amazon EC2 コンソール、コマンドラインインターフェイス、または API を使用して、インスタンスを終了できます。Amazon EC2 を使用してインスタンスを誤って終了できないようにするには、インスタンスの終了保護を有効にできます。`DisableApiTermination` 属性は、インスタンスがコンソール、CLI、または API を使用して終了できるかどうかを制御します。デフォルトでは、インスタンスの終了保護は無効になっています。インスタンスが実行中またはインスタンスが停止中に、インスタンスを起動する際に、この属性の値を設定できます ( Amazon EBS-backed インスタンスの場合 )。

`InstanceInitiatedShutdownBehavior` 属性が設定された場合、`DisableApiTermination` 属性はインスタンスからシャットダウンを開始して ( システムシャットダウン用のオペレーティングシステムコマンドを使用 )、インスタンスを終了できます。詳細については、「[インスタンスによって起動されたシャットダウン動作の変更 \(p. 355\)](#)」を参照してください。

Auto Scaling グループの一部であるインスタンスは、終了保護の対象になりません。詳細については、『*Auto Scaling 開発者ガイド*』の「[Instance Termination Policy for Your Auto Scaling Group](#)」を参照してください。

終了保護は AWS Management Console またはコマンドラインを使用して有効または無効にできます。

起動時にインスタンスに対する終了保護を有効にするには

1. Amazon EC2 コンソールのダッシュボードで、[Launch Instance] をクリックし、ウィザードの指示にしたがいます。
2. [Configure Instance Details] ページで、[Enable termination protection] チェックボックスをオンにします。

実行中または停止中のインスタンスの終了保護を有効にするには

1. インスタンスを選択して、[Actions] をクリックし、次に [Change Termination Protection] をクリックします。
2. [Yes, Enable] をクリックします。

実行中または停止中のインスタンスの終了保護を無効にするには

1. インスタンスを選択して、[Actions] をクリックし、次に [Change Termination Protection] をクリックします。
2. [Yes, Disable] をクリックします。

コマンドラインを使用して終了保護を有効または無効にするには

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

- [modify-instance-attribute](#) ( AWS CLI )
- [ec2-modify-instance-attribute](#) ( Amazon EC2 CLI )
- [Edit-EC2InstanceAttribute](#) ( AWS Tools for Windows PowerShell )

## インスタンスによって起動されたシャットダウン動作の変更

デフォルトでは、shutdown コマンドを使用して、Amazon EBS-Backed インスタンスからシャットダウンを開始すると、インスタンスは停止します。代わりに終了できるように、インスタンスの `InstanceInitiatedShutdownBehavior` 属性を使用して、この動作を変更できます。インスタンスの実行中または停止中に、この属性を更新できます。

`InstanceInitiatedShutdownBehavior` 属性は AWS Management Console またはコマンドラインを使用して更新できます。

コンソールを使用してインスタンスのシャットダウン動作を変更するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. インスタンスを選択して、[Actions] をクリックし、次に [Change Shutdown Behavior] をクリックします。現在の動作は既に選択されています。
4. 動作を変更するには、[Shutdown behavior] リストからオプションを選択し、[Apply] をクリックします。



コマンドラインを使用してインスタンスのシャットダウン動作を変更するには

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

- [modify-instance-attribute](#) ( AWS CLI )
- [ec2-modify-instance-attribute](#) ( Amazon EC2 CLI )
- [Edit-EC2InstanceAttribute](#) ( AWS Tools for Windows PowerShell )

## インスタンスの終了で Amazon EBS ボリュームを保持する

デフォルトでは、Amazon は次の操作を行います。

- 実行中のインスタンスに接続したボリュームは、インスタンスの終了後も保持します。

- インスタンスを停止して再起動する場合、インスタンスに接続したボリュームを保持します。
- 起動時にインスタンスにアタッチするボリュームは、ルートデバイスボリュームを含めて、インスタンスの終了時に削除します。

ボリュームの `DeleteOnTermination` 属性を使用して、この動作を変更できます。この属性の値が `true` の場合、インスタンスが終了した後にボリュームは削除されます。それ以外の場合は、ボリュームは保持されます。ボリュームの `DeleteOnTermination` 属性が `false` の場合、ボリュームは現在の状態を保持します。ボリュームのスナップショットを取得し、別のインスタンスに接続することができます。

起動時にインスタンスにアタッチしたボリュームを解除して、再アタッチする場合、インスタンスが終了された後もボリュームは保持されます。つまり、その `DeleteOnTermination` 属性は `false` に設定されます。

インスタンスに接続されているボリュームの `DeleteOnTermination` 属性の値は、インスタンスのブロックデバイスマッピングで確認できます。詳細については、[インスタンスブロックデバイスマッピングの EBS ボリュームの表示 \(p. 652\)](#) を参照してください。

`DeleteOnTermination` 属性は AWS Management Console またはコマンドラインを使用して更新できます。

## コンソールを使用したルートボリュームの永続的な変更

コンソールを使用して、インスタンスの起動時に `DeleteOnTermination` 属性を変更できます。実行中のインスタンスのこの属性を変更するには、コマンドラインを使用する必要があります。

コンソールを使用して、起動時にインスタンスのルートボリュームが存続するように変更するには

1. Amazon EC2 コンソールを開きます。
2. コンソールダッシュボードで、[Launch Instance] をクリックします。
3. [Choose an Amazon Machine Image (AMI)] ページで AMI を選択し、[Select] をクリックします。
4. ウィザードにしたがって [Choose an Instance Type] ページと [Configure Instance Details] ページを設定します。
5. [Add Storage] ページで、ルートボリュームの [Delete On Termination] チェックボックスの選択を解除します。
6. ウィザードの残りのページを完了した後、[Launch] をクリックします。

インスタンスの詳細ペインでルートデバイスボリュームの詳細を表示することにより、設定を確認できます。[Block devices] の隣にあるルートデバイスボリュームのエントリをクリックします。デフォルトでは、[Delete on termination] は [True] です。デフォルトの動作を変更した場合は、[Delete on termination] が [False] になっています。

## コマンドラインを使用して実行中のインスタンスのルートボリュームが存続するように変更する

次のいずれかのコマンドを使用して、実行中のインスタンスのルートデバイスボリュームを永続化するように変更できます。ルートデバイスは通常、`/dev/sda1` (Linux) または `xvda` (Windows) です。これらのコマンドラインインターフェイスについての詳細は、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `modify-instance-attribute` (AWS CLI)
- `ec2-modify-instance-attribute` (Amazon EC2 CLI)
- `Edit-EC2InstanceAttribute` (AWS Tools for Windows PowerShell)

### Example AWS CLI 用

次のコマンドは、DeleteOnTermination 属性を false に設定して、ルートボリュームを保持します。

```
aws ec2 modify-instance-attribute --instance-id i-5203422c --block-device-mappings [{"DeviceName": "/dev/sda1", "Ebs": {"DeleteOnTermination": false}}]
```

`describe-instances` コマンドを使用して、コマンド出力の `/dev/sda1` で `BlockDeviceMappings` エントリを探すことで、`deleteOnTermination` が false であることを確認します。

### Example Amazon EC2 CLI 用

次のコマンドは、DeleteOnTermination 属性を false に設定して、ルートボリュームを保持します。

```
ec2-modify-instance-attribute i-5203422c -b /dev/sda1::false
```

Windows システム上でコマンドラインツールを使用している場合は、ブロックデバイスマッピングの値を引用符で囲む必要があります。

```
ec2-modify-instance-attribute i-5203422c -b "xvda::false"
```

## コマンドラインを使用して起動時にインスタンスのルートボリュームが存続するように変更する

インスタンスの起動時に、次のコマンドのいずれかを使用して、ルートデバイスボリュームが存続するように変更することができます。ルートデバイスは通常、`/dev/sda1` (Linux) または `xvda` (Windows) です。これらのコマンドラインインターフェイスについての詳細は、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- `run-instances` (AWS CLI)
- `ec2-run-instances` (Amazon EC2 CLI)
- `New-EC2Instance` (AWS Tools for Windows PowerShell)

### Example AWS CLI 用

次のコマンドは、DeleteOnTermination 属性を false に設定して、ルートボリュームを保持します。

```
aws ec2 run-instances --image-id ami-1a2b3c4d --block-device-mappings [{"DeviceName": "/dev/sda1", "Ebs": {"DeleteOnTermination": false}}] other parameters...
```

`describe-instances` コマンドを使用して、コマンド出力の `/dev/sda1` で `BlockDeviceMappings` エントリを探すことで、`deleteOnTermination` が false であることを確認します。

### Example Amazon EC2 CLI 用

次のコマンドは、DeleteOnTermination 属性を false に設定して、ルートボリュームを保持します。

```
ec2-run-instances ami-1a2b3c4d -b /dev/sda1::false other parameters... -v
```

Windows システム上でコマンドラインツールを使用している場合は、ブロックデバイスマッピングの値を引用符で囲む必要があります。

```
ec2-run-instances ami-1a2b3c4d -b "xvda::false" other parameters... -v
```

# Amazon Linux インスタンスを設定する

---

## Abstract

Amazon Linux インスタンスをカスタマイズします。

インスタンスを正常に起動し、ログインしたら、変更できます。インスタンスは、特定のアプリケーションニーズに合わせて、さまざまな方法で構成できます。このセクションには、入門として役立つ一般的な例がいくつか含まれています。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

## Topics

- [一般的な設定シナリオ \(p. 359\)](#)
- [ソフトウェアの管理 \(p. 360\)](#)
- [ユーザーの管理 \(p. 369\)](#)
- [インスタンスの時間を設定します \(p. 370\)](#)
- [インスタンスのホスト名を変更する \(p. 374\)](#)
- [Amazon EC2 で動的 DNS を使用する \(p. 377\)](#)
- [ユーザーデータでインスタンスを起動する \(p. 378\)](#)

## 一般的な設定シナリオ

Amazon Linux のベースのディストリビューションには、基本的なサーバー操作に必要なソフトウェアパッケージとユーティリティが数多く含まれています。ただし、さまざまなソフトウェアリポジトリでさらに多くのソフトウェアパッケージを利用できます。また、ソースコードから、さらに多くのパッケージソースコードを作成できます。これらの場所からソフトウェアをインストールし、作成する方法についての詳細は、[ソフトウェアの管理 \(p. 360\)](#) を参照してください。

Amazon Linux インスタンスには、`ec2-user` アカウントが事前設定されていますが、スーパーユーザー権限を持たない他のユーザーアカウントを追加することがあります。ユーザーアカウントの追加と削除についての詳細は、[ユーザーの管理 \(p. 369\)](#) を参照してください。

Amazon Linux インスタンスのデフォルトの時間設定では、ネットワークタイムプロトコルを利用し、システム時間をインスタンスに設定します。デフォルトの時間帯は UTC です。インスタンスの時間帯の設定または独自のタイムサーバーの利用についての詳細は、[インスタンスの時間を設定します \(p. 370\)](#) を参照してください。

お客様がネットワークを所有し、それにドメイン名を登録している場合、インスタンスのホスト名を変更して、そのドメインに含まれる一部としてインスタンスを識別できます。また、システムプロンプトを変更して、より意味のある名前を表示することもできます。ホスト名設定を変更する必要はありません。詳細については、[インスタンスのホスト名を変更する \(p. 374\)](#) を参照してください。動的 DNS サービスプロバイダを使用するようにインスタンスを設定できます。詳細については、[Amazon EC2 で動的 DNS を使用する \(p. 377\)](#) を参照してください。

Amazon EC2 でインスタンスを起動するとき、起動後にそのインスタンスにユーザーデータを渡し、一般的な設定タスクを実行したり、スクリプトを実行したりできます。2つのタイプのユーザーデータを Amazon EC2 に渡すことができます。cloud-init デイレクティブとシェルスクリプトです。詳細については、[ユーザーデータでインスタンスを起動する \(p. 378\)](#) を参照してください。

## ソフトウェアの管理

### Abstract

Amazon Linux インスタンスでソフトウェアを管理します。

Amazon Linux のベースのディストリビューションには、基本的なサーバー操作に必要なソフトウェアパッケージとユーティリティが数多く含まれています。ただし、さまざまなソフトウェアリポジトリでさらに多くのソフトウェアパッケージを利用できます。また、ソースコードから、さらに多くのパッケージソースコードを作成できます。

### Topics

- [インスタンスソフトウェアの更新 \(p. 361\)](#)
- [リポジトリの追加 \(p. 364\)](#)
- [ソフトウェアパッケージを見つける \(p. 365\)](#)
- [ソフトウェアパッケージのインストール \(p. 367\)](#)
- [ソフトウェアのコンパイル準備 \(p. 367\)](#)

ソフトウェアを最新の状態に維持することが重要です。Linux ディストリビューションの多くのパッケージは頻繁に更新されます。これにより、バグが修正され、機能が追加されて、セキュリティ上の弱点に対する防御措置が行われます。詳細については、[インスタンスソフトウェアの更新 \(p. 361\)](#) を参照してください。

デフォルトで、Amazon Linux インスタンスは、`amzn-main` と `amzn-updates` の2つのリポジトリを有効にして起動します。これらのリポジトリには、Amazon Web Services が更新するさまざまなパッケージがありますが、別のリポジトリに含まれるパッケージをインストールしたい場合があるかもしれません。詳細については、[リポジトリの追加 \(p. 364\)](#) を参照してください。有効なリポジトリでパッケージを探す方法については、[ソフトウェアパッケージを見つける \(p. 365\)](#) を参照してください。Amazon Linux インスタンスにソフトウェアをインストールする方法については、[ソフトウェアパッケージのインストール \(p. 367\)](#) を参照してください。

リポジトリに保管されているソフトウェアパッケージで、すべてのソフトウェアが利用できるわけではありません。一部のソフトウェアは、そのソースコードからインスタンスでコンパイルする必要があります。詳細については、[ソフトウェアのコンパイル準備 \(p. 367\)](#) を参照してください。

Amazon Linux インスタンスは、yum パッケージマネージャを利用してソフトウェアを管理します。yum パッケージマネージャはソフトウェアをインストール、削除、更新し、各パッケージのすべての依存関係を管理できます。Ubuntu のような Debian ベースの Linux ディストリビューションは、apt-get コマンドと dpkg パッケージマネージャを使用するため、次のセクションの yum の例は、このようなディストリビューションには該当しません。

## インスタンスソフトウェアの更新

### Abstract

Amazon Linux インスタンス内で1つのソフトウェアパッケージを更新するか、すべてのパッケージを更新します。

ソフトウェアを最新の状態に維持することが重要です。Linux ディストリビューションの多くのパッケージは頻繁に更新されます。これにより、バグが修正され、機能が追加されて、セキュリティ上の弱点に対する防御措置が行われます。最初に Amazon Linux インスタンスを起動して接続するときに、メッセージが表示され、セキュリティ上の目的からソフトウェアパッケージを更新するように求められる場合があります。このセクションでは、システム全体またはパッケージを1つだけ更新する方法を紹介합니다。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

```
__|  __|_ )
_| ( / Amazon Linux AMI
__|\__|__|

https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/
There are 12 security update(s) out of 25 total update(s) available
Run "sudo yum update" to apply all updates.
[ec2-user ~]$
```

Amazon Linux インスタンスのすべてのパッケージを更新するには

1. (オプション) シェルウィンドウで screen セッションを開始します。ネットワークが遮断され、インスタンスへの SSH 接続が切断されることがあります。この状態が長時間のソフトウェア更新中に発生した場合、インスタンスは混乱した状態になりますが、復元できます。screen セッションを開始しておけば、接続が遮断された場合でも更新を続行でき、後で問題なくセッションに再接続できます。

- a. screen セッションを開始するには screen コマンドを実行します。

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

- b. セッションが中断された場合、インスタンスにログインし直し、利用できる画面を表示します。

```
[ec2-user ~]$ screen -ls
There is a screen on:
 17793.pts-0.ip-12-34-56-78 (Detached)
1 Socket in /var/run/screen/S-ec2-user.
```

- c. `screen -r` コマンドと前のコマンドのプロセス ID を使用して、画面に再接続します。

```
[ec2-user ~]$ screen -r 17793
```

- d. `screen` の使用が終わったら、`exit` コマンドを使用してセッションを閉じます。

```
[ec2-user ~]$ exit  
[screen is terminating]
```

2. `yum update` コマンドを実行します。オプションで、`--security` フラグを追加すれば、セキュリティ更新のみを適用できます。

```
[ec2-user ~]$ sudo yum update  
Loaded plugins: priorities, security, update-motd, upgrade-helper  
amzn-main | 2.1 kB 00:00  
amzn-updates | 2.3 kB 00:00  
Setting up Update Process  
Resolving Dependencies  
--> Running transaction check  
---> Package aws-apitools-ec2.noarch 0:1.6.8.1-1.0.amzn1 will be updated  
---> Package aws-apitools-ec2.noarch 0:1.6.10.0-1.0.amzn1 will be an update  
---> Package gnupg2.x86_64 0:2.0.18-1.16.amzn1 will be updated  
---> Package gnupg2.x86_64 0:2.0.19-8.21.amzn1 will be an update  
---> Package libgcrypt.i686 0:1.4.5-9.10.amzn1 will be updated  
---> Package libgcrypt.x86_64 0:1.4.5-9.10.amzn1 will be updated  
---> Package libgcrypt.i686 0:1.4.5-9.12.amzn1 will be an update  
---> Package libgcrypt.x86_64 0:1.4.5-9.12.amzn1 will be an update  
---> Package openssl.x86_64 1:1.0.1e-4.53.amzn1 will be updated  
---> Package openssl.x86_64 1:1.0.1e-4.54.amzn1 will be an update  
---> Package python-boto.noarch 0:2.9.9-1.0.amzn1 will be updated  
---> Package python-boto.noarch 0:2.13.3-1.0.amzn1 will be an update  
--> Finished Dependency Resolution  
  
Dependencies Resolved  
  
=====
```

Package Size	Arch	Version	Repository
Updating:			
aws-apitools-ec2 14 M	noarch	1.6.10.0-1.0.amzn1	amzn-updates
gnupg2 2.4 M	x86_64	2.0.19-8.21.amzn1	amzn-updates
libgcrypt 248 k	i686	1.4.5-9.12.amzn1	amzn-updates
libgcrypt 262 k	x86_64	1.4.5-9.12.amzn1	amzn-updates
openssl 1.7 M	x86_64	1:1.0.1e-4.54.amzn1	amzn-updates
python-boto 1.6 M	noarch	2.13.3-1.0.amzn1	amzn-updates

```
Transaction Summary
=====
Upgrade           6 Package(s)

Total download size: 20 M
Is this ok [y/N]:
```

- 表示されたパッケージを確認し、yを入力し、Enterを押して更新を承認します。システムのパッケージをすべて更新するには数分かかります。実行中、yum 出力には更新のステータスが表示されます。

```
Downloading Packages:
(1/6): aws-apitools-ec2-1.6.10.0-1.0.amzn1.noarch.rpm | 14 MB 00:00
(2/6): gnupg2-2.0.19-8.21.amzn1.x86_64.rpm | 2.4 MB 00:00
(3/6): libgcrypt-1.4.5-9.12.amzn1.i686.rpm | 248 kB 00:00
(4/6): libgcrypt-1.4.5-9.12.amzn1.x86_64.rpm | 262 kB 00:00
(5/6): openssl-1.0.1e-4.54.amzn1.x86_64.rpm | 1.7 MB 00:00
(6/6): python-boto-2.13.3-1.0.amzn1.noarch.rpm | 1.6 MB 00:00
-----
Total                               28 MB/s | 20 MB 00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : libgcrypt-1.4.5-9.12.amzn1.x86_64
 1/12
  Updating   : gnupg2-2.0.19-8.21.amzn1.x86_64
 2/12
  Updating   : aws-apitools-ec2-1.6.10.0-1.0.amzn1.noarch
 3/12
  Updating   : 1:openssl-1.0.1e-4.54.amzn1.x86_64
 4/12
  ...
Complete!
```

Amazon Linux インスタンスの 1 つのパッケージを更新するには

システム全体ではなく、1 つのパッケージ ( とその依存関係 ) を更新するには、この手順を使用します。

- 更新するパッケージの名前を指定して、yum update コマンドを実行します。

```
[ec2-user ~]$ sudo yum update openssl
Loaded plugins: priorities, security, update-motd, upgrade-helper
amzn-main | 2.1 kB 00:00
amzn-updates | 2.3 kB 00:00
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package openssl.x86_64 1:1.0.1e-4.53.amzn1 will be updated
---> Package openssl.x86_64 1:1.0.1e-4.54.amzn1 will be an update
--> Finished Dependency Resolution
```

```
Dependencies Resolved

=====

Package      Arch      Version      Repository
Size
=====
Updating:
openssl      x86_64     1:1.0.1e-4.54.amzn1  amzn-updates
1.7 M

Transaction Summary
=====
Upgrade      1 Package(s)

Total download size: 1.7 M
Is this ok [y/N]:
```

2. リストにあるパッケージ情報を確認し、yを入力し、Enterを押して更新を承認します。時折、解決する必要があるパッケージ依存関係がある場合、リストには複数のパッケージがあります。実行中、yum 出力には更新のステータスが表示されます。

```
Downloading Packages:
openssl-1.0.1e-4.54.amzn1.x86_64.rpm | 1.7 MB    00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : 1:openssl-1.0.1e-4.54.amzn1.x86_64
  1/2
  Cleanup   : 1:openssl-1.0.1e-4.53.amzn1.x86_64
  2/2
  Verifying : 1:openssl-1.0.1e-4.54.amzn1.x86_64
  1/2
  Verifying : 1:openssl-1.0.1e-4.53.amzn1.x86_64
  2/2

Updated:
  openssl.x86_64 1:1.0.1e-4.54.amzn1

Complete!
```

## リポジトリの追加

### Abstract

Amazon Linux インスタンスにソフトウェアリポジトリを追加します。

デフォルトで、Amazon Linux インスタンスは、`amzn-main` と `amzn-updates` の2つのリポジトリを有効にして起動します。これらのリポジトリには、Amazon Web Services が更新するさまざまなパッケージがありますが、別のリポジトリに含まれるパッケージをインストールしたい場合があるかもしれません。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

yum で異なるリポジトリからパッケージをインストールには、`/etc/yum.conf` ファイル、または `/etc/yum.repos.d` ディレクトリにあるお客様の `repository.repo` ファイルに、リポジトリ情報を追加する必要があります。これは手動で行えますが、ほとんどの yum リポジトリのリポジトリ URL で、独自の `repository.repo` ファイルが提供されています。

yum リポジトリを `/etc/yum.repos.d` に追加するには

1. `.repo` ファイルの場所を見つけます。場所は、追加しているリポジトリによって異なります。例えば、`.repo` ファイルは `https://www.example.com/repository.repo` にあります。
2. `yum-config-manager` コマンドでリポジトリを追加します。

```
[ec2-user ~]$ sudo yum-config-manager --add-repo https://www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/yum.repos.d/repository.repo
repository.repo | 4.0 kB    00:00
repo saved to /etc/yum.repos.d/repository.repo
```

yum リポジトリを `/etc/yum.repos.d` で有効にするには

- `--enable repository` フラグを有効にして、`yum-config-manager` コマンドを使用します。次のコマンドを使用すると、Fedora プロジェクトの Extra Packages for Enterprise Linux ( EPEL ) リポジトリが有効になります。デフォルトでは、このリポジトリは Amazon Linux インスタンスの `/etc/yum.repos.d` にありますが、有効になっていません。

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

## ソフトウェアパッケージを見つける

### Abstract

yum 検索コマンドを使用して、設定したリポジトリで利用できるソフトウェアパッケージを探します。

yum search コマンドを使用すると、設定したリポジトリで利用できるパッケージの説明を検索できます。これは特に、インストールするパッケージの正確な名前がわからない場合に便利です。キーワード検索をコマンドに追加します。複数の単語を検索するには、引用符で検索クエリを囲みます。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

引用符で囲まれた複数の単語検索クエリは、正確なクエリに一致する結果のみを返します。予想されたパッケージが表示されない場合、キーワードを1つに絞って検索し、結果をスキャンします。キーワードの同義語を試して、検索の幅を広げることができます。

```
[ec2-user ~]$ sudo yum search "find"
Loaded plugins: priorities, security, update-motd, upgrade-helper
===== N/S Matched: find =====
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface
                           : to File::Find
perl-Module-Find.noarch : Find and use installed modules in a (sub)category
libpuzzle.i686 : Library to quickly find visually similar images (gif, png,
jpg)
libpuzzle.x86_64 : Library to quickly find visually similar images (gif, png,
                  : jpg)
mlocate.x86_64 : An utility for finding files by name
```

また、yum パッケージマネージャでは、いくつかのパッケージがグループに組み合わされます。それを1つのコマンドでインストールし、ウェブサーバーのインストールやソフトウェアコンパイルのツール作成など、特定のタスクを実行できます。システムに既にインストールされているグループと、インストールできるグループを一覧表示するには、yum grouplist コマンドを使用します。

```
[ec2-user ~]$ sudo yum grouplist
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Group Process
Installed Groups:
  Development Libraries
  Development tools
  Editors
  Legacy UNIX compatibility
  Mail Server
  MySQL Database
  Network Servers
  Networking Tools
  PHP Support
  Perl Support
  System Tools
  Web Server
Available Groups:
  Console internet tools
  DNS Name Server
  FTP Server
  Java Development
  MySQL Database client
  NFS file server
  Performance Tools
  PostgreSQL Database client (version 8)
  PostgreSQL Database server (version 8)
  Scientific support
  TeX support
  Technical Writing
  Web Servlet Engine
Done
```

`yum groupinfo "Group Name"` コマンドを使用すると、グループにさまざまなパッケージを表示できます。`Group Name` には情報を取得するグループの名前を指定します。このコマンドを使用すると、そのグループでインストールできる必須、デフォルト、任意のパッケージがすべて一覧表示されます。

## ソフトウェアパッケージのインストール

### Abstract

`yum` パッケージマネージャを使用して、Amazon Linux インスタンスにパッケージをインストールします。

`yum` パッケージマネージャは、ソフトウェアをインストールするための優れたツールです。有効になっているすべてのリポジトリからさまざまなソフトウェアパッケージを検索し、ソフトウェアインストールプロセスに伴う依存関係を処理できます。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

リポジトリからパッケージをインストールするには、`yum install package` コマンドを使用します。`package` にはインストールするソフトウェアの名前を置き換えます。例えば、links テキストベースウェブブラウザをインストールするには、次のコマンドを入力します。

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

パッケージのグループをインストールするには、`yum groupinstall Group Name` コマンドを使用します。`Group Name` にはインストールするグループの名前を置き換えます。例えば、「Performance Tools」グループをインストールするには、次のコマンドを入力します。

```
[ec2-user@ip-10-161-113-54 ~]$ sudo yum groupinstall "Performance Tools"
```

デフォルトでは、`yum` はグループで表示されている必須とデフォルトのパッケージのみをインストールします。グループの任意のパッケージをインストールする場合も、コマンドに `group_package_types` 設定パラメータを設定できます。コマンドを実行すると、任意のパッケージが追加されます。

```
[ec2-user ~]$ sudo yum --setopt=group_package_types=mandatory,default,optional groupinstall "Performance Tools"
```

また、`yum install` を使用して、インターネットからダウンロードした RPM パッケージファイルをインストールすることもできます。その場合、リポジトリパッケージ名の代わりに、RPM ファイルのパス名をインストールコマンドに追加します。

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

## ソフトウェアのコンパイル準備

### Abstract

ソフトウェアをコンパイルする開発ツールをインストールします。

インターネットには事前コンパイルされていないオープンソースのソフトウェアが豊富に存在します。パッケージリポジトリからダウンロードできます。いずれは、そのソースコードから、自分でコンパイル

ルする必要があるソフトウェアパッケージが判明することがあります。システムでソフトウェアのコンパイルを可能にするには、make、gcc、autoconfなど、いくつかの開発ツールをインストールする必要があります。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

ソフトウェアのコンパイルはすべての Amazon EC2 インスタンスで必要なタスクではないため、そのようなツールはデフォルトでインストールされていません。ただし、「Development Tools」という名前のパッケージグループで利用でき、yum groupinstall コマンドでインスタンスに簡単に追加されます。

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

ソフトウェアのソースコードパッケージは、多くの場合、tarball という圧縮アーカイブファイルとしてダウンロードできます ( <https://github.com/> や <http://sourceforge.net/> などのウェブサイトから )。通常、これらの tarball には .tar.gz というファイル拡張子が付いています。これらのアーカイブは tar コマンドで解凍できます。

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

ソースコードパッケージを解凍したら、ソースコードディレクトリで README ファイルまたは INSTALL ファイルを探します。これらのファイルに、ソースコードのコンパイルとインストールに関する詳細な指示があります。

### Amazon Linux パッケージのソースコードを取得するには

Amazon Web Services は、保守管理されているパッケージのソースコードを提供します。get\_reference\_source コマンドを使用して、インストールされているパッケージのソースコードをダウンロードできます。

- get\_reference\_source -p **package** コマンドを実行して、**package** のソースコードをダウンロードします。例えば、htop パッケージのソースコードをダウンロードするには、次のコマンドを入力します。

```
[ec2-user ~]$ get_reference_source -p htop

Requested package: htop
Found package from local RPM database: htop-1.0.1-2.3.amzn1.x86_64
Corresponding source RPM to found package : htop-1.0.1-2.3.amzn1.src.rpm

Are these parameters correct? Please type 'yes' to continue: yes
Source RPM downloaded to: /usr/src/srpm/debug/htop-1.0.1-2.3.amzn1.src.rpm
```

このコマンドではソース RPM の場所がリストアップされます。この場合、/usr/src/srpm/debug/htop-1.0.1-2.3.amzn1.src.rpm です。

## ユーザーの管理

### Abstract

Amazon Linux インスタンスのユーザーを追加または削除します。

Linux インスタンスタイプは、それぞれデフォルトのユーザーアカウントで起動します。Amazon Linux の場合は、デフォルトのユーザー名は `ec2-user` です。RHEL5 の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。Ubuntu の場合は、ユーザー名は `ubuntu` です。SUSE Linux の場合は、ユーザー名は `root` です。それ以外の場合は、AMI プロバイダーに確認してください。

デフォルトのユーザーアカウントを使用することは多くのアプリケーションで適切ですが、ユーザーアカウントを追加すると、個人が自分だけのファイルと作業場所を確保できます。新しいユーザー用にユーザーアカウントを作成することは、`ec2-user` アカウントへのアクセス権を複数のユーザーに（経験のないユーザーも含めて）与えるよりも、はるかに安全です。`ec2-user` アカウントが不適切に使用された場合、システムにさまざまな損害を与える可能性があるためです。

システムに新しいユーザーを追加するには

Amazon Linux インスタンスにユーザーを追加する作業は、2 つの基本的な操作により効率的に行われます。すなわち、ユーザーをシステムに追加し、そのユーザーにリモートログインする手段を与えます。

1. システムに新しいユーザーを追加するには、`useradd` コマンドを使用します。コマンドの後には、作成するユーザーの名前を指定します。

```
[ec2-user ~]$ sudo useradd newuser
```

このコマンドにより、`newuser` アカウントがシステムに追加され（`/etc/passwd` ファイルにエントリが追加されます）、`newuser` グループが作成されて、`/home/newuser` にアカウントのホームディレクトリが作成されます。

2. このアカウントにリモートアクセス権を与えるには、`newuser` ホームディレクトリで `.ssh` ディレクトリを作成し、パブリックキーを含む「`authorized_keys`」という名前のファイルを `.ssh` ディレクトリに作成する必要があります。
  - a. 新しいアカウントに切り替えると、新しく作成したファイルに適切な所有権が与えられます。

```
[ec2-user ~]$ sudo su - newuser  
[newuser ~]$
```

メッセージには、`ec2-user` ではなく `newuser` と表示される点にご注意ください。シェルセッションを新しいユーザーに切り替えています。

- b. `authorized_keys` ファイルのために `.ssh` ディレクトリが作成されます。

```
[newuser ~]$ mkdir .ssh
```

- c. `.ssh` ディレクトリのファイル権限を `700`（つまり、ファイルの所有者だけがディレクトリを開き、読み書きできます）に変更します。



#### Important

このステップは非常に重要です。厳密なファイル権限がなければ、SSH を使用してこのアカウントにログインできません。

```
[newuser ~]$ chmod 700 .ssh
```

- d. `.ssh` ディレクトリに「`authorized_keys`」という名前のファイルを作成します。

```
[newuser ~]$ touch .ssh/authorized_keys
```

- e. `authorized_keys` ファイルのファイル権限を 600 (つまり、ファイルの所有者だけがファイルを読み書きできます) に変更します。



#### Important

このステップは非常に重要です。厳密なファイル権限がなければ、SSH を使用してこのアカウントにログインできません。

```
[newuser ~]$ chmod 600 .ssh/authorized_keys
```

- f. お好みのテキストエディタで `authorized_keys` ファイルを編集し、キーペアのパブリックキーをファイルに貼り付けます。



#### Note

キーペアを作成する方法や既存のキーペアからパブリックキーを取得する方法については、[Amazon EC2 のキーペア \(p. 470\)](#) を参照してください。

これで、[Step 2.f \(p. 370\)](#) のパブリックキーに一致するプライベートキーを使用して、SSH 経由でインスタンスの `newuser` アカウントにログインできるはずですが、

システムからユーザーを削除するには

ユーザーアカウントが不要になった場合、今後使用されないようにそのアカウントを削除できます。

- ユーザーアカウント、ユーザーのホームディレクトリ、ユーザーのメールスプールを削除するには、削除するユーザーの名前を後に付けて `userdel -r` コマンドを実行します。

```
[ec2-user ~]$ sudo userdel -r olduser
```



#### Note

ユーザーのホームディレクトリとメールスプールを維持するには、`-r` オプションを省略します。

## インスタンスの時間を設定します

### Abstract

時間を設定するか、Amazon Linux インスタンスの時間帯を変更します。

サーバータスクとプロセスの多くには、一貫して正確な時間参照が不可欠です。システムログのほとんどにタイムスタンプが含まれています。これを利用すれば、問題が発生した時刻とイベントの発生順序を判断できます。ネットワークタイムプロトコル ( NTP ) は、デフォルトでは Amazon Linux インスタンスに設定されています。システム時間はインターネットで負荷が分散されたパブリックサーバーのプールと同期し、UTC 時間帯に設定されます。NTP の詳細については、<http://www.ntp.org/> を参照してください。

#### Topics

- [時間帯の変更 \(p. 371\)](#)
- [ネットワークタイムプロトコル \( NTP \) の構成 \(p. 372\)](#)

このセクションの手順は、インスタンスに時間帯を設定する方法と、NTP オペレーションを設定し、検証する方法を示すものです。



#### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

## 時間帯の変更

Amazon Linux インスタンスは、デフォルトでは UTC ( 協定世界時間 ) 時間帯に設定されていますが、インスタンスの時間を現地時間またはネットワークの別の時間帯に変更したい場合があるでしょう。

インスタンスの時間帯を変更するには

1. インスタンスで使用する時間帯を特定します。/usr/share/zoneinfo ディレクトリには、時間帯データファイルの階層が含まれています。その場所でディレクトリ構造を閲覧し、お客様の時間帯のファイルを見つけます。

```
[ec2-user ~]$ ls /usr/share/zoneinfo
Africa      Chile      GB         Indian     Mideast    posixrules US
America     CST6CDT   GB-Eire    Iran        MST         PRC        UTC
Antarctica  Cuba      GMT        iso3166.tab MST7MDT     PST8PDT    WET
Arctic      EET       GMT0       Israel      Navajo     right      W-SU
...
```

この場所にある一部のエントリーはディレクトリです ( など )。そのディレクトリには、特定の都市の時間帯ファイルが含まれています。インスタンスに使用する都市 ( またはお客様の時間帯と同じ都市 ) を見つけます。この例では、ロサンゼルス ( Los Angeles ) の時間帯ファイル ( /usr/share/zoneinfo/America/Los\_Angeles ) を使用できます。

2. /etc/sysconfig/clock ファイルを新しい時間帯で更新します。
  - a. お好みのテキストエディタ ( vim や nano など ) で /etc/sysconfig/clock ファイルを開きます。エディタコマンドで sudo を使用する必要があります。/etc/sysconfig/clock は 所有するためです。
  - b. エントリーを見つけて、それを時間帯ファイルに変更します ( パスの /usr/share/zoneinfo セクションは省略します )。例えば、ロサンゼルス ( Los Angeles ) の時間帯に変更するには、ZONE エントリーを次のように変更します。

```
ZONE="America/Los_Angeles"
```

- c. ファイルを保存し、テキストエディタを終了します。
3. インスタンスが現地時間情報を参照するとき、時間帯ファイルが見つけられるように、`/etc/localtime` と時間帯ファイルの間にシンボリックリンクを作成します。

```
[ec2-user ~]$ sudo ln -sf /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

4. システムを再起動し、すべてのサーバーとアプリケーションで新しい時間帯情報を取得します。

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

## ネットワークタイムプロトコル ( NTP ) の構成

### Abstract

NTP を使って、Amazon Linux インスタンスの時刻を正確に維持します。

ネットワークタイムプロトコル ( NTP ) は、デフォルトにより、Amazon Linux インスタンスに構成済みです。ただし、標準 NTP 構成を機能させるには、インスタンスがインターネットにアクセスする必要があります。このセクションの手順では、デフォルトの NTP 構成が正しく動作していることを確認する方法を示します。インスタンスからインターネットにアクセスできない場合、正確な時刻を維持するために、NTP を構成して、プライベートネットワーク内の別のサーバーにクエリを実行する必要があります。

NTP が適切に動作していることを確認するには

1. インスタンスの NTP サービスのステータスを表示するには、`ntpstat` コマンドを使用します。

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

出力が次のような場合、インスタンスの NTP は適切に動作しています。

```
synchronised to NTP server (12.34.56.78) at stratum 3
time correct to within 399 ms
polling server every 64 s
```

出力ステータスが「`unsynchronised`」の場合、1分待ってから再試行してください。最初の同期が完了するまで、1分かかることがあります。

出力ステータスが「`Unable to talk to NTP daemon. Is it running?`」の場合、NTP サービスを起動して、ブート時に自動的に開始できるようにする必要があります。

2. ( オプション ) NTP サーバーが認識しているピアとその状態の概要を表示するには、`ntpq -p` コマンドを使用します。

```
[ec2-user ~]$ ntpq -p
      remote           refid      st t when poll reach   delay   offset
jitter
=====
+lttleman.deekay 204.9.54.119      2 u  15 128 377   88.649    5.946
```

```
6.876
-bittorrent.tomh 91.189.94.4      3 u  133 128 377 182.673    8.001
1.278
*ntp3.junkemailf 216.218.254.202 2 u   68 128 377 29.377     4.726
11.887
+tesla.selinc.co 149.20.64.28    2 u   31 128 377 28.586    -1.215
1.435
```

### NTP を起動して有効にするには

1. 次のコマンドを使って NTP サービスを起動します。

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

2. chkconfig コマンドを使って、ブート時に NTP を開始できるようにします。

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

3. 次のコマンドを使って、NTP が有効であることを確認します。

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

ここでは実行レベル 2、3、4、および 5 で ntpd がオンになっており、これは適正です。

### NTP サーバーを変更するには

標準 NTP サーバーを使用しないようにすることも、インターネットアクセスを持たないインスタンスに対して、プライベートネットワーク内にあるご自分の NTP サーバーを使用することもできます。

1. お好みのテキストエディタ ( vim や nano など ) で、 /etc/ntp.conf ファイルを開きます。 /etc/ntp.conf は root が所有しているため、エディタコマンドで sudo を使用する必要があります。
2. server セクションを探します。このセクションは、NTP 構成に対してポーリングするサーバーを定義します。

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.amazon.pool.ntp.org iburst
server 1.amazon.pool.ntp.org iburst
server 2.amazon.pool.ntp.org iburst
server 3.amazon.pool.ntp.org iburst
```

3. 使用しないサーバー定義の先頭に「#」文字を追加して、使用しないサーバーをコメントアウトします。

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
#server 0.amazon.pool.ntp.org iburst
```

```
#server 1.amazon.pool.ntp.org iburst  
#server 2.amazon.pool.ntp.org iburst  
#server 3.amazon.pool.ntp.org iburst
```

- 時刻同期のポーリング用に、各サーバーのエントリを追加します。このエントリには DNS 名を使用することも、ドット区切りの IP アドレス ( 10.0.0.254 など ) を使用することもできます。

```
server my-ntp-server.my-domain.com iburst
```

- NTP サービスを再起動して、新しいサーバーを選択します。

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

- 新しい設定が動作すること、また、NTP が機能していることを確認します。

```
[ec2-user ~]$ ntpstat  
synchronised to NTP server (64.246.132.14) at stratum 2  
time correct to within 99 ms
```

## インスタンスのホスト名を変更する

### Abstract

動的 DNS プロバイダを使用し、Amazon Linux インスタンスのホスト名を設定します。

インスタンスを起動すると、プライベートの内部 IP アドレスの形式のホスト名が割り当てられます。典型的な Amazon EC2 プライベート DNS 名は、`ip-12-34-56-78.us-west-2.compute.internal` のような形式になります。この名前は内部ドメイン、サービス ( この例では、`compute` )、リージョン、そしてプライベート IP アドレスで構成されます。インスタンスにログインしたとき、このホスト名の一部がシェルプロンプトで表示されます ( `ip-12-34-56-78` など )。Amazon EC2 インスタンスを停止し、再起動するたびに ( Elastic IP アドレスを使用していない限り )、パブリックとプライベートの IP アドレスが変わり、パブリック DNS 名、システムホスト名、シェルプロンプトも変わります。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

## システムホスト名の変更

インスタンスの IP アドレスにパブリック DNS 名を登録している場合 ( `webserver.mydomain.com` など )、インスタンスがそのドメインに含まれているものとして識別されるように、システムホスト名を設定できます。また、システムホスト名を変更すると、シェルプロンプトも変更され、AWS が提供するホスト名の代わりに、新しいシステムホスト名の最初の部分 ( `ip-12-34-56-78` など ) が表示されます。パブリック DNS 名を登録していない場合でもホスト名は変更できますが、プロセスが少し違います。

### システムホスト名をパブリック DNS 名に変更するには

パブリック DNS 名を登録している場合、この手順を行います。

1. お好みのテキストエディタで `/etc/sysconfig/network` 設定ファイルを開き、`HOSTNAME` エントリを変更して、完全修飾ドメイン名 (`webserver.mydomain.com` など) を反映させます。

```
HOSTNAME=webserver.mydomain.com
```

2. インスタンスを再起動し、新しいホスト名を取得します。

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

3. インスタンスにログインして、ホスト名が更新されていることを確認します。メッセージには、新しいホスト名が表示されるはずですが (最初の「`.`」まで)。`hostname` コマンドで完全修飾ドメイン名が表示されます。

```
[ec2-user@webserver ~]$ hostname  
webserver.mydomain.com
```

### パブリック DNS 名なしでシステムホスト名を変更するには

1. お好みのテキストエディタで `/etc/sysconfig/network` 設定ファイルを開き、`HOSTNAME` エントリを変更して、希望するホスト名を反映させます (`webserver` など)。

```
HOSTNAME=webserver.localdomain
```

2. お好みのテキストエディタで `/etc/hosts` ファイルを開き、下の例と一致する `127.0.0.1` で始まるエントリを変更します。ホスト名は自分のホスト名に置換します。

```
127.0.0.1 webserver.localdomain webserver localhost localhost.localdomain
```

3. インスタンスを再起動し、新しいホスト名を取得します。

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

4. インスタンスにログインして、ホスト名が更新されていることを確認します。メッセージには、新しいホスト名が表示されるはずですが (最初の「`.`」まで)。`hostname` コマンドで完全修飾ドメイン名が表示されます。

```
[ec2-user@webserver ~]$ hostname  
webserver.localdomain
```

## ホスト名に影響を与えずにシェルプロンプトを変更する

インスタンスのホスト名を変更せずに、AWS が提供するプライベート名 ( ip-12-34-56-78 など ) よりも便利なシステム名 ( webserver など ) を表示させる場合、ホスト名の代わりにシステムネットワーク名を表示するようにシェルプロンプト設定ファイルを編集できます。

シェルプロンプトをホストニックネームに変更するには

1. /etc/profile.d で、NICKNAME と呼ばれる環境変数を設定するファイルを作成して、シェルプロンプトに表示する値を設定します。例えば、システムニックネームを webserver に設定するには、次のコマンドを実行します。

```
[ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/profile.d/prompt.sh'
```

2. お好みのテキストエディタ ( vim や nano など ) で /etc/bashrc ファイルを開きます。エディタコマンドで sudo を使用する必要があります。/etc/bashrc は root が所有するためです。
3. ファイルを編集し、ホスト名の代わりにニックネームを表示するようにシェルプロンプト変数 ( PS1 ) を変更します。/etc/bashrc でシェルプロンプトを設定する次の行を見つけます ( 以下には、コンテキストを示すため前後の行も表示されています。[ "\$PS1" で始まる行を探してください ) 。

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@\h \W\\\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

その行の \h ( hostname を表す記号 ) を NICKNAME 変数の値に変更します。

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\s-\v\\\$ " ] && PS1="\u@$NICKNAME \W\\\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

4. ( オプション ) シェルウィンドウのタイトルを新しいニックネームに設定するには、次の手順を完了します。

- a. /etc/sysconfig/bash-prompt-xterm という名前のファイルを作成します。

```
[ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
```

- b. 次のコマンドを使用して、ファイルを実行可能にします。

```
[ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
```

- c. お好みのテキストエディタ ( vim や nano など ) で /etc/sysconfig/bash-prompt-xterm ファイルを開きます。エディタコマンドで sudo を使用する必要があります。/etc/sysconfig/bash-prompt-xterm は root が所有するためです。

- d. 次の行をファイルに追加します。

```
echo -ne "\033]0;${USER}@${NICKNAME}:${PWD/#$HOME/~}\007"
```

5. ログアウトしてから再度ログインし、新しいニックネーム値を取得します。

## Amazon EC2 で動的 DNS を使用する

Amazon EC2 インスタンスを起動すると、パブリック IP アドレスとパブリック DNS (ドメイン名システム) 名が割り当てられます。それらを使用してインターネットからインスタンスにアクセスできます。Amazon Web Services ドメインには数多くのホストが存在するため、これらのパブリック名はそれぞれの名前を一意にするために、かなり長くする必要があります。典型的な Amazon EC2 パブリック DNS 名は、`ec2-12-34-56-78.us-west-2.compute.amazonaws.com` のような形式になります。この名前は Amazon Web Services ドメイン、サービス (この例では、`compute`)、リージョン、パブリック IP アドレスで構成されます。

動的 DNS サービスはそのドメイン領域内でカスタムの DNS ホスト名を提供します。この名前は覚えやすく、ホストのユースケースとの関連性が高くなっています。また、一部の動的 DNS サービスは無料です。Amazon EC2 では動的 DNS プロバイダを利用できます。また、インスタンスを起動するたびに、パブリック DNS 名に関連付けられている IP アドレスを更新するようにインスタンスを設定できます。プロバイダは数多く存在します。また、プロバイダを選択し、それぞれのプロバイダで名前を登録する方法については本ガイドの範囲外です。インターネットで「動的 DNS」で検索すれば、たくさんのサービスプロバイダがヒットします。その多くは動的 DNS サービスを無料で提供しています。



### Important

次の手順は、Amazon Linux での使用を想定していますが、コマンドとファイルの場所は Red Hat および CentOS と似ています。その他のディストリビューションの詳細については、各ドキュメントを参照してください。

### Amazon EC2 で動的 DNS を使用するには

1. 動的 DNS サービスプロバイダにサインアップし、そのサービスでパブリック DNS 名を登録します。この手順では、[no-ip.com/free](https://no-ip.com/free) の無料サービスを利用できます。
2. 動的 DNS 更新クライアントを設定します。動的 DNS サービスプロバイダを選び、そのサービスでパブリック DNS 名を登録したら、その DNS 名をインスタンスの IP アドレスにポイントします。多くのプロバイダでは、ウェブサイトのお客様のアカウントから手動で設定できますが、ソフトウェア更新クライアントで自動実行することもできます。この例では、`noip2` クライアントを使用できます。このクライアントは、[no-ip.com](https://no-ip.com) が提供するサービスでのみ使用できます。



### Note

他のサービスプロバイダは独自のクライアントを提供する場合があります。あるいは、そのサービスとともに使用できる `ddclient` 更新ユーティリティを設定できる場合があります。詳細については、お客様のサービスプロバイダが提供する文書と

<http://sourceforge.net/p/ddclient/wiki/Home/> を参照してください。

- a. `noip` クライアントにアクセスできるように、Extra Packages for Enterprise Linux (EPEL) リポジトリを有効にします。`ddclient` パッケージはこのリポジトリでも利用できますが、後続の設定ステップが異なります。

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

- b. noip パッケージをインストールします。

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

- c. noip2 設定ファイルを作成します。メッセージが表示されたら、ログインおよびパスワード情報を入力して、後から続く質問に答え、クライアントを設定します。

```
[ec2-user ~]$ sudo noip2 -C
```

3. chkconfig コマンドを使用し、noip サービスを有効にします。

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

chkconfig --list コマンドを使用すると、サービスが有効になっていることを確認できます。

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

ここでは、noip はランレベル 2、3、4、5 で on になっています (これは正常です)。これで、更新クライアントは起動のたびに開始し、パブリック DNS レコードを更新して、インスタンスの IP アドレスをポイントします。

4. noip サービスを開始します。

```
[ec2-user ~]$ sudo service noip start  
Starting noip2: [ OK ]
```

このコマンドを使用すると、クライアントが起動します。クライアントは前に作成した設定ファイル ( /etc/no-ip2.conf ) を読み、選択したパブリック DNS 名の IP アドレスを更新します。

5. 更新クライアントが動的 DNS 名に正しい IP アドレスを設定したことを確認します。DNS レコードの更新には数分かかります。その後、この手順で設定したパブリック DNS 名により、SSH 経由でインスタンスに接続します。

## ユーザーデータでインスタンスを起動する

### Abstract

起動時に Amazon Linux インスタンスにユーザーデータを渡します。

Amazon EC2 でインスタンスを起動するとき、起動後にそのインスタンスにユーザーデータを渡し、一般的な自動設定タスクを実行したり、スクリプトを実行したりできます。2 つのタイプのユーザーデータを Amazon EC2 に渡すことができます。シェルスクリプトと cloud-init デイレクティブです。また、このデータはプレーンテキスト、ファイル ( コマンドラインツールでインスタンスを起動する場合に便利です )、または base64 でエンコードされたテキスト ( API 呼出の場合 ) として、起動ウィザードに渡すこともできます。

より複雑な自動化シナリオに興味がある場合、AWS CloudFormation や AWS OpsWorks のご利用を検討してください。詳細については、[AWS CloudFormation ユーザーガイド](#) と [AWS OpsWorks ユーザーガイド](#) を参照してください。

#### Topics

- [ユーザーデータとシェルスクリプト \(p. 379\)](#)
- [ユーザーデータと cloud-init ディレクティブ \(p. 381\)](#)

次の例では、[Installing a LAMP Web Server tutorial \(p. 39\)](#) のコマンドが、シェルスクリプトとインスタンスの起動時に実行される一連の cloud-init ディレクティブに変換されています。各例では、次のタスクがユーザーデータにより実行されます。

- ディストリビューションソフトウェアパッケージが更新されます。
- 必要なウェブサーバー、php、mysql パッケージがインストールされます。
- chkconfig を介して httpd サービスが開始され、オンになります。
- www グループが追加され、ec2-user がそのグループに追加されます。
- ウェブディレクトリとその中に含まれるファイルに対して、適切な所有権とファイル権限が設定されます。
- ウェブサーバーと php エンジン进行测试するために、シンプルなウェブページが作成されます。

#### 前提条件

次の例では、インターネットからアクセス可能なパブリック DNS 名がお客様のインスタンスに設定されているものと仮定しています。詳細については、[Amazon EC2 インスタンスの起動 \(p. 28\)](#) を参照してください。また、SSH (ポート 22)、HTTP (ポート 80)、HTTPS (ポート 443) 接続を許可するように、セキュリティグループを設定する必要もあります。前提条件の詳細については、[Amazon EC2 でのセットアップ \(p. 21\)](#) を参照してください。

また、これらの指示は Amazon Linux での使用を意図しています。他の Linux ディストリビューションの場合、コマンドとディレクティブが動作しないことがあります。cloud-init のサポートなど、その他のディストリビューションについての詳細は、該当するディストリビューションの文書を参照してください。

## ユーザーデータとシェルスクリプト

シェルスクリプトに詳しい場合、シェルスクリプトが起動時に指示を送る最も簡単で完全な方法となります。cloud-init ログファイル ( /var/log/cloud-init.log ) にコンソール出力が記録されるので、インスタンスが意図したとおりに動作しない場合、起動後にスクリプトを簡単にデバッグできます。



#### Important

ユーザーデータのスクリプトおよび cloud-init ディレクティブは、インスタンスを起動するときの初回の起動サイクルでのみ実行されます。

ユーザーデータシェルスクリプトは、#! 文字とスクリプトを読み取るインタープリターへのパス ( 通常、/bin/bash ) で始める必要があります。シェルスクリプトの概要については、Linux Documentation Project ( [tldp.org](http://tldp.org) ) の [BASH Programming HOW-TO](#) を参照してください。

ユーザーデータとして入力されたスクリプトは、root ユーザーとして実行されます。そのため、スクリプトでは sudo コマンドを使用しないでください。作成したファイルはすべて root の所有になることを忘れないでください。非 root ユーザーにファイルアクセスを与える場合、スクリプトで権限を適宜変更する必要があります。また、スクリプトはインタラクティブに実行されないため、ユーザーフィードバックを必要とするコマンド ( -y フラグのない yum update など ) を含めることはできません。

起動時にこれらのタスクを追加すると、インスタンスの起動にかかる時間が増えます。タスクが完了するまでさらに数分待ち、それからユーザースクリプトが正常に完了したことをテストしてください。

ユーザーデータでシェルスクリプトをインスタンスに渡すには

1. [AMIからのインスタンスの起動 \(p. 325\)](#) でインスタンスを起動するための手順を行いますが、[Step 6 \(p. 327\)](#) に到達したら、ユーザーデータスクリプトテキストを [UserData] フィールドに貼り付け、起動手順を完了します。下の例の場合では、スクリプトがウェブサーバーを作成し、設定します。

```
#!/bin/bash
yum update -y
yum groupinstall -y "Web Server" "MySQL Database" "PHP Support"
yum install -y php-mysql
service httpd start
chkconfig httpd on
groupadd www
usermod -a -G www ec2-user
chown -R root:www /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} +
find /var/www -type f -exec chmod 0664 {} +
echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

2. インスタンスが起動し、スクリプトのコマンドを実行するまで十分待ち、それからスクリプトが意図したタスクを完了したことを確認します。例では、ウェブブラウザにスクリプトが作成した PHP テストファイルの URL を入力します。この URL は、インスタンスのパブリック DNS アドレスにスラッシュとファイル名を追加したものです。

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

PHP 情報ページが表示されるはずですが。



#### Tip

PHP 情報ページが表示されない場合、使用しているセキュリティグループに HTTP (ポート 80) トラフィックを許可するルールが含まれていることを確認します。HTTP ルールをセキュリティグループに追加する方法については、[セキュリティグループへのルールの追加 \(p. 482\)](#) を参照してください。

3. (オプション) スクリプトが予定のタスクを完了しなかった場合、あるいはスクリプトがエラーせずにタスクを完了したかを確認するには、`/var/log/cloud-init.log` にある `cloud-init` ログファイルを調べ、エラーメッセージが出力されていないか探します。

デバッグの詳細情報を取得するには、ディレクティブ

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

を指定して `cloud-init` データセクションを含む MIME マルチパートアーカイブを作成します。このディレクティブにより、スクリプトから `/var/log/cloud-init-output.log` にコマンド出力が送信されます。`cloud-init` データ形式と MIME マルチパートアーカイブの作成方法の詳細については、「[cloud-init Formats](#)」を参照してください。

## ユーザーデータと cloud-init デイレクティブ

cloud-init パッケージは、新しい Amazon Linux インスタンスが起動したときに、特定の側面を設定します。具体的には、お客様のプライベートキーでログインできるように、ec2-user の .ssh/authorized\_keys ファイルを設定します。

構文は異なりますが、渡されたスクリプトと同じ方法で cloud-init ユーザーデイレクティブを起動時のインスタンスに渡すことができます。cloud-init の詳細については、<http://cloudinit.readthedocs.org/en/latest/index.html> にアクセスしてください。



### Important

ユーザーデータのスクリプトおよび cloud-init デイレクティブは、インスタンスを起動するときの初回の起動サイクルでのみ実行されます。

cloud-init の Amazon Linux バージョンは、ベースパッケージで利用できるデイレクティブの一部をサポートしません。一部のデイレクティブは名前が変更されています (例えば、apt-upgrade の代わりに repo\_update が使用されています)。Amazon Linux で使用できる cloud-init のサンプルをより完全なセットでお求めの場合、

/usr/share/doc/cloud-init-0.5.15/examples/cloud-init-amzn.txt から Amazon Linux インスタンスを実行するサンプルファイルを参照してください。

起動時にこれらのタスクを追加すると、インスタンスの起動にかかる時間が増えます。タスクが完了するまでさらに数分待ち、それからユーザーデータデイレクティブが完了したことをテストしてください。

ユーザーデータで cloud-init デイレクティブをインスタンスに渡すには

1. [AMI からのインスタンスの起動 \(p. 325\)](#) でインスタンスを起動するための手順を行いますが、[Step 6 \(p. 327\)](#) に到達したら、cloud-init デイレクティブテキストを [User data] フィールドに貼り付けて、起動手順を完了します。下の例では、デイレクティブがウェブサーバーを作成し、設定します。

```
#cloud-config
repo_update: true
repo_upgrade: all

packages:
- httpd
- crypto-utils
- mod_perl
- mod_ssl
- mod_wsgi
- mysql-server
- php
- php-gd
- php-pdo
- php-pear
- php-xml
- php-mysql

runcmd:
- service httpd start
- chkconfig httpd on
- groupadd www
- [ sh, -c, "usermod -a -G www ec2-user" ]
```

```
- [ sh, -c, "chown -R root:www /var/www" ]  
- chmod 2775 /var/www  
- [ find, /var/www, -type, d, -exec, chmod, 2775, {}, + ]  
- [ find, /var/www, -type, f, -exec, chmod, 0664, {}, + ]  
- [ sh, -c, 'echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php' ]
```

2. インスタンスが起動し、ユーザーデータのディレクティブを実行するまで十分待ち、それから意図したタスクをディレクティブが完了したことを確認します。例では、ウェブブラウザにディレクティブが作成した PHP テストファイルの URL を入力します。この URL は、インスタンスのパブリック DNS アドレスにスラッシュとファイル名を追加したものです。

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

PHP 情報ページが表示されるはずですが。



#### Tip

PHP 情報ページが表示されない場合、使用しているセキュリティグループに HTTP (ポート 80) トラフィックを許可するルールが含まれていることを確認します。HTTP ルールをセキュリティグループに追加する方法については、[セキュリティグループへのルールの追加 \(p. 482\)](#) を参照してください。

3. (オプション) ディレクティブが予定のタスクを完了しなかった場合、あるいはディレクティブがエラーなしでタスクを完了したかを確認するには、`/var/log/cloud-init.log` にある cloud-init ログファイルを調べ、エラーメッセージが出力されていないか探します。デバッグの詳細情報を取得するには、ディレクティブに次の行を追加します：

```
output : { all : '| tee -a /var/log/cloud-init-output.log' }
```

このディレクティブにより、`runcmd` 出力が `/var/log/cloud-init-output.log` に送信されません。

# Amazon EC2 のモニタリング

モニタリングは、Amazon Elastic Compute Cloud ( Amazon EC2 ) インスタンスおよび AWS ソリューションの信頼性、可用性、およびパフォーマンスを維持する上で重要な部分です。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。ただし、Amazon EC2 のモニタリングを開始する前に、次の内容を盛り込んだモニタリング計画を作成する必要があります。

- モニタリングの目的とは？
- モニタリングの対象となるリソースとは？
- これらのリソースをモニタリングする頻度は？
- 使用するモニタリングツールは？
- 誰がモニタリングタスクを実行しますか？
- 誰が問題が発生したときに通知を受け取りますか？

モニタリングの目的を定義し、モニタリングの計画を作成したら、次のステップとして、お客様の環境内で通常の Amazon EC2 パフォーマンスのベースラインを確立します。さまざまな時間帯に、さまざまな負荷条件で Amazon EC2 パフォーマンスを測定します。Amazon EC2 をモニタリングしながら、収集したモニタリングデータの履歴を格納する必要があります。現在の Amazon EC2 パフォーマンスをこの履歴データと比較して、通常のパフォーマンスパターンとパフォーマンス異常を識別することで、異常への対処方法を考案することが容易になります。例えば、Amazon EC2 インスタンスの CPU 使用率、ディスク I/O、およびネットワーク使用率をモニタリングすることができます。確立したベースラインからパフォーマンスが外れた場合は、インスタンスの再設定または最適化を行って CPU 使用率の抑制、ディスク I/O の改善、またはネットワークトラフィックの低減を行うことが必要な場合があります。

ベースラインを確立するには、少なくとも、次の項目をモニタリングする必要があります。

モニタリング対象の項目	Amazon EC2 メトリックス	モニタリングスクリプト
CPU 使用率	<a href="#">CPUUtilization</a>	
メモリ使用率		<a href="#">Amazon EC2 インスタンスのモニタリングスクリプト</a>
使用中のメモリ		<a href="#">Amazon EC2 インスタンスのモニタリングスクリプト</a>

モニタリング対象の項目	Amazon EC2 メトリックス	モニタリングスクリプト
利用可能なメモリ		<a href="#">Amazon EC2 インスタンスのモニタリングスクリプト</a>
ネットワーク使用率	<a href="#">NetworkIn</a> <a href="#">NetworkOut</a>	
ディスクパフォーマンス	<a href="#">DiskReadOps</a> <a href="#">DiskWriteOps</a>	
ディスクスワップの使用率 使用中のスワップ (Linux)		<a href="#">Amazon EC2 Linux インスタンスのモニタリングスクリプト</a>
ページファイルの使用率 使用中のページファイル 使用可能なページファイル		<a href="#">Amazon EC2 Windows インスタンスのモニタリングスクリプト</a>
ディスクの読み書き	<a href="#">DiskReadBytes</a> <a href="#">DiskWriteBytes</a>	
ディスクスペースの使用率		<a href="#">Amazon EC2 インスタンスのモニタリングスクリプト</a>
使用中のディスク容量		<a href="#">Amazon EC2 インスタンスのモニタリングスクリプト</a>
利用可能なディスク容量		<a href="#">Amazon EC2 インスタンスのモニタリングスクリプト</a>

## 自動モニタリングと手動モニタリング

### Abstract

AWS ツールを設定して Amazon EC2 インスタンスをモニタリングします。これらのツールの一部は自動化されていますが、手動操作を必要とするものもあります。

AWS では、Amazon EC2 のモニタリングに使用できるさまざまなツールを提供しています。これらのツールの中には、自動モニタリングを設定できるものもあれば、手操作を必要とするものもあります。

### Topics

- [自動モニタリングツール \(p. 384\)](#)
- [手動モニタリングツール \(p. 385\)](#)

## 自動モニタリングツール

次に示す自動化されたモニタリングツールを使用すると、Amazon EC2 の監視が行われ、問題が検出されたときにレポートが返されます。

- [System Status Checks] – インスタンスを使用する際に必要な AWS システムをモニタリングして、AWS システムが正常に実行されていることを確認します。これらのチェックでは、修復において

AWS が関係するインスタンスの問題が検出されます。システムステータスチェックが失敗した場合、AWS によって問題が修正されるのを待つか、自分自身で（例えば、インスタンスを停止、再起動、終了、置換するなどによって）問題を解決できます。システムステータスチェックの失敗の原因となる問題には、次のようなものがあります。

- ネットワーク接続の喪失
- システム電源の喪失
- 物理ホストのソフトウェアの問題
- 物理ホストのハードウェアの問題

詳細については、[Monitoring Instances with Status Checks](#) を参照してください。

- [Instance Status Checks] – 個々のインスタンスのソフトウェアとネットワークの設定をモニタリングします。これらのチェックでは、ユーザーが関与して修復する必要がある問題が検出されます。インスタンスステータスチェックが失敗した場合、通常は自分自身で（例えば、インスタンスを再起動する、オペレーティングシステムを修正するなどによって）問題に対処する必要があります。インスタンスステータスチェックの失敗の原因となる問題には、次のようなものがあります。
  - 失敗したシステムステータスチェック
  - 誤って設定されたネットワークまたは起動設定
  - メモリの枯渇
  - 破損したファイルシステム
  - 互換性のないカーネル

詳細については、[Monitoring Instances with Status Checks](#) を参照してください。

- [Amazon CloudWatch Alarms] – 指定された期間にわたって単一のメトリックスを監視し、複数の期間にわたり既定のしきい値に関連するメトリックス値に基づいて 1 つ以上のアクションを実行します。アクションは、Amazon Simple Notification Service ( Amazon SNS ) トピックまたは Auto Scaling ポリシーに送信される通知です。アラームは、持続的な状態変化の場合のみアクションを呼び出します。CloudWatch アラームはそれが特定の状態にあるという理由だけではアクションを呼び出しません。アクションを呼び出すには、状態が変化し、その状態が特定の期間にわたって維持されている必要があります。詳細については、[Monitoring Your Instances with CloudWatch](#) を参照してください。
- [Amazon EC2 Monitoring Scripts for Linux and Windows] – Linux および Windows インスタンスでのメモリ、ディスク、およびページ/スワップファイルの使用率をモニタリングできる Perl および PowerShell スクリプトです。詳細については、[Amazon EC2 インスタンスのモニタリングスクリプト](#) を参照してください。
- [AWS Management Pack for Microsoft System Center Operations Manager] – Amazon EC2 インスタンスと、これらのインスタンス内で稼働する Microsoft Windows または Linux オペレーティングシステムをリンクします。AWS マネジメントパックは Microsoft System Center Operations Manager 向けの拡張パックです。データセンターの指定されたコンピュータ（監視ノードと呼びます）と Amazon ウェブサービス API を使用して、AWS リソースに関する情報をリモートで検出して収集します。詳細については、[Microsoft System Center Operations Manager 向け AWS マネジメントパック](#) を参照してください。

## 手動モニタリングツール

Amazon EC2 のモニタリングにおけるもう 1 つの重要な部分は、モニタリングスクリプト、ステータスチェック、および CloudWatch アラームで網羅されていない項目を手動でモニタリングすることです。Amazon EC2 および CloudWatch のコンソールダッシュボードには、Amazon EC2 環境の状態が一目でわかるビューが表示されます。

- Amazon EC2 ダッシュボードには次の内容が表示されます。
  - リージョンごとのサービス状態とスケジュールされたイベント
  - インスタンスの状態
  - ステータスチェック

- アラームステータス
- インスタンスメトリックスの詳細 ( ナビゲーションペインで、[Instances] をクリックし、インスタンスを選択して、[Monitoring] タブをクリックします )
- ボリュームメトリックスの詳細 ( ナビゲーションペインの [Volumes] をクリックし、ボリュームを選択して、[Monitoring] タブをクリックします )
- Amazon CloudWatch ダッシュボードには、次の内容が表示されます。
  - 現在のアラームとステータス
  - アラームとリソースのグラフ
  - サービス状態ステータス

さらに、CloudWatch を使用して次のことが行えます。

- Amazon EC2 モニタリングデータをグラフ化して、問題のトラブルシューティングを行い、傾向を確認する
- AWS リソースのすべてのメトリックスを検索して、参照する
- 問題があることを通知するアラームを作成/編集する
- アラームおよび AWS リソースが一目でわかる概要を表示する

## モニタリングのベストプラクティス

### Abstract

Amazon EC2 モニタリングタスクに役立つ、モニタリングのベストプラクティスを説明します。

次に示すモニタリングのベストプラクティスを使用すると、Amazon EC2 のモニタリングタスクが容易になります。

- モニタリングの優先順位を設定し、小さな問題が大きな問題に発展する前に阻止します。
- AWS ソリューションのすべての部分からモニタリングデータを収集するモニタリング計画を作成し、実施すると、マルチポイント障害が発生した場合に、その障害をより簡単にデバッグできます。モニタリング計画には、少なくとも、次の質問に対する回答を盛り込む必要があります。
  - モニタリングの目的とは？
  - モニタリングの対象となるリソースとは？
  - これらのリソースをモニタリングする頻度は？
  - 使用するモニタリングツールは？
  - 誰がモニタリングタスクを実行しますか？
  - 誰が問題が発生したときに通知を受け取りますか？
- モニタリングタスクは可能な限り自動化します。
- EC2 インスタンスでログファイルを確認します。

## インスタンスのステータスのモニタリング

### Abstract

インスタンスのステータスチェックや、インスタンスに予定されたイベントを表示して、Amazon EC2 インスタンスのステータスをモニタリングします。

インスタンスのステータスをモニタリングして、インスタンスのステータスチェックや、インスタンスにスケジュールされたイベントを表示できます。ステータスチェックでは、Amazon EC2 によって実行される自動化されたチェックからの情報が提供されます。これらの自動化されたチェックは、特定の

問題がインスタンスに影響を与えているかどうかを検出します。ステータスチェックの情報と、Amazon CloudWatch で提供されるデータによって、各インスタンスの詳細な動作状況を把握できます。

インスタンスに予定されている特定のイベントに関するステータスも表示できます。イベントでは、インスタンスに計画されている再起動またはリタイアなどの今後のアクティビティに関する情報と、各イベントで予定されている開始時間と終了時間が提供されます。

#### Topics

- [ステータスチェックでインスタンスをモニタリングする \(p. 387\)](#)
- [インスタンスのイベントのモニタリング \(p. 392\)](#)

## ステータスチェックでインスタンスをモニタリングする

### Abstract

ステータスチェックでインスタンスをモニタリングして、Amazon EC2 がインスタンスで何らかの問題を検出しているかどうかを判断します。

インスタンスのステータスのモニタリングでは、Amazon EC2 によってインスタンスによるアプリケーションの実行が妨げられるような問題が検出されたかどうかをすばやく判断できます。Amazon EC2 は、実行されている各 Amazon EC2 インスタンスについて自動化されたチェックを実行して、ハードウェアおよびソフトウェアの問題を識別します。これらのステータスチェックの結果を表示して、具体的に検出可能な問題を識別できます。このデータは、Amazon EC2 が各インスタンスで意図された状態（保留、実行、停止など）に関して既に提供している情報および Amazon CloudWatch が監視している利用メトリックス（CPU 使用率、ネットワークトラフィック、ディスクアクティビティ）を補うものです。

ステータスチェックは、5分ごとに実行され、それぞれが成功または失敗のステータスを返します。すべてのチェックが成功すると、インスタンス全体のステータスが OK になります。1つ以上のチェックが失敗すると、全体のステータスが `impaired` になります。ステータスチェックは Amazon EC2 に組み込まれています。そのため、無効にしたり、削除したりすることはできません。ただし、ステータスチェックの結果に基づいて実行されるアラームを作成または削除することは可能です。例えば、特定のインスタンスでステータスチェックが失敗したときに警告するアラームを作成できます。詳細については、「[ステータスチェックアラームの作成と編集 \(p. 390\)](#)」を参照してください。

ステータスチェックには、システムステータスチェックとインスタンスステータスチェックの2種類があります。

System status checks では、インスタンスを使用するために必要な AWS システムを監視して、AWS システムが正常に実行されていることを確認します。これらのチェックでは、修復において AWS が関係するインスタンスの問題が検出されます。システムステータスチェックが失敗した場合、AWS によって問題が修正されるのを待つが、自分自身で（例えば、インスタンスを停止、再起動、終了、置換するなどによって）問題を解決できます。システムステータスチェックの失敗の原因となる問題には、次のようなものがあります。

- ネットワーク接続の喪失
- システム電源の喪失
- 物理ホストのソフトウェアの問題
- 物理ホストのハードウェアの問題

Instance status checks では、個々のインスタンスのソフトウェアとネットワークの設定を監視します。これらのチェックでは、お客様が修復する必要のある問題が検出されます。インスタンスステータスチェックが失敗した場合、通常はお客様ご自身で（インスタンスの再起動、オペレーティングシステム

の修正など)問題を修復する必要があります。インスタンスステータスチェックの失敗の原因となる問題には、次のようなものがあります。

- 失敗したシステムステータスチェック
- 誤って設定されたネットワークまたは起動設定
- メモリの枯渇
- 破損したファイルシステム
- 互換性のないカーネル



#### Note

インスタンスの再起動、または Windows のインスタンスストアバックトインスタンスがバンドルされている間に生じたステータスチェックは、インスタンスが再度使用可能になるまで、インスタンスステータスチェックの失敗を報告します。

## ステータスを表示する

AWS では、AWS Management Console を使用して API をインタラクティブに操作したり、コマンドラインインターフェイスを使用したりするなど、ステータスチェックを表示し、作業する方法をいくつか提示しています。

### Amazon EC2 コンソール

コンソールを使用してステータスチェックを表示するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. [Navigation] ペインの [Instances] をクリックします。
3. [Instances] ページで、[Status Checks] 列には、各インスタンスの動作状況が表示されます。
4. 各インスタンスのステータスを表示するには、インスタンスを選択して、Status Checks タブをクリックします。

Description Status Checks Monitoring Tags

Status checks detect problems that may impair this instance from running your applications. [Learn more](#) about status checks.

Create Status Check Alarm

System Status Checks ⓘ Instance Status Checks ⓘ

These checks monitor the AWS systems required to use this instance and ensure they are functioning properly.  
System reachability check passed

These checks monitor your software and network configuration for this instance.  
Instance reachability check failed at October 7, 2013 11:52:11 AM UTC+2 (16 minutes ago)  
[Learn more about this issue](#)

Additional Resources

[Submit feedback](#) if our checks do not reflect your experience with this instance or if they do not detect the issues you are having.  
Please note that we will not respond to customer support issues reported via this form. Please post your issue on the [Developer Forums](#) or contact [AWS Support](#) if you need technical assistance with this instance.



#### Note

ステータスチェックが失敗したインスタンスがあり、そのインスタンスに 20 分以上アクセスできない場合は、[Contact AWS Support] をクリックして、サポートのリクエストを送信してください。ご自分でシステムまたはインスタンスのステータスチェック失敗のトラブルシューティングを行う場合は、[ステータスチェックに失敗したインスタンスのトラブルシューティング \(p. 693\)](#) を参照してください。

## コマンドラインインターフェイス

目的	実行するコマンド
すべてのインスタンスのステータスを取得する	<code>describe-instance-status</code>
インスタンスステータスが <code>impaired</code> であるすべてのインスタンスのステータスを取得する	<code>describe-instance-status --filters Name=instance-status.status,Values=impaired</code>
<code>i-15a4417c</code> というインスタンス ID を持つ単一インスタンスのステータスを取得する	<code>describe-instance-status --instance-ids -i-15a4417c</code>

`describe-instance-status` コマンドの使用の詳細については、『*AWS Command Line Interface Reference*』の「[describe-instance-status](#)」を参照してください。



### Note

ステータスチェックが失敗したインスタンスがある場合は、[ステータスチェックに失敗したインスタンスのトラブルシューティング \(p. 693\)](#) を参照してください。

## API

`DescribeInstanceStatus` アクションを使用してインスタンスのステータスを取得できます。詳細については、*Amazon Elastic Compute Cloud API Reference* の [DescribeInstanceStatus](#) を参照してください。

## ステータスの報告

ステータスが `impaired` と表示されていないインスタンスで問題が生じている場合は、インスタンスに関するフィードバックをご提供ください。また、障害のあるインスタンスで発生している問題に関する追加の詳細を AWS に送る場合にも、フィードバックをご提供いただけます。

Amazon では、報告されたフィードバックを使用して、複数のお客様に影響する可能性のある問題を識別していますが、このフォームで報告された個々のアカウントの問題には返答しておりません。フィードバックをご提供いただいても、現在このインスタンスに関して表示されているステータスチェックの結果は変わりません。

アカウントに対して個別に技術的なサポートが必要な場合は、開発者フォーラムに質問を投稿するか、プレミアムサポートにご連絡ください。

## Amazon EC2 コンソール

Console を使用してステータスのフィードバックを報告するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. [Navigation] ペインの [Instances] をクリックします。
3. [Instances] ページで、ステータスを報告するインスタンスをクリックします。
4. [Status Checks] タブをクリックしてから、[Submit Feedback] をクリックします。
5. [Report Instance Status] ページで情報の入力を完了します。

## コマンドラインインターフェイス

コマンドラインインターフェイスを使用してステータスのフィードバックを送信するには、`report-instance-status` コマンドを使用します。コマンドでは次の構文を使用します。

```
aws ec2 report-instance-status [--instances ...] [--status ...] [--reason-codes ..]
```

`report-instance-status` コマンドの使用の詳細については、『*AWS Command Line Interface Reference*』の「[report-instance-status](#)」コマンドを参照してください。

## API

`ReportInstanceStatus` アクションを使用して、実行中のインスタンスのステータスに関するフィードバックを提出できます。インスタンスが `DescribeInstanceStatus` アクションによって返されるインスタンスステータスと異なる場合は、`ReportInstanceStatus` を使用してご利用のインスタンスを報告してください。Amazon EC2 では、これらの情報を収集してステータスチェックの精度を向上させます。詳細については、*Amazon Elastic Compute Cloud API Reference* の [ReportInstanceStatus](#) を参照してください。

## ステータスチェックアラームの作成と編集

インスタンスのステータスとシステムのステータスに関するアラームを作成して、インスタンスのステータスチェックが失敗した場合に通知を受け取ることができます。これらのアラームを作成または変更するには、AWS Management Console またはコマンドラインインターフェイス ( CLI ) を使用できます。

## AWS マネジメントコンソール

ステータスチェックアラームを作成するには

既存のインスタンスに対してステータスチェックアラームを作成し、インスタンスまたはシステムのステータスをモニタリングできます。インスタンスのインスタンスチェックまたはシステムステータスチェックが失敗した場合にメールで通知を受け取るようにアラームを設定できます。

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. [Navigation] ペインの [Instances] をクリックします。
3. インスタンスを選択し、[Status Checks] タブで [Create Status Check Alarm] をクリックします。
4. [Create Alarm] ダイアログボックスで [Send a notification to] チェックボックスをオンにし、このアラームに使用する既存の Amazon Simple Notification Service ( SNS ) トピックを選択するか、新しい SNS トピックを作成します。
5. [With these recipients] ボックスに、ご自身のメールアドレス ( 例: john.stiles@example.com ) と、必要に応じてその他の受信者のメールアドレスをコンマで区切って入力します。
6. [Whenever] ドロップダウンリストで、通知を受け取る必要があるステータスチェック ( 例: Status Check Failed ( Any )、Status Check Failed ( Instance )、Status Check Failed ( System ) ) を選択します。
7. [For at least] ボックスで、評価する期間数 ( 例: 2 ) を設定し、[consecutive periods] ドロップダウンメニューで、アラームをトリガーしてメールを送信するまでの評価の間隔 ( 例: 5 分間 ) を選択します。
8. アラームのデフォルト名を変更するには、[Name of alarm] ボックスにアラームのわかりやすい名前 ( 例: StatusCheckFailed ) を入力し、[Create Alarm] をクリックします。



### Important

受信者のリストにメールアドレスを追加したか、新しいトピックを作成した場合、アラーム作成後ほどなく、追加した各メールアドレスに Amazon SNS から登録の確認メールが送信されます。そのメールに示されているリンクをクリックして登録を確認してください。アラート通知は確認されたアドレスにのみ送信されます。

ステータスチェックアラームを編集するには

インスタンスステータスのアラームを変更する必要がある場合は、そのアラームを編集できます。

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. [Navigation] ペインの [Instances] をクリックします。
3. インスタンスを選択し、[ Actions] をクリックして、[Add/Edit Alarms] をクリックします。
4. [Alarm Details] ダイアログボックスで、アラームの名前をクリックします。
5. [Edit] ダイアログボックスで、希望する変更を行い、[Save Alarm] をクリックします。

## コマンドラインインターフェイス

CLI を使用してステータスチェックアラームを作成するには

AWS CLI を使用して、ステータスチェックアラームを作成できます。次の例では、インスタンス `i-ab12345` のインスタンスチェックまたはシステムステータスチェックが少なくとも 2 つの期間で失敗した場合、ARN `arn:aws:sns:us-east-1:111111111111>StatusCheckNotifications` が設定されている特定の SNS トピックに対して通知を発行します ( メトリックスは `StatusCheckFailed` です )。詳細については、『*AWS Command Line Interface Reference*』の「`put-metric-alarm`」コマンドを参照してください。

1. コマンドプロンプトで「`aws cloudwatch list-metrics`」と入力して、使用している AWS のサービスで利用可能な Amazon CloudWatch メトリックスを一覧表示します。
2. メトリックスのリストで、AWS/EC2 名前空間を持つステータスチェックメトリックスを確認します。これらのメトリックスが、ステータスチェックアラームを作成するために使用できるステータスチェックメトリックスです。
3. コマンドプロンプトで、次のコマンドを入力します。

```
aws cloudwatch put-metric-alarm --alarm-name StatusCheckFailed-Alarm-for-i-ab12345 --alarm-description "Alarm when StatusCheckFailed metric has a value of one for two periods" --metric-name StatusCheckFailed --namespace AWS/EC2 --statistic Maximum --dimensions Name=InstanceId,Value=i-ab12345 --period 300 --unit Count --evaluation-periods 2 --threshold 1 --comparison-operator GreaterThanThreshold --alarm-actions arn:aws:sns:us-east-1:111111111111>StatusCheckNotifications
```

各パラメータの意味は次のとおりです。

`--alarm-name` は、アラームの名前です。これは必須です。

`--alarm-description` は、アラームのわかりやすい説明です。

`--metric-name` は、利用可能なステータスメトリックスの 1 つです ( 例: `StatusCheckFailed`、`StatusCheckFailed_Instance`、`StatusCheckFailed_System` )。これは必須です。

`--namespace` は、メトリックスの名前空間です ( 例: `AWS/EC2` )。これは必須です。

--statistic は、Average、Sum、Minimum、Maximum のいずれかの値です。これは必須です。

--dimensions は、メトリックスと関連付けられています (例: InstanceId=i-ab12345)。

--period は、Amazon CloudWatch メトリックスが収集される期間 (秒単位) です。この例では、300 と入力します (5 分を秒数で指定)。これは必須です。

--unit は、アラームに関連付けられたメトリックスの単位です。

--evaluation-periods は、メトリックスの値がしきい値と比較されなければならない連続した期間の数です。これは必須です。

--threshold は、メトリックスが比較される値です (例: 1)。これは必須です。

--alarm-actions は、このアラームがトリガーされたときに実行するアクションのリストです。各アクションは Amazon リソース番号 (ARN) として指定されます。この例では、Amazon SNS を使用してメールが届くようにします。



#### Note

Amazon SNS コントロールでアラームに使用される Amazon SNS トピックの ARN を検索できます。

1. Amazon SNS コンソール (<https://console.aws.amazon.com/sns/>) を開きます。
2. ナビゲーションページの [My Topics] で、アラームが送信するメールの宛先となるトピックを選択します。
3. ARN は [Topic Details] ページの [Topic ARN] フィールドに表示されます。

--unit は、アラームに使用するメトリックスの単位です (例: Count)。

## インスタンスのイベントのモニタリング

### Abstract

インスタンスに、予定されたイベントをモニタリングします。

インスタンスステータスには、再起動やリタイヤなど、AWS がインスタンスに対して予定することができる特定のイベントが記述されます。このような予定されたイベントは頻繁には発生しません。インスタンスのいずれかが予定されたイベントの影響を受ける場合は、予定されたイベントに先立ち、イベントに関する詳細と開始日および終了日が記載された E メールが届きます。Amazon EC2 コンソール、API、または CLI を使用してインスタンスに予定されたイベントを表示することもできます。詳細については、[予定されたイベントの表示 \(p. 393\)](#) を参照してください。

予定されたイベントには、さまざまな種類があります。

- [Reboot]: 再起動は、インスタンスの再起動またはシステムの再起動をどちらかを指定できます。
- [System maintenance]: インスタンスは、ネットワークメンテナンスまたは電源のメンテナンスの影響を一時的に受ける場合があります。
- [Instance retirement]: リタイヤが予定されたインスタンスは、停止または終了されます。
- [Instance stop]: インスタンスを新しいハードウェアに移行するために、インスタンスの停止が必要な場合もあります。

上記のイベントのいずれかがお客様のインスタンスに予定されている場合、イベントのタイミングを制御したり、あるいはダウンタイムを最小限に抑えたりするなど、お客様が操作を行えます。詳細については、[予定されたイベントのあるインスタンスの使用 \(p. 393\)](#) を参照してください。

#### Topics

- [予定されたイベントの表示 \(p. 393\)](#)
- [予定されたイベントのあるインスタンスの使用 \(p. 393\)](#)

## 予定されたイベントの表示

Amazon EC2 コンソール、コマンドラインインターフェイス ( CLI )、または API を使用して、インスタンスに予定されているイベントを表示できます。

### Amazon EC2 コンソール

コンソールを使用してインスタンスに予定されたイベントを表示するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインの [Events] をクリックします。すべてのリソースリストと、それらに関連付けられたイベントを表示できます。インスタンスまたはボリューム別、あるいは特定のステータスタイプ別で、フィルタリングできます。
3. または、以下を実行して今後予定されているイベントを表示できます。
  - a. ナビゲーションペインの [EC2 Dashboard] をクリックします。
  - b. [Scheduled Events] で、Amazon EC2 インスタンスおよびボリュームに関連付けられたイベントが表示されます。

### コマンドラインインターフェイスと API

インスタンスのステータスを表示するには、`ec2-describe-instance-status` コマンドを使用するか、`DescribeInstanceStatus` API アクションを使用します。

## 予定されたイベントのあるインスタンスの使用

このセクションでは、インスタンスに予定されたイベントがある場合に実行できるタスクを説明します。スケジュールされたイベントの場合、インスタンスのルートデバイスボリュームが Amazon EBS ボリュームか、インスタンスストアボリュームかによって、お客様の一連の手順は異なります。インスタンスのルートデバイスタイプを特定するには、[Instances] ページの詳細ペインにある [Root device type] フィールドの値を確認してください。

### 再起動が予定されたインスタンス

AWS では、基盤となるホストにパッチ、アップグレード、またはメンテナンスを適用するタスクを実行するために、インスタンスに再起動を予定することができます。再起動イベントには、システムの再起動とインスタンスの再起動の 2 種類があります。システムの再起動中、インスタンスと、インスタンスをサポートするハードウェアが再起動されます。インスタンスの再起動中、インスタンスは再起動されませんが、インスタンスをサポートするハードウェアは再起動されません。

予定された再起動イベントは、予定されたメンテナンスウィンドウ内で開始され、通常は数分程度で完了します。再起動が完了したら、インスタンスは使用できるようになります。予定された終了時間まで待つ必要はありません。

再起動が行われたことを確認するには、予定されているイベントを確認し、インスタンスに予定されているイベントがないことを確認します。再起動された後に、アプリケーションが期待した通りに機能しているかを確認することをお勧めします。

再起動が予定されているインスタンスに必要なアクション

いずれかのインスタンスの再起動が予定されている場合は、ユーザー側での作業は必要ありません。予定されているメンテナンスウィンドウ内で再起動が自動的に行われるのを待つことをお勧めします。

ご自分で再起動を実行して、イベントのタイミングを管理することもできます。

インスタンスの再起動

インスタンスの再起動イベントについては、予定された再起動イベントが開始する前に、いつでもご自分で再起動を実行できます (インスタンスをシャットダウンしたり、終了させないように注意してください)。詳細については、[インスタンスの再起動 \(p. 349\)](#) を参照してください。再起動後、基礎となるホストに対する保留中のメンテナンスは自動的に実行され、インスタンスが完全に起動した後で、再度インスタンスの使用を開始できます。



#### Note

再起動を実行した後、インスタンスの再起動を予定しているイベントはすぐにキャンセルされます。イベントの説明は、Amazon EC2 コンソールで更新され、この点が反映されています。

システムの再起動

インスタンスのルートデバイスが Amazon EBS ボリュームである場合は、そのボリュームの停止と起動を行うことができます。詳細については、[インスタンスの停止と起動 \(p. 346\)](#) を参照してください。インスタンスのルートデバイスがインスタンスストアボリュームである場合は、そのボリュームを終了して代替ボリュームを起動することができます。



#### Important

インスタンスストアボリュームに格納されているデータはいずれも、インスタンスが停止または終了されると失われ、復旧することはできません。これには、ルートデバイスとして Amazon EBS ボリュームを所有するインスタンスにアタッチされたインスタンスストアボリュームも該当します。

## 停止またはリタイヤされる予定になっているインスタンス

AWS が、インスタンスをホストしている基盤のハードウェアで、回復不可能な障害を検出すると、インスタンスは停止またはリタイヤするように予定されます。インスタンスは予定されたリタイヤ日になると、AWS によって停止またはリタイヤされます。インスタンスのルートデバイスが Amazon EBS ボリュームである場合、インスタンスは停止されますが、その後いつでも再度起動できます。インスタンスのルートデバイスがインスタンスストアボリュームである場合、インスタンスは終了されると、再使用はできません。

停止またはリタイヤされる予定になっているインスタンスに必要なアクション

インスタンスのルートデバイスが Amazon EBS ボリュームである場合、予定されたイベント開始時刻にインスタンスが停止またはリタイヤされるのを待つことができます。あるいは、お客様がインスタンス自体を停止および起動することもできます。そうした場合、インスタンスは新しいハードウェアに移行され、予期しないダウンタイムの低減に役立ちます。インスタンスの停止や、インスタンスが停止したときにインスタンス設定を変更する方法については、[インスタンスの停止と起動 \(p. 346\)](#) を参照してください。

インスタンスのルートデバイスがインスタンスストアボリュームである場合は、最新の AMI から代替インスタンスを起動し、予定されたリタイヤの前に必要なすべてのデータを代替インスタンスに移

行することをお勧めします。インスタンスをご自分で終了することも、インスタンスがリタイヤされたときにインスタンスが自動的に終了されるのを待つこともできます。

リタイヤが予定されているインスタンスの詳細と管理方法については、「[インスタンスのリタイヤ \(p. 350\)](#)」を参照してください。



#### Important

インスタンスストアボリュームに格納されているデータはいずれも、インスタンスが停止または終了されると失われ、復旧することはできません。これには、ルートデバイスとして Amazon EBS ボリュームを所有するインスタンスにアタッチされたインスタンスストアボリュームも該当します。インスタンスが停止または終了される前に、後で必要となるインスタンスストアボリュームからデータを必ず取得しておきます。

## メンテナンスが予定されているインスタンス

基礎となる Amazon EC2 ハードウェアでメンテナンスが必要な場合は、インスタンスにメンテナンスを予定します。2 種類のメンテナンスイベントがあります。1 つはネットワークメンテナンスで、もう 1 つは電源のメンテナンスです。ネットワークメンテナンス中は、短い期間、予定されたインスタンスのネットワーク接続が切断されます。メンテナンスが終了すると、インスタンスとの通常のネットワーク接続は回復します。電源のメンテナンス中は、短い期間、予定されたインスタンスはオフラインになり、その後再起動されます。

メンテナンスが予定されているインスタンスに必要なアクション

いずれかのインスタンスのメンテナンスが予定されている場合は、ユーザー側での作業は必要ありません。ただし、メンテナンス期間中に通常の操作を維持する場合は、最新の AMI から代替インスタンスを起動し、予定されたメンテナンスが実行される前に、必要なすべてのデータを代替インスタンスに移行します。予定されたネットワークまたは電源のメンテナンスによって、代替インスタンスが影響されることはありません。

電源のメンテナンスの場合、再起動されると、インスタンスの設定内容はすべて維持されます。

# CloudWatch によるインスタンスのモニタリング

## Abstract

Amazon CloudWatch を使用して Amazon EC2 インスタンスをモニタリングし、インスタンスから未加工データを収集して、リアルタイムに近い読み取り可能なメトリックスに加工します。

Amazon CloudWatch を使用して Amazon EC2 インスタンスをモニタリングすることで、Amazon EC2 から未加工データを収集し、リアルタイムに近い読み取り可能なメトリックスに加工することができません。これらの統計は 2 週間記録されるため、履歴情報にアクセスしてウェブアプリケーションまたはサービスの動作をよりの確に把握することができます。デフォルトでは、Amazon EC2 メトリックスデータは 5 分間隔で CloudWatch に自動的に送信されます。ただし、Amazon EC2 インスタンスの詳細モニタリングを有効にすれば、1 分間隔でデータが CloudWatch に送信されます。Amazon CloudWatch の詳細については、[Amazon CloudWatch 開発者ガイド](#) を参照してください。

次の表では、Amazon EC2 インスタンスの基本モニタリングと詳細モニタリングについて説明します。

タイプ	説明
基本	データは 5 分間隔で自動的に取得できます。無料です。

タイプ	説明
詳細	<p>データは 1 分間隔で取得できます。追加料金がかかります。このレベルのデータを取得するには、インスタンスのデータ取得を明確に有効にする必要があります。詳細モニタリングを有効にしたインスタンスでは、同様のインスタンスグループの集約データを取得することもできます。</p> <p>料金表の詳細については、<a href="#">Amazon CloudWatch product page</a> を参照してください。</p>

Amazon EC2 インスタンスのモニタリングデータは、Amazon CloudWatch API または AWS マネジメントコンソールを使用して取得できます。コンソールには、Amazon CloudWatch API の未加工データに基づいて一連のグラフが表示されます。必要に応じて、API のデータまたはコンソールのグラフのいずれかを使用できます。

#### Topics

- [Amazon EC2 インスタンスの詳細モニタリングの有効化または無効化 \(p. 396\)](#)
- [Amazon EC2 メトリックスを表示する \(p. 399\)](#)
- [メトリックスの統計情報を取得する \(p. 406\)](#)
- [メトリックスのグラフ化 \(p. 423\)](#)
- [CloudWatch アラームを作成する \(p. 427\)](#)
- [インスタンスを停止または終了するアラームを作成する \(p. 435\)](#)

## Amazon EC2 インスタンスの詳細モニタリングの有効化または無効化

### Abstract

新しい Amazon EC2 インスタンス、または実行中か停止中のインスタンスで、詳細モニタリングを有効または無効にします。

このセクションでは、新しいインスタンス ( 起動時 ) または実行中または停止中のインスタンスの詳細モニタリングを有効または無効にする方法について説明します。詳細モニタリングを有効にすると、Amazon EC2 コンソールに、インスタンスの 1 分間ごとのモニタリンググラフが表示されます。コンソールまたはコマンドラインインターフェイス ( CLI ) を使用して、詳細モニタリングを有効または無効にすることができます。

## AWS マネジメントコンソール

既存のインスタンスの詳細モニタリングを有効にするには

EC2 インスタンスの詳細モニタリングを有効できます。これにより 1 分間隔でインスタンスのデータが提供されます ( 1 分間隔のモニタリングには追加料金がかかります )。これにより、AWS Management Console コンソールグラフ、または API 経由で、インスタンスの詳細データを取得することができます。このレベルのデータを取得するには、インスタンスのデータ取得を明確に有効にする必要があります。詳細モニタリングを有効にしたインスタンスでは、同様のインスタンスグループの集約データを取得することもできます。詳細モニタリングを有効にするには、インスタンスは実行中または停止中である必要があります。

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. [Navigation] ペインの [Instances] をクリックします。

3. インスタンスのリストで、実行中または停止中のインスタンスを選択し、[Actions] をクリックし、[Enable Detailed Monitoring] をクリックします。
4. [Enable Detailed Monitoring] ダイアログボックスで、[Yes, Enable] をクリックします。
5. 確認の [Enable Detailed Monitoring] ダイアログボックスで、[Close] をクリックします。

これにより、AWS Management Console グラフ、または API 経由で、(1 分間ごとに収集される) インスタンスの詳細データを取得することができます。

#### EC2 インスタンスの起動時に詳細モニタリングを有効にするには

- AWS Management Console を使用してインスタンスを起動する場合は、起動ウィザードの [Configure Instance Details] ページの [Monitoring] チェックボックスをオンにします。

インスタンスを起動したら、コンソールでインスタンスを選択し、下部のペインにあるインスタンスの [Monitoring] タブにそのモニタリンググラフを表示できます。

#### EC2 インスタンスの詳細モニタリングを無効にするには

1 分間隔でインスタンスをモニタリングする必要がなくなった場合は、詳細モニタリングを無効にして、基本モニタリングを使用します。基本モニタリングでは、5 分間隔のデータを無料で取得できます。

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. [Navigation] ペインの [Instances] をクリックします。
3. インスタンスのリストで、実行中または停止中のインスタンスを選択し、[Actions] をクリックして、[Disable Detailed Monitoring] をクリックします。
4. [Disable Detailed Monitoring] ダイアログボックスで、[Yes, Disable] をクリックします。
5. 確認の [Disable Detailed Monitoring] ダイアログボックスで、[Close] をクリックします。

インスタンスの起動については、「[インスタンスの起動 \(p. 324\)](#)」を参照してください。

## コマンドラインインターフェイス

#### 既存のインスタンスで詳細モニタリングを有効にするには

- 1 つまたは複数のインスタンス ID を指定して `monitor-instances` コマンドを使用します。`monitor-instances` コマンドの使用に関する詳細については、『*AWS Command Line Interface Reference*』の「[monitor-instances](#)」を参照してください。

```
PROMPT> aws ec2 monitor-instances --instance-ids i-570e5a28
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-570e5a28",
      "Monitoring": {
        "State": "pending"
      }
    }
  ]
}
```

これにより、AWS Management Console グラフ、または API 経由で、(1 分間ごとに収集される) インスタンスの詳細データを取得することができます。

インスタンスの起動時に詳細モニタリングを有効にするには

- `run-instances` コマンドを、`--monitoring` フラグを付けて使用します。`run-instances` コマンドの使用に関する詳細については、『*AWS Command Line Interface Reference*』の「[run-instances](#)」を参照してください。

```
PROMPT> aws ec2 run-instances --image-id ami-09092360 --key-name MyKeyPair  
--monitoring Enabled=value
```

Amazon EC2 が次の例のような出力を返します。モニタリングのステータスが `pending` として表示されます。

```
{  
  "OwnerId": "111122223333",  
  "ReservationId": "r-25fad905",  
  "Groups": [  
    {  
      "GroupName": "default",  
      "GroupId": "sg-eafelb82"  
    }  
  ],  
  "Instances": [  
    {  
      "Monitoring": {  
        "State": "pending"  
      },  
      "PublicDnsName": null,  
      "Platform": "windows",  
      "State": {  
        "Code": 0,  
        "Name": "pending"  
      },  
      "EbsOptimized": false,  
      "LaunchTime": "2014-02-24T18:02:49.000Z",  
      "ProductCodes": [],  
      "StateTransitionReason": null,  
      "InstanceId": "i-31283b11",  
      "ImageId": "ami-09092360",  
      "PrivateDnsName": null,  
      "KeyName": MyKeyPair,  
      "SecurityGroups": [  
        {  
          "GroupName": "default",  
          "GroupId": "sg-eafelb82"  
        }  
      ],  
      "ClientToken": null,  
      "InstanceType": "m1.small",  
      "NetworkInterfaces": [],  
      "Placement": {  
        "Tenancy": "default",  
        "GroupName": null,  
        "AvailabilityZone": "us-east-1b"  
      },  
      "Hypervisor": "xen",  
      "BlockDeviceMappings": [],  
      "Architecture": "x86_64",  
    }  
  ]  
}
```

```
    "StateReason": {
      "Message": "pending",
      "Code": "pending"
    },
    "VirtualizationType": "hvm",
    "RootDeviceType": "instance-store",
    "AmiLaunchIndex": 0
  }
]
```

インスタンスを実行したら、AWS Management Console グラフ、または API 経由で、( 1 分間ごとに収集される ) インスタンスの詳細データを取得することができます。

インスタンスの詳細モニタリングを無効にするには

- 1 つまたは複数のインスタンス ID を指定して `unmonitor-instances` コマンドを使用します。 `unmonitor-instances` コマンドの使用に関する詳細については、『*AWS Command Line Interface Reference*』の「[unmonitor-instances](#)」を参照してください。

```
PROMPT> aws ec2 unmonitor-instances --instance-ids i-570e5a28
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-570e5a28",
      "Monitoring": {
        "State": "disabling"
      }
    }
  ]
}
```

## Amazon EC2 メトリックスを表示する

### Abstract

アクティビティの過去 5 分間 ( デフォルト ) について、Amazon EC2 が CloudWatch に送信するメトリックスを表示します。

お客様が現在 AWS で使用されているサービスのみがメトリックスを Amazon CloudWatch に送信します。Amazon EC2 から CloudWatch に送信されるメトリックスを表示するには、Amazon CloudWatch コンソール、`mon-list-metrics` コマンド、または `ListMetrics` API を使用できます。詳細モニタリングを有効にした場合、各データポイントはインスタンスの過去 1 分間のアクティビティを対象とします。それ以外の場合、各データポイントはインスタンスの過去 5 分間のアクティビティが対象です。

メトリックス	説明
CPUUtilization	割り当てられた EC2 コンピュートユニットのうち、現在インスタンス上で使用されているものの比率。このメトリックスは、選択されたインスタンス上でアプリケーションを実行するのに必要な処理能力を表します。  単位: <i>Percent</i>

メトリックス	説明
DiskReadOps	<p>このインスタンスで利用できるすべてのエフェメラルディスクでの、完了した読み取り操作の数 ( インスタンスで Amazon EBS を使用している場合は、「<a href="#">Amazon EBS のメトリックス (p. 581)</a>」を参照してください)。</p> <p>このメトリックスは、一定の時間当たりの、アプリケーションがディスクから読み取る回数を表しています。これを利用すると、アプリケーションがハードディスクからデータを読み取る速度がわかります。</p> <p>単位: <i>Count</i></p>
DiskWriteOps	<p>このインスタンスで利用できるすべてのエフェメラルディスクでの、完了した書き込み操作の数 ( インスタンスで Amazon EBS を使用している場合は、「<a href="#">Amazon EBS のメトリックス (p. 581)</a>」を参照してください)。</p> <p>このメトリックスは、一定の時間当たりの、アプリケーションがハードディスクに書き込む回数を表しています。これを利用すると、アプリケーションがハードディスクにデータを保存する速度がわかります。</p> <p>単位: <i>Count</i></p>
DiskReadBytes	<p>このインスタンスで利用できるすべてのエフェメラルディスクでの、読み取りバイト数 ( インスタンスで Amazon EBS を使用している場合は、「<a href="#">Amazon EBS のメトリックス (p. 581)</a>」を参照してください)。</p> <p>このメトリックスを使用すると、このインスタンスのハードディスクからアプリケーションが読み取るデータの量がわかります。これを利用すると、アプリケーションの速度がわかります。</p> <p>単位: <i>Bytes</i></p>
DiskWriteBytes	<p>このインスタンスで利用できるすべてのエフェメラルディスクでの、書き込みバイト数 ( インスタンスで Amazon EBS を使用している場合は、「<a href="#">Amazon EBS のメトリックス (p. 581)</a>」を参照してください)。</p> <p>このメトリックスを使用すると、このインスタンスのハードディスクにアプリケーションが書き込むデータの量がわかります。これを利用すると、アプリケーションの速度がわかります。</p> <p>単位: <i>Bytes</i></p>
NetworkIn	<p>すべてのネットワークインターフェースでの、このインスタンスによって受信されたバイトの数。このメトリックスは、1つのインスタンス上での1つのアプリケーションへのネットワークトラフィックの量を表しています。</p> <p>単位: <i>Bytes</i></p>

メトリックス	説明
NetworkOut	<p>すべてのネットワークインターフェースでの、このインスタンスから送信されたバイトの数。このメトリックスは、1つのインスタンス上での1つのアプリケーションからのネットワークトラフィックの量を表しています。</p> <p>単位: <i>Bytes</i></p>
StatusCheckFailed	<p>StatusCheckFailed_Instance と StatusCheckFailed_System の組み合わせで、どちらかのステータスチェックが失敗したら報告します。このメトリックスの値は0か1です。0はステータスチェックが成功したことを示します。1はステータスチェックの失敗を示します。</p> <p> <b>Note</b></p> <p>ステータスチェックメトリックスは、1分間隔で利用できます。インスタンスを新しく起動した場合は、インスタンスの初期化状態が完了しないと、ステータスチェックのメトリックスを利用できません。ステータスが実行中に変わったら、数分以内にステータスチェックのメトリックスが利用できるようになります。</p> <p>単位: <i>Count</i></p>
StatusCheckFailed_Instance	<p>最近 1 分間にインスタンスが EC2 インスタンスステータスチェックに成功したかどうかを報告します。このメトリックスの値は0か1です。0はステータスチェックが成功したことを示します。1はステータスチェックの失敗を示します。</p> <p> <b>Note</b></p> <p>ステータスチェックメトリックスは、1分間隔で利用できます。インスタンスを新しく起動した場合は、インスタンスの初期化状態が完了しないと、ステータスチェックのメトリックスを利用できません。ステータスが実行中に変わったら、数分以内にステータスチェックのメトリックスが利用できるようになります。</p> <p>単位: <i>Count</i></p>

メトリックス	説明
StatusCheckFailed_System	<p>最近 1 分間にインスタンスが EC2 システムステータスチェックに成功したかどうかを報告します。このメトリックスの値は 0 か 1 です。0 はステータスチェックが成功したことを示します。1 はステータスチェックの失敗を示します。</p> <p> <b>Note</b></p> <p>ステータスチェックメトリックスは、1 分間隔で利用できます。インスタンスを新しく起動した場合は、インスタンスの初期化状態が完了しないと、ステータスチェックのメトリックスを利用できません。ステータスが実行中に変わったら、数分以内にステータスチェックのメトリックスが利用できるようになります。</p> <p>単位: <i>Count</i></p>

次の表のディメンションを使用すると、インスタンスに対して返されるメトリックスを絞り込むことができます。

ディメンション	説明
AutoScalingGroupName	このディメンションを指定すると、リクエストしたデータがフィルタリングされて、指定したキャパシティグループ内のインスタンスのものだけになります。 <i>AutoScalingGroup</i> は、Auto Scaling サービスを使用する場合に定義するインスタンスのコレクションです。このディメンションを EC2 のメトリックスに対して使用できるのは、インスタンスが <i>AutoScalingGroup</i> 内にあるときに限られます。詳細モニタリングまたは基本モニタリングが有効になっているインスタンスに対して使用できます。
ImageId	このディメンションを指定すると、リクエストしたデータがフィルタリングされて、この EC2 Amazon マシンイメージ (AMI) を実行しているインスタンスのものだけになります。詳細モニタリングが有効になっているインスタンスに対して使用できます。
InstanceId	このディメンションを指定すると、リクエストしたデータがフィルタリングされて、指定のインスタンスのものだけになります。これを利用すると、どのインスタンスからのデータをモニタリングするかを指定できます。詳細モニタリングが有効になっているインスタンスに対して使用できます。
InstanceType	このディメンションを指定すると、リクエストしたデータがフィルタリングされて、指定のインスタンスタイプで実行されているインスタンスのものだけになります。これを利用すると、実行されているインスタンスのタイプでデータを分類することができます。例えば、 <i>m1.small</i> インスタンスと <i>m1.large</i> インスタンスのデータを比較して、アプリケーションに対するビジネス価値はどちらが上かを判断します。詳細モニタリングが有効になっているインスタンスに対して使用できます。

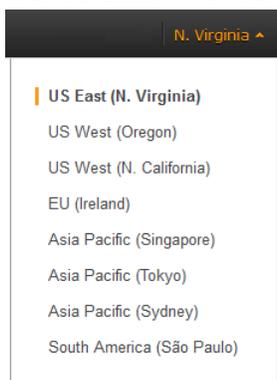
*GetMetricStatistics* アクションの使用の詳細については、「*Amazon CloudWatch API リファレンス*」の「[GetMetricStatistics](#)」を参照してください。

## AWS マネジメントコンソール

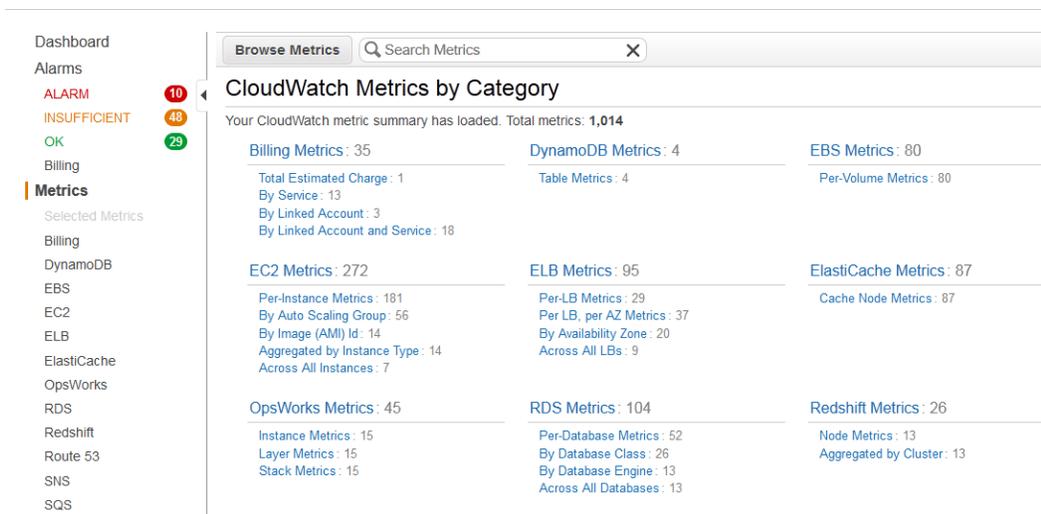
カテゴリごとに取得できるメトリックスを表示するには

カテゴリごとにメトリックスを表示できます。メトリックスはまずネームスペースごとにグループ化され、次に各ネームスペース内の種々のディメンションの組み合わせごとにグループ化されます。例えば、すべての EC2 メトリックスを表示したり、インスタンス ID、インスタンスタイプ、イメージ (AMI) ID、または Auto Scaling グループごとにグループ化された EC2 メトリックスを表示したりできます。

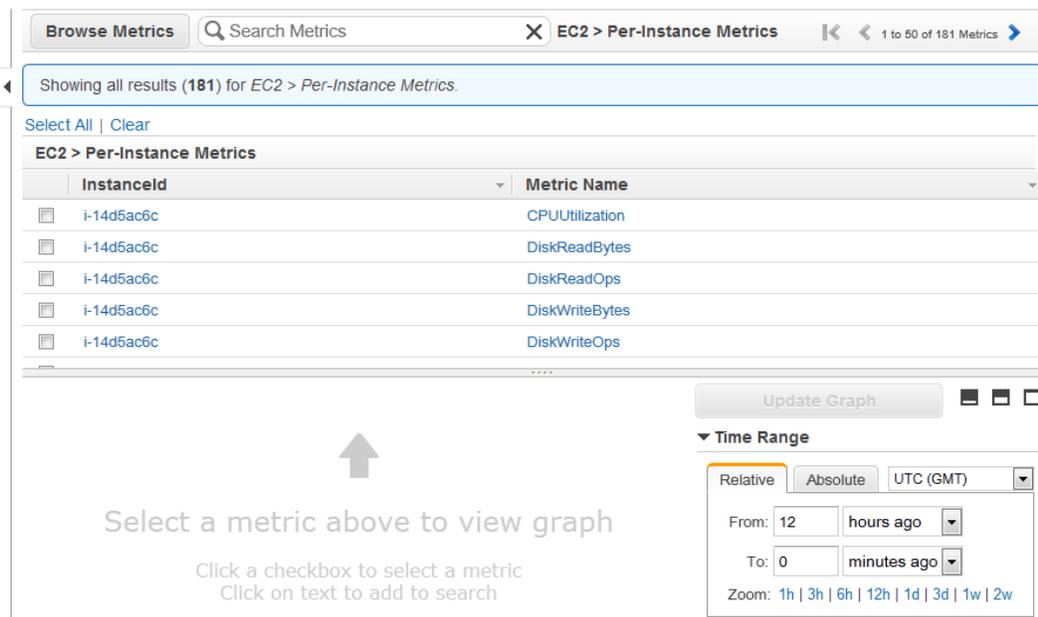
1. Amazon CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションバインの [Metrics] をクリックします。



4. [CloudWatch Metrics by Category] ペインで、[EC2 Metrics] の下にある [Per-Instance Metrics] を選択します。次に上側のペインで下方にスクロールし、メトリックスの完全なリストを確認します。



## コマンドラインインターフェイス

複数の Amazon EC2 インスタンスで使用可能なメトリックスを一覧表示するには

- `list-metrics` コマンドを入力し、結果を Amazon EC2 に限定するために `AWS/EC2` 名前空間を指定します。`list-metrics` コマンドの詳細については、『*AWS Command Line Interface Reference*』の「[list-metrics](#)」を参照してください。

```
Prompt>aws cloudwatch list-metrics --namespace AWS/EC2
```

CloudWatch は次の情報 (一部を表示) を返します。

```
{
  "Namespace": "AWS/EC2",
  "Dimensions": [
    {
      "Name": "InstanceType",
      "Value": "t1.micro"
    }
  ],
  "MetricName": "CPUUtilization"
},
{
  "Namespace": "AWS/EC2",
  "Dimensions": [
    {
      "Name": "InstanceId",
      "Value": "i-570e5a28"
    }
  ],
  "MetricName": "DiskWriteOps"
},
```

```
{
  "Namespace": "AWS/EC2",
  "Dimensions": [
    {
      "Name": "InstanceType",
      "Value": "t1.micro"
    }
  ],
  "MetricName": "NetworkOut"
},
{
  "Namespace": "AWS/EC2",
  "Dimensions": [
    {
      "Name": "ImageId",
      "Value": "ami-6cb90605"
    }
  ],
  "MetricName": "CPUUtilization"
},
{
  "Namespace": "AWS/EC2",
  "Dimensions": [
    {
      "Name": "ImageId",
      "Value": "ami-6cb90605"
    }
  ],
  "MetricName": "NetworkIn"
},
{
  "Namespace": "AWS/EC2",
  "Dimensions": [
    {
      "Name": "InstanceType",
      "Value": "t1.micro"
    }
  ],
  "MetricName": "DiskReadBytes"
},
{
  "Namespace": "AWS/EC2",
  "Dimensions": [
    {
      "Name": "InstanceId",
      "Value": "i-570e5a28"
    }
  ],
  "MetricName": "StatusCheckFailed_System"
},
{
  "Namespace": "AWS/EC2",
  "Dimensions": [
    {
      "Name": "InstanceId",
      "Value": "i-570e5a28"
    }
  ],
  "MetricName": "StatusCheckFailed_System"
}
```

```
    "MetricName": "NetworkOut"
  },
  {
    "Namespace": "AWS/EC2",
    "Dimensions": [
      {
        "Name": "InstanceId",
        "Value": "i-0c986c72"
      }
    ],
    "MetricName": "DiskWriteBytes"
  }
]
```

## メトリックスの統計情報を取得する

### Abstract

Amazon CloudWatch コンソールまたはコマンドラインコマンドを使用して、メトリックスの統計情報を取得する方法に関するシナリオをひとつお説明します。

この一連のシナリオでは、AWS マネジメントコンソール、`get-metric-statistics` コマンド、または `GetMetricStatistics` API を使用して、さまざまな統計を取得する方法について示します。



#### Note

開始時刻と終了時刻は、過去 14 日以内である必要があります。

#### Topics

- [特定の EC2 インスタンスの統計情報を取得する \(p. 406\)](#)
- [複数のインスタンスにわたって統計情報を集計する \(p. 410\)](#)
- [Auto Scaling グループごとに集計された統計情報を取得する \(p. 415\)](#)
- [イメージ \(AMI\) ID ごとに集計された統計情報を取得する \(p. 418\)](#)

## 特定の EC2 インスタンスの統計情報を取得する

### Abstract

説明するシナリオは、特定の EC2 インスタンスの統計情報を取得する方法に関するものです。

次のシナリオでは、AWS Management Console または `get-metric-statistics` コマンドを使用して、特定の EC2 インスタンスの最大 CPU 使用率を特定する方法を順を追って説明します。



#### Note

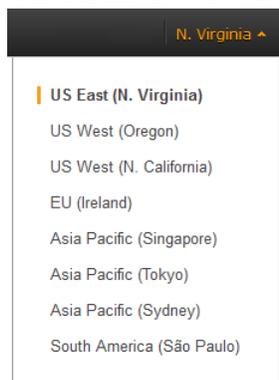
開始時刻と終了時刻は、過去 14 日以内である必要があります。

この例では、EC2 インスタンス ID を持っていることを前提とします。アクティブな EC2 インスタンス ID は、AWS Management Console または `describe-instances` コマンドを使用して取得できます。

## AWS マネジメントコンソール

特定のインスタンスの平均 CPU 使用率を表示するには

1. Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。

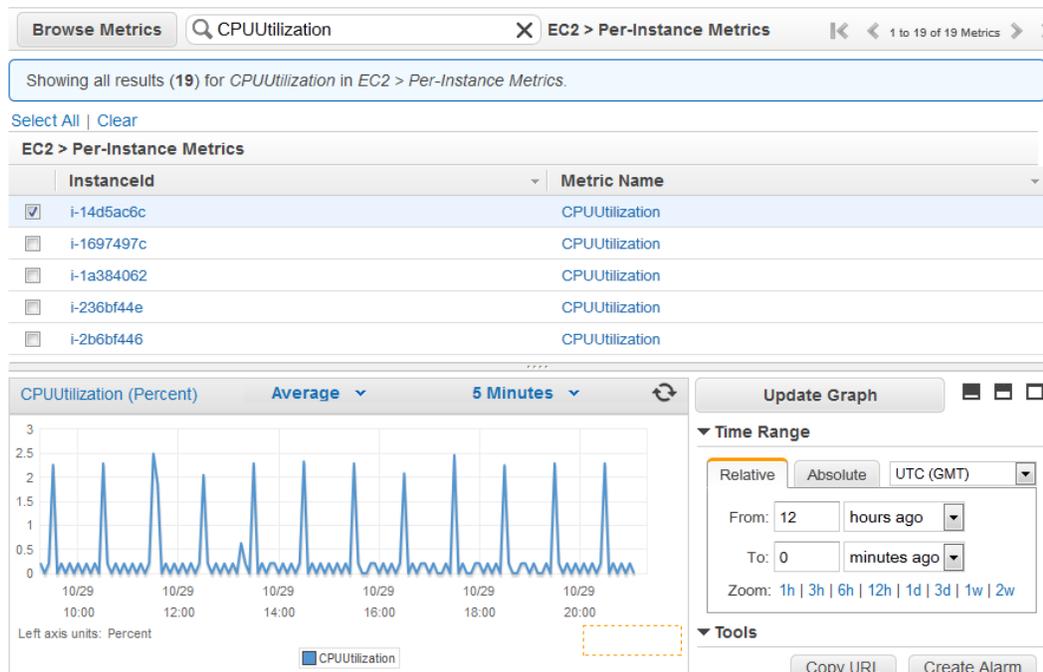


3. ナビゲーションペインの [Metrics] をクリックします。
4. [CloudWatch Metrics by Category] ペインで [EC2: Metrics] を選択します。

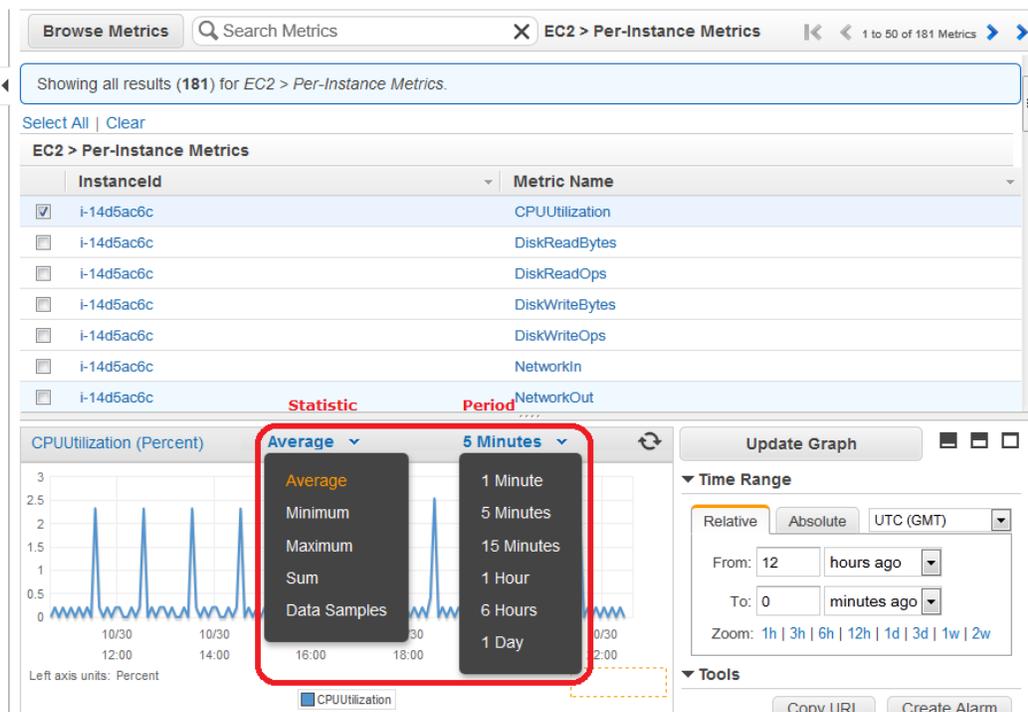
それぞれのインスタンスで利用可能なメトリックスが上側のペインに表示されます。

5. 特定のインスタンス ID の [CPUUtilization] が含まれる行を選択します。

単一のインスタンスの平均 CPUUtilization を表すグラフが詳細ペインに表示されます。



6. メトリックスの [Statistic] の値 ( 例: 平均 ) を変更するには、ポップアップリストから異なる値を選択します。



7. [Period]の値（例:5分）を変更して、さらに粒度の細かい詳細を表示するには、ポップアップリストから異なる値を選択します。

## コマンドラインインターフェイス

EC2 インスタンスごとに CPU 使用率を取得するには

- `get-metric-statistics` コマンドを、以下のパラメータを指定して入力します。  
`get-metric-statistics` コマンドの詳細については、『*AWS Command Line Interface Reference*』の「[get-metric-statistics](#)」を参照してください。

```
Prompt>aws cloudwatch get-metric-statistics --metric-name CPUUtilization -  
--start-time 2014-02-18T23:18:00 --end-time 2014-02-19T23:18:00 --period 3600  
--namespace AWS/EC2 --statistics Maximum --dimensions Name=Inst  
stanceId,Value=<your-instance-id>
```

AWS CLI は以下の値を返します。

```
{  
  "Datapoints": [  
    {  
      "Timestamp": "2014-02-19T00:18:00Z",  
      "Maximum": 0.33000000000000002,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-19T03:18:00Z",  
      "Maximum": 99.670000000000002,  
      "Unit": "Percent"  
    }  
  ]  
}
```

```
},
{
  "Timestamp": "2014-02-19T07:18:00Z",
  "Maximum": 0.34000000000000002,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T12:18:00Z",
  "Maximum": 0.34000000000000002,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T02:18:00Z",
  "Maximum": 0.34000000000000002,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T01:18:00Z",
  "Maximum": 0.34000000000000002,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T17:18:00Z",
  "Maximum": 3.3900000000000001,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T13:18:00Z",
  "Maximum": 0.33000000000000002,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-18T23:18:00Z",
  "Maximum": 0.67000000000000004,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T06:18:00Z",
  "Maximum": 0.34000000000000002,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T11:18:00Z",
  "Maximum": 0.34000000000000002,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T10:18:00Z",
  "Maximum": 0.34000000000000002,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T19:18:00Z",
  "Maximum": 8.0,
  "Unit": "Percent"
},
{
  "Timestamp": "2014-02-19T15:18:00Z",
```

```
    "Maximum": 0.34000000000000002,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T14:18:00Z",  
    "Maximum": 0.34000000000000002,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T16:18:00Z",  
    "Maximum": 0.34000000000000002,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T09:18:00Z",  
    "Maximum": 0.34000000000000002,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T04:18:00Z",  
    "Maximum": 2.0,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T08:18:00Z",  
    "Maximum": 0.68000000000000005,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T05:18:00Z",  
    "Maximum": 0.33000000000000002,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-19T18:18:00Z",  
    "Maximum": 6.6699999999999999,  
    "Unit": "Percent"  
  }  
],  
"Label": "CPUUtilization"  
}
```

時間間隔として 2 日が要求された場合、返される統計情報は 6 分値となります。各値は、単一の EC2 インスタンスにおける最大 CPU 使用率 (パーセンテージ) を示します。

## 複数のインスタンスにわたって統計情報を集計する

### Abstract

次に説明するシナリオは、インスタンスとリージョンにわたって統計情報を取得し、集計する方法についてひととおり説明しています。

集計統計は、詳細モニタリングが有効になっているインスタンスで使用できます。基本モニタリングを使用しているインスタンスは集計対象になりません。さらに、Amazon CloudWatch は複数のリージョンにわたってデータを集計しません。したがって、メトリックスはリージョン間で完全に独立しています。複数のインスタンスにわたって集計された統計情報を取得するには、1 分間隔でデータが提供される詳細モニタリングを事前に有効にしておく必要があります (追加料金がかかります)。このシナリオ

は、AWS マネジメントコンソール、GetMetricStatistics API、または `get-metric-statistics` コマンドで詳細モニタリングを使用して EC2 インスタンスの平均 CPU 使用率を取得する方法を紹介します。ディメンションが指定されていないため、CloudWatch は `AWS/EC2` ネームスペース内のすべてのディメンションについて統計情報を返します。他のメトリックスの統計情報を取得するには、「[Amazon CloudWatch の名前空間、ディメンション、メトリックスのリファレンス](#)」を参照してください。



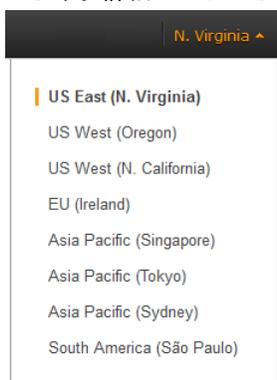
#### Important

AWS ネームスペースですべてのディメンションを取得するこの技術は、Amazon CloudWatch に発行するカスタムネームスペースでは機能しません。カスタムネームスペースの場合、特定のデータポイントに関連付けられた完全なディメンションセットを指定して、そのデータポイントを含む統計情報を取得する必要があります。

## AWS マネジメントコンソール

Amazon EC2 インスタンスの平均 CPU 使用率を表示するには

1. Amazon CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。

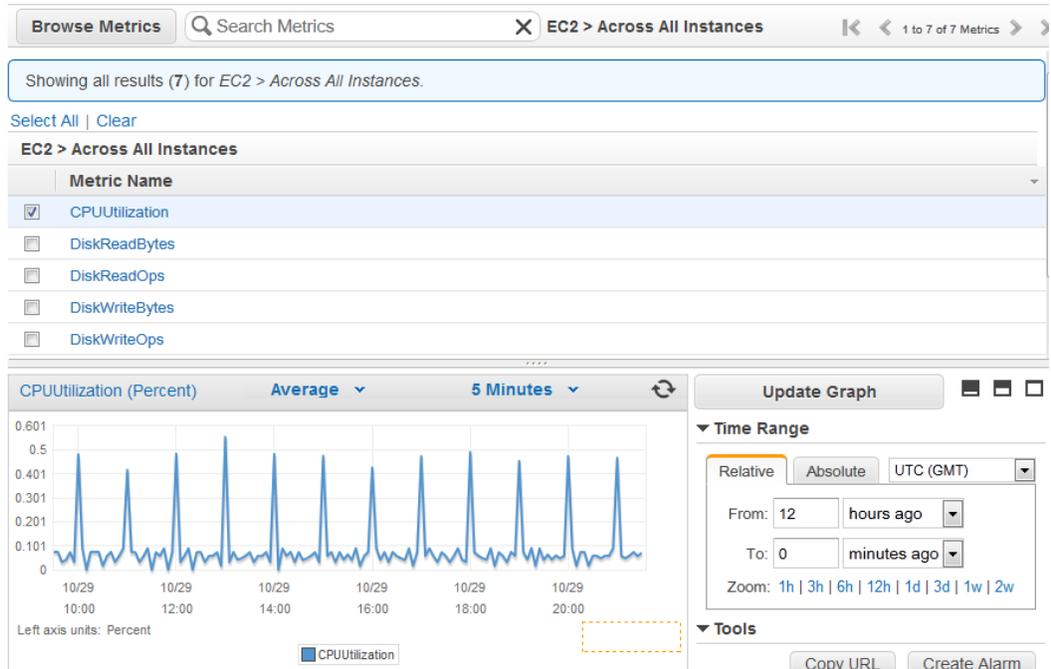


3. ナビゲーションペインの [Metrics] をクリックします。
4. [CloudWatch Metrics by Category] ペインで、[EC2 Metrics] の下にある [Across All Instances] を選択します。

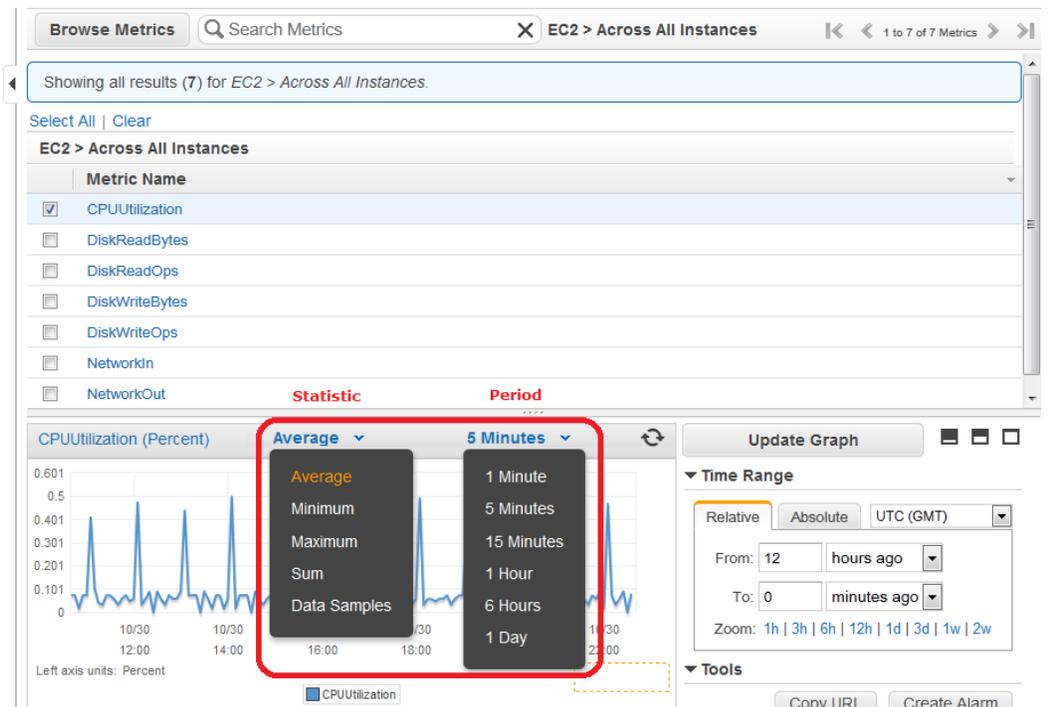
すべてのインスタンスで使用できるメトリックスが上側のペインに表示されます。

5. 上側のペインで、[CPUUtilization] が含まれている行を選択します。

EC2 インスタンスの `CPUUtilization` を示すグラフが詳細ペインに表示されます。



6. メトリックスの [Statistic] の値 (例: 平均) を変更するには、ポップアップリストから異なる値を選択します。



7. [Period] の値 (例: 5分) を変更して、さらに粒度の細かい詳細を表示するには、ポップアップリストから異なる値を選択します。

## コマンドラインインターフェイス

Amazon EC2 インスタンスで平均 CPU 使用率を取得するには

- `get-metric-statistics` コマンドを、以下のパラメータを指定して入力します。  
`get-metric-statistics` コマンドの詳細については、『*AWS Command Line Interface Reference*』の「[get-metric-statistics](#)」を参照してください。

```
Prompt>aws cloudwatch get-metric-statistics --metric-name CPUUtilization -  
-start-time 2014-02-11T23:18:00 --end-time 2014-02-12T23:18:00 --period 3600  
--namespace AWS/EC2 --statistics "Average" "SampleCount"
```

AWS CLI は以下の値を返します。

```
{  
  "Datapoints": [  
    {  
      "SampleCount": 238.0,  
      "Timestamp": "2014-02-12T07:18:00Z",  
      "Average": 0.038235294117647062,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 240.0,  
      "Timestamp": "2014-02-12T09:18:00Z",  
      "Average": 0.16670833333333332,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 238.0,  
      "Timestamp": "2014-02-11T23:18:00Z",  
      "Average": 0.041596638655462197,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 240.0,  
      "Timestamp": "2014-02-12T16:18:00Z",  
      "Average": 0.039458333333333345,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 239.0,  
      "Timestamp": "2014-02-12T21:18:00Z",  
      "Average": 0.041255230125523033,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 240.0,  
      "Timestamp": "2014-02-12T01:18:00Z",  
      "Average": 0.044583333333333336,  
      "Unit": "Percent"  
    },  
    {  
      "SampleCount": 239.0,  
      "Timestamp": "2014-02-12T18:18:00Z",  
      "Average": 0.043054393305439344,  
      "Unit": "Percent"  
    }  
  ]  
}
```

```
},
{
  "SampleCount": 240.0,
  "Timestamp": "2014-02-12T13:18:00Z",
  "Average": 0.039458333333333345,
  "Unit": "Percent"
},
{
  "SampleCount": 238.0,
  "Timestamp": "2014-02-12T15:18:00Z",
  "Average": 0.041260504201680689,
  "Unit": "Percent"
},
{
  "SampleCount": 240.0,
  "Timestamp": "2014-02-12T19:18:00Z",
  "Average": 0.037666666666666668,
  "Unit": "Percent"
},
{
  "SampleCount": 240.0,
  "Timestamp": "2014-02-12T06:18:00Z",
  "Average": 0.037541666666666675,
  "Unit": "Percent"
},
{
  "SampleCount": 240.0,
  "Timestamp": "2014-02-12T20:18:00Z",
  "Average": 0.039333333333333338,
  "Unit": "Percent"
},
{
  "SampleCount": 240.0,
  "Timestamp": "2014-02-12T08:18:00Z",
  "Average": 0.039250000000000014,
  "Unit": "Percent"
},
{
  "SampleCount": 239.0,
  "Timestamp": "2014-02-12T03:18:00Z",
  "Average": 0.037740585774058588,
  "Unit": "Percent"
},
{
  "SampleCount": 240.0,
  "Timestamp": "2014-02-12T11:18:00Z",
  "Average": 0.039500000000000007,
  "Unit": "Percent"
},
{
  "SampleCount": 238.0,
  "Timestamp": "2014-02-12T02:18:00Z",
  "Average": 0.039789915966386563,
  "Unit": "Percent"
},
{
  "SampleCount": 238.0,
  "Timestamp": "2014-02-12T22:18:00Z",
```

```
    "Average": 0.039705882352941181,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T14:18:00Z",  
    "Average": 0.082458333333333328,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T05:18:00Z",  
    "Average": 0.042875000000000001,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T17:18:00Z",  
    "Average": 0.039458333333333345,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T10:18:00Z",  
    "Average": 0.083416666666666667,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 236.0,  
    "Timestamp": "2014-02-12T00:18:00Z",  
    "Average": 0.036567796610169498,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T12:18:00Z",  
    "Average": 0.039541666666666676,  
    "Unit": "Percent"  
  },  
  {  
    "SampleCount": 240.0,  
    "Timestamp": "2014-02-12T04:18:00Z",  
    "Average": 0.043000000000000003,  
    "Unit": "Percent"  
  }  
],  
"Label": "CPUUtilization"  
}
```

## Auto Scaling グループごとに集計された統計情報を取得する

### Abstract

次に説明するシナリオは、Auto Scaling グループに沿って集計された統計情報を取得する方法に関するものです。

このシナリオでは、AWS マネジメントコンソール、`get-metric-statistics` コマンド、または `GetMetricStatistics` API を `DiskWriteBytes` メトリックス指定で実行して、1 つの Auto Scaling グループのディスクに書き込まれる合計バイト数を取得する方法について示します。合計は、指定された `AutoScalingGroupName` の EC2 インスタンス全体で、24 時間 1 分おきで計算されます。



#### Note

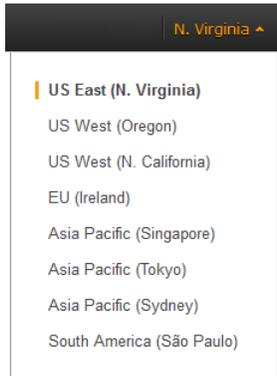
開始時刻と終了時刻は、過去 14 日以内である必要があります。

この例では、EC2 アプリケーションは動作中であり、`test-group-1` という名前の Auto Scaling グループを所有していることを前提にしています。

## AWS マネジメントコンソール

オートスケールされている EC2 アプリケーションの合計 `DiskWriteBytes` を表示するには

1. Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。

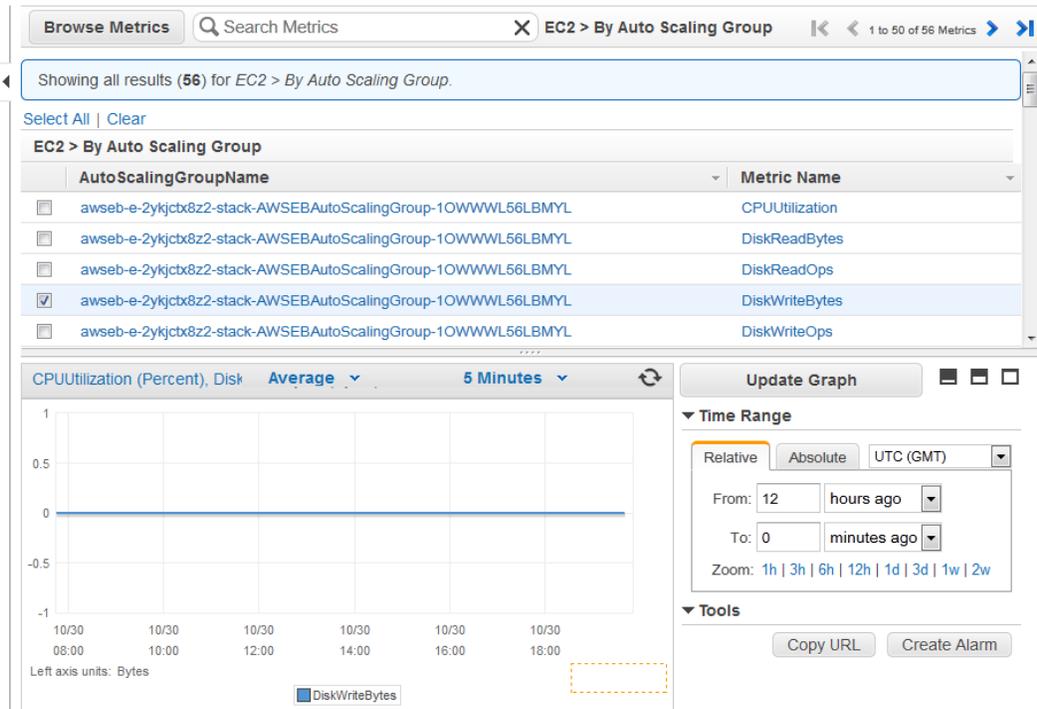


3. ナビゲーションペインの [Metrics] をクリックします。
4. [CloudWatch Metrics by Category] ペインで、[EC2 Metrics] の下にある [By Auto Scaling Group] を選択します。

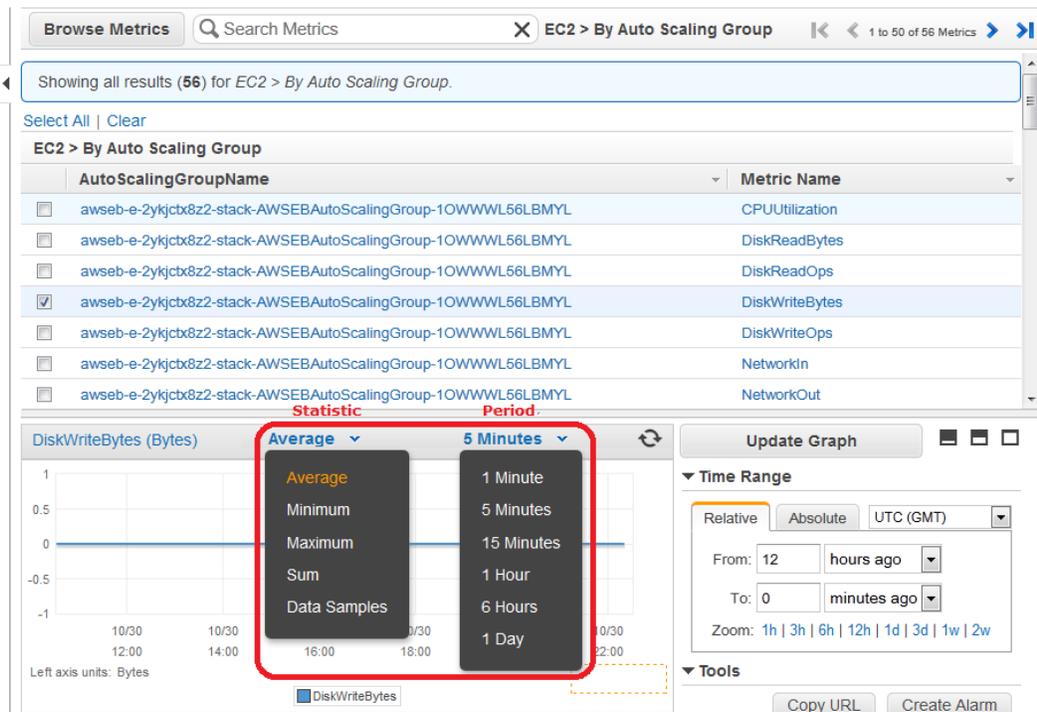
Auto Scaling グループで使用できるメトリックスは上側のペインに表示されます。

5. [DiskWriteBytes] が含まれている行を選択します。

すべての EC2 インスタンスの `DiskWriteBytes` を示すグラフが詳細ペインに表示されます。



6. メトリックスの [Statistic] の値 (例: 平均) を変更するには、ポップアップリストから異なる値を選択します。



7. [Period] の値 (例: 5分) を変更して、さらに粒度の細かい詳細を表示するには、ポップアップリストから異なる値を選択します。

## コマンドラインインターフェイス

オートスケールされている EC2 アプリケーションの合計 DiskWriteBytes を取得するには

- `get-metric-statistics` コマンドを、以下のパラメータを指定して入力します。  
`get-metric-statistics` コマンドの詳細については、『*AWS Command Line Interface Reference*』の「`get-metric-statistics`」を参照してください。

```
Prompt>aws cloudwatch get-metric-statistics --metric-name DiskWriteBytes -  
--start-time 2014-02-16T23:18:00 --end-time 2014-02-18T23:18:00 --period 360  
--namespace AWS/EC2 --statistics "Sum" "SampleCount" --dimensions  
Name=AutoScalingGroupName,Value=test-group-1
```

AWS CLI は以下の値を返します。

```
{  
  "Datapoints": [  
    {  
      "SampleCount": 18.0,  
      "Timestamp": "2014-02-19T21:36:00Z",  
      "Sum": 0.0,  
      "Unit": "Bytes"  
    },  
    {  
      "SampleCount": 5.0,  
      "Timestamp": "2014-02-19T21:42:00Z",  
      "Sum": 0.0,  
      "Unit": "Bytes"  
    }  
  ],  
  "Label": "DiskWriteBytes"  
}
```

## イメージ (AMI) ID ごとに集計された統計情報を取得する

### Abstract

次に説明するシナリオは、イメージ ID に沿って集計された統計情報を取得する方法に関するものです。

このシナリオでは、AWS マネジメントコンソール、`get-metric-statistics` コマンド、または `GetMetricStatistics` API を使用して、特定のイメージ ID に対応するすべてのインスタンスの平均 CPU 使用率を明らかにします。60 秒間隔で得た値の 1 日ごとの平均となります。



### Note

開始時刻と終了時刻は、過去 14 日以内である必要があります。

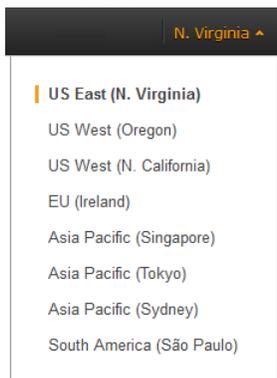
このシナリオでは、EC2 インスタンスは `ami-c5e40dac` のイメージ ID を実行しています。

## AWS マネジメントコンソール

イメージ ID の平均 CPU 使用率を表示するには

1. Amazon CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。

- 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。

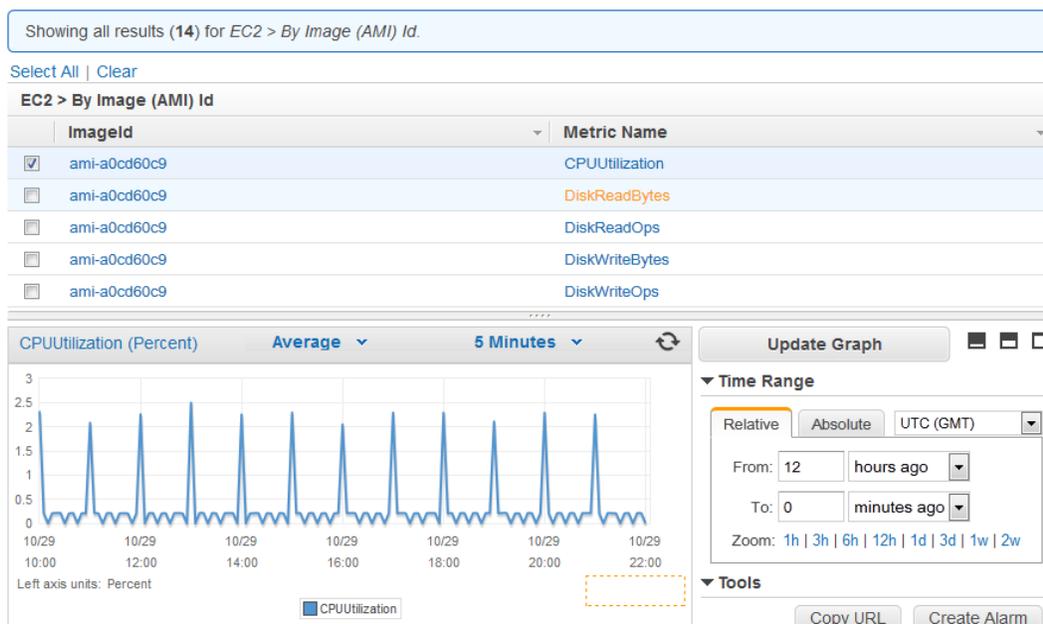


- ナビゲーションペインの [Metrics] をクリックします。
- [CloudWatch Metrics by Category] ペインで、[EC2 Metrics] の下にある [By Image (AMI) Id] を選択します。

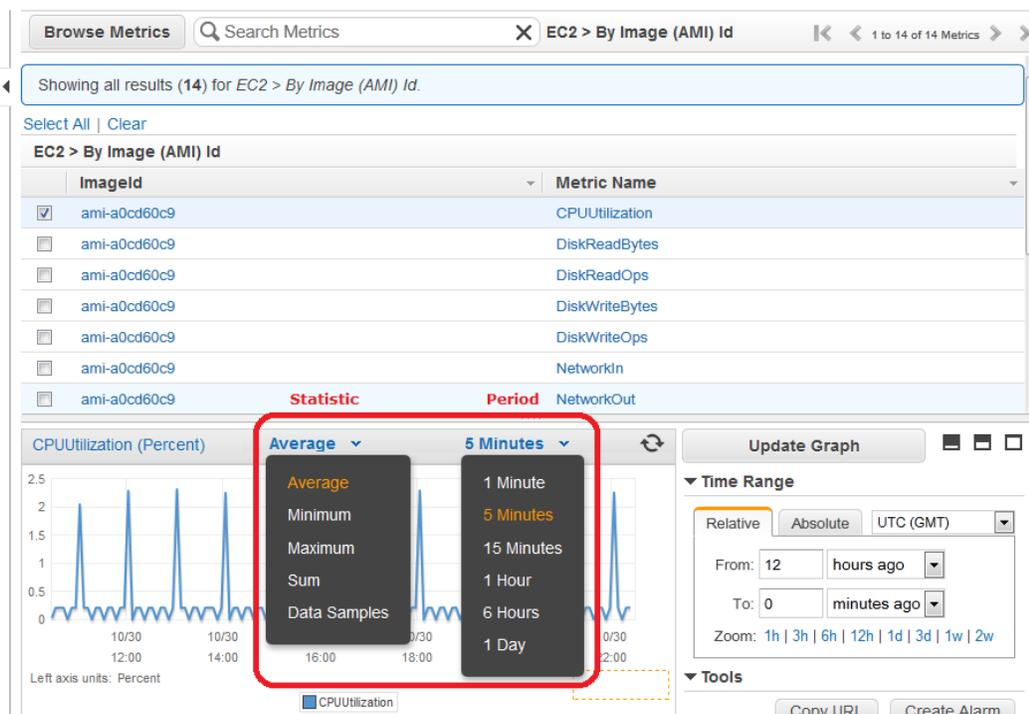
イメージ ID で利用可能なメトリックスが上側のペインに表示されます。

- [CPUUtilization] とイメージ ID が含まれる行を選択します。

ami-c5e40dac に基づいてすべての EC2 インスタンスの平均 CPUUtilization を示すグラフが詳細ペインに表示されます。



- メトリックスの [Statistic] の値 (例: 平均) を変更するには、ポップアップリストから異なる値を選択します。



- [Period] の値 (例: 5 分) を変更して、さらに粒度の細かい詳細を表示するには、ポップアップリストから異なる値を選択します。

## コマンドラインインターフェイス

イメージ ID の平均 CPU 使用率を取得するには

- 以下の例のように、`get-metric-statistics` コマンドを入力します。`get-metric-statistics` コマンドの詳細については、『*AWS Command Line Interface Reference*』の「[get-metric-statistics](#)」を参照してください。

```
Prompt>aws cloudwatch get-metric-statistics --metric-name CPUUtilization -
--start-time 2014-02-10T00:00:00 --end-time 2014-02-11T00:00:00 --period 3600
--statistics Average --namespace AWS/EC2 --dimensions Name="ImageId",Value="ami-3c47a355"
```

AWS CLI は以下の値を返します。

```
{
  "Datapoints": [
    {
      "Timestamp": "2014-02-10T07:00:00Z",
      "Average": 0.041000000000000009,
      "Unit": "Percent"
    },
    {
      "Timestamp": "2014-02-10T14:00:00Z",
      "Average": 0.079579831932773085,
      "Unit": "Percent"
    }
  ]
}
```

```
    },  
    {  
      "Timestamp": "2014-02-10T06:00:00Z",  
      "Average": 0.036000000000000011,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T13:00:00Z",  
      "Average": 0.037625000000000013,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T18:00:00Z",  
      "Average": 0.042750000000000003,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T21:00:00Z",  
      "Average": 0.039705882352941188,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T20:00:00Z",  
      "Average": 0.039375000000000007,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T02:00:00Z",  
      "Average": 0.041041666666666671,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T01:00:00Z",  
      "Average": 0.041083333333333354,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T23:00:00Z",  
      "Average": 0.038016877637130804,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T15:00:00Z",  
      "Average": 0.037666666666666668,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T12:00:00Z",  
      "Average": 0.039291666666666676,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T03:00:00Z",  
      "Average": 0.036000000000000004,  
      "Unit": "Percent"  
    },  
    {  
      "Timestamp": "2014-02-10T04:00:00Z",
```

```
    "Average": 0.042666666666666672,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T19:00:00Z",  
    "Average": 0.038305084745762719,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T22:00:00Z",  
    "Average": 0.039291666666666676,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T09:00:00Z",  
    "Average": 0.17126050420168065,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T08:00:00Z",  
    "Average": 0.041166666666666678,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T11:00:00Z",  
    "Average": 0.082374999999999962,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T17:00:00Z",  
    "Average": 0.037625000000000013,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T10:00:00Z",  
    "Average": 0.039458333333333345,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T05:00:00Z",  
    "Average": 0.039250000000000007,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T00:00:00Z",  
    "Average": 0.037625000000000013,  
    "Unit": "Percent"  
  },  
  {  
    "Timestamp": "2014-02-10T16:00:00Z",  
    "Average": 0.041512605042016815,  
    "Unit": "Percent"  
  }  
],  
"Label": "CPUUtilization"  
}
```

オペレーションからは、1分値の1日ごとの統計情報が返されます。各値は、指定されたマシンイメージを実行している EC2 インスタンスの平均 CPU 使用率 (パーセンテージ) を示します。

## メトリックスのグラフ化

インスタンスを起動したら、Amazon EC2 コンソールに移動し、インスタンスのモニタリンググラフを表示できます。これらのグラフは、EC2 ダッシュボードの Instances ページでインスタンスを選択すると表示されます。インスタンスの Description タブの隣りに、Monitoring タブが表示されます。以下のグラフが利用可能です。

- 平均 CPU 使用率 (パーセント)
- 平均ディスク読み込み (バイト)
- 平均ディスク書き込み (バイト)
- 最大ネットワーク受信 (バイト)
- 最大ネットワーク送信 (バイト)
- 要約ディスク読み込み操作 (カウント)
- 要約ディスク書き込み操作 (カウント)
- 要約ステータス (任意)
- 要約ステータスインスタンス (カウント)
- 要約ステータスシステム (カウント)

また、CloudWatch コンソールを使用すれば、Amazon EC2 およびその他の AWS サービスによって生成されたメトリックスデータをグラフ化して、何が起きているのかをより簡単に確認できるようになります。次の手順を使用すると、CloudWatch でメトリックスをグラフ化できます。

### Topics

- [メトリックスをグラフ化する \(p. 423\)](#)
- [複数のリソースにおける単一のメトリックスをグラフ化する \(p. 424\)](#)

## メトリックスをグラフ化する

### Abstract

次に説明するシナリオは、CloudWatch でメトリックスを選択し、データのグラフを作成する方法に関するものです。

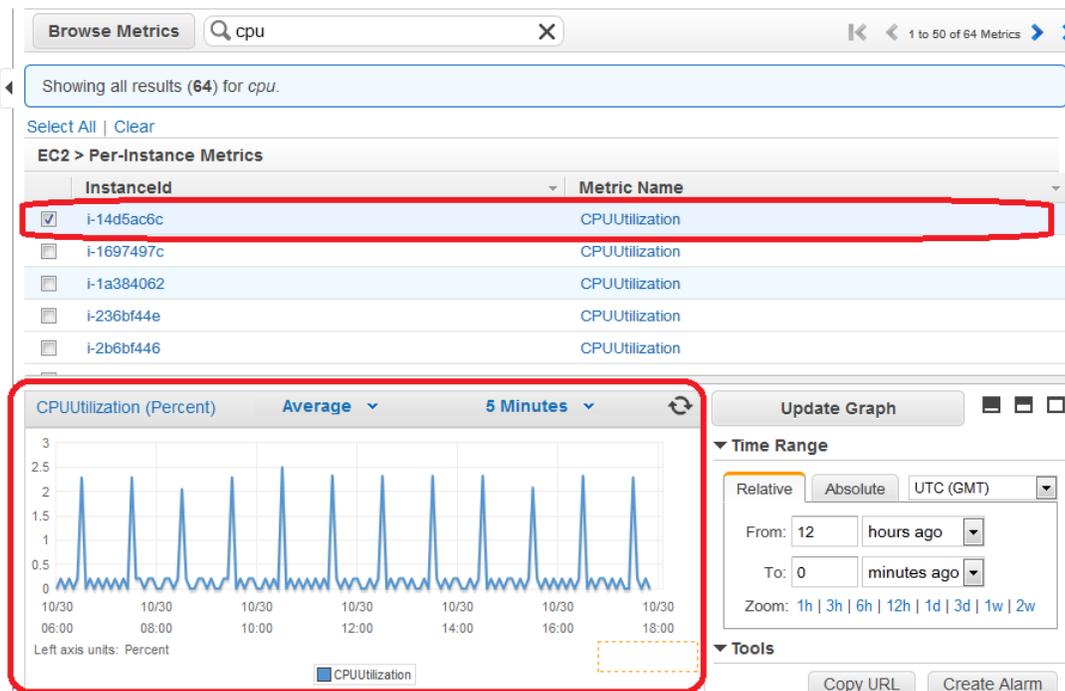
CloudWatch でメトリックを選択して、グラフを作成することができます。例えば、Amazon EC2 インスタンスの CPUUtilization メトリックスを選択して、そのインスタンスの一定期間の CPU 使用率をグラフで表示することができます。

メトリックスをグラフ化するには

1. Amazon CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、*Amazon Web Services General Reference* の [Regions and Endpoints](#) を参照してください。



3. ナビゲーションペインの [Metrics] をクリックします。
4. [CloudWatch Metrics by Category] ペインで、[Search Metrics] ボックスとカテゴリを使用して、メトリックス名、AWS リソース、またはその他のメタデータ別にメトリックスを見つけます。
5. スクロールバーやメトリックスリストの上にある進む矢印および戻る矢印を使用して、メトリックスの完全なリスト内のページを移動します。
6. 表示するメトリックス (例: CPUUtilization) を選択します。詳細ペインにグラフが表示されます。



7. このグラフを保存して後からアクセスするには、詳細ペインの [Tools] の下にある [Copy URL] をクリックし、[Copy Graph URL] ダイアログボックスで URL をコピーして、それをブラウザに貼り付けます。

## 複数のリソースにおける単一のメトリックスをグラフ化する

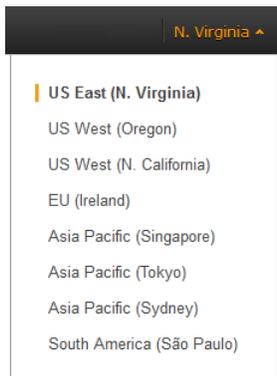
### Abstract

次に説明するシナリオは、CloudWatch ですべてのリソースにおける単一のメトリックスを選択し、そのデータのグラフを作成する方法に関するものです。

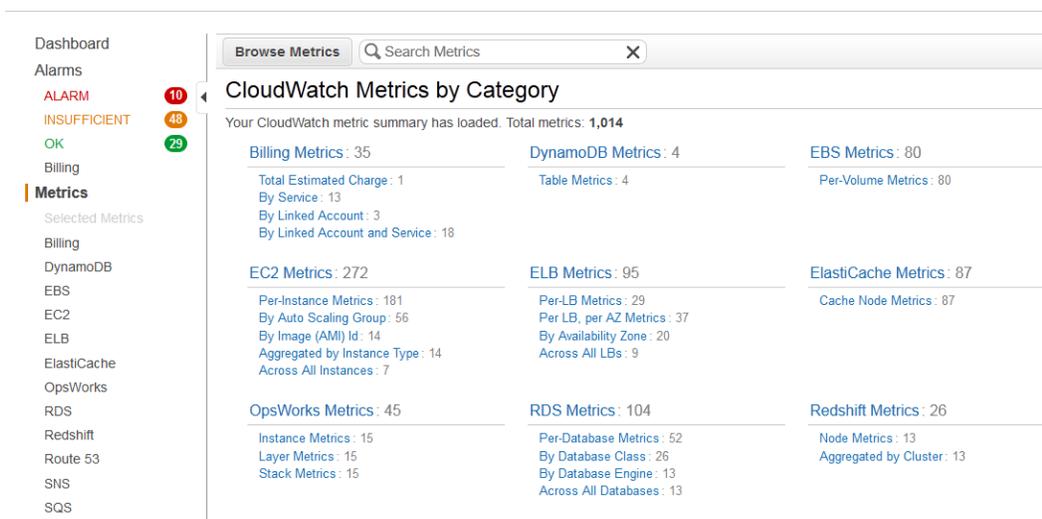
すべてのリソースにおける単一のメトリックスをグラフ化すれば、1つのグラフですべてのを確認できます。例えば、すべての Amazon EC2 インスタンスの CPUUtilization メトリックスを1つのグラフ上に表示することができます。

複数のリソースにおける単一のメトリックスをグラフ化するには

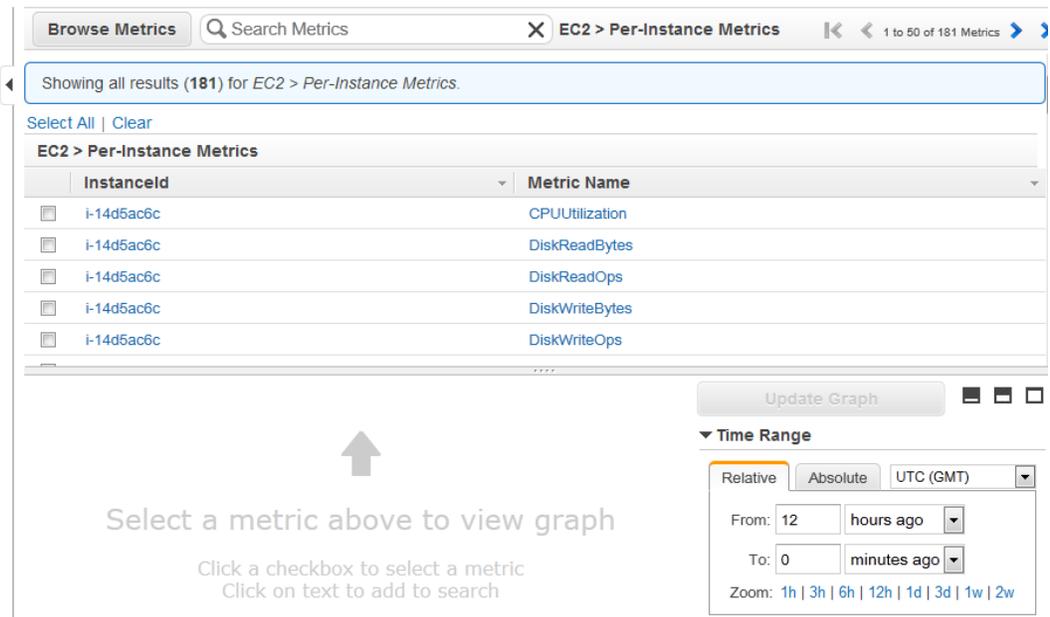
1. Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションバインの [Metrics] をクリックします。

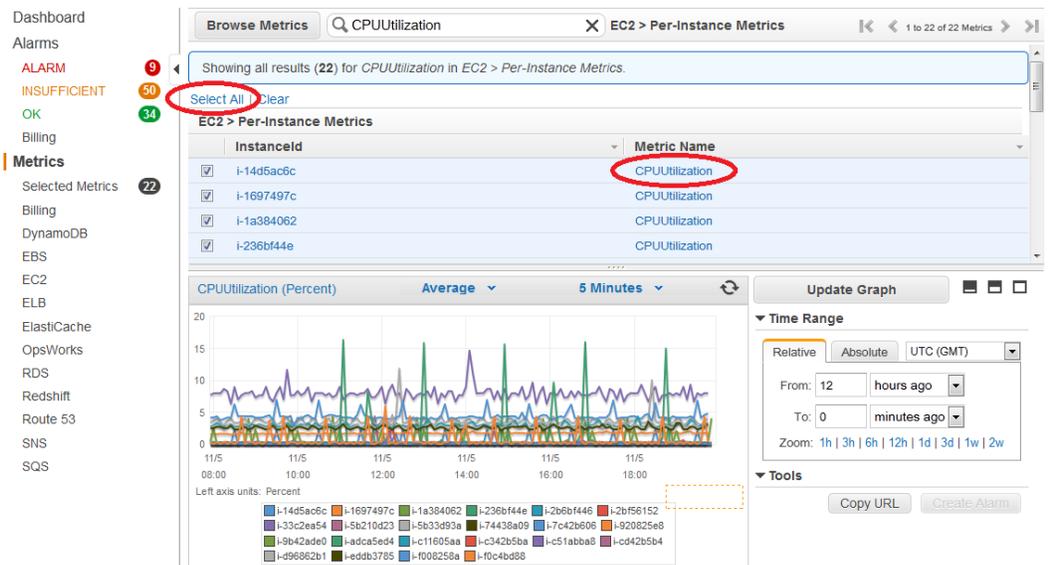


4. [CloudWatch Metrics by Category] ペインで、メトリックスのカテゴリを選択します。例えば、[EC2 Metrics] の下にある [Per-Instance Metrics] を選択します。



5. メトリックスリストの [Metric Name] 列で、メトリックスをクリックします。例えば、[CPUUtilization] を選択します。
6. メトリックスリストの先頭で、[Select All] をクリックします。

グラフには、選択したメトリックスのすべての発生に関する全データが表示されます。次の例では、すべての Amazon EC2 インスタンスの CPUUtilization が表示されています。



7. このグラフを保存して後からアクセスするには、詳細ペインの [Tools] の下にある [Copy URL] をクリックし、[Copy Graph URL] ダイアログボックスで URL をコピーして、それをブラウザに貼り付けます。

## CloudWatch アラームを作成する

Amazon EC2 インスタンスの CloudWatch メトリックスのいずれかをモニタリングする Amazon CloudWatch アラームを作成できます。CloudWatch は、指定したしきい値にメトリックスが達すると、自動的に通知を送信します。AWS Management Console の Amazon EC2 コンソールで CloudWatch アラームを作成するか、または CloudWatch コンソールを使用して、より詳細なオプションを設定することができます。

### Topics

- [CPU 使用率のアラームに基づいて E メールを送信する \(p. 427\)](#)
- [ロードバランサーアラームに基づいて E メールを送信する \(p. 429\)](#)
- [ストレージスループットアラームに基づいて E メールを送信する \(p. 432\)](#)

## CPU 使用率のアラームに基づいて E メールを送信する

### Abstract

次に説明するシナリオは、CPU 使用率のアラーム状態に基づいて E メールを送信するアラームを作成する方法に関するものです。

このシナリオでは、AWS Management Console またはコマンドラインインターフェイスを使用して、アラームの状態が OK から ALARM に変化したときに、Amazon Simple Notification Service E メールメッセージを送信する Amazon CloudWatch アラームを作成する方法について順を追って説明します。

このシナリオでは、5 分間隔で連続して 2 回、EC2 インスタンスの平均 CPU 使用が 70 パーセントを超えたときに、アラームが ALARM 状態に変化するように設定します。

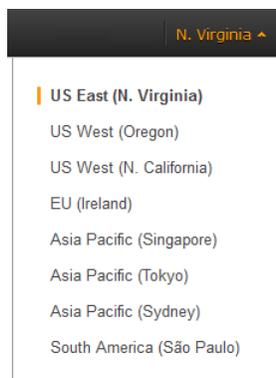
### Topics

- [AWS マネジメントコンソール \(p. 427\)](#)
- [コマンドラインインターフェイス \(p. 429\)](#)

## AWS マネジメントコンソール

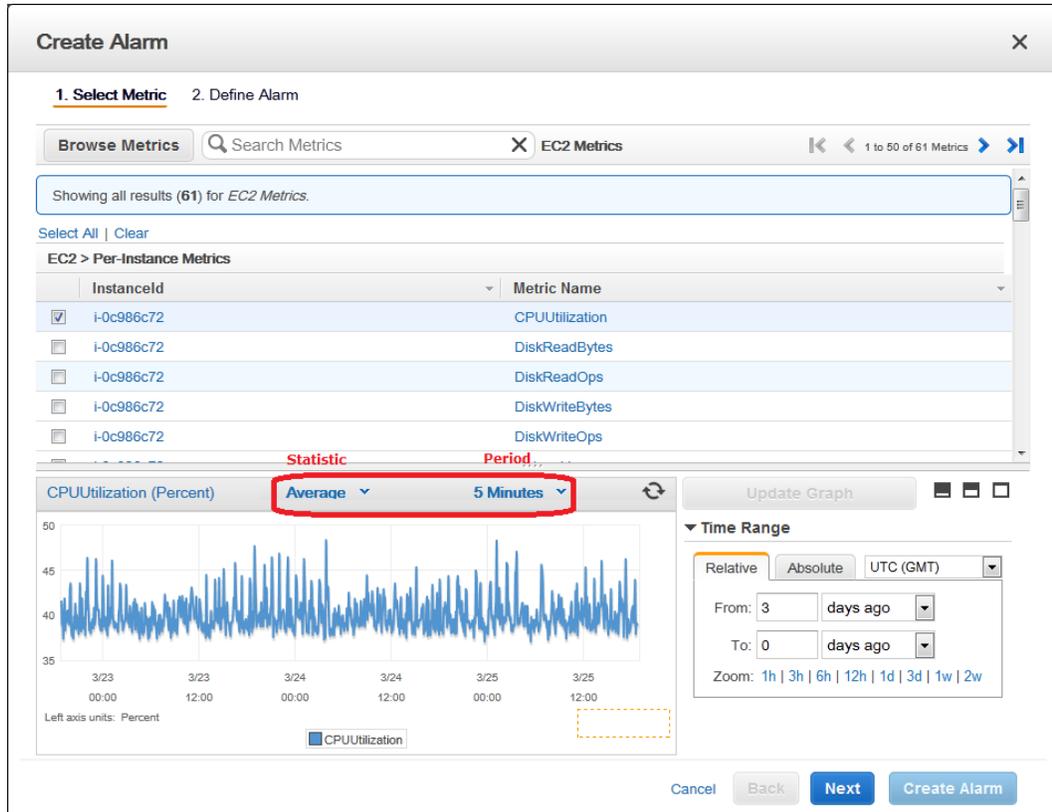
CPU 使用率に基づいて E メールを送信するアラームを作成するには

1. Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションペインの [Alarms] をクリックします。

- [Create Alarm] をクリックし、[CloudWatch Metrics by Category] ペインでメトリックスのカテゴリ (たとえば [EC2 Metrics]) を選択します。
- メトリックスのリストで、特定のインスタンス ID の [CPUUtilization] が含まれる行を選択します。  
単一のインスタンスの平均 CPUUtilization を表すグラフが下側のペインに表示されます。



- [Statistic] ドロップダウンリストから [Average] を選択します。
- [Period] ドロップダウンリストから、期間 (5 など) を選択します。
- [Next] をクリックし、[Alarm Threshold] の [Name] フィールドに一意のアラーム名 (例: 「myHighCpuAlarm」) を入力します。
- [Description] フィールドには、アラームの説明 (例: 「CPU 70」) を入力します。
- [is] ドロップダウンリストで [>] を選択します。
- [is] ドロップダウンリストの隣にあるボックスに 「70」 を入力し、[for] フィールドに 「10」 を入力します。  
[Alarm Preview] の下に、しきい値が図で表示されます。
- [Actions] の [Whenever this alarm] ドロップダウンリストで、[State is ALARM] を選択します。
- [Send notification to] ドロップダウンリストにある既存の Amazon SNS トピックを選択するか、新しい Amazon SNS トピックを作成します。
- 新しい Amazon SNS トピックを作成するには、[New list] を選択します。  
[Send notification to] フィールドに、新しい Amazon SNS トピックの名前 (例: 「myHighCpuAlarm」) を入力し、[Email list] フィールドに、アラームが ALARM 状態に変化したときに通知を受ける E メールアドレスをカンマで区切って入力します。
- アラームの作成を完了するには、[Create Alarm] をクリックします。

## コマンドラインインターフェイス

CPU 使用率が 70 パーセントを超えたときに、Amazon Simple Notification Service E メールメッセージを送信するには

1. Amazon Simple Notification Service トピックを設定するか、使用するトピックのトピックリソース名を取得します。Amazon Simple Notification Service トピックの設定については、[Set Up Amazon Simple Notification Service](#) を参照してください。
2. `put-metric-alarm` コマンドを使用してアラームを作成します。`put-metric-alarm` コマンドの詳細については、『*AWS Command Line Interface Reference*』の「`put-metric-alarm`」を参照してください。次に示す例の値を使用します。ただし、`InstanceID` と `alarm-actions` の値はご自分の値に置き換えます。

```
Prompt>aws cloudwatch
      put-metric-alarm --alarm-name cpu-mon --alarm-description
"Alarm when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2
--statistic Average --period 300
      --threshold 70 --comparison-operator GreaterThanThreshold -
--dimensions Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-
actions arn:aws:sns:us-east-1:111122223333:MyTopic --unit Percent
```

コマンドが成功した場合、AWS CLI はコマンドプロンプトに戻ります。

3. `set-alarm-state` コマンドを使用してアラームの状態を強制的に変化させ、アラームをテストします。
  - a. アラームの状態を `INSUFFICIENT_DATA` から `OK` に変更します。

```
Prompt>aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-
reason "initializing" --state-value OK
```

コマンドが成功した場合、AWS CLI はコマンドプロンプトに戻ります。

- b. アラームの状態を `OK` から `ALARM` に変更します。

```
Prompt>aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-
reason "initializing" --state-value ALARM
```

コマンドが成功した場合、AWS CLI はコマンドプロンプトに戻ります。

- c. E メールが届いていることを確認します。

## ロードバランサーアラームに基づいて E メールを送信する

### Abstract

次に説明するシナリオは、アラームの状態に基づいて E メールを送信するロードバランサーを設定する方法に関するものです。

このシナリオでは、AWS マネジメントコンソールまたはコマンドラインインターフェイスを使用して Amazon Simple Notification Service 通知をセットアップし、さらに 100 ms を超えるロードバランサーレイテンシーをモニタリングするアラームを設定する方法を説明します。

### Topics

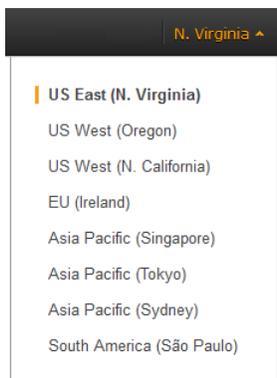
- [AWS マネジメントコンソール \(p. 430\)](#)

- [コマンドラインインターフェイス \(p. 432\)](#)

## AWS マネジメントコンソール

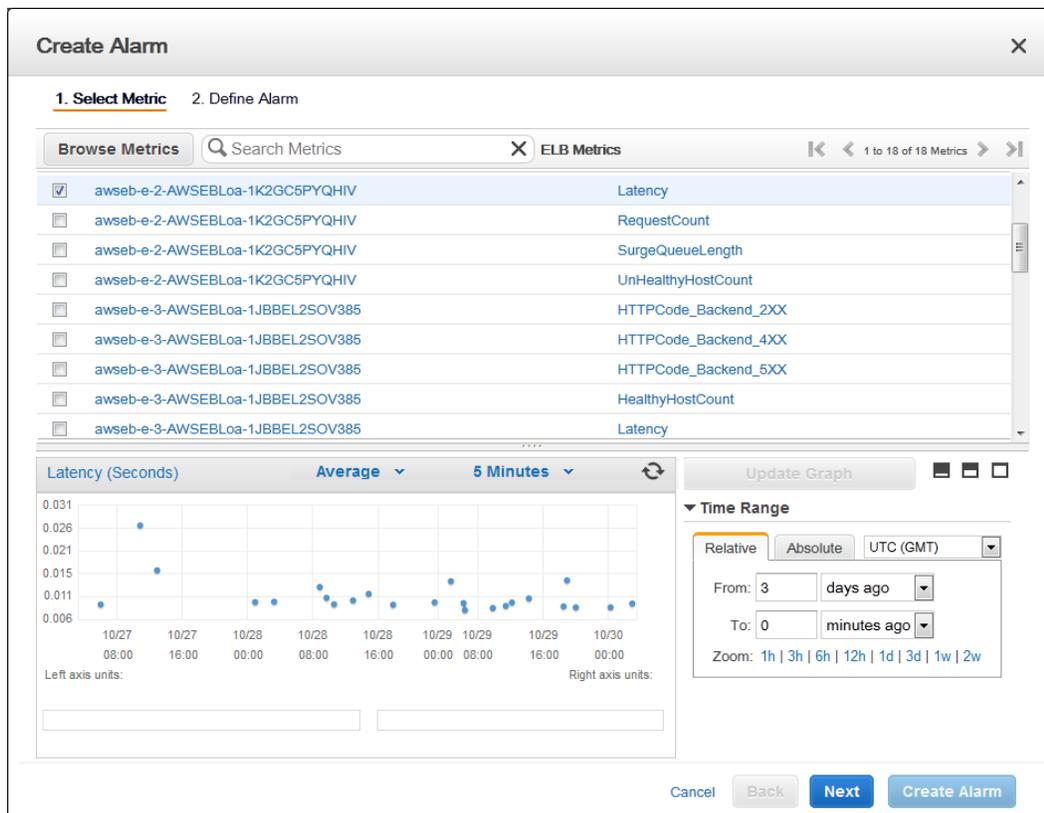
E メールを送信するロードバランサーアラームを作成するには

1. Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションペインの [Alarms] をクリックします。
4. [Create Alarm] をクリックし、[CloudWatch Metrics by Category] ペインでメトリックのカテゴリ ( 例: [ELB Metrics] ) を選択します。
5. メトリックスのリストで、特定のロードバランサーの [Latency] が含まれる行を選択します。

単一のロードバランサーの平均 Latency を表すグラフが下側のペインに表示されます。



6. [Statistic] ドロップダウンリストから [Average] を選択します。
7. [Period] ドロップダウンリストから [1 Minute] を選択します。
8. [Next] をクリックし、[Alarm Threshold] の [Name] フィールドに一意のアラーム名 ( 例: 「myHighCpuAlarm」 ) を入力します。
9. [Description] フィールドには、アラームの説明 ( 例: 「Latency 100ms」 ) を入力します。
10. [is] ドロップダウンリストで [>] を選択します。
11. [is] ドロップダウンリストの隣にあるボックスに「0.1」を入力し、[for] フィールドに「3」を入力します。

[Alarm Preview] の下に、しきい値が図で表示されます。

12. [Actions] の [Whenever this alarm] ドロップダウンリストで、[State is ALARM] を選択します。
13. [Send notification to] ドロップダウンリストにある既存の Amazon SNS トピックを選択するか、新しい Amazon SNS トピックを作成します。
14. 新しい Amazon SNS トピックを作成するには、[New list] を選択します。

[Send notification to] フィールドに、新しい Amazon SNS トピックの名前 ( 例: 「myHighCpuAlarm」 ) を入力し、[Email list] フィールドに、アラームが ALARM 状態に変化したときに通知を受ける E メールアドレスをカンマで区切って入力します。

15. アラームの作成を完了するには、[Create Alarm] をクリックします。

## コマンドラインインターフェイス

ロードバランサーの Latency が 100 ms を超えたときに、Amazon Simple Notification Service E メールメッセージを送信するには

1. Amazon Simple Notification Service トピックを作成します。Amazon SNS トピックの作成手順については、[Set Up Amazon Simple Notification Service](#) を参照してください。
2. `put-metric-alarm` コマンドを使用してアラームを作成します。`put-metric-alarm` コマンドの詳細については、『*AWS Command Line Interface Reference*』の「[put-metric-alarm](#)」を参照してください。

```
Prompt>aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description
"Alarm when Latency exceeds 100ms" --metric-name Latency --namespace AWS/ELB
--statistic Average --period 60 --threshold 100 --comparison-operator
GreaterThanThreshold --dimensions Name=LoadBalancerName,Value=my-server --
evaluation-periods 3 --alarm-actions arn:aws:sns:us-east-1:1234567890:my-
topic --unit Milliseconds
```

コマンドが成功した場合、AWS CLI はコマンドプロンプトに戻ります。

3. アラームをテストします。
  - アラームの状態を強制的に ALARM に変化させます。

```
Prompt>aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
"initializing" --state OK
Prompt>aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
"initializing" --state ALARM
```

コマンドが成功した場合、AWS CLI はコマンドプロンプトに戻ります。

- E メールが届いていることを確認します。

## ストレージスループットアラームに基づいてEメールを送信する

### Abstract

次に説明するシナリオは、アラームの状態に基づいてEメールを送信するストレージスループットアラームを作成する方法に関するものです。

このシナリオでは、AWS Management Console またはコマンドラインインターフェイスを使用して Amazon Simple Notification Service 通知をセットアップし、さらに EBS が 100 MB スループットを上回ったときに E メールを送信するアラームを設定します。

### Topics

- [AWS マネジメントコンソール \(p. 433\)](#)
- [コマンドラインインターフェイス \(p. 434\)](#)

## AWS マネジメントコンソール

E メールを送信するストレージスループットアラームを作成するには

1. Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーで、ニーズに合ったリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションペインの [Alarms] をクリックします。
4. [Create Alarm] をクリックし、[CloudWatch Metrics by Category] ペインで、メトリックスのカテゴリ ( 例: [EBS Metrics] ) を選択します。
5. メトリックスのリストで、特定のボリューム ID の [VolumeWriteBytes] が含まれる行を選択します。

単一のボリュームの平均 VolumeWriteBytes を表すグラフが下側のペインに表示されます。

VolumeID	Metric Name
vol-75192701	VolumeIdleTime
vol-75192701	VolumeQueueLength
vol-75192701	VolumeReadOps
vol-75192701	VolumeTotalWriteTime
vol-75192701	VolumeWriteBytes

6. [Statistic] ドロップダウンリストから [Average] を選択します。
7. [Period] ドロップダウンリストから [5 Minutes] を選択します。
8. [Next] をクリックし、[Alarm Threshold] の [Name] フィールドに一意のアラーム名 (例: 「myHighWriteAlarm」) を入力します。
9. [Description] フィールドには、アラームの説明 (例: 「volumeWriteBytes 100,000 KiB/s」) を入力します。
10. [is] ドロップダウンリストで [>] を選択します。
11. [is] ドロップダウンリストの隣にあるボックスに 「100000」 を入力し、[for] フィールドに 「15」 を入力します。

[Alarm Preview] の下に、しきい値が図で表示されます。

12. [Actions] の [Whenever this alarm] ドロップダウンリストで、[State is ALARM] を選択します。
13. [Send notification to] ドロップダウンリストにある既存の Amazon SNS トピックを選択するか、新しい Amazon SNS トピックを作成します。
14. 新しい Amazon SNS トピックを作成するには、[New list] を選択します。

[Send notification to] フィールドに、新しい Amazon SNS トピックの名前 (例: 「myHighCpuAlarm」) を入力し、[Email list] フィールドに、アラームが ALARM 状態に変化したときに通知を受ける E メールアドレスをカンマで区切って入力します。

15. アラームの作成を完了するには、[Create Alarm] をクリックします。

## コマンドラインインターフェイス

EBS が 100 MB スループットを上回ったときに Amazon Simple Notification Service E メールメッセージを送信するには

1. Amazon Simple Notification Service トピックを作成します。Amazon SNS トピックの作成手順については、[Set Up Amazon Simple Notification Service](#) を参照してください。
2. `put-metric-alarm` コマンドを使用してアラームを作成します。`put-metric-alarm` コマンドの詳細については、『*AWS Command Line Interface Reference*』の「`put-metric-alarm`」を参照してください。

```
Prompt>aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description "Alarm when EBS volume exceeds 100MB throughput" --metric-name VolumeReadBytes --namespace AWS/EBS --statistic Average --period 300 --threshold 100000000 --comparison-operator GreaterThanThreshold --dimensions Name=VolumeId,Value=my-volume-id --evaluation-periods 3 --alarm-actions arn:aws:sns:us-east-1:1234567890:my-alarm-topic --insufficient-data-actions arn:aws:sns:us-east-1:1234567890:my-insufficient-data-topic
```

コマンドが成功した場合、AWS CLI はコマンドプロンプトに戻ります。

3. アラームをテストします。
  - アラームの状態を強制的に ALARM に変化させます。

```
Prompt>aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "initializing" --state-value OK
Prompt>aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "initializing" --state-value ALARM
Prompt>aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason "initializing" --state-value INSUFFICIENT_DATA
```

- E メールが 2 通届いていることを確認します。

# インスタンスを停止または終了するアラームを作成する

## Abstract

Amazon EC2 インスタンスを自動的に停止または終了するアラームを作成します。

Amazon CloudWatch アラームアクションを使用すると、Amazon Elastic Compute Cloud ( Amazon EC2 ) インスタンスの実行を継続する必要がなくなった場合に、そのインスタンスを自動的に停止または終了するアラームを作成することができます。例えば、バッチ給与計算処理ジョブまたは科学計算タスクを専用に行うインスタンスを使用している場合が挙げられます。これらのインスタンスは一定期間動作して仕事を完了します。このようなインスタンスは、アイドル状態 ( 課金されている状態 ) にせず、停止または終了するとコスト削減につながります。停止アラームアクションと終了アラームアクションの主な違いは、停止したインスタンスを後から再起動する必要が生じたときに簡単に実行でき、さらに同じインスタンス ID とルートボリュームを維持できることです。一方、終了したインスタンスは再起動できません。代わりに、新しいインスタンスを起動する必要があります。

停止アラームまたは終了アラームのアクションは、Amazon EC2 インスタンスメトリックスに設定されている任意のアラームに追加できます。例えば、( AWS/EC2 ネームスペースで ) Amazon CloudWatch によって提供される基本モニタリングメトリックスと詳細モニタリングメトリックス、また「InstanceID=」ディメンションを含む任意のカスタムメトリックスに設定されているアラームが対象です。ただし、InstanceID 値が実行中の有効な Amazon EC2 インスタンスを参照している場合に限りです。

## Topics

- [Amazon CloudWatch アラームへのアクションの追加 \(p. 435\)](#)
- [Amazon CloudWatch アラームアクションシナリオ \(p. 447\)](#)

## Amazon CloudWatch アラームへのアクションの追加

アラームアクションを設定するには、Amazon EC2 コンソールまたは Amazon CloudWatch コンソールのいずれかを使用できます。あるいは、Amazon CloudWatch コマンドラインインターフェイス ( CLI )、API、または AWS SDK を使用することもできます。Amazon CloudWatch API を AWS SDK と合わせて使用する詳細については、[サンプルコード&ライブラリ](#)を参照してください。

## Amazon EC2 コンソールを使用して、インスタンスを停止するアラームを作成する

特定のしきい値を超えた場合に Amazon EC2 インスタンスを停止するアラームを作成できます。例えば、開発またはテスト用のインスタンスを実行したまま、終了するのを忘れることがたまにあります。CPU 平均使用率が 24 時間にわたって 10 パーセントを下回った場合にトリガされ、インスタンスがアイドル状態にあり使用されていないことを通知するアラームを作成できます。必要に応じて、しきい値、期間、および間隔を調整できます。さらに、アラームがトリガされたときに E メールが届くように Amazon Simple Notification Service ( Amazon SNS ) 通知を追加できます。

ルートデバイスとして Amazon Elastic Block Store ボリュームを使用している Amazon EC2 インスタンスは、停止することも終了することもできます。一方、ルートデバイスとしてインスタンスストアを使用しているインスタンスは、終了することのみ可能です。



### Note

AWS Identity and Access Management ( IAM ) アカウントを使用してアラームを作成または変更する場合、アラームアクションが実行されるためには、Amazon EC2 の許可 ( ec2:DescribeInstanceStatus、ec2:DescribeInstances、ec2:StopInstances、ec2:TerminateInstances ) が必要です。Amazon CloudWatch にする読み取り/書き込み許

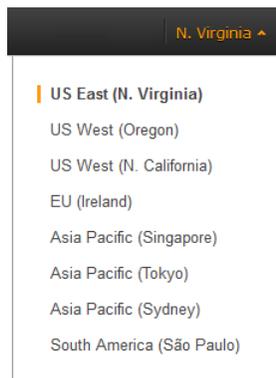
可はあっても、Amazon EC2 に対しては許可がない場合でもアラームは作成できます。しかし、Amazon EC2 インスタンスで停止または終了アクションは実行されません。ただし、関連付けられている Amazon EC2 API の使用許可が後で付与される場合、以前に作成したアラームアクションは実行されるようになります。IAM 許可の詳細については、[IAM を使用するの Permissions and Policies](#) を参照してください。

IAM ロール（例えば、Amazon EC2 インスタンスプロファイル）を使用している場合は、アラームアクションを使用してインスタンスを停止または終了することはできません。ただし、アラームの状態を確認して、Amazon SNS 通知や Auto Scaling ポリシーなど、その他のアクションを実行することはできます。

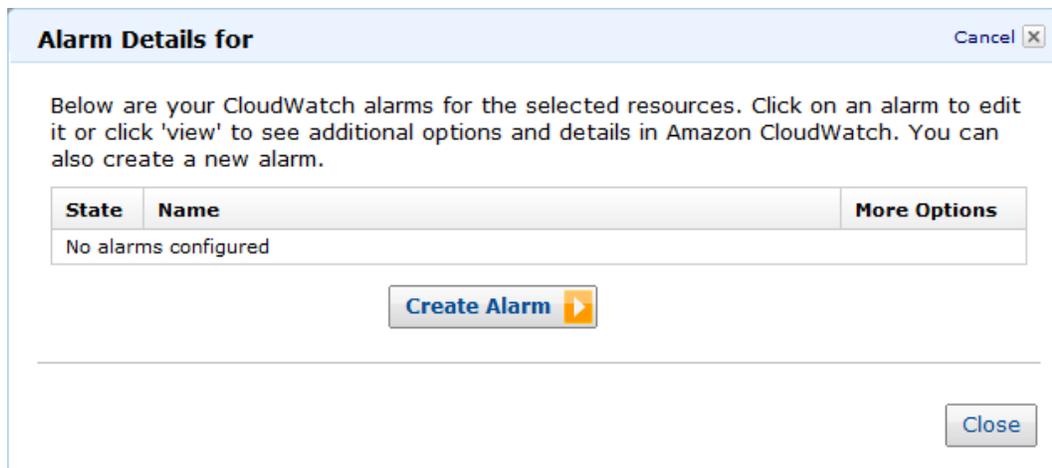
AWS Security Token Service (AWS STS) を用いて許可された一時的な認証情報を使用している場合は、アラームアクションを用いて Amazon EC2 インスタンスを停止または終了することはできません。

アイドル状態のインスタンスを停止するアラームを作成するには

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーから、インスタンスが動作しているリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションペインで、[INSTANCES] の下にある [Instances] をクリックします。
4. 上側のペインで、インスタンスを右クリックし、[Add/Edit Alarms] をクリックします。  
または、インスタンスを選択し、下側のペインの [Monitoring] タブで [Create Alarm] をクリックします。
5. [Alarm Details for] ダイアログボックスで、[Create Alarm] をクリックします。



- アラームがトリガされたときに E メールが届くようにする場合は、[Create Alarm for] ダイアログボックスの [Send a notification to] ボックスで、既存の Amazon SNS トピックを選択するか、または [Create Topic] をクリックして新しい Amazon SNS トピックを作成します。

新しいトピックを作成する場合は、[Send a notification to] ボックスでトピックの名前を入力し、[With these recipients] ボックスで、該当する受信者の E メールアドレス (コンマで区切る) を入力します。その後、アラームを作成すると、このトピックの E メールを受信する前に、許可する必要がある受信登録確認メールが送信されます。

The screenshot shows the 'Create Alarm for' dialog box. It contains the following fields and options:

- Send a notification to:** EC2InstanceStop (with a 'cancel' link)
- With these recipients:** john.stiles@example.com
- Take the action:**  Stop  Terminate this instance.
- Whenever:** Average of CPU Utilization
- Is:** < 10 Percent
- For at least:** 24 consecutive period(s) of 1 hour
- Name of alarm:** EC2InstanceStop

At the bottom right, there are 'Cancel' and 'Create Alarm' buttons. A line graph on the right shows 'CPU Utilization (Percent)' with a red threshold line at 10% and a blue line representing the metric data.

- [Take the action] チェックボックスをオンにし、[Stop] ラジオボタンを選択します。
- [Whenever] ボックスで、使用する統計を選択し、メトリックスを選択します。この例では、[Average] と [CPU Utilization] を選択します。
- [Is] ボックスで、メトリックスのしきい値を定義します。この例では、[10] パーセントを入力します。
- [For at least] ボックスで、アラームのサンプリング期間を選択します。この例では、1 時間の期間の連続数として [24] を入力します。
- アラームの名前を変更するには、[Name this alarm] ボックスに新しい名前を入力します。

アラーム名を入力しない場合は、Amazon CloudWatch によってアラーム名が自動的に作成されます。



#### Note

アラーム設定は、アラームを作成する前に実際の要件に基づいて調整することも、アラーム作成後に編集することもできます。その対象となる設定には、メトリックス、しきい値、期間、アクション、および通知の設定などがあります。ただし、アラームの作成後は、アラームの名前を編集できなくなります。

- [Create Alarm] をクリックします。

## Amazon EC2 コンソールを使用して、インスタンスを終了するアラームを作成する

特定のしきい値を超えたときに EC2 インスタンスを自動的に終了するアラームを作成できます (インスタンスの終了保護が有効になっていない場合)。例えば、インスタンスが仕事を終え、再びそのインスタンスを使用する必要がない場合は、インスタンスを終了することをお勧めします。インスタンスを後で使用できる可能性がある場合は、インスタンスを終了するのではなく停止します。インスタンスの終了

保護の有効化および無効化の詳細については、*Amazon Elastic Compute Cloud ユーザーガイド* の [Enabling Termination Protection for an Instance](#) を参照してください。



#### Note

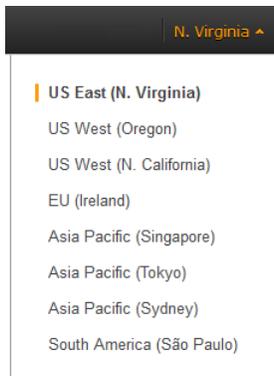
AWS Identity and Access Management ( IAM ) アカウントを使用してアラームを作成または変更する場合は、Amazon EC2 許可となる `ec2:DescribeInstanceStatus`、`ec2:DescribeInstances`、`ec2:StopInstances`、およびアラームアクションを実行するための `ec2:TerminateInstances` が必要です。Amazon CloudWatch にする読み取り/書き込み許可はあっても、Amazon EC2 に対しては許可がない場合でもアラームは作成できます。しかし、Amazon EC2 インスタンスで停止または終了アクションは実行されません。ただし、関連付けられている Amazon EC2 API の使用許可が後で付与される場合、以前に作成したアラームアクションは実行されるようになります。IAM 許可の詳細については、[IAM を使用するの Permissions and Policies](#) を参照してください。

IAM ロール (例えば、Amazon EC2 インスタンスプロファイル) を使用している場合は、アラームアクションを使用してインスタンスを停止または終了することはできません。ただし、アラームの状態を確認して、Amazon SNS 通知や Auto Scaling ポリシーなど、その他のアクションを実行することはできます。

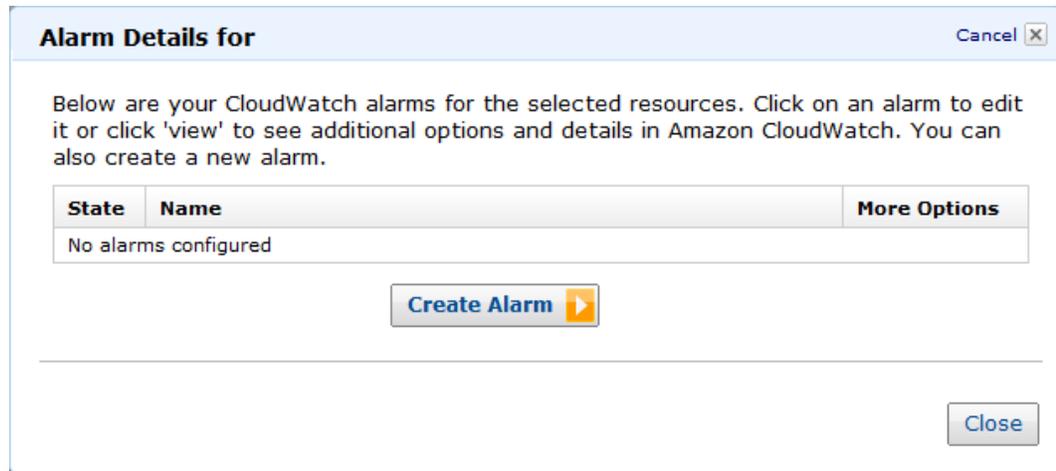
AWS Security Token Service (AWS STS) を用いて許可された一時的な認証情報を使用している場合は、アラームアクションを用いて Amazon EC2 インスタンスを停止または終了することはできません。

アイドル状態のインスタンスを終了するアラームを作成するには

1. Amazon EC2 コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーから、インスタンスが動作しているリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。

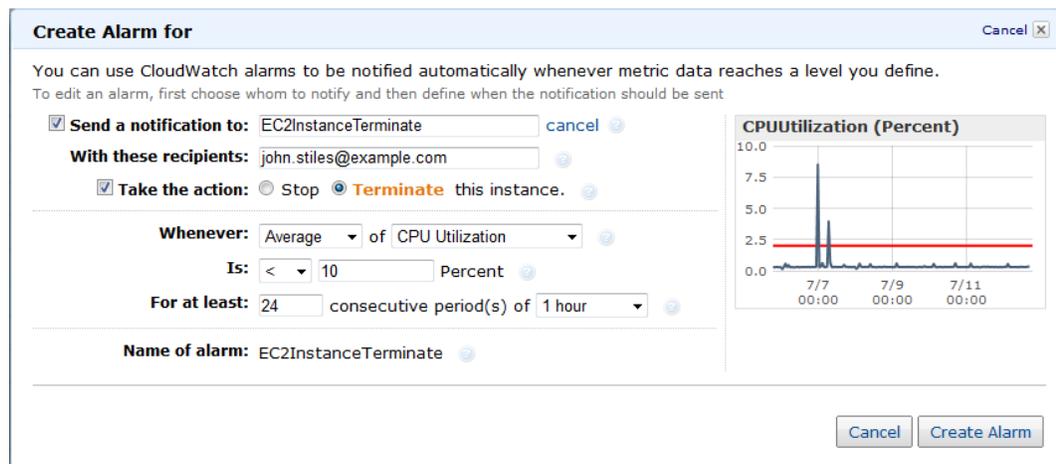


3. ナビゲーションペインで、[INSTANCES] の下にある [Instances] をクリックします。
4. 上側のペインで、インスタンスを右クリックし、[Add/Edit Alarms] をクリックします。  
または、インスタンスを選択し、下側のペインの [Monitoring] タブで、[Create Alarm] をクリックします。
5. [Alarm Details for] ダイアログボックスで、[Create Alarm] をクリックします。



- アラームがトリガされたときに E メールが届くようにする場合は、[Create Alarm for] ダイアログボックスの [Send a notification to] ボックスで既存の SNS トピックを選択するか、[Create Topic] をクリックして新しい SNS トピックを作成します。

新しいトピックを作成する場合は、[Send a notification to] ボックスでトピックの名前を入力し、[With these recipients] ボックスで、該当する受信者の E メールアドレス (コンマで区切る) を入力します。その後、アラームを作成すると、このトピックの E メールを受信する前に、許可する必要がある受信登録確認メールが送信されます。



- [Take the action] チェックボックスをオンにし、[Terminate] ラジオボタンを選択します。
- [Whenever] ボックスで、使用する統計を選択し、メトリックスを選択します。この例では、[Average] と [CPU Utilization] を選択します。
- [Is] ボックスで、メトリックスのしきい値を定義します。この例では、[10] パーセントを入力します。
- [For at least] ボックスで、アラームのサンプリング期間を選択します。この例では、1 時間の期間の連続数として [24] を入力します。
- アラームの名前を変更するには、[Name this alarm] ボックスに新しい名前を入力します。

アラーム名を入力しない場合は、Amazon CloudWatch によってアラーム名が自動的に作成されます。



#### Note

アラーム設定は、アラームを作成する前に実際の要件に基づいて調整することも、アラーム作成後に編集することもできます。その対象となる設定には、メトリックス、しきい値、期間、アクション、および通知の設定などがあります。ただし、アラームの作成後は、アラームの名前を編集できなくなります。

12. [Create Alarm] をクリックします。

## Amazon CloudWatch コンソールを使用して、インスタンスを停止するアラームを作成する

特定のしきい値を超えた場合に Amazon EC2 インスタンスを停止するアラームを作成できます。例えば、開発またはテスト用のインスタンスを実行したまま、終了するのを忘れることがたまにあります。CPU 平均使用率が 24 時間にわたって 10 パーセントを下回った場合にトリガされ、インスタンスがアイドル状態にあり使用されていないことを通知するアラームを作成できます。必要に応じて、しきい値、期間、および間隔を調整できます。さらに、アラームがトリガされたときに E メールが届くように Amazon Simple Notification Service ( Amazon SNS ) 通知を追加できます。

Amazon CloudWatch アラームアクションは、EBS-Backed Amazon EC2 インスタンスを停止できますが、Instance store-Backed Amazon EC2 インスタンスを停止することはできません。ただし、Amazon CloudWatch アラームアクションは、どのタイプの Amazon EC2 インスタンスでも終了することができます。



#### Note

AWS Identity and Access Management ( IAM ) アカウントを使用してアラームを作成または変更する場合、アラームアクションが実行されるためには、Amazon EC2 の許可 ( `ec2:DescribeInstanceStatus`、`ec2:DescribeInstances`、`ec2:StopInstances`、`ec2:TerminateInstances` ) が必要です。Amazon CloudWatch にする読み取り/書き込み許可はあっても、Amazon EC2 に対しては許可がない場合でもアラームは作成できます。しかし、Amazon EC2 インスタンスで停止または終了アクションは実行されません。ただし、関連付けられている Amazon EC2 API の使用許可が後で付与される場合、以前に作成したアラームアクションは実行されるようになります。IAM 許可の詳細については、[IAM を使用するの Permissions and Policies](#) を参照してください。

IAM ロール ( 例えば、Amazon EC2 インスタンスプロファイル ) を使用している場合は、アラームアクションを使用してインスタンスを停止または終了することはできません。ただし、アラームの状態を確認して、Amazon SNS 通知や Auto Scaling ポリシーなど、その他のアクションを実行することはできます。

AWS Security Token Service ( AWS STS ) を用いて許可された一時的な認証情報を使用している場合は、アラームアクションを用いて Amazon EC2 インスタンスを停止または終了することはできません。

アイドル状態のインスタンスを停止するアラームを作成するには

1. Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーから、インスタンスが動作しているリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションペインの [Alarms] をクリックします。
4. [Create Alarm] をクリックし、[CloudWatch Metrics by Category] ペインで、[EC2 Metrics] の下にある [Per-Instance Metrics] を選択します。
5. メトリックスのリストで、アラームを作成するインスタンスとメトリックスを選択します。また、検索ボックスにインスタンス ID を入力して希望するインスタンスを見つけることができます。
6. [Statistic] ドロップダウンリストから [Average] を選択します。
7. [Period] ドロップダウンリストから、期間 ( 1 など ) を選択します。
8. [Next] をクリックし、[Alarm Threshold] の [Name] フィールドに一意的なアラーム名 ( 例: 「stop EC2 instance」 ( EC2 インスタンスを停止する ) ) を入力します。
9. [Description] フィールドにアラームの説明 ( 「CPU EC2 」など ) を入力します。
10. [is] ドロップダウンリストで [<] を選択します。
11. [is] ドロップダウンリストの隣にあるボックスに 「10」 を入力し、[for] フィールドに 「1440」 を入力します。

[Alarm Preview] の下に、しきい値が図で表示されます。

12. [Actions] の下にある [EC2 Action] をクリックします。
13. [Whenever this alarm] ドロップダウンリストで、[State is ALARM] を選択します。
14. [Take this action] ドロップダウンリストで、[Stop this instance] を選択します。
15. [Notification] をクリックし、[Send notification to] ドロップダウンリストにある既存の Amazon SNS トピックを選択するか、新しい Amazon SNS トピックを作成します。
16. 新しい Amazon SNS トピックを作成するには、[New list] を選択します。

[Send notification to] フィールドに、新しい Amazon SNS トピックの名前 ( 「stop\_EC2\_Instance」 など ) を入力し、[Email list] フィールドに、アラームが ALARM 状態に変化したときに通知を送信する E メールアドレスをカンマで区切って入力します。



#### Important

新しいトピックを作成する場合や E メールアドレスを既存のトピックに追加する場合は、追加する各 E メールアドレスにトピックの受信登録の確認メールが送信されます。お客様が確認メールに含まれたリンクをクリックし、受信登録を確認すると、新しい E メールアドレスに通知が送信されるようになります。

17. アラームの作成を完了するには、[Create Alarm] をクリックします。

## Amazon CloudWatch コンソールを使用して、アイドル状態のインスタンスを終了するアラームを作成する

特定のしきい値を超えたときに Amazon EC2 インスタンスを自動的に終了するアラームを作成できません (インスタンスの終了保護が無効になっていない場合)。例えば、インスタンスが仕事を終え、再びそのインスタンスを使用する必要がない場合は、インスタンスを終了することをお勧めします。インスタンスを後で使用できる可能性がある場合は、インスタンスを終了するのではなく停止します。インスタンスの終了保護の無効化の詳細については、*Amazon Elastic Compute Cloud ユーザーガイド* の [Enabling Termination Protection for an Instance](#) を参照してください。



### Note

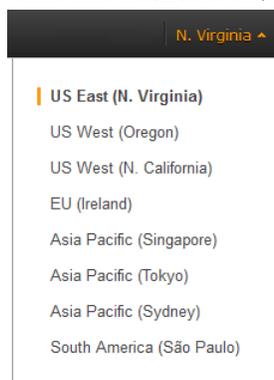
AWS Identity and Access Management (IAM) アカウントを使用してアラームを作成または変更する場合は、Amazon EC2 許可となる `ec2:DescribeInstanceStatus`、`ec2:DescribeInstances`、`ec2:StopInstances`、およびアラームアクションを実行するための `ec2:TerminateInstances` が必要です。Amazon CloudWatch にする読み取り/書き込み許可はあっても、Amazon EC2 に対しては許可がない場合でもアラームは作成できます。しかし、Amazon EC2 インスタンスで停止または終了アクションは実行されません。ただし、関連付けられている Amazon EC2 API の使用許可が後で付与される場合、以前に作成したアラームアクションは実行されるようになります。IAM 許可の詳細については、*IAM を使用するの Permissions and Policies* を参照してください。

IAM ロール (例えば、Amazon EC2 インスタンスプロファイル) を使用している場合は、アラームアクションを使用してインスタンスを停止または終了することはできません。ただし、アラームの状態を確認して、Amazon SNS 通知や Auto Scaling ポリシーなど、その他のアクションを実行することはできます。

AWS Security Token Service (AWS STS) を用いて許可された一時的な認証情報を使用している場合は、アラームアクションを用いて Amazon EC2 インスタンスを停止または終了することはできません。

アイドル状態のインスタンスを終了するアラームを作成するには

1. Amazon CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーから、インスタンスが動作しているリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションペインの [Alarms] をクリックします。
4. ペイン上の [Create Alarm] をクリックします。
5. [Create Alarm] をクリックし、[CloudWatch Metrics by Category] ペインで、[EC2 Metrics] の下にある [Per-Instance Metrics] を選択します。
6. メトリックスのリストで、アラームを作成するインスタンスとメトリックスを選択します。また、検索ボックスにインスタンス ID を入力して希望するインスタンスを見つけることができます。

7. [Statistic] ドロップダウンリストから [Average] を選択します。
8. [Period] ドロップダウンリストから、期間 ( 1 など ) を選択します。
9. [Next] をクリックし、[Alarm Threshold] の [Name] フィールドに一意のアラーム名 ( 例: 「`Terminate EC2 instance`」 ( EC2 インスタンスを終了する ) ) を入力します。
10. [Description] フィールドにアラームの説明 ( 「CPU `EC2` 」 など ) を入力します。
11. [is] ドロップダウンリストで [<] を選択します。
12. [is] ドロップダウンリストの隣にあるボックスに 「10」 を入力し、[for] フィールドに 「1440」 を入力します。

[Alarm Preview] の下に、しきい値が図で表示されます。

13. [Actions] の下で、[EC2 Action] をクリックします。
14. [Whenever this alarm] ドロップダウンリストで、[State is ALARM] を選択します。
15. [Take this action] ドロップダウンリストで、[Terminate this instance] をクリックします。
16. [Notification] をクリックし、[Send notification to] ドロップダウンリストにある既存の Amazon SNS トピックを選択するか、新しい Amazon SNS トピックを作成します。
17. 新しい Amazon SNS トピックを作成するには、[New list] を選択します。

[Send notification to] フィールドに、新しい Amazon SNS トピックの名前 ( 「`Terminate_EC2_Instance`」 など ) を入力し、[Email list] フィールドに、アラームが `ALARM` 状態に変化したときに通知を送信する E メールアドレスをカンマで区切って入力します。



#### Important

新しいトピックを作成する場合や E メールアドレスを既存のトピックに追加する場合は、追加する各 E メールアドレスにトピックの受信登録の確認メールが送信されます。お客様が確認メールに含まれたリンクをクリックし、受信登録を確認すると、新しい E メールアドレスに通知が送信されるようになります。

18. アラームの作成を完了するには、[Create Alarm] をクリックします。

## Amazon CloudWatch コンソールを使用して、トリガされたアラームとアクションの履歴を表示する

Amazon CloudWatch コンソールではアラームとアクションの履歴を表示できます。Amazon CloudWatch では過去 2 週間分のアラームとアクションの履歴を維持します。

トリガされたアラームとアクションの履歴を表示するには

1. Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
2. 必要ならば、リージョンを変更します。ナビゲーションバーから、インスタンスが動作しているリージョンを選択します。詳細については、[Regions and Endpoints](#) を参照してください。



3. ナビゲーションペインの [Alarms] をクリックします。
4. 上側のペインで、履歴を表示するアラームを選択します。
5. 下側のペインの [Details] タブには、最新の状態の遷移に加えて、時刻とメトリクス値が表示されます。
6. [History] タブをクリックすると、最新の履歴エントリが表示されます。

## CLI または API を使用して、インスタンスを停止または終了するアラームを作成する

AWS CLI または CloudWatch API を使用する場合、あるいは AWS SDK を API と一緒に使用する場合は、Amazon EC2 インスタンスメトリクスを使用して Amazon CloudWatch アラームを作成し、さらにアクション専用の Amazon リソース名前 (ARN) を使用してアクションを追加することができます。任意のアラーム状態にアクションを追加できます。また、アクションごとにリージョンを指定できます。このリージョンは、put-metric-alarm リクエストの送信先であるリージョンと一致する必要があります。

Action	ARN (リージョンを伴う)
<i>Stop</i>	arn:aws:automate:us-east-1:ec2:stop
<i>Terminate</i>	arn:aws:automate:us-east-1:ec2:terminate

Amazon CloudWatch API を AWS SDK と合わせて使用する詳細については、[サンプルコード&ライブラリ](#)を参照してください。



### Note

AWS Identity and Access Management (IAM) アカウントを使用してアラームを作成または変更する場合は、Amazon EC2 許可となる `ec2:DescribeInstanceStatus`、`ec2:DescribeInstances`、`ec2:StopInstances`、およびアラームアクションを実行するための `ec2:TerminateInstances` が必要です。Amazon CloudWatch にする読み取り/書き込み許可はあっても、Amazon EC2 に対しては許可がない場合でもアラームは作成できます。しかし、Amazon EC2 インスタンスで停止または終了アクションは実行されません。ただし、関連付けられている Amazon EC2 API の使用許可が後で付与される場合、以前に作成したアラームアクションは実行されるようになります。IAM 許可の詳細については、[IAM を使用するの Permissions and Policies](#) を参照してください。

IAM ロール (例えば、Amazon EC2 インスタンスプロファイル) を使用している場合は、アラームアクションを使用してインスタンスを停止または終了することはできません。ただし、アラームの状態を確認して、Amazon SNS 通知や Auto Scaling ポリシーなど、その他のアクションを実行することはできます。

AWS Security Token Service (AWS STS) を用いて許可された一時的な認証情報を使用している場合は、アラームアクションを用いて Amazon EC2 インスタンスを停止または終了することはできません。

CLI を使用してインスタンスを停止するアラームを作成するには

arn:aws:automate:us-east-1:ec2:stop ARN を使用して、Amazon EC2 インスタンスを停止することができます。次の例は、CPU 使用率の平均が 24 時間にわたって 10 パーセント未満である場合、インスタンスを停止する方法を示したものです。

- コマンドプロンプトで、次のように入力します。

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-description
  "Stop the instance when it is idle for a day" --namespace "AWS/EC2" --di
  mensions Name=InstanceId,Value=i-abc123 --statistic Average --metric-name
  CPUUtilization --comparison-operator LessThanThreshold --threshold 10 --
  period 86400 --evaluation-periods 4 --alarm-actions arn:aws:automate:us-
  east-1:ec2:stop
```

CLI を使用してインスタンスを終了するアラームを作成するには

- コマンドプロンプトで、次のように入力します。

```
% aws cloudwatch put-metric-alarm --alarm-name my-Alarm --alarm-description
  "Terminate the instance when it is idle for a day" --namespace "AWS/EC2"
  --dimensions Name=InstanceId,Value=i-abc123" --statistic Average --metric-
  name CPUUtilization --comparison-operator LessThanThreshold --threshold 1
  --period 86400 --evaluation-periods 4 -- alarm-actions arn:aws:automate:us-
  east-1:ec2:terminate
```

API を使用して、アラームを作成およびインスタンスを停止するには

次のリクエストの例では、Amazon EC2 インスタンスを停止するアラームの作成方法を示します。

- 次のリクエストを作成します。

```
http://monitoring.amazonaws.com/

?SignatureVersion=2

&Action=PutMetricAlarm

&Version=2009-05-15

&Namespace=AWS/EC2

&MetricName=CPUUtilization

&Dimension.member.1.Name=instance-id

&Dimension.member.1.Value=i-abc123
```

```
&Period=86400

&Statistic=Average

&AlarmName=Stop-EC2-Instance

&ComparisonOperator=LessThanThreshold

&Threshold=10

&EvaluationPeriods=4

&StartTime=2009-01-16T00:00:00

&EndTime=2009-01-16T00:02:00

&Timestamp=2009-01-08-18

&AWSAccessKeyId=XXX YOUR ACCESS KEY XXX

&Signature=%XXX YOUR SIGNATURE XXX%3D

&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:stop
```

API を使用して、アラームを作成し、インスタンスを終了するには

次のリクエストの例では、Amazon EC2 インスタンスを終了するアラームの作成方法を示します。

- 次のリクエストを作成します。

```
http://monitoring.amazonaws.com/

?SignatureVersion=2

&Action=PutMetricAlarm

&Version=2009-05-15

&Namespace=AWS/EC2

&MetricName=CPUUtilization

&Dimension.member.1.Name=instance-id

&Dimension.member.1.Value=i-abc123

&Period=86400

&Statistic=Average

&AlarmName=Terminate-EC2-Instance

&ComparisonOperator=LessThanThreshold

&Threshold=10
```

```
&EvaluationPeriods=4

&StartTime=2009-01-16T00:00:00

&EndTime=2009-01-16T00:02:00

&Timestamp=2009-01-08-18

&AWSSecretKeyId=XXX YOUR SECRET KEY XXX

&Signature=%XXX YOUR SIGNATURE XXX%3D

&AlarmActions.member.1=arn:aws:automate:us-east-1:ec2:terminate
```

## Amazon CloudWatch アラームアクションシナリオ

Amazon Elastic Compute Cloud ( Amazon EC2 ) コンソールを使用すると、特定の条件が満たされたときに Amazon EC2 インスタンスを停止または終了するアラームアクションを作成できます。アラームアクションが設定されるコンソールページを示す次のスクリーンキャプチャで、設定に番号を振りしました。また、適切なアクションの作成をサポートするため、後続のシナリオ内の設定にも番号を振りしました。

**Create Alarm for** Cancel X

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.  
To create an alarm, first choose whom to notify and then define when the notification should be sent.

**Send a notification to:**  create topic

**Take the action:**  Stop  Terminate this instance.

**Whenever:**  **2** of  **3**

**Is:**  **4**  **5** Percent

**For at least:**  **6** consecutive period(s) of  **7**

**Name this alarm:**  cancel

Cancel Create Alarm

**CPUUtilization (Percent)**

40  
30  
20  
10  
0

12/5 14:00 12/5 16:00 12/5 18:00

### シナリオ 1: アイドル状態にある開発用およびテスト用のインスタンスを停止する

ソフトウェアの開発またはテストに使用されるインスタンスが1時間以上アイドル状態になっている場合、該当するインスタンスを停止するアラームを作成します。

設定	値
<b>1</b>	停止
<b>2</b>	最大
<b>3</b>	CPUUtilization

設定	値
4	<=
5	10%
6	60 分
7	1

### シナリオ 2: アイドル状態のインスタンスを停止する

インスタンスが 24 時間にわたってアイドル状態になっている場合、インスタンスを停止し、E メールを送信するアラームを作成します。

設定	値
1	停止および E メール
2	平均
3	CPUUtilization
4	<=
5	5%
6	60 分
7	24

### シナリオ 3: トラフィック量が異常に多いウェブサーバーを停止する

インスタンスの 1 日あたりのアウトバウンドネットワークトラフィックが 10 GB を超えた場合、E メールを送信するアラームを作成します。

設定	値
1	E メール
2	合計
3	NetworkOut
4	>
5	10 GB

設定	値
6	1 日
7	1

#### シナリオ 4: トラフィック量が異常に多いウェブサーバーを停止する

アウトバウンドトラフィックが 1 時間あたり 1 GB を超えた場合、インスタンスを停止してテキストメッセージ (SMS) を送信するアラームを作成します。

設定	値
1	SMS を停止および送信する
2	合計
3	NetworkOut
4	>
5	1 GB
6	1 時間
7	1

#### シナリオ 5: メモリリークが発生しているインスタンスを停止する

メモリの使用率が 90 パーセント以上になった場合、インスタンスを停止するアラームを作成します。トラブルシューティングのためのアプリケーションログを取得できます。



##### Note

MemoryUtilization メトリックスはカスタムメトリックスです。MemoryUtilization メトリックスを使用するには、[Amazon EC2 インスタンスのモニタリングスクリプト \(p. 451\)](#) をインストールする必要があります。

設定	値
1	停止
2	最大
3	MemoryUtilization
4	>=

設定	値
5	90%
6	1 分
7	1

### シナリオ 6: 障害のあるインスタンスを停止する

ステータスチェック ( 5 分間隔で実行 ) に 3 回続けて失敗したインスタンスを停止するアラームを作成します。

設定	値
1	停止
2	平均
3	StatusCheckFailed_System
4	>=
5	1
6	15 分
7	1

### シナリオ 7: バッチ処理ジョブが完了したときにインスタンスを終了する

バッチジョブを実行するインスタンスが結果データを送信しなくなったときに、そのインスタンスを終了するアラームを作成します。

設定	値
1	終了
2	最大
3	NetworkOut
4	<=
5	100,000 バイト
6	5 分

設定	値
7	1

また、前述シナリオは、Amazon CloudWatch コンソールを使用して実行することもできます。Amazon EC2 コンソールの付番済みの設定および前に説明したシナリオに対応するように、Amazon CloudWatch コンソールでの設定にも番号を振りました。したがって、比較しながら、適切なアクションを伴うアラームを作成できます。

## Amazon EC2 インスタンスのモニタリングスクリプト

### Abstract

Amazon EC2 の Linux または Windows インスタンスのモニタリングスクリプトを使用して、Amazon CloudWatch カスタムメトリックスを作成し利用します。

Amazon CloudWatch Monitoring Scripts for Amazon Elastic Compute Cloud ( Amazon EC2 ) の Linux ベースおよび Windows ベースのインスタンスは、Amazon CloudWatch カスタムメトリックスを作成/利用する方法を示しています。Linux 用のスクリプトは Perl スクリプトのサンプルで、Linux インスタンスに関するメモリ、スワップ、およびディスクスペースの使用状況メトリックスをレポートする、完全に機能する例で構成されます。Windows 用のスクリプトは PowerShell スクリプトのサンプルで、Windows インスタンスに関する、メモリ、ページファイル、およびディスクスペースの使用状況メ

リックスをレポートする、完全に機能する例で構成されます。Linux 用および Windows 用の CloudWatch Monitoring Scripts は、アマゾンウェブサービス (AWS) サンプルコードライブラリからダウンロードして、Linux ベースまたは Windows ベースのインスタンスにインストールできます。



#### Important

これらのスクリプトは、サンプルとして提供されています。そのため、「この状態のまま」スクリプトは提供され、サポートはされません。



#### Note

これらのスクリプトの利用については、カスタムメトリックスの通常の Amazon CloudWatch 無料利用枠の量と利用料金が適用されます。詳細については、[Amazon CloudWatch 製品ページ](#)を参照してください。

#### Topics

- [Amazon CloudWatch Monitoring Scripts for Linux \(p. 452\)](#)
- [Amazon CloudWatch Monitoring Scripts for Windows \(p. 458\)](#)

## Amazon CloudWatch Monitoring Scripts for Linux

### Abstract

Amazon EC2 の Linux インスタンスのモニタリングスクリプトを使用して、Amazon CloudWatch カスタムメトリックスを作成し利用します。

Amazon CloudWatch Monitoring Scripts for Linux は、Amazon CloudWatch カスタムメトリックスを作成し利用する方法を示した Perl スクリプトのサンプル集です。このスクリプトは、Elastic Compute Cloud (Amazon EC2) Linux インスタンスのメモリ、スワップ、およびディスクスペースの使用状況メトリックスをレポートする、完全に機能する例で構成されます。

#### Topics

- [前提条件 \(p. 453\)](#)
- [使用開始 \(p. 453\)](#)
- [スクリプトの使用 \(p. 454\)](#)
- [AWS マネジメントコンソールでのカスタムメトリックスの表示 \(p. 458\)](#)

[Amazon CloudWatch Monitoring Scripts for Linux](#) は、AWS のサンプルコードライブラリからダウンロードできます。CloudWatchMonitoringScripts-v1.1.0.zip パッケージには、以下のファイルが含まれています。

- CloudWatchClient.pm – 他のスクリプトからの Amazon CloudWatch の呼び出しを簡易化する共有 Perl モジュールです。
- mon-put-instance-data.pl – Amazon EC2 インスタンスでシステムメトリックス (メモリ、スワップ、ディスクスペースの使用状況) を収集し、Amazon CloudWatch に送信します。
- mon-get-instance-stats.pl – Amazon CloudWatch に対してクエリを実行し、このスクリプトの実行対象である EC2 インスタンスの最新の使用状況統計を表示します。
- – お客様のアクセスキー ID とシークレットアクセスキーを格納する AWS 認証情報のファイルテンプレートです。
- [LICENSE.txt] – Apache 2.0 ライセンスが記載されたテキストファイルです。
- [NOTICE.txt] – 著作権情報です。

これらのモニタリングスクリプトは、Linux オペレーティングシステムを実行している Amazon EC2 インスタンスで使用することを目的として記述されています。これらのスクリプトは、以下の Amazon マシンイメージ (AMI) の 32 ビットおよび 64 ビットバージョンでテスト済みです。

- Amazon Linux 2011.09.02
- Red Hat Enterprise Linux 6.3
- Ubuntu Server 12.04
- SUSE Linux Enterprise Server 11

## 前提条件

Linux のバージョンによっては、追加の手順を実行する必要があります。

### Amazon Linux AMI

Amazon Linux AMI バージョン 2014.03 以降を実行している場合は、スクリプトが動作していることをモニタリングするために、追加の Perl モジュールをいくつか追加する必要があります。サーバーを設定するには、次の手順に従います。

- Amazon Linux AMI インスタンスにログオンし、以下のパッケージをインストールします。

```
sudo yum install perl-Switch perl-Sys-Syslog perl-LWP-Protocol-https
```

### Red Hat Enterprise Linux または SUSE Linux

Red Hat Enterprise Linux または SUSE Linux Enterprise Server を実行している場合は、以下の手順を実行してサーバーを設定してください。

1. コマンドプロンプトで `cd /` と入力して、ルートディレクトリに移動します。
2. すべてのプロンプトで `perl -MCPAN -e shell` と入力し、その後 `yes` と入力します。
3. `cpan[1]>` メッセージで `install Bundle::LWP5_837 LWP` と入力して、LWP バンドルをインストールし LWP バージョン 6.03 に更新します。すべてのメッセージで `yes` と入力します。

### Ubuntu Server

Ubuntu Server を実行している場合は、以下の手順を実行してサーバーを設定してください。

- Ubuntu Server インスタンスにログオンし、以下のパッケージをインストールします。

```
sudo apt-get install unzip libwww-perl libcrypt-ssleay-perl
```

Amazon EC2 Linux インスタンスに接続する方法については、*Amazon Elastic Compute Cloud ユーザーガイド* の [インスタンスへの接続](#) を参照してください。

## 使用開始

次のステップでは、CloudWatch モニタリングスクリプトをダウンロードして解凍し、EC2 Linux インスタンスで設定する方法を説明しています。

スクリプトをダウンロード、インストール、設定するには

- コマンドプロンプトを開き、スクリプトを保存したいフォルダに移動し、以下を入力します。

```
wget http://ec2-downloads.s3.amazonaws.com/cloudwatch-samples/CloudWatchMonitoringScripts-v1.1.0.zip
unzip CloudWatchMonitoringScripts-v1.1.0.zip
rm CloudWatchMonitoringScripts-v1.1.0.zip
cd aws-scripts-mon
```

## スクリプトの使用

### mon-put-instance-data.pl

このスクリプトは、現在のシステムのメモリ、スワップ、およびディスクスペースの使用状況データを収集します。その後、収集したデータをカスタムメトリックスとしてレポートするため、Amazon CloudWatch に対してリモートコールを実行します。

#### オプション

名前	説明
<code>--mem-util</code>	パーセンテージで表した MemoryUtilization メトリックスを収集し送信します。このオプションでは、アプリケーションとオペレーティングシステムによって割り当てられたメモリのみをレポートし、キャッシュとバッファ内のメモリは除外します。
<code>--mem-used</code>	メガバイトで表した MemoryUsed メトリックスを収集し送信します。このオプションでは、アプリケーションとオペレーティングシステムによって割り当てられたメモリのみをレポートし、キャッシュとバッファ内のメモリは除外します。
<code>--mem-avail</code>	メガバイトで表した MemoryAvailable メトリックスを収集し送信します。このオプションでは、アプリケーションとオペレーティングシステムで使用可能なメモリについてレポートします。
<code>--swap-util</code>	パーセンテージで表した SwapUtilization メトリックスを収集し送信します。
<code>--swap-used</code>	メガバイトで表した SwapUsed メトリックスを収集し送信します。
<code>--disk-path=PATH</code>	レポートの対象とするディスクを選択します。 PATHでは、レポートする必要があるファイルシステムのマウントポイントか、またはマウントポイント上でのファイルの場所を指定することができます。複数のディスクを選択する場合は、それぞれ <code>--disk-path=PATH</code> を指定します。 / および /home にマウントされたファイルシステムのディスクを選択するには、以下のパラメータを使用します。 <code>--disk-path=/</code> <code>--disk-path=/home</code>
<code>--disk-space-util</code>	選択したディスクについて、DiskSpaceUtilization メトリックを収集し送信します。メトリックはパーセンテージでレポートされません。

名前	説明
<code>--disk-space-used</code>	<p>選択したディスクについて、DiskSpaceUsed メトリックを収集し送信します。メトリックは、デフォルトにより、ギガバイトでレポートされます。</p> <p>Linux オペレーティングシステムにディスクスペースが予約されるため、使用されているディスクスペースと利用可能なディスクスペースを合算しても、ディスクスペースの合計と一致しない場合があります。</p>
<code>--disk-space-avail</code>	<p>選択したディスクについて、DiskSpaceAvailable メトリックを収集し送信します。メトリックはギガバイトでレポートされます。</p> <p>Linux オペレーティングシステムにディスクスペースが予約されるため、使用されているディスクスペースと利用可能なディスクスペースを合算しても、ディスクスペースの合計と一致しない場合があります。</p>
<code>--memory-units=UNITS</code>	<p>メモリの使用状況をレポートする際の単位を指定します。指定しない場合は、メモリはメガバイトでレポートされます。UNITS に指定できるのは、バイト、キロバイト、メガバイト、またはギガバイトです。</p>
<code>--disk-space-units=UNITS</code>	<p>ディスクスペースの使用状況をレポートする際の単位を指定します。指定しない場合は、ディスクスペースはギガバイトでレポートされます。UNITS に指定できるのは、バイト、キロバイト、メガバイト、またはギガバイトです。</p>
<code>--aws-credential-file=PATH</code>	<p>AWS 認証情報が格納されているファイルの場所を指定します。このパラメータは、<code>--aws-access-key-id</code> および <code>--aws-secret-key</code> パラメータとともに使用できません。</p>
<code>--aws-access-key-id=VALUE</code>	<p>AWS アクセスキーID を指定して、発信者を特定するために指定します。<code>--aws-secret-key</code> オプションとともに使用する必要があります。このオプションを <code>--aws-credential-file</code> パラメータとともに使用しないでください。</p>
<code>--aws-secret-key=VALUE</code>	<p>AWS のシークレットアクセスキーを指定して、CloudWatch へのリクエストへの署名に使用します。<code>--aws-access-key-id</code> オプションとともに使用する必要があります。このオプションを <code>--aws-credential-file</code> パラメータとともに使用しないでください。</p>
<code>--verify</code>	<p>メトリックスを収集するスクリプトのテストを実行し、完全な HTTP リクエストを準備します。ただし、実際に CloudWatch コールを実行して、データをレポートしません。また、このオプションは、認証情報が提供されているかどうかもチェックします。このオプションを冗長モードで実行すると、CloudWatch に送信されるメトリックスを出力します。</p>
<code>--from-cron</code>	<p>cron からスクリプトを呼び出す際に、このオプションを使用します。このオプションを使用すると、すべての診断出力が抑制されますが、エラーメッセージはユーザーアカウントのローカルシステムログに送信されます。</p>
<code>--verbose</code>	<p>スクリプトが実行する内容についての詳細情報を表示します。</p>
<code>--help</code>	<p>使用状況情報を表示します。</p>

名前	説明
<code>--version</code>	スクリプトのバージョン番号を表示します。

## 例

以下の例では、有効な AWS 認証情報を使用して `awscreds.conf` ファイルが更新されていることを前提としています。`awscreds.conf` ファイルを使用しない場合は、`--aws-access-key-id` および `--aws-secret-key` 引数を使用して、認証情報を提供してください。

CloudWatch にデータを投稿せずに、単純なテストを実行するには

- 次のコマンドを実行します。

```
./mon-put-instance-data.pl --mem-util --verify --verbose
```

利用可能なメモリメトリックスをすべて収集して CloudWatch に送信するには

- 次のコマンドを実行します。

```
./mon-put-instance-data.pl --mem-util --mem-used --mem-avail
```

CloudWatch にレポートするメトリックスの cron スケジュールを設定するには

- 以下のコマンドを使用して、`crontab` の編集を開始します。

```
crontab -e
```

- 以下のコマンドを追加して、メモリとディスクスペースの使用状況を5分ごとに CloudWatch にレポートします。

```
*/5 * * * * ~/aws-scripts-mon/mon-put-instance-data.pl --mem-util --disk-space-util --disk-path=/ --from-cron
```

スクリプトがエラーを検出すると、スクリプトはシステムログにエラーメッセージを書き込みます。

## mon-get-instance-stats.pl

このスクリプトは CloudWatch に対してクエリを実行し、メモリ、スワップ、およびディスクスペースのメトリックスの統計を要求します。対象期間は直近の時間数で指定します。これらのデータは、スクリプトの実行対象である Amazon EC2 インスタンスに提供されます。

## オプション

名前	説明
<code>--recent-hours=N</code>	レポート対象期間とする直近の時間数を <code>N</code> で指定します。 <code>N</code> は整数です。

名前	説明
<code>--aws-credential-file=PATH</code>	AWS 認証情報が格納されているファイルの場所を指定します。
<code>--aws-access-key-id=VALUE</code>	AWS アクセスキー ID を指定して、発信者を特定するために指定します。 <code>--aws-secret-key</code> オプションとともに使用する必要があります。このオプションを <code>--aws-credential-file</code> オプションとともに使用しないでください。
<code>--aws-secret-key=VALUE</code>	AWS のシークレットアクセスキーを指定して、CloudWatch へのリクエストへの署名に使用します。 <code>--aws-access-key-id</code> オプションとともに使用する必要があります。このオプションを <code>--aws-credential-file</code> オプションとともに使用しないでください。
<code>--verify</code>	メトリクスを収集するスクリプトのテストを実行し、完全な HTTP リクエストを準備します。ただし、実際に CloudWatch コールを実行して、データをレポートしません。また、このオプションは、認証情報が提供されているかどうかもチェックします。このオプションを冗長モードで実行すると、CloudWatch に送信されるメトリクスを出力します。
<code>--verbose</code>	スクリプトが実行する内容についての詳細情報を表示します。
<code>--help</code>	使用状況情報を表示します。
<code>--version</code>	スクリプトのバージョン番号を表示します。

## 例

直近 12 時間の使用状況統計を取得するには

- 次のコマンドを実行します。

```
mon-get-instance-stats.pl --recent-hours=12
```

以下の出力例のような応答が返されます。

```
Instance metric statistics for the last 12 hours.

CPU Utilization
  Average: 1.06%, Minimum: 0.00%, Maximum: 15.22%

Memory Utilization
  Average: 6.84%, Minimum: 6.82%, Maximum: 6.89%

Swap Utilization
  Average: N/A, Minimum: N/A, Maximum: N/A

Disk Space Utilization on /dev/xvda1 mounted as /
  Average: 9.69%, Minimum: 9.69%, Maximum: 9.69%
```

## AWS マネジメントコンソールでのカスタムメトリックスの表示

mon-put-instance-data.pl スクリプトを正常に呼び出した場合、AWS マネジメントコンソールを使用して、投稿したカスタムメトリックスを Amazon CloudWatch コンソール内で表示できます。

カスタムメトリックスを表示するには

1. 先ほど説明したように、mon-put-instance-data.pl を実行します。
2. AWS マネジメントコンソールにサインインし、Amazon CloudWatch コンソール ( <https://console.aws.amazon.com/cloudwatch/> ) を開きます。
3. [View Metrics] をクリックします。
4. [Viewing] ドロップダウンリストに、スクリプトにより投稿されたカスタムメトリックスが System/Linux のプレフィックス付きで表示されます。

## Amazon CloudWatch Monitoring Scripts for Windows

### Abstract

Amazon EC2 Windows インスタンスのモニタリングスクリプトを使用して、Amazon CloudWatch カスタムメトリックスを作成し利用します。

Amazon CloudWatch Monitoring Scripts for Windows は、Amazon CloudWatch カスタムメトリックスを作成し利用する方法を示した PowerShell スクリプトのサンプルです。このスクリプトは、Elastic Compute Cloud ( Amazon EC2 ) Windows インスタンスのメモリ、ページファイル、およびディスクスペースの使用状況メトリックスをレポートする、完全に機能する例で構成されます。

### Topics

- [使用開始 \(p. 459\)](#)
- [スクリプトの使用 \(p. 459\)](#)

[Amazon CloudWatch Monitoring Scripts for Microsoft Windows Server](#) は、アマゾン ウェブ サービス ( AWS ) のサンプルコードライブラリからダウンロードできます。AmazonCloudWatchMonitoringWindows.zip パッケージには、以下のファイルが含まれています。

- [mon-put-metrics-mem.ps1] – Amazon EC2 Windows インスタンスでシステムメトリックス ( メモリ、ページファイルの使用状況 ) を収集し、Amazon CloudWatch に送信します。
- [mon-put-metrics-disk.ps1] – EC2; インスタンスでシステムメトリックス ( ディスクスペースの使用状況 ) を収集し、Amazon CloudWatch に送信します。
- [mon-put-metrics-perfmon.ps1] – EC; インスタンスで PerfMon カウンタを収集し、Amazon CloudWatch に送信します。
- [mon-get-instance-stats.ps1] – CWLong; に対してクエリを実行し、このスクリプトの実行対象である EC2 インスタンスの最新の使用状況統計を表示します。
- [awscreds.conf] – ユーザーのアクセスキー ID とシークレットアクセスキーを格納する AWS 認証情報のファイルテンプレートです。
- [LICENSE.txt] – Apache 2.0 ライセンスが記載されたテキストファイルです。
- [NOTICE.txt] – 著作権情報です。

これらのモニタリングスクリプトは、Microsoft Windows Server を実行している Amazon EC2 インスタンスで使用することを目的として記述されています。これらのスクリプトは、以下の Amazon マシンイメージ ( AMI ) の 32 ビットおよび 64 ビットバージョンでテスト済みです。

- Windows Server 2003 R2
- Windows Server 2008
- Windows Server 2008 R2

## 使用開始

次のステップでは、Amazon CloudWatch モニタリングスクリプトをダウンロードして解凍し、Amazon EC2 Windows インスタンスで設定する方法を説明しています。

スクリプトをダウンロード、インストール、設定するには

1. Amazon EC2 Windows インスタンスに接続します。Amazon EC2 Windows インスタンスに接続する方法については、*Amazon Elastic Compute Cloud ユーザーガイド* の [Connecting to Windows Instances](#) を参照してください。
2. [AWS SDK for .NET](#) をダウンロードし、監視対象とする EC2 インスタンスにインストールします。
3. [Amazon CloudWatch Monitoring Scripts for Microsoft Windows Server](#) が格納されている .zip ファイルを EC2 インスタンスにダウンロードし、お好みの場所で解凍します。
4. 以前ダウンロードした `awscreds.conf` ファイルを更新してください。このファイルのコンテンツは、以下の形式を使用する必要があります。

```
AWSAccessKeyId=YourAccessKeyID
```

```
AWSSecretKey=YourSecretAccessKey
```



### Note

認証情報のファイルを既に作成済みの場合、この手順はオプションになります。既存のファイルを使用する場合は、スクリプトを呼び出すときにコマンドラインでそのファイルの場所を指定します。または、環境変数 `AWS_CREDENTIAL_FILE` を設定して、AWS 認証情報で該当するファイルを参照できます。

認証情報にアクセスするには、以下の手順を実行します。

ベストプラクティスとして、ルートの認証情報は使用しないでください。代わりに、Amazon CloudWatch のオペレーションのみに制限するポリシーを使用して IAM ( Identity and Access Management ) ユーザーを作成してください。詳細については、[Controlling User Access to Your AWS Account](#) を参照してください。

## スクリプトの使用

### mon-put-metrics-mem.ps1

このスクリプトは、現在のシステムでのメモリおよびページファイルの使用状況データを収集します。その後、収集したデータをカスタムメトリックスとしてレポートするため、Amazon CloudWatch に対してリモートコールを実行します。

### オプション

名前	説明
<code>-mem_util</code>	パーセンテージで表した MemoryUtilization メトリックスを収集し送信します。このオプションでは、アプリケーションとオペレーティングシステムによって割り当てられたメモリのみをレポートし、キャッシュとバッファ内のメモリは除外します。

名前	説明
<code>-mem_used</code>	メガバイトで表した MemoryUsed メトリックスを収集し送信します。このオプションでは、アプリケーションとオペレーティングシステムによって割り当てられたメモリのみをレポートし、キャッシュとバッファ内のメモリは除外します。
<code>-mem_avail</code>	メガバイトで表した MemoryAvailable メトリックスを収集し送信します。このオプションでは、アプリケーションとオペレーティングシステムで使用可能なメモリについてレポートします。
<code>-page_util</code>	パーセンテージで表した PageUtilization メトリックスを収集し送信します。PageUtilization は、Windows インスタンスのページファイルごとにレポートされます。
<code>-page_used</code>	メガバイトで表した PageUsed メトリックスを収集し送信します。
<code>-page_avail</code>	すべてのディスクについて、ページファイル内で利用可能なスペースをレポートします。
<code>-memory_units UNITS</code>	メモリの使用状況をレポートする際の単位を指定します。指定しない場合は、メモリはメガバイトでレポートされます。UNITS に指定できるのは、バイト、キロバイト、メガバイト、またはギガバイトです。
<code>-aws_credential_file=PATH</code>	AWS 認証情報が格納されているファイルの場所を指定します。このパラメータは、 <code>-aws_access_id</code> および <code>-aws_secret_key</code> パラメータとともに使用できません。
<code>-aws_access_id=VALUE</code>	AWS アクセスキー ID を指定して、発信者を特定するために使用します。 <code>-aws_secret_key</code> オプションとともに使用する必要があります。このオプションを <code>-aws_credential_file</code> オプションとともに使用しないでください。
<code>-aws_secret_key=VALUE</code>	AWS のシークレットアクセスキーを指定して、Amazon CloudWatch へのリクエストへの署名に使用します。 <code>-aws_access_key_id</code> オプションとともに使用する必要があります。このオプションを <code>-aws_credential_file</code> オプションとともに使用しないでください。
<code>-whatif</code>	メトリックスを収集するスクリプトのテストを実行します。ただし、実際に CloudWatch コールを実行して、データをレポートしません。また、このオプションは、認証情報が提供されているかどうかもチェックします。
<code>-from_scheduler</code>	タスクスケジューラからスクリプトを呼び出す際に、このオプションを使用します。このオプションを使用すると、すべての診断出力が抑制されますが、エラーメッセージはログファイルに送信されません。
<code>-verbose</code>	スクリプトが実行する内容についての詳細情報を表示します。
<code>Get-help mon-put-metrics-mem.ps1</code>	使用状況情報を表示します。
<code>-version</code>	スクリプトのバージョン番号を表示します。

名前	説明
<code>-logfile</code>	logfile はエラーメッセージをログするために使用します。これは、「 <code>-from_scheduler</code> 」オプションとともに使用します。logfile に値を指定しない場合、デフォルトにより、スクリプトと同じ名前に <code>.log</code> 拡張子を付けたファイルが作成されます。

## 例

以下の例では、有効な AWS 認証情報を使用して `awscreds.conf` ファイルが更新されていることを前提としています。`awscreds.conf` ファイルを使用しない場合は、`-aws_access_id` および `-aws_secret_key` 引数を使用して、認証情報を提供してください。

利用可能なすべてのメモリメトリックスを、インラインアクセス ID とシークレットキーを使用して収集し、データを CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-mem.ps1 -aws_access_id ThisIsMyAccessKey -aws_secret_key ThisIsMySecretKey -mem_util -mem_avail -page_avail -page_used -page_util -memory_units Megabytes
```

利用可能なすべてのメモリメトリックスを、認証情報ファイルを使用して収集し、データを CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-mem.ps1 -aws_credential_file C:\awscreds.conf -mem_util -mem_used -mem_avail -page_avail -page_used -page_util -memory_units Megabytes
```

利用可能なすべてのメモリメトリックスを、環境変数に格納されている認証情報ファイルを使用して収集し、データを CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-mem.ps1 -mem_util -mem_used -mem_avail -page_avail -page_used -page_util -memory_units Megabytes
```

## mon-put-metrics-disk.ps1

このスクリプトは、現在のシステムでのディスクスペースの使用状況データを収集します。その後、収集したデータをカスタムメトリックスとしてレポートするため、Amazon CloudWatch に対してリモートコールを実行します。

## オプション

名前	説明
<code>-disk_space_util</code>	選択したディスクについて、DiskSpaceUtilization メトリックを収集し送信します。メトリックはパーセンテージでレポートされます。
<code>-disk_space_used</code>	選択したディスクについて、DiskSpaceUsed メトリックを収集し送信します。メトリックはデフォルトでギガバイトでレポートされます。
<code>-disk_space_avail</code>	選択したディスクについて、DiskSpaceAvailable メトリックを収集し送信します。メトリックはギガバイトでレポートされます。
<code>-disk_space_units UNITS</code>	メモリの使用状況をレポートする際の単位を指定します。指定しない場合は、メモリはギガバイトでレポートされます。UNITS に指定できるのは、バイト、キロバイト、メガバイト、またはギガバイトです。
<code>-disk_drive</code>	レポートの対象とするドライブ文字を選択します。c および d ドライブのメトリックスをレポートするには、以下のオプションを使用します: <code>-disk_drive C:, D:</code> 。値はコンマで区切ります。
<code>-aws_credential_file PATH</code>	AWS 認証情報が格納されているファイルの場所を指定します。このパラメータは、 <code>-aws_access_id</code> および <code>-aws_secret_key</code> パラメータとともに使用できません。
<code>-aws_access_id VALUE</code>	AWS アクセスキー ID を指定して、発信者を特定するために使用します。 <code>-aws_secret_key</code> オプションとともに使用する必要があります。このオプションを <code>-aws_credential_file</code> オプションとともに使用しないでください。
<code>-aws_secret_key VALUE</code>	AWS のシークレットアクセスキーを指定して、Amazon CloudWatch へのリクエストへの署名に使用します。 <code>-aws_access_id</code> オプションとともに使用する必要があります。このオプションを <code>-aws_credential_file</code> オプションとともに使用しないでください。
<code>-whatif</code>	メトリックスを収集するスクリプトのテストを実行します。ただし、実際に CloudWatch コールを実行して、データをレポートしません。また、このオプションは、認証情報が提供されているかどうかもチェックします。
<code>-from_scheduler</code>	タスクスケジューラからスクリプトを呼び出す際に、このオプションを使用します。このオプションを使用すると、すべての診断出力が抑制されますが、エラーメッセージはログファイルに送信されます。
<code>-verbose</code>	スクリプトが実行する内容についての詳細情報を表示します。
<code>Get-help mon-put-metrics-disk.ps1</code>	使用状況情報を表示します。
<code>-version</code>	スクリプトのバージョン番号を表示します。

名前	説明
<code>-logfile</code>	logfile はエラーメッセージをログするために使用します。これは <code>-from_scheduler</code> オプションとともに使用します。logfile に値を指定しない場合、デフォルトにより、スクリプトと同じ名前に <code>.log</code> 拡張子を付けたデフォルトファイルが作成されます。

## 例

利用可能なすべてのディスクメトリックスを、インラインアクセス ID とシークレットキーを使用して収集し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-disk.ps1 -aws_access_id ThisIsMyAccessKey -aws_secret_key ThisIsMySecretKey -disk_space_util -disk_space_avail -disk_space_units Gigabytes
```

利用可能なすべてのディスクメトリックスを、認証情報ファイルを使用して収集し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-disk.ps1 -aws_credential_file C:\awscreds.conf -disk_drive C:, D: -disk_space_util -disk_space_used -disk_space_avail -disk_space_units Gigabytes
```

利用可能なすべてのディスクメトリックスを、環境変数に格納されている認証情報ファイルを使用して収集し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-disk.ps1 -disk_drive C:, D: -disk_space_util -disk_space_used -disk_space_avail -disk_space_units Gigabytes
```

## mon-put-metrics-perfmon.ps1

このスクリプトは、現在のシステムで PerfMon カウンタを収集します。その後、収集したデータをカスタムメトリックスとしてレポートするため、Amazon CloudWatch に対してリモートコールを実行します。

### オプション

名前	説明
<code>-processor_queue</code>	現在のプロセッサキューカウンタをレポートします。

名前	説明
<code>-pages_input</code>	メモリページ/入力メモリカウンタをレポートします。
<code>-aws_credential_file PATH</code>	AWS 認証情報が格納されているファイルの場所を指定します。このパラメータは、 <code>-aws_access_id</code> および <code>-aws_secret_key</code> パラメータとともに使用できません。
<code>-aws_access_id VALUE</code>	AWS アクセスキー ID を指定して、発信者を特定するために使用します。 <code>-aws_secret_key</code> オプションとともに使用する必要があります。このオプションを <code>-aws_credential_file</code> オプションとともに使用しないでください。
<code>-aws_secret_key VALUE</code>	AWS のシークレットアクセスキーを指定して、Amazon CloudWatch へのリクエストへの署名に使用します。 <code>-aws_access_id</code> オプションとともに使用する必要があります。このオプションを <code>-aws_credential_file</code> オプションとともに使用しないでください。
<code>-whatif</code>	メトリックスを収集するスクリプトのテストを実行します。ただし、実際に CloudWatch コールを実行して、データをレポートしません。また、このオプションは、認証情報が提供されているかどうかもチェックします。
<code>-from_scheduler</code>	タスクスケジューラからスクリプトを呼び出す際に、このオプションを使用します。このオプションを使用すると、すべての診断出力が抑制されますが、エラーメッセージはログファイルに送信されません。
<code>-verbose</code>	スクリプトが実行する内容についての詳細情報を表示します。
<code>Get-help mon-put-metrics-disk.ps1</code>	使用状況情報を表示します。
<code>-version</code>	スクリプトのバージョン番号を表示します。
<code>-logfile</code>	logfile はエラーメッセージをログするために使用します。これは <code>-from_scheduler</code> オプションとともに使用します。logfile に値を指定しない場合、デフォルトにより、スクリプトと同じ名前に <code>.log</code> 拡張子を付けたデフォルトファイルが作成されます。

## 例

スクリプト内の設定済みの PerfMon カウンタを、インラインアクセス ID とシークレットキーを使用して収集し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-perfmon.ps1 -aws_access_id ThisIsMyAccessKey -aws_secret_key ThisIsMySecretKey -pages_input -processor_queue
```

スクリプト内の設定済みの PerfMon カウンタを、認証情報ファイルを使用して収集し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-perfmon.ps1 -aws_credential_file C:\awscreds.conf -pages_input -processor_queue
```

スクリプト内の設定済みの PerfMon カウンタを、環境変数に格納されている認証情報ファイルを使用して収集し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-put-metrics-perfmon.ps1 -pages_input -processor_queue
```

プッシュするカウンタを Amazon CloudWatch に追加するには

1. スクリプトをメモ帳などのテキストエディタで開き、72行目にある以下のコメント付きのセクションを見つけます。

```
### Add More counters here.  
#$Counters.Add('\Memory\Cache Bytes','Bytes')  
#$Counters.Add('\localhost\physicaldisk(0 c:)\% disk time','Percent')
```



#### Note

最初のパラメータ ( \$Counters.Add など ) は PerfMon カウンタです。2 番目のパラメータ ( ('Memory\Cache Bytes','Bytes') など ) は、カウンタが提供するデータの単位です。

2. スクリプトを編集し、上記にあるように、独自の PerfMon カウンタを追加します。スクリプトにカスタム PerfMon カウンタを追加した後は、認証情報以外のパラメータを使用しなくても、スクリプトを実行できます。



#### Note

PerfMon カウンタを追加できるのは、コンピューター上のスクリプトのみです。PerfMon カウンタをテストするには、Get-Counter コマンドを使用します。詳細については、Microsoft TechNet ウェブサイトの [Get-Counter](#) を参照してください。

## mon-get-instance-stats.ps1

このスクリプトは Amazon CloudWatch に対してクエリを実行し、メモリ、ページファイル、およびディスクスペースのメトリックスの統計を要求します。対象期間は直近の時間数で指定します。これらのデータは、スクリプトの実行対象である Amazon EC2 インスタンスに提供されます。

## オプション

名前	説明
<code>-recent-hours N</code>	レポート対象期間とする直近の時間数を N で指定します。N は整数です。
<code>-aws_credential_file PATH</code>	AWS 認証情報が格納されているファイルの場所を指定します。このパラメータは、 <code>-aws_access_id</code> および <code>-aws_secret_key</code> パラメータとともに使用できません。
<code>-aws_access_id VALUE</code>	AWS アクセスキー ID を指定して、発信者を特定するために使用します。 <code>-aws_secret_key</code> オプションとともに使用する必要があります。このオプションを <code>-aws_credential_file</code> オプションとともに使用しないでください。
<code>-aws_secret_key VALUE</code>	AWS のシークレットアクセスキーを指定して、Amazon CloudWatch へのリクエストへの署名に使用します。 <code>-aws_access_id</code> オプションとともに使用する必要があります。このオプションを <code>-aws_credential_file</code> オプションとともに使用しないでください。
<code>-verbose</code>	スクリプトが実行する内容についての詳細情報を表示します。
<code>Get-help</code> <code>mon-get-instance-stat</code> <code>s.ps1</code>	使用状況情報を表示します。
<code>-version</code>	スクリプトのバージョン番号を表示します。

## 例

直近の 12 時間の使用状況統計を、インラインアクセス ID とシークレットキーを使用して取得し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\ mon-get-instance-stats.ps1 -aws_access_id
    ThisIsMyAccessKey -aws_secret_key ThisIsMySecretKey -re
cent_hours 12
```

直近の 12 時間の使用状況統計を、認証情報ファイルを使用して取得し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-get-instance-stats.ps1 -aws_credential_file C:\awscreds.conf -re
cent_hours 12
```

直近の 12 時間の使用状況統計を、環境変数に格納されている認証情報ファイルを使用して取得し、データを Amazon CloudWatch に送信するには

- 次のコマンドを実行します。

```
.\mon-get-instance-stats.ps1 -recent_hours 12
```

以下の出力例のような応答が返されます。

```
Assembly Loaded
Instance Metrics for last 12 hours.
CPU Utilization
Average: 4.69 % Maximum: 10.47 % Minimum: 1.16 %

Memory Utilization
Average: 14.45 % Maximum: 14.77 % Minimum: 14.38 %

pagefileUtilization(c:\pagefile.sys)
Average: 0.00 % Maximum: 0.00 % Minimum: 0.00 %

Volume Utilization C:
Average: 17.28 % Maximum: 17.28 % Minimum: 17.28 %

Volume Utilization D:
Average: 1.41 % Maximum: 1.41 % Minimum: 1.41 %

pagefileUtilization(f:\pagefile.sys)
Average: 0.00 % Maximum: 0.00 % Minimum: 0.00 %
pagefileUtilization(f:\pagefile.sys)
Average: 0 Maximum: 0 Minimum: 0

pagefileUtilization(f:\pagefile.sys)
Average: 0 Maximum: 0 Minimum: 0
```

## メトリックスレポートを Amazon CloudWatch に送信するためのタスクスケジューラの設定

Windows タスクスケジューラを使用して、メトリックスレポートを定期的に Amazon CloudWatch に送信することができます。

メトリックスレポートを Amazon CloudWatch に送信するためにタスクスケジューラを設定するには

1. Windows Server インスタンスで [Start] をクリックし、[Administrative Tools]、[Task Scheduler] の順にクリックします。
2. [Action] メニューの [Create Task] をクリックします。
3. [Create Task] ダイアログボックスの [General] タブにある [Name] ボックスにタスクの名前を入力し、[Run whether user is logged on or not] を選択します。
4. [Triggers] タブで [New] をクリックします。
5. [New Trigger] ダイアログボックスの [Settings] で [One time] を選択します。
6. [Advanced settings] の [Repeat task every] として、ドロップダウンメニューから [5 minutes] を選択します。
7. [for a duration of] ドロップダウンメニューから [Indefinitely] を選択し、[OK] をクリックします。



#### Note

この設定により、スクリプトを永久に5分おきに起動するトリガが作成されます。このタスクを特定の日数だけ実行させるように変更するには、[Expire] チェックボックスを使用します。

8. [Actions] タブで [New] をクリックします。
9. [Action] ドロップダウンメニューから [Start a program] を選択します。
10. [Settings] の [Program/script] ボックスに [Powershell.exe] と入力します。
11. [Add arguments (optional)] ボックスに 「-command "C:\scripts\mon-put-metrics-disk.ps1 -disk\_drive C:,d -disk\_space\_util -disk\_space\_units gigabytes -from\_scheduler -logfile C:\mylogfile.log"」 と入力して、[OK] をクリックします。
12. [Create Task] ダイアログボックスで [OK] をクリックします。

ユーザーアカウントを選択してこのタスクを実行した場合、タスクスケジューラによってユーザー認証情報を求められます。タスクを実行するアカウントのユーザー名とパスワードを入力し、[OK] をクリックします。



#### Note

使用する PerfMon カウンタが管理者権限を必要としない場合は、管理者アカウントではなく、システムアカウントを使用してこのタスクを実行できます。[Create Task] ダイアログボックスの [General] タブで、[Change User or Group] をクリックし、システムアカウントを選択します。

---

# ネットワークとセキュリティ

---

このセクションでは、Amazon EC2 に関連するネットワークとセキュリティの主要な機能について説明します。

## Topics

- [Amazon EC2 のキーペア](#) (p. 470)
- [Amazon EC2 セキュリティグループ](#) (p. 477)
- [Amazon EC2 のリソースに対するアクセスの制御](#) (p. 485)
- [Amazon EC2 と Amazon Virtual Private Cloud \( VPC \)](#) (p. 524)
- [Amazon EC2 インスタンスの IP アドレッシング](#) (p. 529)
- [Elastic IP アドレス \( EIP \)](#) (p. 540)
- [Elastic Network Interface \( ENI \)](#) (p. 545)
- [VPC 内の Linux インスタンスでの拡張ネットワーキングの有効化](#) (p. 559)

コマンドラインツールまたは API を使用して Amazon EC2 にアクセスする場合は、アクセスキー ID とシークレットアクセスキーが必要です。詳細については、「[How Do I Get Security Credentials?](#)」を参照してください

インスタンスは、プラットフォーム EC2-Classic または EC2-VPC で起動できます。EC2-Classic に起動されるインスタンスまたはデフォルト VPC には、パブリック IP アドレスが自動的に割り当てられません。デフォルト以外の VPC に起動されるインスタンスには、起動時にパブリック IP アドレスを割り当てることができます。EC2-Classic と EC2-VPC の詳細については、「[サポートされているプラットフォーム](#)」を参照してください。

インスタンスは、制御不能な理由によりエラーを起こしたり終了したりすることがあります。インスタンスが異常終了したために別のインスタンスを起動した場合は、そのインスタンスのパブリック IP アドレスは元の IP アドレスとは異なります。ただし、アプリケーションが静的な IP アドレスを必要とする場合は、*Elastic IP* アドレスを使用できます。

インスタンスにアクセスできるユーザーの制御にはセキュリティグループを使います。これは、着信ネットワークファイアウォール ( どのプロトコル、ポート、ソース IP 範囲にインスタンスへの到達を許可するかを指定する ) に似ています。複数のセキュリティグループを作成してそれぞれに異なるルールを割り当てることもできます。その後で、各インスタンスを1つまたは複数のセキュリティグループに割り当てます。どのトラフィックが各インスタンスに到達できるかは、ルールを使用して決定されます。セキュリティグループを設定することで、特定の IP アドレスや特定のセキュリティグループのみにインスタンスへのアクセスを許可することができます。

# Amazon EC2 のキーペア

## Abstract

パブリックキー暗号ペアを使用して、Amazon EC2 インスタンスのログイン情報の暗号化と復号化を行います。

Amazon EC2 はパブリックキー暗号ペアを使用して、ログイン情報の暗号化と復号化を行います。パブリックキー暗号ペアは、パブリックキーを使用してパスワードなどのデータを暗号化し、受信者はプライベートキーを使用してデータを復号化します。パブリックキーとプライベートキーは、キーペアと呼ばれます。

インスタンスにログインするには、キーペアを作成し、インスタンスを起動するときにキーペアの名前を指定して、インスタンスに接続するときにプライベートキーを指定する必要があります。Linux/Unix インスタンスにはパスワードがなく、キーペアを使用して SSH によってログインします。Windows インスタンスでは、キーペアを使用して管理者パスワードを取得してから、RDP を使用してログインします。

### キーペアを作成する

Amazon EC2 を使用してキーペアを作成できます。詳細については、[Amazon EC2 を使用してキーペアを作成する \(p. 471\)](#) を参照してください。

または、サードパーティ製のツールで、パブリックキーを Amazon EC2 にインポートすることもできます。詳細については、[独自のキーペアを Amazon EC2 にインポートする \(p. 472\)](#) を参照してください。

それぞれのキーペアには名前が必要です。覚えやすい名前を選択するようにしてください。Amazon EC2 は、キー名として指定した名前にパブリックキーを関連付けます。

Amazon EC2 はパブリックキーのみを保存し、お客様はプライベートキーを保存します。お客様のプライベートキーを持っていれば誰でもお客様のログイン情報を復号化できるので、プライベートキーを安全な場所に保存することが重要です。

Amazon EC2 が使用するキーは、1024-bit SSH-2 RSA キーです。リージョンあたり最大 5000 のキーペアを持つことができます。

### インスタンスを起動し、接続する

インスタンスを起動するときは、インスタンスへの接続に使用するキーペアの名前を指定する必要があります。インスタンスの起動時に既存のキーペアを指定しない場合、インスタンスに接続することはできません。インスタンスに接続するときは、インスタンスの起動時に指定したキーペアに対応するプライベートキーを指定する必要があります。Amazon EC2 ではプライベートキーのコピーが保持されないため、プライベートキーを失った場合、復元することはできません。Instance store-Backed インスタンスのプライベートキーを失った場合は、インスタンスにアクセスできなくなります。そのため、インスタンスを終了し、新しいキーペアを使用して、別のインスタンスを起動する必要があります。EBS-Backed インスタンスのプライベートキーを失った場合は、インスタンスへのアクセス権を回復することができます。詳細については、[プライベートキーを失った場合にインスタンスに接続する \(p. 475\)](#) を参照してください。

### Topics

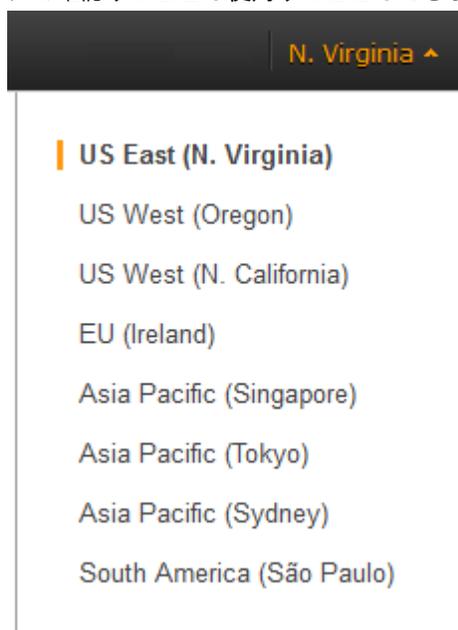
- [Amazon EC2 を使用してキーペアを作成する \(p. 471\)](#)
- [独自のキーペアを Amazon EC2 にインポートする \(p. 472\)](#)
- [キーペアのパブリックキーを取得する \(p. 473\)](#)
- [キーペアの削除 \(p. 474\)](#)
- [プライベートキーを失った場合にインスタンスに接続する \(p. 475\)](#)

## Amazon EC2 を使用してキーペアを作成する

Amazon EC2 コンソールまたはコマンドラインを使用して、キーペアを作成できます。

コンソールを使用してキーペアを作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーで、キーペアを生成するリージョンを選択します。お客様は場所に関係なく、使用できるリージョンをどれでも選択できます。この選択は重要です。複数の Amazon EC2 リソースはリージョン間で共有することができますが、キーペアの場合はそれができないからです。例えば、米国西部 (オレゴン) リージョン でキーペアを作成した場合、そのキーペアは別のリージョンで確認することも使用することもできません。



3. ナビゲーションペインで [Key Pairs] をクリックします。
4. [Create Key Pair] をクリックします。
5. [Create Key Pair] ダイアログボックスの [Key Pair Name] フィールドに新しいキーペアの名前を入力し、[Create] をクリックします。
6. ブラウザによって秘密キーファイルが自動的にダウンロードされます。ベースファイル名はキーペアの名前として指定した名前となり、ファイル名の拡張子は `.pem` となります。プライベートキーファイルを安全な場所に保存します。



### Important

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

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

```
$ chmod 400 my-key-pair.pem
```

コマンドラインを使用してキーペアを作成するには

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

- [create-key-pair](#) ( AWS CLI )
- [ec2-create-keypair](#) ( Amazon EC2 CLI )
- [New-EC2KeyPair](#) ( AWS Tools for Windows PowerShell )

## 独自のキーペアを Amazon EC2 にインポートする

前のセクションで説明したように、Amazon EC2 を使用してキーペアを作成すれば、インスタンスを起動できる状態になります。または、Amazon EC2 を使用してキーペアを作成する代わりに、サードパーティ製のツールで RSA キーペアを作成してから、パブリックキーを Amazon EC2 にインポートすることもできます。例えば、`ssh-keygen` ( 標準 OpenSSH インストールで提供されるツール ) を使用して、キーペアを作成できます。また、Java、Ruby、Python などのさまざまなプログラミング言語では、RSA キーペアの作成に使用できる標準ライブラリが提供されています。

Amazon EC2 では、次の形式を使用できます。

- OpenSSH パブリックキー形式 ( `~/.ssh/authorized_keys` の形式 )
- Base64 でエンコードされた DER 形式
- SSH パブリックキーファイル形式 ( [RFC4716](#) で指定 )

Amazon EC2 では、DSA キーを使用できません。キージェネレータが、RSA キーを作成するように設定されていることを確認してください。

サポートされている長さ: 1024、2048、および 4096。

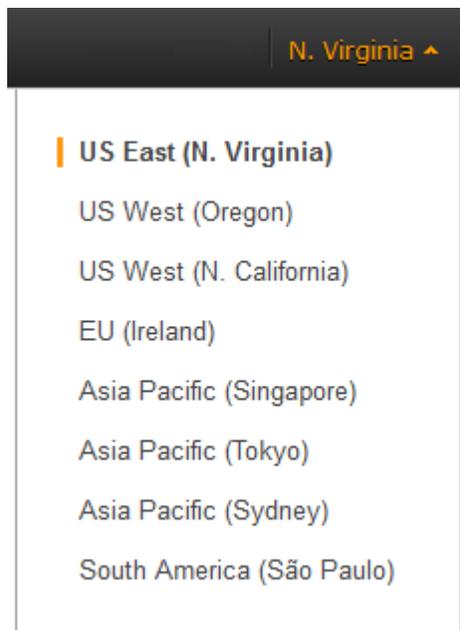
キーペアを作成するにはサードパーティ製のツールでキーペアを作成するには

1. 選択したサードパーティ製のツールでキーペアを生成します。
2. ローカルファイルにパブリックキーを保存します。例えば、`C:\keys\my-key-pair.pub` です。このファイル名の拡張子は重要ではありません。
3. `.pem` 拡張子を持つ別のローカルファイルにプライベートキーを保存します。例えば、`C:\keys\my-key-pair.pem` です。プライベートキーファイルを安全な場所に保存します。インスタンスと対応するプライベートキーの起動時には、毎回インスタンスに接続するたびに、キーペアの名前を入力する必要があります。

Amazon EC2 コンソールを使用してキーペアをインポートするには、次のステップを使用します ( 必要に応じて、[ec2-import-keypair](#) コマンドまたは [ImportKeyPair](#) アクションを使用してパブリックキーをインポートできます )。

パブリックキーをインポートするには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーで、キーペアを生成するリージョンを選択します。キーペアリソースはリージョン間で共有できないので、この選択は重要です。例えば、キーペアを米国西部 ( オレゴン ) リージョンにインポートした場合、別のリージョンでこのキーペアを表示または使用することはできません。



3. ナビゲーションペインで [Key Pairs] をクリックします。
4. [Import Key Pair] をクリックします。
5. [Import Key Pair] ダイアログボックスで [Browse] をクリックし、前に保存したパブリックキーファイルを選択します。[Key pair name] フィールドにキーペアの名前を入力し、[Import] をクリックします。

パブリックキーファイルをインポートしたら、次のように Amazon EC2 コンソールを使用して、キーペアが正しくインポートされたかどうかを確認できます。(必要に応じて、[ec2-describe-keypairs](#) コマンドまたは [DescribeKeyPairs](#) アクションを使用してキーペアをリストできます。)

キーペアがインポートされたことを確認するには

1. ナビゲーションバーから、キーペアを作成したリージョンを選択します。
2. ナビゲーションペインで [Key Pairs] をクリックします。
3. インポートしたキーペアが、表示されているキーペアのリストに含まれていることを確認します。

## キーペアのパブリックキーを取得する

Linux/Unix インスタンスでは、パブリックキーコンテンツは `~/.ssh/authorized_keys` 内のエントリに配置されます。この配置は起動時に行われ、パスワードを使用しなくても、安全にインスタンスにアクセスできるようになります。このファイルをエディタで開くと、キーペアのパブリックキーを表示できます。以下に、`&examplekeypairname` という名前のキーペアのエントリの例を示します。パブリックキーの後にキーペアの名前が続く構成になっています。以下に例を示します。

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS7O6Vhz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RJhJOI0iBxrlsLnBITntckij7FbtXJMXLvvwJryDUilBMTjYtwB+QhYXUMOzce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZqaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqc0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3RbBQoQzdv8v7yeb70z1PnWOyN0qFU0XA246RA8QFYiCNYwiI3f05p6KLxEXAMPLE my-key-pair
```

Linux/Unix では、`ssh-keygen` を使用してキーペアのパブリックキーを取得できます。プライベートキーをダウンロードしたコンピュータで、次のコマンドを実行します。

```
$ ssh-keygen -y
```

ssh-keygen によってメッセージが表示されたら、.pem ファイルのパスを指定します。コマンドがパブリックキーを返します。以下に例を示します。

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706Vhz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RjHJOI0iBXRlsLnBItnctckiJ7FbtXJMXLvvwJryDUilBMTjYtwB+QhYXUM0zce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZqaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3RbBQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE
```

Windows では、PuTTYgen を使用してキーペアのパブリックキーを取得できます。PuTTYgen を起動して、[Load] をクリックし、.ppk または .pem ファイルを選択します。PuTTYgen によってパブリックキーが表示されます。

インスタンスの起動時に指定したパブリックキーも、そのインスタンスメタデータを介して表示できます。インスタンスの起動時に指定したパブリックキーを表示するには、インスタンスから次のコマンドを使用します。

```
$ GET http://169.254.169.254/latest/public-keys/0/openssh-key
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQClKsfkNkuSevGj3eYhCe53pcjqP3maAhDFcvBS706Vhz2ItxCih+PnDSUaw+WNQn/mZphTk/a/gU8jEzoOWbkM4yxyb/wB96xbiFveSFJuOp/d6RjHJOI0iBXRlsLnBItnctckiJ7FbtXJMXLvvwJryDUilBMTjYtwB+QhYXUM0zce5Pjz5/i8SeJtjnV3iAoG/cQk+0FzZqaeJAAHco+CY/5WrUBkrHmFJr6HcXkvJdWPkYQS3xqC0+FmUZofz221CBt5IMucxXPkX4rWi+z7wB3RbBQoQzd8v7yeb7Oz1PnWOyN0qFU0XA246RA8QFYiCNYwI3f05p6KLxEXAMPLE my-key-pair
```

インスタンスへの接続に使用するキーペアを変更した場合、このページの次のセクションに示すように、インスタンスメタデータが更新されて新しいパブリックキーが表示されないで、ご注意ください。インスタンスメタデータには、インスタンスの起動時に指定したキーペアのパブリックキーが引き続き表示されます。インスタンスメタデータの詳細については、[インスタンスメタデータの取得 \(p. 291\)](#) を参照してください。

## キーペアの削除

キーペアを削除すると、Amazon EC2 のパブリックキーのコピーのみが削除されます。キーペアの削除は、コンピュータのプライベートキーにも、このキーペアを使用して既に起動している各インスタンスのパブリックキーにも影響しません。削除したキーペアを使用して新しいインスタンスを起動することはできませんが、プライベートキー (.pem) ファイルを保持している間は、削除したキーペアを使用して起動した各インスタンスに引き続き接続することができます。

Amazon EC2 コンソールまたはコマンドラインを使用して、キーペアを削除できます。

コンソールを使用してキーペアを削除するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Key Pairs] をクリックします。
3. キーペアを選択し、[Delete] をクリックします。
4. プロンプトが表示されたら、[Yes] をクリックします。

コマンドラインを使用してキーペアを削除するには

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

- [delete-key-pair](#) ( AWS CLI )
- [ec2-delete-keypair](#) ( Amazon EC2 CLI )
- [Remove-EC2KeyPair](#) ( AWS Tools for Windows PowerShell )

## プライベートキーを失った場合にインスタンスに接続する

EBS-Backed インスタンスのプライベートキーを失った場合は、インスタンスへのアクセス権を回復することができます。インスタンスを停止し、そのルートボリュームをデタッチし、データボリュームとして別のインスタンスにアタッチし、`authorized_keys` ファイルを変更して、ボリュームを元のインスタンスに戻し、インスタンスを再起動する必要があります。この手順は、Instance store-Backed インスタンス、またはルートボリュームに AWS Marketplace 製品コードが含まれるインスタンスではサポートされていません。インスタンスの起動、接続、および停止の詳細については、[インスタンスのライフサイクル](#) (p. 321) を参照してください。

インスタンスのルートデバイスタイプを判断するには、Amazon EC2 コンソールを開き、[Instances] をクリックしてインスタンスを選択し、詳細ペインで [Root device type] の値をチェックします。この値は `ebs` または `instance store` のどちらかです。ルートデバイスがインスタンスストアボリュームである場合、インスタンスに接続するにはプライベートキーが必要です。また、詳細ペインで [Product codes] の値を確認して、AWS Marketplace 製品コードを使用しているかどうかを判断することもできます。

### 前提条件

Amazon EC2 コンソールまたはサードパーティ製のツールで、新しいキーペアを作成します。

別のキーペアを使用して EBS-Backed インスタンスに接続するには

1. Amazon EC2 コンソールのナビゲーションペインで [Instances] をクリックし、接続先にするインスタンスを選択します ( このインスタンスを「元のインスタンス」と呼びます )。
2. この手順を完了するために必要となる以下の情報を控えておきます。
  - 元のインスタンスのインスタンス ID ( `i-xxxxxxx` )、AMI ID ( `ami-xxxxxxx` )、およびアベイラビリティゾーンを書き留めます。
  - 詳細ペインの [Block devices] で `sda1` ( ルートボリューム ) のエントリをクリックし、ボリューム ID ( `vol-xxxxxxx` ) を書き留めます。
  - [EC2-Classic] 元のインスタンスに Elastic IP アドレスが関連付けられている場合は、詳細ペインの [Elastic IP] に表示される Elastic IP アドレスを書き留めます。
3. [Actions] をクリックして、[Stop] をクリックします。[Stop] が無効になっている場合は、インスタンスが既に停止しているか、またはルートボリュームがインスタンスストアボリュームです。
4. 以下のオプションでコンソールを使用して一時インスタンスを起動してください。
  - [Choose an AMI] ページで、元のインスタンスを起動するのに使用したのと同じ AMI を選択します。その AMI を使用できない場合は、停止したインスタンスから使用可能な AMI を作成できません。詳細については、「[Amazon EBS-Backed Linux AMI の作成](#) (p. 77)」を参照してください。
  - [Choose and Instance Type] ページで、`t1.micro` インスタンスを選択します。
  - [Configure Instance Details] ページで、接続するインスタンスと同じアベイラビリティゾーンを指定します。VPC のインスタンスを起動する場合、このアベイラビリティゾーンのサブネットを選択します。
  - [Tag Instance] ページで、一時インスタンスであることを示すために、インスタンスに `Name=Temporary` タグを追加します。

- [Review] ページで、[Launch] をクリックします。作成した新しいキーペアを指定し、[Launch Instances] をクリックします。
- 5. ナビゲーションペインで [Volumes] をクリックし、元のインスタンスのルートデバイスボリュームを選択します ( 前のステップでそのボリューム ID を書き留めました )。[Actions] をクリックし、[Detach Volume] をクリックします。ボリュームの状態が available になるまで待ちます ( [Refresh] アイコンをクリックしなければならない場合があります )。
- 6. ボリュームを選択したまま [Actions] をクリックし、[Attach Volume] をクリックします。一時インスタンスのインスタンス ID を選択し、[Device] で指定したデバイス名 ( /dev/sdf など ) を書き留めて、[Yes, Attach] をクリックします。
- 7. 一時インスタンスに接続します。
- 8. 一時インスタンスから、そのファイルシステムにアクセスできるように、インスタンスにアタッチしたボリュームをマウントします。例えば、デバイス名が /dev/sdf の場合、次のコマンドを使用してボリュームを /mnt/tempvol としてマウントします。

```
$ sudo mkdir /mnt/tempvol
$ sudo mount /dev/sdf /mnt/tempvol
```

- 9. 一時インスタンスから、次のコマンドを使用して、/dev/sdf の authorized\_keys を一時インスタンスの authorized\_keys の新しいパブリックキーで更新します。

```
$ cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

このコピーが正常に終了すると、次のステップに進むことができます。

( オプション ) または、/mnt/tempvol のファイルを編集するアクセス許可がない場合、sudo を使用してファイルを更新してから、ファイルに対するアクセス許可を確認して、元のインスタンスにログインできるかどうかを確認する必要があります。次のコマンドを使用して、ファイルに対するアクセス許可を確認します。

```
$ sudo ls -l /mnt/home/ec2-user/.ssh
total 4
-rw----- 1 222 500 398 Sep 13 22:54 authorized_keys
```

この出力例では、222 はユーザー ID、500 はグループ ID です。次に、sudo を実行して、失敗したコピーコマンドを再実行します。

```
$ sudo cp .ssh/authorized_keys /mnt/tempvol/home/ec2-user/.ssh/authorized_keys
```

次のコマンドを再度実行して、アクセス許可が変更されているかどうかを判断します。

```
$ sudo ls -l /mnt/home/ec2-user/.ssh
```

ユーザー ID とグループ ID が変更されている場合は、次のコマンドを実行して復元します。

```
$ sudo chown 222:500 /mnt/home/ec2-user/.ssh/authorized_keys
```

- 10. 一時インスタンスから、元のインスタンスに再アタッチできるように、アタッチしたボリュームをアンマウントします。例えば、次のコマンドを使用して、/dev/sdf をアンマウントします。

```
$ sudo umount -d /dev/sdf
```

- Amazon EC2 コンソールから、書き留めたボリューム ID を持つボリュームを選択して、[Actions] をクリックし、[Detach Volume] をクリックします。ボリュームの状態が `available` になるまで待ちます ([Refresh] アイコンをクリックしなければならない場合があります)。
- ボリュームを選択したまま [Actions] をクリックし、[Attach Volume] をクリックします。元のインスタンスのインスタンス ID を選択し、デバイス名 `/dev/sda1` を指定して、[Yes, Attach] をクリックします。



#### Warning

`sda1` をデバイス名として指定しない場合、元のインスタンスを起動することはできません。これは、Amazon EC2 では `sda1` のルートデバイスボリュームが想定されているためです。

- 元のインスタンスを選択して、[Actions] をクリックし、[Start] をクリックします。インスタンスが `running` 状態になったら、新しいキーペアのプライベートキーファイルを使用して、そのインスタンスに接続できます。
- [EC2-Classic] 停止する前に元のインスタンスに Elastic IP アドレスが関連付けられていた場合は、以下のようにして、このアドレスをインスタンスに再度関連付ける必要があります。
  - ナビゲーションペインで、[Elastic IPs] をクリックします。
  - この手順の最初に書き留めた Elastic IP アドレスを選択します。
  - [Associate Address] をクリックします。
  - 元のインスタンスの ID を選択し、[Associate] をクリックします。
- ( オプション ) 一時インスタンスをそれ以上使用しない場合は、終了できます。一時インスタンスを選択して、[Actions] をクリックし、[Terminate] をクリックします。

## Amazon EC2 セキュリティグループ

### Abstract

1つ以上のインスタンスのトラフィックを制御するファイアウォールとして、セキュリティグループとセキュリティグループルールを使用します。

セキュリティグループは、1つ以上のインスタンスのトラフィックを制御する仮想ファイアウォールとして機能します。インスタンスを起動するときに、1つ以上のセキュリティグループとインスタンスを関連付けます。各セキュリティグループに対してルールを追加し、関連付けられたインスタンスに対するトラフィックを許可します。セキュリティグループルールはいつでも変更できます。新しいルールは、セキュリティグループに関連付けられているインスタンスすべてに自動的に適用されます。インスタンスに到達できるトラフィックを許可するかどうかの判断では、インスタンスに関連付けられているすべてのセキュリティグループのすべてのルールが評価されます。

このトピックでは、セキュリティグループとセキュリティグループルールについて説明します。

### Topics

- [EC2-Classic 用セキュリティグループ \(p. 478\)](#)
- [EC2-VPC 用セキュリティグループ \(p. 478\)](#)
- [セキュリティグループのルール \(p. 479\)](#)
- [デフォルトのセキュリティグループ \(p. 480\)](#)

- [カスタムのセキュリティグループ \(p. 480\)](#)
- [セキュリティグループを作成する \(p. 481\)](#)
- [セキュリティグループについて説明する \(p. 482\)](#)
- [セキュリティグループへのルールの追加 \(p. 482\)](#)
- [ルールの削除 \(セキュリティグループから\) \(p. 484\)](#)
- [セキュリティグループの削除 \(p. 484\)](#)
- [API とコマンドの概要 \(p. 484\)](#)

お客様の要件がセキュリティグループでは満たされない場合は、セキュリティグループを使用した上で、どのインスタンスでも、お客様独自のファイアウォールを維持できます。

## EC2-Classic 用セキュリティグループ

EC2-Classic を使用している場合は、EC2-Classic 用に作成したセキュリティグループを使用する必要があります。EC2-Classic でインスタンスを起動する場合は、インスタンスと同じリージョンのセキュリティグループを指定する必要があります。EC2-Classic でインスタンスを起動する場合は、VPC 用に作成したセキュリティグループは指定できません。

EC2-Classic でインスタンスを起動した後でセキュリティグループを変更することはできません。ただし、セキュリティグループルールの追加または削除は可能です。これらの変更は、セキュリティグループに関連付けられているすべてのインスタンスに自動的に適用されます。



### Note

EC2-Classic では、1 つのインスタンスを最大 500 のセキュリティグループと関連付けることができ、また 1 つのセキュリティグループには最大 100 のルールを追加できます。

## EC2-VPC 用セキュリティグループ

EC2-VPC を使用している場合は、VPC 用に作成したセキュリティグループを使用する必要があります。VPC でインスタンスを起動する場合は、その VPC 用のセキュリティグループを指定する必要があります。VPC でインスタンスを起動する場合は、EC2-Classic 用に作成したセキュリティグループは指定できません。

VPC でインスタンスを起動した後に、そのセキュリティグループを変更することができます。セキュリティグループルールも変更できます。これらの変更は、セキュリティグループに関連付けられているすべてのインスタンスに自動的に適用されます。



### Note

EC2-VPC では、1 つのインスタンスを最大 5 つのセキュリティグループと関連付けることができ、また 1 つのセキュリティグループには最大 50 のルールを追加できます。

CLI または API アクションに対して、デフォルト以外の VPC 用のセキュリティグループを指定する場合、セキュリティグループを識別するには、セキュリティグループ名ではなく、セキュリティグループ ID を使用する必要があります。

セキュリティグループを識別する EC2-VPC のセキュリティグループには追加の機能がありますが、この機能は EC2-Classic のセキュリティグループではサポートされていません。EC2-VPC のセキュリティグループの詳細については、[Amazon Virtual Private Cloud ユーザーガイドの Security Groups for Your VPC](#) を参照してください。

## セキュリティグループのルール

セキュリティグループのルールは、セキュリティグループに関連付けられたインスタンスに到達することを許可されるインバウンドトラフィックと、外に向かうことを許可されるアウトバウンドトラフィックを制御します。デフォルトで、セキュリティグループはすべてのアウトバウンドトラフィックを許可します。

ルールの追加と削除は随時行うことができます。変更は、短時間が経過した後、セキュリティグループに関連付けられたインスタンスに自動的に適用されます。セキュリティグループの既存のルールを編集または削除したり、新しいルールを追加したりすることができます。既存のセキュリティグループから新しいセキュリティグループにルールをコピーできます。EC2-Classicのアウトバウンドルールは変更できません。セキュリティグループのルールは常にパーミッシブです。アクセスを拒否するルールを作成することはできません。

ルールごとに、以下の点について指定します。

- 許可するプロトコル ( TCP、UDP、ICMP など )。
- TCP および UDP、またはカスタムプロトコル: 許可するポートの範囲
- ICMP: ICMP タイプおよびコード
- 1つのオプション、または送信元 ( インバウンドルール ) または送信先 ( アウトバウンドルール ) の以下のオプション:
  - 個別の IP アドレス ( CIDR 表記 )。IP アドレスの後には必ず /32 プレフィックスを付けてください。IP アドレスの後に /0 プレフィックスを付けると、ポートがすべてのユーザーに開放されます。例えば、IP アドレス 203.0.113.1 は 203.0.113.1/32 のように指定します。
  - CIDR 表記の IP アドレスの範囲 ( 例えば、203.0.113.0/24 )。
  - セキュリティグループの名前 ( EC2-Classic ) または ID ( EC2-Classic または EC2-VPC )。これにより、指定セキュリティグループに関連付けられたインスタンスから、このセキュリティグループに関連付けられたインスタンスへのアクセスが許可されます。( その際、送信元セキュリティグループからこのセキュリティグループにルールが追加されることはありません。 ) 以下のセキュリティグループの 1 つを指定できます:
    - 現在のセキュリティグループ。
    - EC2-Classic: 同じリージョンの EC2-Classic の異なるセキュリティグループ
    - EC2-VPC: 同じ VPC の異なるセキュリティグループ
    - EC2-Classic: 同じリージョンの別の AWS アカウントのセキュリティグループ ( AWS アカウント ID をプレフィックスとして追加。例えば、111122223333/sg-edcd9784 )

ルールの送信元または送信先としてセキュリティグループを指定する場合、ルールはセキュリティグループと関連付けられるすべてのインスタンスに影響します。たとえば、受信トラフィックは、送信元セキュリティグループと関連付けられたインスタンスのプライベート IP アドレスに基づいて許可されます。

特定のポートに複数のルールがある場合、最も許容度の大きいルールを適用します。例えば、IP アドレス 203.0.113.1 からの TCP ポート 22 ( SSH ) に対するアクセスを許可するルールがあり、全員からの TCP ポート 22 に対するアクセスを許可する別のルールがある場合、全員が TCP ポート 22 にアクセスできます。

複数のセキュリティグループをインスタンスに関連付けると、各セキュリティグループのルールが効率的に集約され、1つのルールセットが作成されます。このルールセットを使用して、アクセスを許可するかどうかを判断します。



### Caution

複数のセキュリティグループを 1つのインスタンスに割り当てることができるため、インスタンスには数百単位のルールを適用できます。結果として、インスタンスにアクセスするときに

問題が発生する可能性があります。そのため、ルールは可能な限り要約することをお勧めします。

IP アドレスについては、[Amazon EC2 インスタンスの IP アドレッシング \(p. 529\)](#) を参照してください。

## デフォルトのセキュリティグループ

AWS アカウントには、EC2-Classic のリージョンごとにデフォルトのセキュリティグループが自動的に設定されます。VPC を作成する場合、VPC のデフォルトセキュリティグループが自動的に作成されます。インスタンスを起動するときに別のセキュリティグループを指定しないと、そのインスタンスは適切なデフォルトのセキュリティグループに自動的に関連付けられます。

デフォルトのセキュリティグループには `default` と名前が付けられ、AWS によって ID が割り当てられます。デフォルトのセキュリティグループごとの初期設定を次に示します。

- デフォルトのセキュリティグループと関連付けられている他のインスタンスからのインバウンドトラフィックのみを許可する
- インスタンスからのすべてのアウトバウンドトラフィックを許可する

デフォルトのセキュリティグループは、それ自体を、そのインバウンドルール内の送信元セキュリティグループとして指定します。これは、デフォルトのセキュリティグループと関連付けられているインスタンスが、デフォルトのセキュリティグループと関連付けられている他のインスタンスと通信することを許可するものです。

デフォルトのセキュリティグループルールは変更できます。例えば、特定のホストがインスタンスを管理できるように、SSH またはリモートデスクトップ接続を許可するインバウンドルールを追加できます。

デフォルトのセキュリティグループを削除することはできません。

## カスタムのセキュリティグループ

すべてのインスタンスでデフォルトのセキュリティグループを使用することを望まない場合、独自のセキュリティグループを作成して、インスタンスの起動時にそれらを指定することができます。複数のセキュリティグループを作成して、インスタンスが果たすさまざまな役割（例えば、Web サーバーまたはデータベースサーバー）を反映させることができます。Web サーバーまたはデータベースサーバーのセキュリティグループを作成する手順については、[Amazon Virtual Private Cloud ユーザーガイドの「Recommended Security Groups」](#) を参照してください。



### Note

EC2-Classic では、アカウントごとに各リージョンに最大 500 個のセキュリティグループを作成できます。EC2-VPC では、VPC ごとに最大 100 個のセキュリティグループを作成できます。EC2-VPC のセキュリティグループ制限に対して、EC2-Classic のセキュリティグループ数はカウントされません。

セキュリティグループを作成する場合、名前と説明を指定する必要があります。セキュリティグループには、255 文字以下の名前と説明を指定できます。また、次の特徴の制限があります。

- EC2-Classic: ASCII 文字
- EC2-VPC: a-z、A-Z、0-9、スペース、およびピリオド (.) \_-:/()#,@[]+=&:{}!\$\*

AWS は各セキュリティグループに `sg-xxxxxxx` の形式で一意的 ID を割り当てます。作成するセキュリティグループの初期設定を次に示します。

- 着信トラフィックを許可しない
- すべての発信トラフィックを許可する

セキュリティグループを作成したら、関連するインスタンスに到達できる着信トラフィックのタイプを反映するように着信ルールを変更できます。EC2-VPC では、アウトバウンドルールも変更できます。

同じセキュリティグループのインスタンス同士が通信できるようにするには、そのためのルールを明示的に追加する必要があります。以下の表に、EC2-Classic のインスタンスの通信を許可するために、セキュリティグループに追加する必要のあるルールを示します。

インバウンド			
送信元	プロトコル	ポート範囲	コメント
セキュリティグループの ID	ICMP	すべて	このセキュリティグループに関連付けられている他のインスタンスからのインバウンド ICMP アクセスを許可する
セキュリティグループの ID	TCP	0 – 65535	このセキュリティグループに関連付けられている他のインスタンスからのインバウンド TCP アクセスを許可する
セキュリティグループの ID	UDP	0 – 65535	このセキュリティグループに関連付けられている他のインスタンスからのインバウンド UDP アクセスを許可する

以下の表に、VPC のインスタンスの通信を許可するためにセキュリティグループに追加する必要のあるルールを示します。

インバウンド			
送信元	プロトコル	ポート範囲	コメント
セキュリティグループの ID	すべて	すべて	このセキュリティグループに関連付けられている他のインスタンスからのインバウンドトラフィックを許可する

## セキュリティグループを作成する

新しいセキュリティグループを作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] をクリックします。
3. [Create Security Group] をクリックします。
4. セキュリティグループの名前と説明を指定します。[VPC] の場合、[No VPC] を選択して、EC2-Classic のセキュリティグループを作成します。その VPC のセキュリティグループを作成するには、VPC ID を選択します。

5. ルールの追加を開始するか、[Create] をクリックして、セキュリティグループを作成できます ( ルールは後で追加できます )。ルールの追加の詳細については、[セキュリティグループへのルールの追加 \(p. 482\)](#) を参照してください。

セキュリティグループをコピーするには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] をクリックします。
3. コピーするセキュリティグループを選択します。[Actions] をクリックし、[Copy to new] を選択します。
4. [Create Security Group] ダイアログが開き、既存のセキュリティグループのルールが自動入力されます。新しいセキュリティグループの名前と説明を指定します。[VPC] リストで、[No VPC] を選択して、EC2-Classic のセキュリティグループを作成します。その VPC のセキュリティグループを作成するには、VPC ID を選択します。終了したら [Create] をクリックします。

インスタンスを起動する際に、インスタンスにセキュリティグループを割り当てることができます。ルールを追加または削除すると、それらの変更は、そのセキュリティグループを割り当てたすべてのインスタンスに自動的に適用されます。

EC2-Classic でインスタンスを起動した後でセキュリティグループを変更することはできません。VPC でインスタンスを起動した後、そのセキュリティグループを変更することができます。詳細については、[Amazon Virtual Private Cloud ユーザーガイドの Changing an Instance's Security Groups](#) を参照してください。

## セキュリティグループについて説明する

EC2-Classic のセキュリティグループについて説明するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] をクリックします。
3. フィルタリストから、[EC2 security groups] を選択します。
4. セキュリティグループを選択します。[Description] タブには一般情報が、[Inbound] タブにはインバウンドルールが表示されます。

EC2-VPC のセキュリティグループについて説明するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] をクリックします。
3. フィルタリストから、[VPC security groups] を選択します。
4. セキュリティグループを選択します。[Description] タブには一般情報が、[Inbound] タブにはインバウンドルールが、[Outbound] タブにはアウトバウンドルールが表示されます。

## セキュリティグループへのルールの追加

ルールをセキュリティグループに追加すると、セキュリティグループに関連付けられているすべてのインスタンスに新しいルールが自動的に適用されます。

セキュリティグループにルールを追加するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] をクリックします。

3. セキュリティグループを選択します。
4. Web サーバーを許可して、すべての HTTP および HTTPS インバウンドトラフィックを受信します。[Inbound] タブで、[Edit] をクリックします。ダイアログで、[Add Rule] をクリックします。[Type] リストから [HTTP] を選択し、ソースを [Anywhere] のままにしておきます (0.0.0.0/0)。HTTPS プロトコルに類似のルールを追加します。

Type	Protocol	Port Range	Source
HTTP	TCP	80	Anywhere (0.0.0.0/0)
HTTPS	TCP	443	Anywhere (0.0.0.0/0)

5. Linux インスタンスに接続するには、SSH トラフィックを許可する必要があります。[Add Rule] をクリックし、[Type] リストから [SSH] を選択します。

[Source] フィールドで、コンピュータのパブリック IP アドレスを CIDR 表記で指定します。例えば、IP アドレスが 203.0.113.25 の場合、この単一の IP アドレスを CIDR 表記でリストするには 203.0.113.25/32 と指定します。会社が特定の範囲からアドレスを割り当てている場合、範囲全体 (203.0.113.0/24 など) を指定します。[Source] リストから [My IP] を選択すると、お使いのコンピュータの IP アドレスが自動的にフィールドに入力されます。ただし、ISP 経由で、またはファイアウォールの内側から静的な IP アドレスなしで接続している場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。

**STOP** Caution

0.0.0.0/0 を使うと、すべての IP アドレスから SSH 経由でインスタンスにアクセスすることが許可されます。これはテスト環境で短時間なら許容できますが、実稼働環境で行うのは安全ではありません。実稼働環境では、特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを限定します。

6. このセキュリティグループに関連付けられているすべてのインスタンス間、または他のセキュリティグループに関連付けたインスタンスとこのセキュリティグループに関連付けられたインスタンス間の通信を許可できます。[Add Rule] をクリックし、[All ICMP] を選択して、[Source] 内にセキュリティグループの ID を入力し始めると、セキュリティグループのリストが表示されます。リストからセキュリティグループを選択します。TCP と UDP プロトコルの手順を繰り返します。完了したら [Save] をクリックします。

Type	Protocol	Port Range	Source
HTTP	TCP	80	Anywhere (0.0.0.0/0)
HTTPS	TCP	443	Anywhere (0.0.0.0/0)
All ICMP	ICMP	0 - 65535	Custom IP (sg-ed9f5f86)
All TCP	TCP	0 - 65535	Custom IP (sg-ed9f5f86)
All UDP	UDP	0 - 65535	Custom IP (sg-ed9f5f86)

7. VPC のセキュリティグループを作成する場合は、アウトバウンドルールも指定できます。例については、*Amazon Virtual Private Cloud ユーザーガイド* の「[Adding and Removing Rules](#)」を参照してください。

## ルールの削除 ( セキュリティグループから )

セキュリティグループからルールを削除すると、その変更内容が自動的にセキュリティグループに関連付けられているインスタンスに適用されます。

セキュリティグループルールを削除するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] をクリックします。
3. セキュリティグループを選択します。
4. [Edit] をクリックし、削除する各ルールの隣にある [Delete] アイコンをクリックします。
5. [Save] をクリックします。

## セキュリティグループの削除

インスタンスに関連付けられているセキュリティグループを削除することはできません。デフォルトセキュリティグループを削除することはできません。

セキュリティグループを削除するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Security Groups] をクリックします。
3. セキュリティグループを選択し、Delete をクリックします。
4. [Yes, Delete] をクリックします。

## API とコマンドの概要

このページで説明しているタスクは、コマンドラインまたは API を使用して実行できます。コマンドラインインターフェイスの詳細および利用できる API の一覧については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

セキュリティグループを作成する

- [create-security-group](#) ( AWS CLI )
- [ec2-create-group](#) ( Amazon EC2 CLI )
- [New-EC2SecurityGroup](#) ( AWS Tools for Windows PowerShell )

1 つ以上の Ingress ルールをセキュリティグループに追加する

- [authorize-security-group-ingress](#) ( AWS CLI )
- [ec2-authorize](#) ( Amazon EC2 CLI )
- [Grant-EC2SecurityGroupIngress](#) ( AWS Tools for Windows PowerShell )

[EC2-VPC] 1 つ以上の Egress ルールをセキュリティグループに追加する

- [authorize-security-group-egress](#) ( AWS CLI )

- [ec2-authorize](#) ( Amazon EC2 CLI )
- [Grant-EC2SecurityGroupIngress](#) ( AWS Tools for Windows PowerShell )

1 つまたは複数のセキュリティグループについて説明する

- [describe-security-groups](#) ( AWS CLI )
- [ec2-describe-group](#) ( Amazon EC2 CLI )
- [Get-EC2SecurityGroup](#) ( AWS Tools for Windows PowerShell )

[EC2-VPC] インスタンスのセキュリティグループを変更する

- [modify-instance-attribute](#) ( AWS CLI )
- [ec2-modify-instance-attribute](#) ( Amazon EC2 CLI )
- [Edit-EC2InstanceAttribute](#) ( AWS Tools for Windows PowerShell )

セキュリティグループから 1 つ以上の Ingress ルールを削除する

- [revoke-security-group-ingress](#) ( AWS CLI )
- [ec2-revoke](#) ( Amazon EC2 CLI )
- [Revoke-EC2SecurityGroupIngress](#) ( AWS Tools for Windows PowerShell )

[EC2-VPC] 1 つ以上の Egress ルールをセキュリティグループから削除する

- [revoke-security-group-egress](#) ( AWS CLI )
- [ec2-revoke](#) ( Amazon EC2 CLI )
- [Revoke-EC2SecurityGroupEgress](#) ( AWS Tools for Windows PowerShell )

セキュリティグループを削除する

- [delete-security-group](#) ( AWS CLI )
- [ec2-delete-group](#) ( Amazon EC2 CLI )
- [Remove-EC2SecurityGroup](#) ( AWS Tools for Windows PowerShell )

## Amazon EC2 のリソースに対するアクセスの制御

### Abstract

セキュリティグループを設定し、IAM を使用して、Amazon EC2 リソースへのアクセスを制御します。

セキュリティ認証情報により、AWS のサービスに対してお客様の身分が証明され、Amazon EC2 リソースなどの AWS リソースを無制限に使用できる許可が与えられます。Amazon EC2 および AWS Identity and Access Management ( IAM ) の機能を使用して、その他のユーザー、サービス、およびアプリケーションがお客様の Amazon EC2 のリソースを使用できるようにします。その際、お客様のセキュリティ認証情報は共有されません。Amazon EC2 のリソースの完全使用または制限付き使用のどちらを許可するか選択できます。

### Topics

- [インスタンスへのネットワークアクセス \(p. 486\)](#)
- [Amazon EC2 のアクセス許可属性 \(p. 486\)](#)
- [IAM と Amazon EC2 \(p. 486\)](#)

- [Amazon EC2 の IAM ポリシー \(p. 488\)](#)
- [Amazon EC2 の IAM ロール \(p. 516\)](#)
- [インスタンス用の受信トラフィックの認可 \(p. 522\)](#)

## インスタンスへのネットワークアクセス

セキュリティグループには、インスタンスに到達できるトラフィックを制御するファイアウォールとしての働きがあります。インスタンスを起動するときに、そのインスタンスに1つまたは複数のセキュリティグループを割り当てることができます。セキュリティグループのそれぞれに、そのインスタンスへのトラフィックを制御するルールを追加できます。セキュリティグループルールはいつでも変更できます。新しいルールは、そのセキュリティグループが割り当てられているインスタンスすべてに自動的に適用されます。

詳細については、[インスタンス用の受信トラフィックの認可 \(p. 522\)](#) を参照してください。

## Amazon EC2 のアクセス許可属性

お客様の組織には複数の AWS アカウントがある場合があります。Amazon EC2 では、Amazon マシンイメージ (AMI) および Amazon EBS スナップショットを使用できる追加の AWS アカウントを指定できます。このアクセス許可は AWS アカウントレベルでのみ有効です。特定の AWS アカウント内の特定ユーザーのアクセス許可を制限することはできません。指定した AWS アカウントのすべてのユーザーが、AMI またはスナップショットを使用できます。

AMI ごとに `LaunchPermission` 属性があり、AMI にアクセスできる AWS アカウントを制御します。詳細については、[AMI を一般公開する \(p. 65\)](#) を参照してください。

Amazon EBS スナップショットごとに `createVolumePermission` 属性があり、スナップショットを使用できる AWS アカウントを制御します。詳細については、[スナップショットの共有 \(p. 610\)](#) を参照してください。

## IAM と Amazon EC2

IAM を使って以下を行えます。

- お客様の AWS アカウントでユーザーとグループを作成する
- お客様の AWS アカウントでユーザーごとに固有のセキュリティ認証情報を割り当てる
- AWS のリソースを使用してタスクを実行するために各ユーザーのアクセス許可を制御する
- 別の AWS アカウントのユーザーがお客様の AWS のリソースを共有できるようにする
- AWS アカウントにロールを作成し、それを行えるユーザーまたはサービスを定義する
- お客様の企業用の既存のアイデンティティを使用し、AWS のリソースを使用してタスクを実行するようにアクセス許可を与える

Amazon EC2 で IAM を使用することにより、組織のユーザーが特定の Amazon EC2 API アクションを使用してタスクを実行できるかどうか、また、特定の AWS のリソースを使用できるかどうかを制御できます。

このトピックには、以下の質問に対する回答があります。

- IAM でグループとユーザーを作成するには、どうすればよいですか？
- ポリシーを作成するには、どうすればよいですか？
- Amazon EC2 でタスクを実行するには、どのような IAM ポリシーが必要ですか？
- Amazon EC2 でアクションを実行するための許可を与えるには、どうすればよいですか？

- Amazon EC2 の特定のリソースでアクションを実行するための許可を与えるには、どうすればよいですか？

## IAM のグループとユーザーを作成する

IAM のグループとユーザーを作成するには

1. IAM コンソールを開きます。
2. ダッシュボードから [Create a New Group of Users] をクリックします。
3. [GROUP NAME] ページで、グループ名を指定します。
4. [PERMISSIONS] ページで、グループのポリシーを指定します。ポリシーテンプレートを選択するか、カスタムポリシーを作成できます。例えば、Amazon EC2 では以下のポリシーテンプレートのいずれかが、ニーズを満たす場合があります。

- パワーユーザーアクセス
- 読み取り専用アクセス
- Amazon EC2 フルアクセス
- Amazon EC2 読み取り専用アクセス

カスタムポリシーを作成する詳細については、[Amazon EC2 の IAM ポリシー \(p. 488\)](#) を参照してください。

5. [USERS] ページで、1 つ以上のユーザー名を入力します。ユーザーが CLI または API を使用する場合、[Generate an access key for each User] を選択します。[Continue] をクリックします。
6. アクセスキーを生成する IAM がある場合、[Download Credentials] または [Show User Security Credentials] をクリックしてアクセスキーを保存します。ダイアログボックスで示されているように、シークレットアクセスキーを取得して保存できるのはこの機会だけです。
7. ユーザーがコンソールを使用する場合、ナビゲーションペインで [Users] をクリックして、各ユーザーに対して以下を実行します。
  - a. 対象のユーザーを選択します。
  - b. 詳細ペインの [Security Credentials] タブをクリックします。
  - c. [Sign-In Credentials] で、[Manage Password] をクリックします。
  - d. [Manage Password] ダイアログボックスで、オプションを選択して [Apply] をクリックします。
  - e. [Download Credentials] または [Show User Security Credentials] をクリックしてパスワードを保存します。
8. ユーザーごとに認証情報 ( アクセスキーとパスワード ) を与えます。これにより、IAM グループ用に指定したアクセス許可に基づいてサービスを使用できるようになります。

## 関連トピック

IAM の詳細については、以下のものを参照してください。

- [Amazon EC2 の IAM ポリシー \(p. 488\)](#)
- [Amazon EC2 の IAM ロール \(p. 516\)](#)
- [Identity and Access Management \( IAM \)](#)
- [IAM を使用する](#)

# Amazon EC2 の IAM ポリシー

## Abstract

IAM ユーザーが、どの Amazon EC2 リソースを表示、作成、および変更できるようにするかを制御します。

デフォルトでは、IAM ユーザーには Amazon EC2 リソースを作成または変更、または Amazon EC2 API を使用するタスクを実行する権限がありません。(つまり、Amazon EC2 コンソールまたは CLI を使用して実行することもできません。) IAM ユーザーがリソースを作成または変更、およびタスクを実行できるようにするには、IAM ポリシーを作成する必要があります。これによって、必要な特定のリソースおよび API アクションを使用するためのアクセス許可を IAM ユーザーに付与し、その後、ポリシーをそのアクセス許可が必要な IAM ユーザーまたはグループにアタッチします。

ポリシーをユーザーまたはユーザーグループにアタッチする場合、ポリシーによって特定リソースの特定タスクを実行するユーザーの権限が許可または拒否されます。IAM ポリシーの一般的な情報については、IAM を使用するの [Permissions and Policies](#) を参照してください。

## Topics

- [ポリシー構文 \(p. 488\)](#)
- [Amazon EC2 のアクション \(p. 489\)](#)
- [Amazon EC2 用の Amazon リソースネーム \(p. 489\)](#)
- [Amazon EC2 の条件キー \(p. 492\)](#)
- [ユーザーが必要なアクセス許可を持っているかどうかを確認する \(p. 494\)](#)
- [Amazon EC2 API アクションでサポートされるリソースと条件 \(p. 494\)](#)
- [Amazon EC2 のポリシーステートメントの例 \(p. 501\)](#)

## ポリシー構文

IAM ポリシーは 1 つ以上のステートメントで構成される JSON ドキュメントです。各ステートメントは次のように構成されます。

```
{
  "Statement": [ {
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  }
]
```

ステートメントはさまざまなエレメントで構成されます。

- [Effect]: *effect* は、Allow または Deny にすることができます。デフォルトでは、IAM ユーザーはリソースおよび API アクションを使用するアクセス許可がないため、リクエストはすべて拒否されます。明示的な許可はデフォルトに優先します。明示的な拒否はすべての許可に優先します。
- [Action]: *action* は、アクセス許可を付与または拒否する対象とする、特定の API アクションです。*action* の指定については、[Amazon EC2 のアクション \(p. 489\)](#) を参照してください。
- [Resource]: アクションによって影響を及ぼされるリソースです。Amazon EC2 API アクションの中には、アクションによって作成/変更できるリソースをポリシー内で特定できるものもあります。ス

ステートメント内でリソースを指定するには、Amazon リソースネーム (ARN) を使用する必要があります。*arn* 値の指定については、[Amazon EC2 用の Amazon リソースネーム \(p. 489\)](#) を参照してください。どの API がどの ARN をサポートするかについては、[Amazon EC2 API アクションでサポートされるリソースと条件 \(p. 494\)](#) を参照してください。API アクションが ARN をサポートしていない場合は、\* ワイルドカードを使用すると、アクションがすべてのリソースに影響するように指定できません。

- [Condition]: condition はオプションです。ポリシーの発効条件を指定するために使用します。Amazon EC2 の条件を指定する方法については、[Amazon EC2 の条件キー \(p. 492\)](#) を参照してください。

Amazon EC2 の IAM ポリシーステートメント例については、[Amazon EC2 のポリシーステートメントの例 \(p. 501\)](#) を参照してください。

## Amazon EC2 のアクション

IAM ポリシーステートメントで、IAM をサポートするすべてのサービスから任意の API アクションを指定できます。Amazon EC2 の場合、API アクション *ec2:* の名前で次のプレフィックスを使用します。例: *ec2:RunInstances* および *ec2:CreateImage*。

単一のステートメントに複数のアクションを指定するには、次のようにコンマで区切ります。

```
"Action": ["ec2:action1", "ec2:action2"]
```

ワイルドカードを使用して複数のアクションを指定することもできます。例えば、以下のように「Describe」という単語で始まる名前のすべてのアクションを指定できます。

```
"Action": "ec2:Describe*"
```

Amazon EC2 API アクションをすべて指定するには、\* ワイルドカードを以下のように使用します。

```
"Action": "ec2:*"
```

Amazon EC2 アクションのリストについては、[Amazon Elastic Compute Cloud API Reference](#) の [Actions](#) を参照してください。

## Amazon EC2 用の Amazon リソースネーム

各 IAM ポリシーステートメントは、ARN を使用して指定したリソースに適用されます。



### Important

現時点では、すべての API アクションが個々の ARN をサポートしているわけではありません。今後、Amazon EC2 リソースに対する API アクションおよび ARN のサポートを追加していきます。どの Amazon EC2 API アクションでどの ARN を使用できるかについて、また各 ARN でサポートされる条件キーについては、[Amazon EC2 API アクションでサポートされるリソースと条件 \(p. 494\)](#) を参照してください。

ARN には以下の一般的な構文があります。

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

*service*

サービス (例: *ec2*)。

**リージョン**

リソースのリージョン ( 例: us-east-1 )。

**アカウント**

ハイフンなしの AWS アカウント ID ( 例: 123456789012 )。

**resourceType**

リソースの種類 ( 例: instance )。

**resourcePath**

リソースを識別するパス。パスにワイルドカードの \* が使用できます。

例えば、以下の要領で ARN を使用して、ステートメント内で特定のインスタンス ( i-1a2b3c4d ) を指定することができます。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/i-1a2b3c4d"
```

また、特定のアカウントに属するすべてのインスタンスを指定するには、以下の要領で \* ワイルドカードを使用します。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

すべてのリソースを指定する場合、または特定の API アクションが ARN をサポートしていない場合は、以下の要領で、Resource エlement 内で \* ワイルドカードを使用します。

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

以下の表では、Amazon EC2 API アクションによって使用される各リソースの種類の ARN を説明しています。

リソースタイプ	ARN
すべての Amazon EC2 リソース	arn:aws:ec2:*
特定リージョンの特定アカウントが所有するすべての Amazon EC2 リソース	arn:aws:ec2:region:account:*
カスタマーゲートウェイ	arn:aws:ec2:region:account:customer-gateway/cgw-id cgw-id には cgw-xxxxxxx が入ります
DHCP オプションセット	arn:aws:ec2:region:account:dhcp-options/dhcp-options-id dhcp-options-id には dopt-xxxxxxx が入ります
Image	arn:aws:ec2:region::image/image-id image-id には AMI、AKI または ARI の ID が入り、account は使用されません
インスタンス	arn:aws:ec2:region:account:instance/instance-id instance-id には i-xxxxxxx が入ります
インスタンスプロファイル	arn:aws:iam::account:instance-profile/instance-profile-name instance-profile-name にはインスタンスプロファイルの名前が入り、region は使用されません

リソースタイプ	ARN
インターネットゲートウェイ	arn:aws:ec2:region:account:internet-gateway/igw-id igw-id には igw-xxxxxxx が入ります
キーペア	arn:aws:ec2:region:account:key-pair/key-pair-name key-pair-name にはキーペア名が入ります ( 例: gsg-keypair )
ネットワーク ACL	arn:aws:ec2:region:account:network-acl/nacl-id nacl-id には acl-xxxxxxx が入ります
ネットワークインターフェイス	arn:aws:ec2:region:account:network-interface/eni-id eni-id には eni-xxxxxxx が入ります
配置グループ	arn:aws:ec2:region:account:placement-group/placement-group-name placement-group-name にはプレースメントグループ名が入ります ( 例: my-cluster )
ルートテーブル	arn:aws:ec2:region:account:route-table/route-table-id route-table-id には rtb-xxxxxxx が入ります
セキュリティグループ	arn:aws:ec2:region:account:security-group/security-group-id security-group-id には sg-xxxxxxx が入ります
スナップショット	arn:aws:ec2:region::snapshot/snapshot-id snapshot-id には snap-xxxxxxx が入り、account は使用されません
サブネット	arn:aws:ec2:region:account:subnet/subnet-id subnet-id には subnet-xxxxxxx が入ります
ボリューム	arn:aws:ec2:region:account:volume/volume-id volume-id には vol-xxxxxxx が入ります
VPC	arn:aws:ec2:region:account:vpc/vpc-id vpc-id には vpc-xxxxxxx が入ります
VPC ピア接続	arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id このとき、vpc-peering connection-id は pcx-xxxxxxx です。

Amazon EC2 API アクションの多くが複数のリソースと関連します。例えば、AttachVolume では Amazon EBS ボリュームをインスタンスにアタッチするため、IAM ユーザーはボリュームおよびインスタンスを使用するアクセス許可が必要です。単一のステートメントに複数のリソースを指定するには、以下のようにコンマで ARN を区切ります。

```
"Resource": ["arn1", "arn2"]
```

ARN の一般的な情報については、*Amazon Web Services General Reference* の [Amazon Resource Names \(ARN\) and AWS Service Namespaces](#) を参照してください。Amazon EC2 アクションによって作成または変更されるリソースの詳細について、および IAM ポリシーステートメントで使用できる ARN の詳細については、*Amazon Elastic Compute Cloud API Reference* の [IAM ユーザーへの、Amazon EC2 リソースに対するアクセス許可の付与](#) を参照してください。

## Amazon EC2 の条件キー

ポリシーステートメントでは、オプションで有効になるタイミングを制御する条件を指定できます。各条件には 1 つ以上のキーと値のペアが含まれます。条件キーは大文字小文字を区別しません。私たちは AWS 範囲の条件キーに加え、追加のサービス固有の条件キーを定義しました。

複数の条件、または単一の条件に複数のキーを指定する場合、論理 AND 演算を使用してそれら进行评估します。1 つのキーに複数の値を使用して単一の条件を指定する場合、論理 OR 演算を使用して条件进行评估します。アクセス許可が付与されるには、すべての条件を満たしている必要があります。

条件を指定する際にプレースホルダーも使用できます。例えば、IAM ユーザーに、そのユーザーの IAM ユーザー名を指定したタグ付きのリソースを使用するアクセス許可を与えることができます。詳細については、*IAM を使用するガイド* の [Policy Variables](#) を参照してください。

Amazon EC2 には AWS 範囲の条件キー ( [Available Keys](#) を参照 ) に加え、以下のサービス固有の条件キーが実装されています。( Amazon EC2 用のサービス固有の条件キーのサポートは今後さらに追加される予定です )

条件キー	キー/値のペア	評価の種類
ec2:AccepterVpc	"ec2:AccepterVpc":" <i>vpc-arn</i> "  このとき、 <i>vpc-arn</i> はピア VPC の VPC ARN です。	ARN、Null
ec2:AvailabilityZone	"ec2:AvailabilityZone":" <i>az-api-name</i> "  <i>az-api-name</i> にはアベイラビリティゾーン名が入ります ( 例: us-west-2a )。  アベイラビリティゾーンを一覧表示するには、 <a href="#">ec2-describe-availability-zones</a> を使用します。	文字列、Null
ec2:EbsOptimized	"ec2:EbsOptimized":" <i>optimized-flag</i> "  <i>optimized-flag</i> には true   false が入ります	ブール値、Null
ec2:ImageType	"ec2:ImageType":" <i>image-type-api-name</i> "  <i>image-type-api-name</i> には ami   aki   ari が入ります	文字列、Null
ec2:InstanceProfile	"ec2:InstanceProfile":" <i>instance-profile-arn</i> "  <i>instance-profile-arn</i> にはインスタンスプロファイル ARN が入ります	ARN、Null
ec2:InstanceType	"ec2:InstanceType":" <i>instance-type-api-name</i> "  <i>instance-type-api-name</i> にはインスタンスタイプの名前 ( t1.micro   m1.small   m1.medium   m1.large   m1.xlarge   c1.medium   c1.xlarge   m2.xlarge   m2.2xlarge   m2.4xlarge   hi1.4xlarge   cc1.4xlarge   cg1.4xlarge   cc2.8xlarge ) が入ります	文字列、Null

条件キー	キー/値のペア	評価の種類
ec2:Owner	"ec2:Owner":" <i>account-id</i> " <i>account-id</i> には <code>amazon</code>   <code>aws-account-id</code> が入ります	文字列、Null
ec2:ParentSnapshot	"ec2:ParentSnapshot":" <i>snapshot-arn</i> " <i>snapshot-arn</i> にはスナップショット ARN が入ります	ARN、Null
ec2:ParentVolume	"ec2:ParentVolume":" <i>volume-arn</i> " <i>volume-arn</i> にはボリューム ARN が入ります	ARN、Null
ec2:PlacementGroup	"ec2:PlacementGroup":" <i>placement-group-arn</i> " <i>placement-group-arn</i> にはプレースメントグループ ARN が入ります	ARN、Null
ec2:PlacementGroupStrategy	"ec2:PlacementGroupStrategy":" <i>placement-group-strategy</i> " <i>placement-group-strategy</i> には <code>cluster</code> が入ります	文字列、Null
ec2:Public	"ec2:Public":" <i>public-flag</i> " <i>public-flag</i> には <code>true</code>   <code>false</code> が入ります	ブール値、Null
ec2:Region	"ec2:Region":" <i>region-name</i> " <i>region-name</i> にはリージョン名が入ります ( 例: <code>us-west-2</code> ) リージョンを一覧表示するには、 <a href="#">ec2-describe-regions</a> を使用します。	文字列、Null
ec2:RequesterVpc	"ec2:RequesterVpc":" <i>vpc-arn</i> " このとき、 <i>vpc-arn</i> はリクエストの VPC の VPC ARN です。	ARN、Null
ec2:ResourceTag/ <i>tag-key</i>	"ec2:ResourceTag/ <i>tag-key</i> ":" <i>tag-value</i> " <i>tag-key</i> と <i>tag-value</i> にはタグとキーのペアが入ります	文字列、Null
ec2:RootDeviceType	"ec2:RootDeviceType":" <i>root-device-type-name</i> " <i>root-device-type-name</i> には <code>ebs</code>   <code>instance-store</code> が入ります	文字列、Null
ec2:Subnet	"ec2:Subnet":" <i>subnet-arn</i> " <i>subnet-arn</i> にはサブネット ARN が入ります	ARN、Null
ec2:Tenancy	"ec2:Tenancy":" <i>tenancy-attribute</i> " <i>tenancy-attribute</i> には <code>default</code>   <code>dedicated</code> が入ります	文字列、Null
ec2:Volumelops	"ec2:Volumelops":" <i>volume-iops</i> " <i>volume-iops</i> には 1 秒あたりの入力/出力オペレーション ( IOPS ) が入ります。範囲は 100 から 4000 です	数値、Null
ec2:VolumeSize	"ec2:VolumeSize":" <i>volume-size</i> " <i>volume-size</i> にはボリュームのサイズが入ります ( GiB 単位 )	数値、Null

条件キー	キー/値のペア	評価の種類
ec2:VolumeType	"ec2:VolumeType": <i>volume-type-name</i>  <i>volume-type-name</i> には、標準 Amazon EBS ボリュームを表す <i>standard</i> か、プロビジョンド IOPS ボリュームを表す <i>io1</i> が入ります。	文字列、Null
ec2:Vpc	"ec2:Vpc": <i>vpc-arn</i>  <i>vpc-arn</i> には VPC ARN が入ります	ARN、Null

アクションごとに、どの Amazon EC2 リソースでどの条件キーが使用できるかについては、[Amazon EC2 API アクションでサポートされるリソースと条件 \(p. 494\)](#) を参照してください。Amazon EC2 のポリシーステートメント例については、[Amazon EC2 のポリシーステートメントの例 \(p. 501\)](#) を参照してください。

## ユーザーが必要なアクセス許可を持っているかどうかを確認する

IAM ポリシーを作成したら、ポリシーを本稼働環境に置く前に、そのポリシーがユーザーに特定の API アクションおよび必要なリソースを使用するアクセス許可を付与しているかどうかを確認することをお勧めします。

まずテスト目的の IAM ユーザーを作成し、作成した IAM ポリシーをテストユーザーにアタッチします。次に、テストユーザーとしてリクエストを作成します。

テストしているアクションがリソースを変更する場合、DryRun パラメーターを使用してリクエストを作成する（または、`--auth-dry-run` オプションで CLI を実行する）必要があります。この場合、発信者は認証チェックを行います。操作は完了しません。例えば、実際に終了させることなく、ユーザーが特定のインスタンスを終了できるかどうかを確認できます。テストユーザーに必要なアクセス許可がある場合、リクエストで `DryRunOperation` が返されます。必要なアクセス許可がない場合は `UnauthorizedOperation` が返されます。

ポリシーが想定したアクセス許可をユーザーに付与していない場合、または過度に許可されている場合、必要に応じてポリシーを調整し、必要な結果を得るまで再テストできます。



### Important

ポリシーの変更が反映され、有効になるには数分間かかります。このため、ポリシーの更新をテストするには 5 分かかると見ておいてください。

認証チェックが失敗した場合、リクエストでは診断情報でエンコードされたメッセージが返されます。DecodeAuthorizationMessage アクションを使用してメッセージをデコードできます。詳細については、「AWS Security Token Service API リファレンス」の「[DecodeAuthorizationMessage](#)」、および「AWS Command Line Interface Reference」の「[decode-authorization-message](#)」を参照してください。

## Amazon EC2 API アクションでサポートされるリソースと条件

次の表では、現在リソースレベルのアクセス許可をサポートしている Amazon EC2 API アクションと、サポートされるリソースとその ARN、および各アクションの条件キーについて説明しています（サポートされるアクション、ARN、および条件キーは、今後さらに追加される予定です）。

API アクション	リソース	条件キー
AcceptVpcPeeringConnection	VPC ピア接続 <i>arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id</i>	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
	VPC <i>arn:aws:ec2:region:account:vpc/vpc-id</i> このとき、 <i>vpc-id</i> はアクセプタが所有する VPC です。	ec2:ResourceTag/tag-key ec2:Region ec2:Tenancy
AttachVolume	インスタンス <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
	ボリューム <i>arn:aws:ec2:region:account:volume/volume-id</i>	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:ResourceTag/tag-key ec2:VolumeIops ec2:VolumeSize ec2:VolumeType
AuthorizeSecurityGroupEgress	セキュリティグループ <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
AuthorizeSecurityGroupIngress	セキュリティグループ <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

API アクション	リソース	条件キー
CreateVpcPeeringConnection	VPC  <i>arn:aws:ec2:region:account:vpc/vpc-id</i>  このとき、 <i>vpc-id</i> はリクエスト VPC です。	<i>ec2:ResourceTag/tag-key</i>  <i>ec2:Region</i>  <i>ec2:Tenancy</i>
	VPC ピア接続  <i>arn:aws:ec2:region:account:vpc-peering-connection*</i>	<i>ec2:AcceptorVpc</i>  <i>ec2:Region</i>  <i>ec2:RequesterVpc</i>
DeleteCustomerGateway	カスタマーゲートウェイ  <i>arn:aws:ec2:region:account:customer-gateway/gw-id</i>	<i>ec2:Region</i>  <i>ec2:ResourceTag/tag-key</i>
DeleteDhcpOptions	DHCP オプションセット  <i>arn:aws:ec2:region:account:dhcp-options/dhcp-options-id</i>	<i>ec2:Region</i>  <i>ec2:ResourceTag/tag-key</i>
DeleteInternetGateway	インターネットゲートウェイ  <i>arn:aws:ec2:region:account:internet-gateway/igw-id</i>	<i>ec2:Region</i>  <i>ec2:ResourceTag/tag-key</i>
DeleteNetworkAcl	ネットワーク ACL  <i>arn:aws:ec2:region:account:network-acl/nac-id</i>	<i>ec2:Region</i>  <i>ec2:ResourceTag/tag-key</i>  <i>ec2:Vpc</i>
DeleteNetworkAclEntry	ネットワーク ACL  <i>arn:aws:ec2:region:account:network-acl/nac-id</i>	<i>ec2:Region</i>  <i>ec2:ResourceTag/tag-key</i>  <i>ec2:Vpc</i>
DeleteRoute	ルートテーブル  <i>arn:aws:ec2:region:account:route-table/route-table-id</i>	<i>ec2:Region</i>  <i>ec2:ResourceTag/tag-key</i>  <i>ec2:Vpc</i>
DeleteRouteTable	ルートテーブル  <i>arn:aws:ec2:region:account:route-table/route-table-id</i>	<i>ec2:Region</i>  <i>ec2:ResourceTag/tag-key</i>  <i>ec2:Vpc</i>
DeleteSecurityGroup	セキュリティグループ  <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	<i>ec2:Region</i>  <i>ec2:ResourceTag/tag-key</i>  <i>ec2:Vpc</i>

API アクション	リソース	条件キー
DeleteVolume	<p>ボリューム</p> <p><i>arn:aws:ec2:region:account:volume/volume-id</i></p>	<p>ec2:AvailabilityZone</p> <p>ec2:ParentSnapshot</p> <p>ec2:Region</p> <p>ec2:ResourceTag/tag-key</p> <p>ec2:Volumelops</p> <p>ec2:VolumeSize</p> <p>ec2:VolumeType</p>
DeleteVpcPeeringConnection	<p>VPC ピア接続</p> <p><i>arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id</i></p>	<p>ec2:AccepterVpc</p> <p>ec2:Region</p> <p>ec2:ResourceTag/tag-key</p> <p>ec2:RequesterVpc</p>
DetachVolume	<p>インスタンス</p> <p><i>arn:aws:ec2:region:account:instance/instance-id</i></p>	<p>ec2:AvailabilityZone</p> <p>ec2:EbsOptimized</p> <p>ec2:InstanceProfile</p> <p>ec2:InstanceType</p> <p>ec2:PlacementGroup</p> <p>ec2:Region</p> <p>ec2:ResourceTag/tag-key</p> <p>ec2:RootDeviceType</p> <p>ec2:Tenancy</p>
	<p>ボリューム</p> <p><i>arn:aws:ec2:region:account:volume/volume-id</i></p>	<p>ec2:AvailabilityZone</p> <p>ec2:ParentSnapshot</p> <p>ec2:Region</p> <p>ec2:ResourceTag/tag-key</p> <p>ec2:Volumelops</p> <p>ec2:VolumeSize</p> <p>ec2:VolumeType</p>

API アクション	リソース	条件キー
RebootInstances	インスタンス <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy
RejectVpcPeeringConnection	VPC ピア接続 <i>arn:aws:ec2:region:account:vpc-peering-connection/vpc-peering-connection-id</i>	ec2:AcceptorVpc ec2:Region ec2:ResourceTag/tag-key ec2:RequesterVpc
RevokeSecurityGroupEgress	セキュリティグループ <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
RevokeSecurityGroupIngress	セキュリティグループ <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc

API アクション	リソース	条件キー
RunInstances	Image <i>arn:aws:ec2:region::image/image-id</i>	ec2:ImageType ec2:Owner ec2:Public ec2:Region ec2:RootDeviceType ec2:ResourceTag/tag-key
	インスタンス <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:RootDeviceType ec2:Tenancy
	キーペア <i>arn:aws:ec2:region:account:key-pair/key-pair-name</i>	ec2:Region
	ネットワークインターフェイス <i>arn:aws:ec2:region:account:network-interface/*</i> <i>arn:aws:ec2:region:account:network-interface/eni-id</i>	ec2:AvailabilityZone ec2:Region ec2:Subnet ec2:ResourceTag/tag-key ec2:Vpc
	配置グループ <i>arn:aws:ec2:region:account:placement-group/placement-group-name</i>	ec2:Region ec2:PlacementGroupStrategy
	セキュリティグループ <i>arn:aws:ec2:region:account:security-group/security-group-id</i>	ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	スナップショット <i>arn:aws:ec2:region::snapshot/snapshot-id</i>	

API アクション	リソース	条件キー
		ec2:Owner ec2:ParentVolume ec2:Region ec2:SnapshotTime ec2:ResourceTag/tag-key ec2:VolumeSize
	サブネット am:aws:ec2:region:account:subnet/subnet-id	ec2:AvailabilityZone ec2:Region ec2:ResourceTag/tag-key ec2:Vpc
	ボリューム am:aws:ec2:region:account:volume/volume-id	ec2:AvailabilityZone ec2:ParentSnapshot ec2:Region ec2:Volumelops ec2:VolumeSize ec2:VolumeType
StartInstances	インスタンス am:aws:ec2:region:account:instance/instance-id	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/tag-key ec2:RootDeviceType ec2:Tenancy

API アクション	リソース	条件キー
StopInstances	インスタンス <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy
TerminateInstances	インスタンス <i>arn:aws:ec2:region:account:instance/instance-id</i>	ec2:AvailabilityZone ec2:EbsOptimized ec2:InstanceProfile ec2:InstanceType ec2:PlacementGroup ec2:Region ec2:ResourceTag/ <i>tag-key</i> ec2:RootDeviceType ec2:Tenancy

## Amazon EC2 のポリシーステートメントの例

以下の例では、Amazon EC2 に対して IAM ユーザーが所有するアクセス許可を制御するために使用できるポリシーステートメントを示しています。Amazon VPC 固有の IAM ポリシーの詳細については、[Controlling Access to Amazon VPC Resources](#) を参照してください。

- 1: ユーザーが AWS アカウントに属する Amazon EC2 リソースを一覧表示できるようにする (p. 501)
- 2: ユーザーがすべてのインスタンスを記述、起動、停止、開始、および終了できるようにする (p. 502)
- 3: ユーザーがすべてのインスタンスを記述でき、また特定のインスタンスのみを停止、開始、および終了できるようにする (p. 502)
- 4: ユーザーが特定インスタンスの特定ボリュームを管理できるようにする (p. 503)
- 5: 特定の設定でインスタンスを起動する許可をユーザーに与える (p. 504)
- 6: Amazon EC2 コンソールで作業する (p. 509)

Example 1: ユーザーが AWS アカウントに属する Amazon EC2 リソースを一覧表示できるようにする

次のポリシーでは、名前が `Describe` で始まるすべての Amazon EC2 API アクションを使用できるアクセス許可をユーザーに与えます。Resource エレメントにワイルドカードを使用します。これは、

ユーザーが API アクションですべてのリソースを指定できることを示します。また、API アクションがリソースレベルのアクセス許可をサポートしていない場合も、\*ワイルドカードが必要です。どの Amazon EC2 API アクションでどの ARN を使用できるかについては、[Amazon EC2 API アクションでサポートされるリソースと条件 \(p. 494\)](#) を参照してください。

デフォルトで API アクションを使用するアクセス許可が拒否されているため、ユーザーには (別のステートメントでアクセス許可が与えられない限り) そのリソースに対してアクションを実行するアクセス許可がありません。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:Describe*",
    "Resource": "*"
  }]
}
```

**Example 2: ユーザーがすべてのインスタンスを記述、起動、停止、開始、および終了できるようにする**

次のポリシーでは、Action エレメントで指定された API アクションを使用するアクセス許可をユーザーに与えます。Resource エレメントでは \*ワイルドカードを使用して、ユーザーが API アクションですべてのリソースを指定できることを示します。また、API アクションがリソースレベルのアクセス許可をサポートしていない場合も、\*ワイルドカードが必要です。どの Amazon EC2 API アクションでどの ARN を使用できるかについては、[Amazon EC2 API アクションでサポートされるリソースと条件 \(p. 494\)](#) を参照してください。

デフォルトにより、ユーザーは API アクションを使用するアクセス許可を拒否されているため、ユーザーには (別のステートメントでユーザーにそのアクセス許可を与えない限り) その他の API アクションを使用するアクセス許可がありません。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances", "ec2:DescribeImages",
      "ec2:DescribeKeyPairs", "ec2:DescribeSecurityGroups",
      "ec2:DescribeAvailabilityZones",
      "ec2:RunInstances", "ec2:TerminateInstances",
      "ec2:StopInstances", "ec2:StartInstances"
    ],
    "Resource": "*"
  }]
}
```

**Example 3: ユーザーがすべてのインスタンスを記述でき、また特定のインスタンスのみを停止、開始、および終了できるようにする**

次のポリシーでは、すべてのインスタンスを記述し、i-123abc12 と i-4c3b2a1 インスタンスのみを開始および停止し、米国東部 (バージニア北部) リージョン (us-east-1) 内でリソースタグ "purpose=test" の付いたインスタンスのみを終了する許可をユーザーに与えます。

ステートメントの最初の部分では、Resource エlement に \*ワイルドカードを使用して、ユーザーがそのアクションにすべてのリソースを指定できることを示しています。この場合、すべてのインスタスを一覧表示できます。また、API アクションがリソースレベルのアクセス許可をサポートしていない場合も、\*ワイルドカードが必要です (この場合は、ec2:DescribeInstances)。どの Amazon EC2 API アクションでどの ARN を使用できるかについては、[Amazon EC2 API アクションでサポートされるリソースと条件 \(p. 494\)](#) を参照してください。

このステートメントの 2 番目の部分では、StopInstances および StartInstances アクションに対してリソースレベルのアクセス許可を使用しています。Resource Element 内で、ARN によって特定のインスタスが指定されています。

ステートメントの 3 番目の部分は、特定の AWS アカウントに属する 米国東部 (バージニア北部) リージョン (us-east-1) 内にあり、タグ "purpose=test" が付けられているすべてのインスタスを終了する許可をユーザーに与えています。Condition Element は、ポリシーステートメントの発効条件を指定します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances"
      ],
      "Resource": [
        "arn:aws:ec2:us-east-1:123456789012:instance/i-123abc12",
        "arn:aws:ec2:us-east-1:123456789012:instance/i-4c3b2a1"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:TerminateInstances",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/purpose": "test"
        }
      }
    }
  ]
}
```

#### Example 4: ユーザーが特定インスタスの特定ボリュームを管理できるようにする

API アクションが複数のリソースを指定するために発信者を必要とする場合、ユーザーがすべての必要なリソースにアクセスできるようにポリシーステートメントを作成する必要があります。1 つ以上のリソースで Condition Element を使用する必要がある場合、この例のとおり複数のステートメントを作成する必要があります。

以下のポリシーでは、ユーザーがタグ「volume\_user=iam-user-name」の付いたボリュームを、タグ「department=dev」の付いたインスタスにアタッチしたり、またインスタスからボリュームを

デタッチしたりできるようにします。このポリシーを IAM グループにアタッチする場合、aws:username ポリシー変数によってグループの IAM ユーザーに、値として IAM ユーザー名を持つタグ名が volume\_user のインスタンスからボリュームをアタッチまたはデタッチするためのアクセス許可が付与されます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/department": "dev"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:us-east-1:123456789012:volume/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/volume_user": "${aws:username}"
      }
    }
  }
]
}
```

#### Example 5: 特定の設定でインスタンスを起動する許可をユーザーに与える

[RunInstances](#) API アクションでは、1 つ以上のインスタンスを起動します。RunInstances は AMI を必要とし、インスタンスを作成します。ユーザーは、リクエスト内でキーペアとセキュリティグループを指定できます。EC2-VPC 内に起動するにはサブネットが必要であり、起動されるとネットワークインターフェイスが作成されます。Amazon EBS-Backed AMI から起動すると、ボリュームが作成されます。そのため、ユーザーにはこれらの Amazon EC2 リソースを使用するアクセス許可が必要です。発信者は、インスタンスタイプおよびサブネットなどの RunInstances に対してオプションのパラメーターを使用して、インスタンスを設定することもできます。ユーザーがオプションのパラメーターを指定する必要がある、またはユーザーからパラメーターの特定の値を制限するポリシーステートメントを作成できます。このセクションの例では、ユーザーが起動できるインスタンスの設定を制御できるさまざまな方法について説明します。

デフォルトでは、ユーザーに作成したインスタンスを記述、開始、停止、または終了するアクセス許可はありません。作成したインスタンスを管理するアクセス許可をユーザーに付与する 1 つの方法としては、インスタンスごとに特定のタグを作成し、そのタグでインスタンスを管理できるようにステートメントを作成します。詳細については、「[例 3: ユーザーが特定のインスタンスのみを停止および開始できるようにする \(p. 502\)](#)」を参照してください。

[a. AMI]

次のポリシーでは、インスタンスに関連付けられた特定のタグ「department=dev」のある AMI のみを使用して、インスタンスを起動できます。最初のステートメントの Condition エレメントはユーザーがこのタグのある AMI を指定している必要があるため、ユーザーがその他の AMI を使用してインスタンスを起動することはできません。また、このポリシーはサブネットとネットワークインターフェイスリソースへのアクセス許可を与えていないため、ユーザーはサブネットに起動することができません。ただし、EC2-Classic に起動することはできます。2 番目のステートメントではワイルドカードを使用してユーザーがインスタンスリソースを作成できるようにしており、ユーザーはキーペア project\_keypair およびセキュリティグループ sg-1a2b3c4d を指定する必要があります。ただし、ユーザーはキーペアがなくてもインスタンスを起動できます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/department": "dev"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/project_keypair",
      "arn:aws:ec2:region:account:security-group/sg-1a2b3c4d"
    ]
  }
]
```

また、次のポリシーでは、ユーザーが指定された AMI ( ami-9e1670f7 および ami-45cf5c3c ) のみを使用してインスタンスを起動できます。ユーザーは他の AMI を使用してインスタンスを起動することはできず ( 他のステートメントがそのような許可をユーザーに与えている場合はその限りではありません )、ユーザーはインスタンスをサブネットに起動することはできません。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-9e1670f7",
      "arn:aws:ec2:region::image/ami-45cf5c3c",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group/*"
    ]
  }
]
```

```
]
}
```

一方、以下のポリシーは、Amazon が所有するすべての AMI からインスタンスを起動することをユーザーに許可します。最初のステートメントの Condition エレメントは、ec2:Owner が amazon であるかどうかをテストします。(別のステートメントでユーザーに起動するアクセス許可が付与されない限り) ユーザーは、その他のAMIを使用してインスタンスを起動することはできません。ユーザーは、インスタンスをサブネットに起動することができます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:Owner": "amazon"
      }
    }
  }],
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:subnet/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group*"
    ]
  }
]
```

#### [b. Instance type]

次のポリシーにより、ユーザーは t1.micro または m1.small インスタンスタイプのみを使用してインスタンスを起動できます。これにより、コストを管理することができます。最初のステートメントの Condition エレメントは、ec2:InstanceType が t1.micro または m1.small のどちらであるかをテストするため、ユーザーは大きなインスタンスを起動することはできません。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:instance/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:InstanceType": ["t1.micro", "m1.small"]
      }
    }
  }]
```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account:subnet/*",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group/*"
    ]
  }
]
}

```

### [c. Subnet]

次のポリシーにより、ユーザーは指定したサブネット `subnet-12345678` のみを使用してインスタンスを起動できます。グループは、インスタンスを他のサブネットに起動することはできません（他のステートメントがそのような許可をユーザーに与えている場合はその限りではありません）。ただし、ユーザーは EC2-Classic にインスタンスを起動できます。

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:subnet/subnet-12345678",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group/*"
    ]
  }
]
}

```

また、ユーザーがその他のサブネットにインスタンスを起動するアクセス許可を拒否するポリシーを作成することもできます。ステートメントでは、サブネット `subnet-12345678` が指定されている場合以外は、ネットワークインターフェイスの作成を拒否することでこれを実行します。この拒否は、他のサブネットへのインスタンスの起動を許可する他のすべてのポリシーよりも優先されます。ただし、ユーザーは EC2-Classic にインスタンスを起動できます。

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Deny",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region:account:network-interface/*"
    ]
  }
]
}

```

```
    ],
    "Condition": {
      "ArnNotEquals": {
        "ec2:Subnet": "arn:aws:ec2:region:account:subnet/subnet-12345678"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account:network-interface/*",
      "arn:aws:ec2:region:account:instance/*",
      "arn:aws:ec2:region:account:subnet/*",
      "arn:aws:ec2:region:account:volume/*",
      "arn:aws:ec2:region:account:key-pair/*",
      "arn:aws:ec2:region:account:security-group*"
    ]
  }
]
```

## Example 6: Amazon EC2 コンソールで作業する

IAM ポリシーを使用して、Amazon EC2 コンソールで特定のリソースを表示、および操作するアクセス許可をユーザーに付与することができます。上記のポリシーの例を使用することはできませんが、これらは AWS CLI、Amazon EC2 CLI、または AWS SDK で作成されたリクエスト用に設計されています。コンソールではこの機能を実行するために追加の API アクションを使用するので、これらのポリシーは正常に動作しない可能性があります。たとえば、DescribeVolumes API アクションのみを使用するアクセス許可を持つユーザーがコンソールでボリュームを表示しようとすると、エラーが発生します。このセクションでは、コンソールの特定の部分をユーザーが操作できるようになるポリシーを説明します。



## Note

コンソールでタスクを実行するために必要な API アクションを探すには、AWS CloudTrail などのサービスを使用できます。詳細については、[AWS CloudTrail User Guide](#) を参照してください。ポリシーにより特定のリソースを作成または変更するアクセス許可が付与されない場合、コンソールではエンコードされた診断情報のメッセージが表示されます。AWS STS の [DecodeAuthorizationMessage](#) API アクション、または AWS CLI の [decode-authorization-message](#) コマンドを使用してメッセージをデコードできます。

## a. 読み取り専用アクセス

ユーザーが Amazon EC2 コンソールですべてのリソースを表示するには、上記の例と同じポリシーを使用します: [1: ユーザーが AWS アカウントに属する Amazon EC2 リソースを一覧表示できるようにする \(p. 501\)](#)。別のステートメントによりユーザーにアクセス許可が与えられない限り、ユーザーはリソースのアクションを実行したり新しいリソースを作成することができません。

代わりに、リソースのサブセットへの読み取り専用アクセスを提供できます。これを行うには、ec2:Describe API アクションの \* (ワイルドカード) を各リソースの固有の ec2:Describe アクションに置き換えます。次のポリシーによりユーザーは Amazon EC2 コンソールですべてのインスタンス、AMI、およびスナップショットを表示できます。ec2:DescribeTags アクションにより、ユーザーはパブリック AMI を表示することができます。プライベート AMI のみを表示する場合は、このアクションを削除できます。

```
{
  "Version": "2012-10-17",
  "Statement": [ {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances", "ec2:DescribeImages",
      "ec2:DescribeTags", "ec2:DescribeSnapshots"
    ],
    "Resource": "*"
  } ]
}
```



## Note

現在、Amazon EC2 ec2:Describe\* API アクションは、リソースレベルのアクセス許可をサポートしていません。そのため、ユーザーがコンソールで表示できる個人のリソースを制御できません。したがって、上記のステートメントの Resource エlement には、\* (ワイルドカード) が必要です。どの Amazon EC2 API アクションでどの ARN を使用できるかについては、[Amazon EC2 API アクションでサポートされるリソースと条件 \(p. 494\)](#) を参照してください。

## b. EC2 起動ウィザードを使用する

Amazon EC2 起動ウィザードは、インスタンスを設定し、起動するためのオプションを提供する一連の画面です。ユーザーがウィザードのオプションを操作できるように、APIアクションを使用するアクセス許可をポリシーに含める必要があります。ポリシーにそれらのアクションを使用するアクセス許可が含まれない場合、ウィザードの一部の項目は適切にロードされず、ユーザーは起動を完了できません。

起動を正常に完了させるには、ユーザーに `ec2:RunInstances` API アクションを使用するアクセス許可を付与し、少なくとも以下の API アクションを使用できるようにする必要があります。

- `ec2:DescribeImages`: AMI を表示して選択します。
- `ec2:DescribeVPCs`: 利用可能なネットワークオプション ( EC2-Classic と VPC の一覧 ) を表示します。これは、VPC 内で起動しない場合でも必須です。
- `ec2:DescribeSubnets`: VPC 内で起動している場合、選択した VPC で利用可能なすべてのサブネットを表示します。
- `ec2:DescribeSecurityGroups` または `ec2:CreateSecurityGroup`: 既存のセキュリティグループを選択する、または新しいセキュリティグループを作成します。
- `ec2:DescribeKeyPairs` または `ec2:CreateKeyPair`: 既存のキーペアを選択する、または新しいキーペアを作成します。

```
{
  "Version": "2012-10-17",
  "Statement": [ {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances", "ec2:DescribeImages",
      "ec2:DescribeKeyPairs", "ec2:DescribeVpcs", "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "*"
  }
 ]
}
```

ポリシーに次のような API アクションを追加して、ユーザーに追加のオプションを提供できます。

- `ec2:DescribeAvailabilityZones`: EC2 Classic 内で起動している場合、特定のアベイラビリティゾーンを表示して選択します。
- `ec2:DescribeNetworkInterfaces`: VPC 内で起動している場合、選択したサブネット内に存在するネットワークインターフェイスを表示して選択します。
- `ec2:CreateTags`: インスタンスにタグを追加します。起動ウィザードはデフォルトで Name のキーを含むタグをインスタンスに追加しようとします。このアクションを使用するアクセス許可を持たないユーザーには、このタグをインスタンスに適用できなかったという警告が表示されます。ただし、これが起動の成功に影響を与えることはありません。したがって絶対に必要な場合のみユーザーにこのアクションを使用するアクセス許可を付与することをお勧めします。



#### Important

ユーザーに `ec2:CreateTags` アクションを使用するアクセス許可を付与するには注意が必要です。これにより、`ec2:ResourceTag` 条件キーを使用する能力が限定され、他のリソースの使用が制限されます。ユーザーは、リソースのタグを変更してその制限を回避できません。

現在、Amazon EC2 `Describe*` API アクションは、リソースレベルのアクセス許可をサポートしていません。そのため、ユーザーが起動ウィザードで表示できる個人のリソースを制限することはできません。ただし、`ec2:RunInstances` API アクションにリソースレベルのアクセス許可を適用して、ユーザーがインスタンスの起動に使用できるリソースを制限できます。ユーザーが使用する権限がないオプションを選択すると、起動は失敗します。

次のポリシーにより、ユーザーは Amazon が所有する AMI を使用して `m1.small` インスタンスを特定のサブネット (`subnet-1a2b3c4d`) でのみ起動することができます。ユーザーは `sa-east-1` リージョンでのみ起動できます。ユーザーが異なるリージョンを選択するか、起動ウィザードで異なるインスタンスタイプ、AMI、またはサブネットを選択すると、起動は失敗します。

ステートメントの最初の部分では、上記の例に示したように、起動ウィザードでオプションを表示するアクセス許可がユーザーに付与されます。ステートメントの 2 番目の部分では、`ec2:RunInstances` アクションでネットワークインターフェイス、ボリューム、キーペア、セキュリティグループ、サブネットリソースを使用するアクセス許可が付与されます。これは、ユーザーが VPC でインスタンスを起動するために必要です。`ec2:RunInstances` アクションの使用方法の詳細については、「[5: 特定の設定でインスタンスを起動する許可をユーザーに与える \(p. 504\)](#)」を参照してください。ステートメントの 3 番目と 4 番目の部分では、インスタンスと AMI リソースを使用するアクセス許可がそれぞれ付与されますが、インスタンスが `m1.small` インスタンスの場合のみ、および AMI が Amazon によって所有されている場合のみ付与されます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeInstances", "ec2:DescribeImages",
      "ec2:DescribeKeyPairs", "ec2:DescribeVpcs", "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:sa-east-1:111122223333:network-interface/*",
      "arn:aws:ec2:sa-east-1:111122223333:volume/*",
      "arn:aws:ec2:sa-east-1:111122223333:key-pair/*",
      "arn:aws:ec2:sa-east-1:111122223333:security-group/*",
      "arn:aws:ec2:sa-east-1:111122223333:subnet/subnet-1a2b3c4d"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:sa-east-1:111122223333:instance/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:InstanceType": "m1.small"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:sa-east-1::image/ami-*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:Owner": "amazon"
      }
    }
  }
  ]
}
```

### c. ボリュームを操作する

次のポリシーは、ボリュームを表示して作成し、特定のインスタンスにボリュームをアタッチ、およびデタッチするアクセス許可をユーザーに付与します。

ユーザーは、"purpose=test"というタグを含むインスタンスに対してどのボリュームもアタッチできます。同様に、それらのインスタンスからボリュームをデタッチすることもできます。Amazon EC2 コンソールを使用してボリュームをアタッチするには、ユーザーに `ec2:DescribeInstances` アクションを使用するアクセス許可があると、[Attach Volume] ダイアログボックスのあらかじめ用意されたリストからインスタンスを選択できるため、役立ちます。ただし、これにより、コンソールの [Instances] ページでもすべてのインスタンスが表示されるため、このアクションを省略することもできます。

ステートメントの最初の部分の `ec2:DescribeVolumeStatus` および `ec2:DescribeAvailabilityZones` アクションは、コンソールでボリュームを正しく表示するために必須です。

```
{
  "Version": "2012-10-17",
  "Statement": [ {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVolumes", "ec2:DescribeVolumeStatus",
      "ec2:DescribeAvailabilityZones", "ec2:CreateVolume",
      "ec2:DescribeInstances"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:instance/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/purpose": "test"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AttachVolume",
      "ec2:DetachVolume"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:volume/*"
  }
  ]
}
```

#### d. セキュリティグループを操作する

次のポリシーは、Amazon EC2 コンソールでセキュリティグループを表示し、タグ `Department=Test` を含む既存のセキュリティグループに対してインバウンドおよびアウトバウンドのルールを追加および削除するアクセス許可をユーザーに付与します。



#### Note

EC2-Classical セキュリティグループのアウトバウンドルールを変更することはできません。セキュリティグループの詳細については、[Amazon EC2 セキュリティグループ \(p. 477\)](#) を参照してください。

ステートメントの最初の部分の `ec2:DescribeTags` アクションにより、ユーザーはコンソールでタグを表示できます。これにより、ユーザーは変更できるセキュリティグループをより簡単に識別できます。

```
{
  "Version": "2012-10-17",
  "Statement": [ {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeSecurityGroups", "ec2:DescribeTags"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AuthorizeSecurityGroupIngress", "ec2:RevokeSecurityGroupIngress",
      "ec2:AuthorizeSecurityGroupEgress", "ec2:RevokeSecurityGroupEgress"
    ],
    "Resource": [
      "arn:aws:ec2:region:111122223333:security-group/*"
    ],
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/Department": "Test"
      }
    }
  }
 ]
}
```

ユーザーが Amazon EC2 コンソールの [Create Security Group] ダイアログボックスを使用して作業できるようにするポリシーを作成できます。このダイアログボックスを使用するには、ユーザーに少なくとも以下の API アクションを使用するアクセス許可を付与する必要があります。

- `ec2:CreateSecurityGroup`: 新しいセキュリティグループを作成するには。
- `ec2:DescribeVpcs`: [VPC] リストに既存の VPC のリストを表示します。VPC のセキュリティグループを作成していなくてもこのアクションは必須です。

これらのアクセス許可で、ユーザーは新しいセキュリティグループを正常に作成できますが、ルールを追加することはできません。[Create Security Group] ダイアログボックスでルールを操作するには、ポリシーに次の API アクションを追加します。

- `ec2:AuthorizeSecurityGroupIngress`: インバウンドルールを追加します。
- `ec2:AuthorizeSecurityGroupEgress`: VPC セキュリティグループにアウトバウンドルールを追加します。
- `ec2:RevokeSecurityGroupIngress`: 既存のインバウンドルールを変更または削除します。これは、ユーザーがコンソールで [Copy to new] 機能を使用する場合に役に立ちます。この機能により、[Create Security Group] ダイアログボックスが開き、選択したセキュリティグループと同じルールが追加されます。
- `ec2:RevokeSecurityGroupEgress`: VPC セキュリティグループのアウトバウンドルールを変更または削除します。これは、すべてのアウトバウンドトラフィックを許可するデフォルトのアウトバウンドルールを変更または削除する場合に役に立ちます。
- `ec2>DeleteSecurityGroup`: 無効なルールを保存できない状況に対応します。ユーザーが無効なルールを持つセキュリティグループを作成する場合、コンソールは最初にセキュリティグループを作成し、ルールを追加しようとします。ルールの追加に失敗した後、セキュリティグループは削除されます。ユーザーには [Create Security Group] ダイアログボックスにエラーが表示されたままです。ルールは一覧に表示されたままなので、ユーザーは無効なルールを修正し、セキュリティグループを再作成して試みることができます。この API アクションは必須ではありませんが、ユーザーがこのアクションを使用するアクセス許可が付与されておらず、無効なルールを持つセキュリティグループを作成しようとすると、ルールのないセキュリティグループが作成され、後でルールを追加することが必要になります。

現在、`ec2:CreateSecurityGroup` API アクションは、リソースのレベルのアクセス許可をサポートしていません。ただし、`ec2:AuthorizeSecurityGroupIngress` および `ec2:AuthorizeSecurityGroupEgress` アクションにリソースレベルのアクセス許可を適用してルールを作成する方法を制御できます。

次のポリシーは、[Create Security Group] ダイアログボックスを使用し、特定の VPC (`vpc-1a2b3c4d`) に関連付けられたセキュリティグループに対してインバウンドおよびアウトバウンドのルールを作成するアクセス許可をユーザーに付与します。ユーザーは EC2-Classical または別の VPC のセキュリティグループを作成できますが、ルールを追加することはできません。同様に、ユーザーは VPC `vpc-1a2b3c4d` に関連付けられていない既存のセキュリティグループにルールを追加することもできません。ユーザーには、コンソールですべてのセキュリティグループを表示するアクセス許可も付与されます。これにより、ユーザーはインバウンドルールを追加するセキュリティグループをより簡単に識別できるようになります。

このポリシーは、ユーザーに VPC `vpc-1a2b3c4d` に関連付けられたセキュリティグループを削除するアクセス許可も付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeSecurityGroups", "ec2:CreateSecurityGroup", "ec2:De
scribeVpcs"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2>DeleteSecurityGroup", "ec2:AuthorizeSecurityGroupIngress",
      "ec2:AuthorizeSecurityGroupEgress"
    ],
    "Resource": "arn:aws:ec2:region:111122223333:security-group/*",
    "Condition": {
      "ArnEquals": {
        "ec2:Vpc": "arn:aws:ec2:region:111122223333:vpc/vpc-1a2b3c4d"
      }
    }
  }
]
}
```

## Amazon EC2 の IAM ロール

### Abstract

IAM ロールを作成して、それを Amazon EC2 インスタンスに割り当てると、アプリケーションは、セキュリティが確保された API リクエストを実行することを許可します。

アプリケーションは AWS 認証情報で API リクエストに署名する必要があります。したがって、アプリケーション開発者である場合、EC2 インスタンスで実行するアプリケーションの認証情報を管理する戦略が必要です。例えば、インスタンスに AWS 認証情報を安全に配布することができ、他のユーザーから保護しながら、インスタンスのアプリケーションを有効にし、リクエストに署名するために認証情報を使用できます。ただし、各インスタンスに認証情報を安全に配布することは難しく、特に AWS が代理で作成するスポットインスタンスや Auto Scaling グループのインスタンスなどではそれが顕著です。また、AWS 認証情報を循環させる場合、各インスタンスの認証情報を更新できる必要もあります。

アプリケーションが使用するセキュリティ認証情報をお客様が管理する必要なく、アプリケーションがインスタンスから API リクエストを安全に作成できるように、IAM ロールをデザインしました。AWS 認証情報を作成および配布する代わりに、以下の IAM ロールを使用して API リクエストを作成するアクセス許可を委任できます。

1. IAM ロールを作成する
2. ロールを行うアカウントまたは AWS サービスを定義する
3. ロールを仮定してからアプリケーションが使用できる API アクションおよびリソースを定義する
4. インスタンスを起動する際にロールを指定する
5. 一時的な認証情報のセットを取得およびそれらを使用するアプリケーションを入手する

例えば、IAM ロールを使用し、Amazon S3 のバケットを使用する必要があるインスタンスで実行中のアプリケーションに、アクセス許可を与えることができます。

JSON 形式のポリシーを作成することにより、IAM ロールのアクセス許可を指定できます。これらのポリシーは、IAM ユーザー用に作成するポリシーに類似しています。ロールに変更を加えると、認証情報の管理が単純化され、変更がすべてのインスタンスに反映されます。



#### Note

ロールを既存のインスタンスに割り当てることはできません。ロールは新しいインスタンスを起動するときのみ指定できます。

IAM ロールの作成と使用の詳細については、*IAM を使用するガイド*の [Roles](#) を参照してください。

#### Topics

- [インスタンスメタデータからセキュリティ認証情報を取得する \(p. 517\)](#)
- [IAM ロールを使用してインスタンスを起動するための、IAM ユーザーアクセス許可の付与 \(p. 518\)](#)
- [コンソールから IAM ロールを使用してインスタンスを起動する \(p. 518\)](#)
- [AWS CLI から IAM ロールを使用してインスタンスを起動する \(p. 519\)](#)
- [AWS SDK を使用して IAM ロールでインスタンスを起動する \(p. 521\)](#)

## インスタンスメタデータからセキュリティ認証情報を取得する

インスタンスのアプリケーションは、インスタンスメタデータアイテム `iam/security-credentials/role-name` のロールから提供されたセキュリティ認証情報を取得します。アプリケーションには、ロールに関連付けられたセキュリティ認証情報によって、ロールに対して定義したアクションおよびリソースのアクセス許可が付与されます。これらのセキュリティ認証情報は一時的なものであり、私たちが自動的に循環させます。新しい認証情報は、古い認証情報が失効する少なくとも 5 分前から有効になるようにします。



#### Warning

IAM ロールでインスタンスメタデータを使用するサービスを使用する場合は、サービスが HTTP 呼び出しを行うときに、お客様の認証情報が公開されないように注意する必要があります。認証情報を公開できるサービスには、HTTP プロキシ、HTML/CSS 検証サービス、および XML インクルードをサポートする XML プロセッサなどがあります。

以下のコマンドでは、`s3access` という名前の IAM ロールのセキュリティ認証情報を取得します。

```
GET http://169.254.169.254/latest/meta-data/iam/security-credentials/s3access
```

出力例を次に示します。

```
{
  "Code" : "Success",
  "LastUpdated" : "2012-04-26T16:39:16Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "AKIAIOSFODNN7EXAMPLE",
  "SecretAccessKey" : "wJalrXUtnFEMI/K7MDENG/bPxrFfiCYEXAMPLEKEY",
  "Token" : "token",
  "Expiration" : "2012-04-27T22:39:16Z"
}
```

インスタンスのメタデータの詳細については、「[インスタンスメタデータとユーザーデータ \(p. 289\)](#)」を参照してください。一時的な認証情報の詳細については、IAM ドキュメントの [一時的なセキュリティ認証情報の使用](#) を参照してください。

## IAM ロールを使用してインスタンスを起動するための、IAM ユーザーアクセス許可の付与

IAM ユーザーが IAM ロールでインスタンスを起動できるようにするには、ロールをインスタンスにわたすユーザーアクセス許可を与える必要があります。

例えば以下の IAM ポリシーは、`s3access` という名前の IAM ロールを使用してインスタンスを起動するアクセス許可をユーザーに与えます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::123456789012:role/s3access"
  }]
}
```

または、ポリシー内でリソースを「\*」と指定することで、IAM ユーザーに対し、お客様のすべてのロールへのアクセス許可を与えることができます。ただし、お客様のロール（既存のロールおよび今後作成するロールを含む）を使用してインスタンスを起動するユーザーに、必要ではない、または与えるべきではないアクセス許可が与えられる可能性があることを考慮してください。

詳細については、IAM を使用するガイドの [Permissions Required for Using Roles with Amazon EC2](#) を参照してください。

## コンソールから IAM ロールを使用してインスタンスを起動する

ロールでインスタンスを起動するには、事前に IAM ロールを作成する必要があります。



### Important

IAM ロールを作成した後、適切なアクセス許可が反映されるまで数秒ほどかかります。ロールを使用した最初のインスタンスの起動が失敗した場合は、数秒待ってからもう一度試してください。詳細については、IAM を使用するの [Troubleshooting Working with Roles](#) を参照してください。

AWS Management Console から IAM ロールを使用してインスタンスを起動するには

1. IAM ロールを作成する
  - a. IAM コンソールを開きます。
  - b. ナビゲーションペインで [Roles] をクリックし、続いて [Create New Role] をクリックします。
  - c. 最初の [CONFIGURE ROLE] ページで、役割の名前を入力して [Continue] をクリックします。
  - d. 2 番目の [CONFIGURE ROLE] ページで、[Amazon EC2] の隣にある [Select] をクリックします。
  - e. [SET PERMISSIONS] ページで、グループのポリシーを指定します。ポリシーテンプレートを選択するか、カスタムポリシーを作成できます。例えば、Amazon EC2 では以下のポリシーテンプレートのいずれかが、ニーズを満たす場合があります。
    - パワーユーザーアクセス
    - 読み取り専用アクセス
    - Amazon EC2 フルアクセス
    - Amazon EC2 読み取り専用アクセス

カスタムポリシーを作成する詳細については、[Amazon EC2 の IAM ポリシー \(p. 488\)](#) を参照してください。

- f. 2 番目の [SET PERMISSIONS] ページでは、自動的に生成されたポリシー名を任意の名前に変更できます。ポリシードキュメントで詳細を確認し、[Continue] をクリックします。
  - g. ロール情報を確認し、必要に応じてロールを編集し、[Create Role] をクリックします。
2. IAM ロールを使用してインスタンスを起動します。
    - a. Amazon EC2 コンソールを開きます。
    - b. ダッシュボードで、[Launch Instance] をクリックします。
    - c. AMI を選択し、インスタンスタイプを選択してから、[Next: Configure Instance Details] をクリックします。
    - d. [Configure Instance Details] ページで、作成した IAM ロールを [IAM role] リストから選択します。
    - e. その他の詳細を設定し、ウィザードの残りの部分の指示にしたがうか、または [Review and Launch] をクリックしてデフォルト設定を許可し、直接 [Review Instance Launch] ページに移動します。
    - f. 設定を確認し、[Launch] をクリックして、キーペアを選択し、インスタンスを起動します。
  3. アプリケーションで Amazon EC2 API アクションを使用している場合、インスタンスで有効にされている AWS セキュリティ認証情報を取得し、それを使用しリクエストに署名します。これは AWS SDK によって実行されますのでご注意ください。

```
GET http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

## AWS CLI から IAM ロールを使用してインスタンスを起動する

ロールでインスタンスを起動するには、事前に IAM ロールを作成する必要があります。



### Important

IAM ロールを作成した後、適切なアクセス許可が反映されるまで数秒ほどかかります。ロールを使用した最初のインスタンスの起動が失敗した場合は、数秒待ってからもう一度試してください。詳細については、[IAM を使用するの Troubleshooting Working with Roles](#) を参照してください。

IAM および Amazon EC2 CLI を使用して IAM ロールでインスタンスを起動するには

1. Amazon S3 バケットの使用を許可するポリシーを設定した IAM ロールを作成します。
  - a. 以下の信頼ポリシーを作成し、`ec2-role-trust-policy.json` という名前のテキストファイルに保存します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com" },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
}  
]  
}
```

- b. s3access ロールを作成します。作成した信頼ポリシーを指定します。

```
aws iam create-role --role-name s3access --assume-role-policy-document  
file://ec2-role-trust-policy.json  
{  
  "Role": {  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Action": "sts:AssumeRole",  
          "Effect": "Allow",  
          "Principal": {  
            "Service": "ec2.amazonaws.com"  
          }  
        }  
      ]  
    },  
    "RoleId": "AROAIIZKPBKS2LEXAMPLE",  
    "CreateDate": "2013-12-12T23:46:37.247Z",  
    "RoleName": "s3access",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:role/s3access"  
  }  
}
```

- c. アクセスポリシーを作成し、ec2-role-access-policy.jsonという名前のテキストファイルに保存します。例えば、このポリシーは、インスタンスで実行しているアプリケーションに対し、Amazon S3 の管理権限を与えます。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": ["s3:*"],  
      "Resource": ["*"]  
    }  
  ]  
}
```

- d. アクセスポリシーをロールに付与します。

```
aws iam put-role-policy --role-name s3access --policy-name S3-Permissions  
--policy-document file://ec2-role-access-policy.json
```

- e. s3access-profile という名前のインスタンスプロファイルを作成します。

```
aws iam create-instance-profile --instance-profile-name S3-Permissions
{
  "InstanceProfile": {
    "InstanceId": "AIPAJTLPJLEGREXAMPLE",
    "Roles": [],
    "CreateDate": "2013-12-12T23:53:34.093Z",
    "InstanceProfileName": "S3-Permissions",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:instance-profile/S3-Permissions"
  }
}
```

- f. s3access-profile インスタンスプロファイルに s3access ロールを追加します。

```
aws iam add-role-to-instance-profile --instance-profile-name S3-Permissions --role-name s3access
```

これらのコマンドの詳細については、*AWS Command Line Interface Reference* にある [create-role](#)、[put-role-policy](#) および [create-instance-profile](#) を参照してください。

2. インスタンスプロファイルを使用してインスタンスを起動します。以下の例では、インスタンスプロファイルを使用して t1.micro インスタンスを起動する方法を示しています。

```
aws ec2 run-instances --instance-type t1.micro --image-id ami-e565ba8c --iam-instance-profile Name="S3-Permissions" --key-name my-key-pair --security-groups my-security-group
```

詳細については、*AWS Command Line Interface Reference* の [run-instances](#) を参照してください。

3. アプリケーションで Amazon EC2 API アクションを使用している場合、インスタンスで有効にされている AWS セキュリティ認証情報を取得し、それを使用しリクエストに署名します。これは AWS SDK によって実行されますのでご注意ください。

```
GET http://169.254.169.254/latest/meta-data/iam/security-credentials/role_name
```

## AWS SDK を使用して IAM ロールでインスタンスを起動する

AWS SDK を使用しアプリケーションに書き込みを行う場合、現在のインスタンスと関連付けられたロールから一時的なセキュリティ認証情報を自動的に取得します。AWS SDK ドキュメントには、アプリケーションが Amazon S3 バケットを読み取れるように IAM ロールからセキュリティ認証情報をどのように使用できるかを示すチュートリアルが含まれています。詳細については、次の SDK ドキュメントのトピックを参照してください。

- [Using IAM Roles for EC2 Instances with the SDK for Java](#)
- [Using IAM Roles for EC2 Instances with the SDK for .NET](#)
- [Using IAM Roles for EC2 Instances with the SDK for PHP](#)
- [Using IAM Roles for EC2 Instances with the SDK for Ruby](#)

# インスタンス用の受信トラフィックの認可

## Abstract

セキュリティグループを作成して、Amazon EC2 インスタンスに到達可能なネットワークトラフィックを制御します。

インスタンスへのネットワークアクセスを有効にするには、インスタンスへのインバウンドトラフィックを許可する必要があります。インバウンドトラフィック用のポートを開くには、起動時にインスタンスに関連付けたセキュリティグループにルールを追加します。

以下の表に、トラフィックを許可するために設定する必要があるルールを示します。ドキュメントには、コンピュータのパブリック IP アドレスからのトラフィックを許可する方法が示されています。追加の IP アドレス範囲からのトラフィックを許可するには、許可する範囲ごとにルールを追加します。

お客様のインスタンス	プロトコル	ドキュメント
Linux	SSH (ポート 22)	<a href="#">受信 SSH トラフィックのルールの追加Linux インスタンスへ (p. 522)</a>
Windows	RDP (ポート 3389)	<a href="#">受信 RDP トラフィックのルールを Windows インスタンスに追加する (p. 523)</a>

## 開始する前に

例えば、単一のホストや信頼する特定のネットワークなど、インスタンスへのアクセスを必要とするユーザーを決定します。この場合、Amazon は、お客様のローカルシステムのパブリック IP アドレスを使用します。サービスを使用して、ローカルコンピュータのパブリック IP アドレスを取得できます。例えば、次のサービスが提供されています。<http://checkip.amazonaws.com/> IP アドレスを提供する別のサービスを検索するには、検索フレーズ「what is my IP address」を使用します。ISP 経由で、またはファイアウォールの内側から静的な IP アドレスなしで接続している場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。



### Caution

0.0.0.0/0 を使うと、すべての IP アドレスから SSH や RDP 経由でインスタンスにアクセスすることが許可されます。これはテスト環境で短時間なら許容できますが、実稼働環境で行うのは安全ではありません。実稼働環境では、特定の IP アドレスまたは特定のアドレス範囲のみ、インスタンスへのアクセスを限定します。

セキュリティグループの詳細については、「[Amazon EC2 セキュリティグループ \(p. 477\)](#)」を参照してください。

## 受信 SSH トラフィックのルールの追加Linux インスタンスへ

セキュリティグループは、関連付けられたインスタンスのファイアウォールとして動作し、インバウンドトラフィックとアウトバウンドトラフィックの両方をインスタンスレベルでコントロールします。SSH を使用して IP アドレスからインスタンスに接続できるようにするためのルールをセキュリティグループに追加します。

コンソールを使用して、インバウンド SSH トラフィック用のルールをセキュリティグループに追加するには

1. Amazon EC2 コンソールのナビゲーションペインで [Instances] をクリックします。インスタンスを選択し、[Description] タブを確認します。[Security groups] リストに、インスタンスに関連付け

られたセキュリティグループが表示されます。[view rules] をクリックして、インスタンスに対して有効なルールのリストを表示します。

2. ナビゲーションペインで、[Security Groups] をクリックします。インスタンスに関連付けられているセキュリティグループのいずれかを選択します。
3. 詳細ペインの [Inbound] タブで、[Edit] をクリックします。ダイアログで [Add Rule] をクリックし、[Type] リストから [SSH] を選択します。
4. [Source] フィールドで、コンピュータのパブリック IP アドレスを CIDR 表記で指定します。例えば、IP アドレスが 203.0.113.25 の場合、この単一の IP アドレスを CIDR 表記でリストするには 203.0.113.25/32 と指定します。会社が特定の範囲からアドレスを割り当てている場合、範囲全体 ( 203.0.113.0/24 など ) を指定します。

IP アドレスを見つける方法については、[開始する前に \(p. 522\)](#) を参照してください。

5. [Save] をクリックします。

コマンドラインを使用してセキュリティグループにルールを追加するには

次のコマンドの1つを使用できます。このコマンドは、インスタンスではなく、ローカルシステムで実行されていることを確認してください。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [authorize-security-group-ingress](#) ( AWS CLI )
- [ec2-authorize](#) ( Amazon EC2 CLI )
- [Grant-EC2SecurityGroupIngress](#) ( AWS Tools for Windows PowerShell )

## 受信 RDP トラフィックのルールを Windows インスタンスに追加する

セキュリティグループは、関連付けられたインスタンスのファイアウォールとして動作し、インバウンドトラフィックとアウトバウンドトラフィックの両方をインスタンスレベルでコントロールします。RDP を使用して IP アドレスから Windows インスタンスに接続できるようにするためのルールをセキュリティグループに追加します。

コンソールを使用して、インバウンド RDP トラフィック用のルールをセキュリティグループに追加するには

1. Amazon EC2 コンソールのナビゲーションペインで [Instances] をクリックします。インスタンスを選択し、[Description] タブを確認します。[Security groups] リストに、インスタンスに関連付けられたセキュリティグループが表示されます。[view rules] をクリックして、インスタンスに対して有効なルールのリストを表示します。
2. ナビゲーションペインで、[Security Groups] をクリックします。インスタンスに関連付けられているセキュリティグループのいずれかを選択します。
3. 詳細ペインの [Inbound] タブで、[Edit] をクリックします。ダイアログで [Add Rule] をクリックし、[Type] リストから [RDP] を選択します。
4. [Source] フィールドで、コンピュータのパブリック IP アドレスを CIDR 表記で指定します。例えば、IP アドレスが 203.0.113.25 の場合、この単一の IP アドレスを CIDR 表記でリストするには 203.0.113.25/32 と指定します。会社が特定の範囲からアドレスを割り当てている場合、範囲全体 ( 203.0.113.0/24 など ) を指定します。

IP アドレスを見つける方法については、[開始する前に \(p. 522\)](#) を参照してください。

5. [Save] をクリックします。

コマンドラインを使用してセキュリティグループにルールを追加するには

次のコマンドの1つを使用できます。このコマンドは、インスタンスではなく、ローカルシステムで実行されていることを確認してください。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [authorize-security-group-ingress](#) ( AWS CLI )
- [ec2-authorize](#) ( Amazon EC2 CLI )
- [Grant-EC2SecurityGroupIngress](#) ( AWS Tools for Windows PowerShell )

## インスタンスへのセキュリティグループの割り当て

インスタンスを起動する際に、インスタンスにセキュリティグループを割り当てることができます。ルールを追加または削除すると、それらの変更は、そのセキュリティグループを割り当てたすべてのインスタンスに自動的に適用されます。

EC2-Classic でインスタンスを起動した後でセキュリティグループを変更することはできません。VPC でインスタンスを起動した後、そのセキュリティグループを変更することができます。詳細については、*Amazon Virtual Private Cloud ユーザーガイド* の [Changing an Instance's Security Groups](#) を参照してください。

# Amazon EC2 と Amazon Virtual Private Cloud ( VPC )

## Abstract

Amazon EC2 インスタンスを VPC で起動して、インスタンスで追加の機能を利用します。

Amazon Virtual Private Cloud ( Amazon VPC ) を使用すると、アマゾン ウェブ サービス ( AWS ) クラウド内の独自の論理的に分離された領域仮想ネットワーク ( *Virtual Private Cloud ( VPC )* と呼ばれます ) を定義できます。AWS のリソース ( インスタンスなど ) を VPC 内部で起動できます。VPC は、お客様自身のデータセンターで運用される可能性のある従来のネットワークによく似ていますが、AWS のスケーラブルなインフラストラクチャを使用できるというメリットがあります。お客様の VPC はお客様が設定できます。例えば、IP アドレス範囲の選択、サブネットの作成、ルートテーブル、ネットワークゲートウェイ、セキュリティの設定などが可能です。VPC のインスタンスをインターネットに接続できます。VPC を自社のデータセンターに接続できるので、AWS クラウドをデータセンターの拡張のように活用できます。各サブネットでのリソースの保護には、セキュリティグループ、ネットワークアクセスコントロールリストなど、複数のセキュリティレイヤーを使用できます。詳細については、[Amazon Virtual Private Cloud ユーザーガイド](#) を参照してください。

## VPC を使用する利点

EC2-Classic の代わりに VPC でインスタンスを起動すると、次が可能になります。

- 開始から停止までの間に維持される静的プライベート IP アドレスをインスタンスに割り当てる
- 複数の IP アドレスをインスタンスに割り当てる
- ネットワークインターフェイスを定義し、1つまたは複数のネットワークインターフェイスをインスタンスに割り当てる
- 実行中にインスタンスのセキュリティグループメンバーシップを変更する
- インスタンスからのアウトバウンドトラフィックを制御し ( egress フィルタリング )、インスタンスへのインバウンドトラフィックを制御する ( ingress フィルタリング )

- ネットワークアクセスコントロールリスト (ACL) の形でインスタンスにアクセス制御の層を追加する
- 単一テナントハードウェアでインスタンスを実行する

## EC2-Classic と EC2-VPC の違い

インスタンスはサポートされる 2 つのプラットフォームである、EC2-Classic と EC2-VPC のいずれかで実行されます。お客様の AWS アカウントでは、インスタンスを両方のプラットフォームで起動できる場合と、EC2-VPC だけで起動できる場合があり、どちらになるかはリージョンごとに異なります。インスタンスを起動できるのが EC2-VPC だけである場合は、デフォルトの VPC が自動的に作成されます。デフォルトの VPC では、EC2-VPC の高度な機能と EC2-Classic の使いやすさという両方のメリットを得られます。詳細については、「[サポートされるプラットフォーム](#)」を参照してください。

次の表は、EC2-Classic で起動したインスタンス、デフォルト VPC で起動したインスタンス、デフォルトではない VPC で起動したインスタンスの違いをまとめたものです。

特徴	EC2-Classic	デフォルト VPC	デフォルトではない VPC
パブリック IP アドレス (Amazon のパブリック IP アドレスプールより)	インスタンスはパブリック IP アドレスを受け取ります。	デフォルトサブネットで起動したインスタンスは、パブリック IP アドレスをデフォルトで受け取ります。ただし、起動時に別のアドレスを指定した場合を除きます。	インスタンスは、起動時に別のアドレスを指定した場合を除き、パブリック IP アドレスをデフォルトで受け取りません。
プライベート IP アドレス	インスタンスは、起動するたびに、EC2-Classic の範囲に含まれるプライベート IP アドレスを受け取ります。	インスタンスはデフォルト VPC のアドレス範囲から静的プライベート IP アドレスを受け取ります。	インスタンスは VPC のアドレス範囲から静的プライベート IP アドレスを受け取ります。
複数のプライベート IP アドレス	1 つのインスタンスには 1 つのプライベート IP アドレスを選択します。複数の IP アドレスはサポートされません。	複数のプライベート IP アドレスを 1 つのインスタンスに割り当てることができます。	複数のプライベート IP アドレスを 1 つのインスタンスに割り当てることができます。
Elastic IP アドレス	停止すると、EIP とインスタンスの関連付けが解除されます。	停止しても、EIP とインスタンスの関連付けが維持されます。	停止しても、EIP とインスタンスの関連付けが維持されます。
DNS ホスト名	DNS ホスト名はデフォルトで有効化されています。	DNS ホスト名はデフォルトで有効化されています。	DNS ホスト名はデフォルトで無効化されています。
セキュリティグループ	セキュリティグループは、その他の AWS アカウントに属するセキュリティグループを参照できます。  リージョンごとに最大 500 のセキュリティグループを作成できます。	セキュリティグループは、VPC のみのセキュリティグループを参照できます。  VPC ごとに最大 100 のセキュリティグループを作成できます。	セキュリティグループは、VPC のみのセキュリティグループを参照できます。  VPC ごとに最大 100 のセキュリティグループを作成できます。

特徴	EC2-Classical	デフォルト VPC	デフォルトではない VPC
セキュリティグループの関連付け	<p>インスタンスを起動する際、セキュリティグループをいくつでも割り当てることができます。</p> <p>実行中のインスタンスのセキュリティグループは変更できません。割り当て済みのセキュリティグループのルールを変更するか、インスタンスを新しいインスタンスと置き換える必要があります (置き換えるには、まずインスタンスから AMI を作成し、この AMI から、目的のセキュリティグループを使用して新しいインスタンスを起動し、元のインスタンスから Elastic IP アドレスに関連付けられているものをすべて解除し、新しいインスタンスに関連付け、その後、元のインスタンスを終了します)。</p>	<p>最大 5 つのセキュリティグループを 1 つのインスタンスに割り当てることができます。</p> <p>セキュリティグループをインスタンスに割り当てられるのは、インスタンスの起動時と実行中です。</p>	<p>最大 5 つのセキュリティグループを 1 つのインスタンスに割り当てることができます。</p> <p>セキュリティグループをインスタンスに割り当てられるのは、インスタンスの起動時と実行中です。</p>
セキュリティグループのルール	<p>インバウンドトラフィックのみにルールを追加できます。</p> <p>最大 100 のルールをセキュリティグループに追加できます。</p>	<p>インバウンドトラフィックとアウトバウンドトラフィックのルールを追加できます。</p> <p>最大 50 のルールをセキュリティグループに追加できます。</p>	<p>インバウンドトラフィックとアウトバウンドトラフィックのルールを追加できます。</p> <p>最大 50 のルールをセキュリティグループに追加できます。</p>
Instance Option	<p>インスタンスは共有ハードウェアで実行されます。</p>	<p>共有ハードウェアまたはシングルテナントハードウェアでインスタンスを実行できます。</p>	<p>共有ハードウェアまたはシングルテナントハードウェアでインスタンスを実行できます。</p>

## Amazon VPC ドキュメント

Amazon VPC に関する詳細は、Amazon VPC ドキュメントを参照してください。

ガイド	説明
<a href="#">Amazon Virtual Private Cloud 入門ガイド</a>	Amazon VPC の実践的入門を提供します。
<a href="#">Amazon Virtual Private Cloud ユーザーガイド</a>	Amazon VPC の使用方法に関する詳細を提供します。
<a href="#">Amazon Virtual Private Cloud ネットワーク管理者ガイド</a>	ネットワーク管理者がカスタマーゲートウェイを設定する際に役立ちます。

## サポートされているプラットフォーム

### Abstract

AWS アカウントによってサポートされるプラットフォームでインスタンスを起動します。

Amazon EC2 は以下のプラットフォームをサポートします。お客様の AWS アカウントでは、インスタンスを両方のプラットフォームで起動できる場合と、EC2-VPC だけで起動できる場合があり、どちらになるかはリージョンごとに異なります。

プラットフォーム	導入時期	説明
EC2-Classic	Amazon EC2 のオリジナルリリース	お客様のインスタンスは他のユーザー様と共有する単一のフラットネットワーク内で稼働します。
EC2-VPC	Amazon VPC のオリジナルリリース	お客様のインスタンスはご自分の AWS アカウントから論理的に独立した仮想プライベートクラウド ( VPC ) 内で稼働します。

## Amazon EC2 コンソールでサポートされるプラットフォーム

Amazon EC2 コンソールを見れば、指定したリージョンでインスタンスを起動可能なプラットフォームや、そのリージョンにデフォルト VPC があるかどうかわかります。

使用するリージョンがナビゲーションバーで選択されていることを確認してください。Amazon EC2 コンソールダッシュボードの [Account Attributes] の下にある [Supported Platforms] を探します。そこに値が 2 つ ( EC2 と VPC ) あれば、どちらのプラットフォームにもインスタンスを起動できます。値が 1 つ ( VPC ) なら、EC2-VPC にもインスタンスを起動できます。

インスタンスを起動できるのが EC2-VPC だけである場合は、デフォルトの VPC が自動的に作成されます。これで、お客様がインスタンスを起動するときは、デフォルトの VPC で起動されるようになります。ただし、お客様が非デフォルト VPC を作成してその VPC でインスタンスを起動するよう指定した場合は除きます。

### EC2-VPC

ダッシュボードの [Account Attributes] に次のように表示される場合は、アカウントが EC2-VPC プラットフォームのみをサポートし、識別子 vpc-1a2b3c4d のデフォルトの VPC があることを示します。

Supported Platforms  
VPC  
Default VPC  
vpc-1a2b3c4d

アカウントが EC2-VPC だけをサポートしている場合は、起動ウィザードを使用してインスタンスを起動するときに、[Network] リストで VPC を選択し、[Subnet] リストでサブネットを選択できます。

Network ⓘ	vpc-1a2b3c4d (172.31.0.0/16) (default)	⊞ Create new VPC
Subnet ⓘ	No preference (default subnet in any Availability Zor	Create new subnet

### EC2-Classic、EC2-VPC

ダッシュボードの [Account Attributes] に次のように表示される場合は、アカウントが EC2-Classic と EC2-VPC の両方のプラットフォームをサポートすることを示します。

#### Supported Platforms

EC2  
VPC

アカウントが EC2-Classic と EC2-VPC をサポートしている場合は、[Network] リストから [Launch into EC2-Classic] を選択すると、起動ウィザードが EC2-Classic を起動します。VPC で起動するには、[Network] リストで VPC を選択し、[Subnet] リストでサブネットを選択します。

## 関連トピック

インスタンスを起動できるプラットフォームを確認する方法の詳細については、『*Amazon Virtual Private Cloud ユーザーガイド*』の「[Detecting Your Supported Platforms](#)」を参照してください。

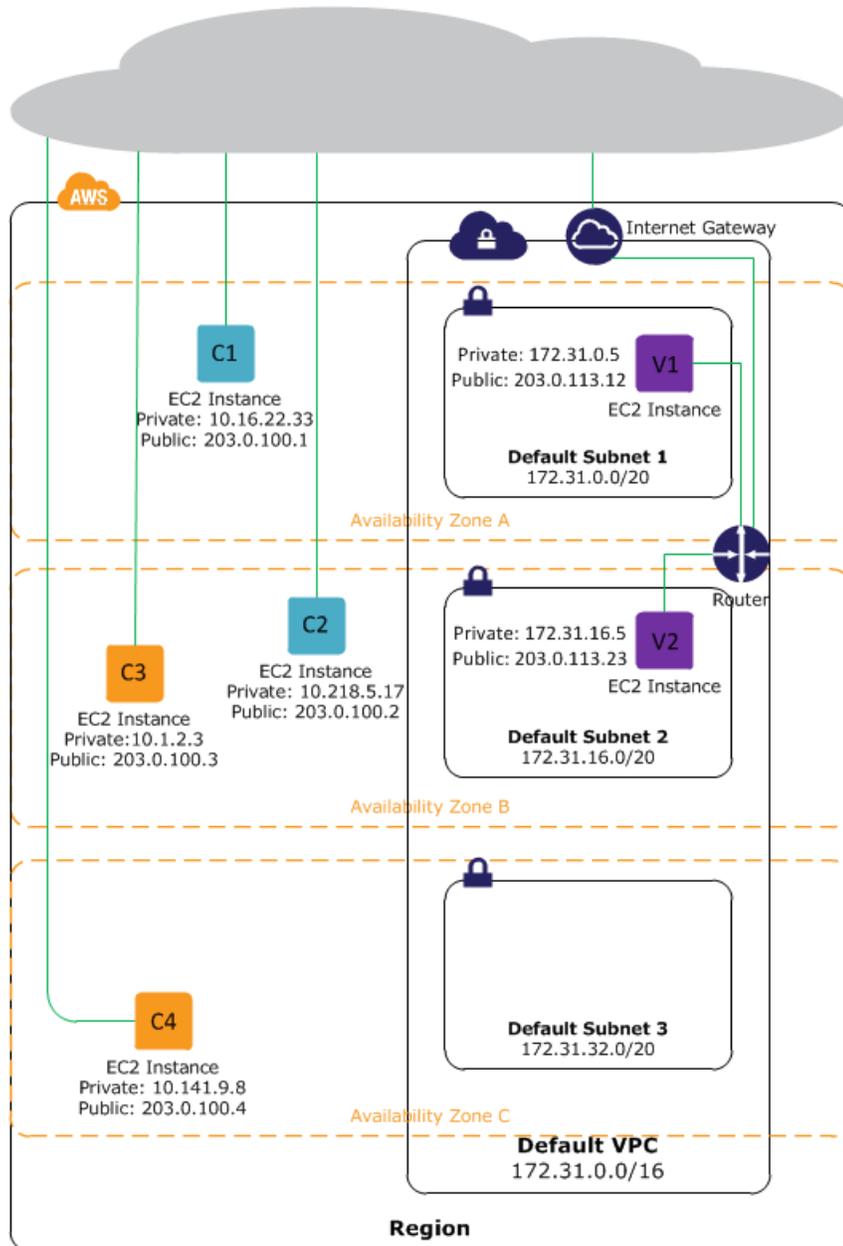
## EC2-Classic と EC2-VPC の違い

EC2-Classic では、共有プライベート IP アドレス範囲のプライベート IP アドレスが各インスタンスに割り当てられます。また、Amazon は各インスタンスに、Amazon のパブリック IP アドレスのプールからパブリック IP アドレスを割り当てます。インスタンスは、AWS ネットワークエッジを通してインターネットに直接アクセスします。

EC2-VPC では、お客様の VPC のプライベート IP アドレス範囲のプライベート IP アドレスが各インスタンスに割り当てられます。お客様は、お客様の VPC の IP アドレス範囲、サブネット、ルーティング、ネットワークゲートウェイ、ネットワーク ACL、セキュリティグループを制御できます。インスタンスが起動している間にパブリック IP アドレスを受け取るかどうかを指定できます。パブリック IP アドレスまたは Elastic IP アドレスが割り当てられたインスタンスは、AWS ネットワークエッジにアタッチされている論理インターネットゲートウェイを通してインターネットにアクセスできます。EC2-VPC の詳細については、「[What is Amazon VPC?](#)」を参照してください (『*Amazon Virtual Private Cloud ユーザーガイド*』)。

次の図では、各プラットフォームのインスタンスを示します。次のことに注意してください。

- インスタンス C1、C2、C3、C4 は、EC2-Classic プラットフォーム内にあります。C1 と C2 は 1 つのアカウントによって起動され、C3 と C4 はそれとは異なるアカウントによって起動されました。これらのインスタンスは、相互に通信したり、インターネットに直接アクセスしたり、Amazon Simple Storage Service ( Amazon S3 ) などの他のサービスにアクセスしたりできます。
- インスタンス V1 と V2 は、EC2-VPC プラットフォームの同じ VPC 内の異なるサブネットにあります。これらは VPC を所有するアカウントによって起動されました。他のアカウントはこの VPC ではインスタンスを起動できません。これらのインスタンスは、相互に通信でき、インターネットゲートウェイを通して EC2-Classic 内のインスタンス、その他のサービス ( Amazon S3 など )、インターネットにアクセスできます。



EC2-Classical と EC2-VPC の相違点の詳細については、「[Amazon EC2 と Amazon Virtual Private Cloud \(VPC\)](#) (p. 524)」を参照してください。

## Amazon EC2 インスタンスの IP アドレッシング

### Abstract

インスタンスが EC2 または VPC のどちらで起動されたかに応じて、Amazon EC2 がパブリック IP アドレス、プライベート IP アドレス、および DNS ホスト名をインスタンスに提供する方法を説明します。

インスタンスには、IP アドレスと DNS ホスト名が付与されます。これらは、インスタンスが EC2-Classic プラットフォームと Virtual Private Cloud ( VPC ) のどちらで起動されたかに応じて異なることがあります。

EC2-Classic プラットフォームと EC2-VPC プラットフォームの詳細については、「[サポートされているプラットフォーム \(p. 527\)](#)」を参照してください。Amazon VPC の詳細については、「[What is Amazon VPC?](#)」を参照してください (「[Amazon Virtual Private Cloud ユーザーガイド](#)」)。

#### Topics

- [プライベートアドレスと内部 DNS ホスト名 \(p. 530\)](#)
- [パブリック IP アドレスと外部 DNS ホスト名 \(p. 531\)](#)
- [EC2-Classic と EC2-VPC の違い \(p. 532\)](#)
- [パブリック IP アドレス、プライベート IP アドレス、Elastic IP アドレスの決定 \(p. 532\)](#)
- [パブリック ID アドレスの割り当て \(p. 533\)](#)
- [複数のプライベート IP アドレス \(p. 534\)](#)

## プライベートアドレスと内部 DNS ホスト名

同じネットワーク内 ( EC2-Classic または VPC ) のインスタンス間通信には、プライベート IP アドレスと内部 DNS ホスト名を使用できます。プライベート IP アドレスには、インターネットからは到達できません。プライベート IP アドレスについては、[RFC 1918](#) を参照してください。

インスタンスが起動されるときに、そのインスタンスのプライベート IP アドレスが DHCP を使用して割り当てられます。

VPC 内で起動される各インスタンスに、デフォルトのネットワークインターフェイスがあります。このネットワークインターフェイスでは、そのインスタンスのプライマリプライベート IP アドレスが指定されます。プライマリプライベート IP アドレスが選択されていない場合は、サブネットの範囲内の利用可能な IP アドレスの 1 つが自動的に選択されます。追加のプライベート IP アドレス ( セカンダリプライベート IP アドレス ) を指定します。プライマリプライベート IP アドレスとは異なり、セカンダリプライベート IP アドレスは、別のインスタンスに割り当て直すことができます。詳細については、「[複数のプライベート IP アドレス \(p. 534\)](#)」を参照してください。

各インスタンスには内部 DNS ホスト名が付与されます。この名前を解決すると、EC2-Classic または VPC 内のインスタンスのプライベート IP アドレスとなります。この DNS ホスト名は、そのインスタンスが存在するネットワークの外部では解決できません。

EC2-Classic でカスタムファイアウォール構成を作成する場合、Amazon DNS サーバーのアドレスからのポート 53 ( エフェメラル範囲からの送信先ポート ) からのインバウンドトラフィックを許可する必要があります。そうしない場合、インスタンスからの内部 DNS 解決に失敗します。ファイアウォールが自動的に DNS クエリレスポンスを自動的に許可しない場合は、Amazon DNS サーバーの IP アドレスからのトラフィックを許可する必要があります。Linux の Amazon DNS サーバーの IP アドレスを取得するには、コマンド `grep nameserver /etc/resolv.conf` を使用します。Windows の Amazon DNS サーバーの IP アドレスを取得するには、コマンド `ipconfig /all | findstr /c:"DNS Servers"` を使用します。

EC2-Classic で起動されたインスタンスの場合、終了または停止するまで、プライベート IP アドレスはそのインスタンスと関連付けられます。

VPC で起動されたインスタンスの場合、インスタンスが終了し、再開された場合は、プライベート IP アドレスとネットワークインターフェイスの関連付けが維持され、インスタンスが停止された場合は解放されます。

## パブリック IP アドレスと外部 DNS ホスト名

インスタンスと、インターネットまたは Amazon Simple Storage Service ( Amazon S3 ) などのその他の AWS 製品との間の通信には、パブリック IP アドレスと外部 DNS ホスト名を使用できます。パブリック IP アドレスには、インターネットから到達できます。

インスタンスを EC2-Classic で起動すると、そのインスタンスにはパブリック IP アドレスが自動的に割り当てられます。インスタンスを EC2-VPC に起動する場合は、インスタンスにパブリック IP アドレスを送信するかどうかを制御できます。パブリック IP アドレスは、eth0 ネットワークインターフェイス ( プライマリネットワークインターフェイス ) に割り当てられます。

パブリック IP アドレスは、Amazon のパブリック IP アドレスプールからインスタンスに割り当てられ、お客様の AWS アカウントには関連付けられません。パブリック IP アドレスをインスタンスから割り当て解除すると、そのパブリック IP アドレスはパブリック IP アドレスプールに戻され、再利用することはできません。

手動でパブリック IP アドレスをインスタンスに関連付けること、また、手動でインスタンスから割り当て解除することはできません。場合によって、パブリック IP アドレスはインスタンスから解放されたり、新しいインスタンスに割り当てられたりします。

- インスタンスが停止または終了されると、インスタンスのパブリック IP アドレスは解放されます。停止していたインスタンスが再起動されると、そのインスタンスには新しいパブリック IP アドレスが送信されます。
- Elastic IP アドレス ( EIP ) をインスタンスに関連付けた場合、または EIP を VPC 内のインスタンスのプライマリネットワークインターフェイス ( eth0 ) に関連付けた場合、インスタンスのパブリック IP アドレスは解放されます。EIP をインスタンスから割り当て解除すると、そのインスタンスには新しいパブリック IP アドレスが送信されます。
- VPC 内のインスタンスのパブリック IP アドレスが既に解放されている場合には、複数のネットワークインターフェイスがインスタンスにアタッチされていると、インスタンスに新しいパブリック IP アドレスは送信されません。

必要に応じて、インスタンスに関連付けおよびインスタンスから関連付けできる永続的なパブリック IP アドレスが必要な場合は、Elastic IP アドレス ( EIP ) を使用します。独自の EIP を割り当てて、それをインスタンスに関連付けることができます。詳細については、[Elastic IP アドレス \( EIP \) \( p. 540 \)](#) を参照してください。

デフォルト以外のサブネットで起動するインスタンスには、パブリック IP アドレスが自動的に割り当てられません。そのため、デフォルト以外のサブネットのインスタンスでインターネットと通信する場合、起動時にパブリック IP アドレス機能を有効にするか、そのインスタンスのネットワークインターフェイスに割り当てられたプライマリまたはセカンダリプライベート IP アドレスに Elastic IP アドレスを関連付ける必要があります。

パブリック IP アドレスを持つインスタンスのそれぞれに、外部 DNS ホスト名が付与されます。外部 DNS ホスト名を解決すると、インスタンスのパブリック IP アドレス ( インスタンスのネットワークの外部の場合 ) およびインスタンスのプライベート IP アドレス ( インスタンスのネットワーク内からの場合 ) となります。VPC 内のインスタンスで、Elastic IP アドレスを割り当てると、DNS ホスト名が有効な場合は DNS ホスト名を受け取ります。詳細については、「[Amazon Virtual Private Cloud ユーザーガイド](#)」の「[VPC での DNS の使用](#)」を参照してください。

インスタンスのプライベート IP アドレスとパブリック IP アドレスは、ネットワークアドレス変換 ( NAT ) を介して別のアドレスに直接マッピングされます。NAT の詳細については、[RFC 1631: The IP Network Address Translator \( NAT \)](#) を参照してください。



#### Note

インスタンスがパブリック NAT IP アドレスを使用して他のインスタンスにアクセスする場合、アクセス先のインスタンスが同じリージョンにあるかどうかによって、リージョンデータ転送またはインターネットデータ転送に対して課金されます。

## EC2-Classic と EC2-VPC の違い

次の表では、EC2-Classic で起動されたインスタンス、デフォルトサブネット で起動されたインスタンス、デフォルト以外のサブネット で起動されたインスタンスの IP アドレスの違いをまとめています。

特徴	EC2-Classic	デフォルトのサブネット	デフォルト以外のサブネット
パブリック IP アドレス ( Amazon のパブリック IP アドレスプールより )	インスタンスはパブリック IP アドレスを受け取ります。	デフォルトサブネット で起動したインスタンスは、パブリック IP アドレスをデフォルトで受け取ります。ただし、起動時に別のアドレスを指定した場合を除きます。	インスタンスは、起動時に別のアドレスを指定した場合を除き、パブリック IP アドレスをデフォルトで受け取りません。
プライベート IP アドレス	インスタンスは、起動するたびに、EC2-Classic の範囲に含まれるプライベート IP アドレスを受け取ります。	インスタンスはデフォルト VPC のアドレス範囲から静的プライベート IP アドレスを受け取ります。	インスタンスは VPC のアドレス範囲から静的プライベート IP アドレスを受け取ります。
複数の IP アドレス	1 つのインスタンスには 1 つのプライベート IP アドレスを選択します。複数の IP アドレスはサポートされません。	複数のプライベート IP アドレスを 1 つのインスタンスに割り当てることができます。	複数のプライベート IP アドレスを 1 つのインスタンスに割り当てることができます。
ネットワークインターフェイス	IP アドレスが 1 つのインスタンスに関連付けられません。ネットワークインターフェイスはサポートされません。	IP アドレスはネットワークインターフェイスに関連付けられます。各インスタンスには 1 つまたは複数のネットワークインターフェイスがあります。	IP アドレスはネットワークインターフェイスに関連付けられます。各インスタンスには 1 つまたは複数のネットワークインターフェイスがあります。
Elastic IP アドレス	停止すると、EIP とインスタンスの関連付けが解除されます。	停止しても、EIP とインスタンスの関連付けが維持されます。	停止しても、EIP とインスタンスの関連付けが維持されます。
DNS ホスト名	DNS ホスト名はデフォルトで有効化されています。	DNS ホスト名はデフォルトで有効化されています。	DNS ホスト名はデフォルトで無効化されています。

## パブリック IP アドレス、プライベート IP アドレス、Elastic IP アドレスの決定

EC2 コンソールを使用して、インスタンスのパブリック IP アドレス、プライベート IP アドレス、および EIP を決定できます。

コンソールを使用してインスタンスの IP アドレスを判断するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. インスタンスを選択します。コンソールの下部ペインにインスタンスに関する情報が表示されます。
4. [Public IP] フィールドからパブリック IP アドレスを取得します。
5. EIP がインスタンスに関連付けられている場合は、[Elastic IP] フィールドから EIP を取得します。
6. [Private IP] フィールドからプライベート IP アドレスを取得します。

インスタンスのメタデータを使用して、インスタンスのパブリック IP アドレスとプライベート IP アドレスを判断することもできます。インスタンスのメタデータの詳細については、「[インスタンスメタデータとユーザーデータ \(p. 289\)](#)」を参照してください。

インスタンスのメタデータを使用してインスタンスの IP アドレスを判断するには

1. インスタンスに接続します。
2. プライベート IP アドレスにアクセスするには、次のコマンドを使用します。

```
GET http://169.254.169.254/latest/meta-data/local-ipv4
```

3. パブリック IP アドレスにアクセスするには、次のコマンドを使用します。

```
GET http://169.254.169.254/latest/meta-data/public-ipv4
```

EIP がインスタンスに関連付けられると、返される値は EIP の値になることにご注意ください。

## パブリック ID アドレスの割り当て

EC2-Classical でインスタンスを起動すると、そのインスタンスにはパブリック IP アドレスがデフォルトで割り当てられます。この動作を変更することはできません。

VPC でインスタンスを起動すると、パブリック IP アドレス機能が利用可能になり、インスタンスにパブリック IP アドレスを割り当てるかどうかを制御できます。パブリック IP アドレスは、Amazon のパブリック IP アドレスプールから割り当てられ、デバイスインデックス eth0 を持つネットワークインターフェイスに割り当てられます。この機能は、インスタンス起動時の特定の条件により異なります。



### Important

起動後、パブリック IP アドレスをインスタンスから手動で割り当て解除することはできません。場合によって、パブリック IP アドレスは自動的に解放されます。解放されたパブリック IP アドレスは再利用できません。詳細については、[パブリック IP アドレスと外部 DNS ホスト名 \(p. 531\)](#) を参照してください。お客様の意志で関連付けたり関連付けを解除したりできる永続的なパブリック IP アドレスを必要とする場合は、起動してからインスタンスに Elastic IP アドレスを割り当てます。詳細については、[Elastic IP アドレス \( EIP \) \(p. 540\)](#) を参照してください。

インスタンス起動時にパブリック IP アドレス機能にアクセスするには

1. Amazon EC2 コンソールを開きます。
2. [Launch Instance] をクリックします。

- AMI を選択してその [Select] ボタンをクリックし、インスタンスタイプを選択して [Next: Configure Instance Details] をクリックします。
- [Configure Instance Details] ページで、[Network] リストから VPC を選択します。[Public IP] チェックボックスが表示されます。このチェックボックスをオンにして、インスタンスにパブリック IP アドレスを割り当てます。デフォルトサブネットを選択すると、デフォルトでは [Public IP] チェックボックスが選択されます。

以下のルールが適用されます。

- パブリック IP アドレスのみ、インデックス eth0 を持つデバイスの単一のネットワークインターフェイスに割り当てることができます。複数のネットワークインターフェイスで起動しようとしている場合、[Public IP] チェックボックスは選択できません。また、eth1 ネットワークインターフェイスで選択できません。
  - パブリック IP アドレスのみ、既存のではなく、新しいネットワークインターフェイスと関連付けることができます。
- ウィザードの後続ページに表示されるステップにしたがって、インスタンスのセットアップを最後まで実行します。ウィザード設定オプションの詳細については、[インスタンスの起動 \(p. 325\)](#) を参照してください。最終ページの [Review Instance Launch] で、設定内容を確認します。[Launch] をクリックして、キーペアを選択し、インスタンスを起動します。
  - [Instances] ページで、新しいインスタンスを選択し、そのパブリック IP アドレスを、詳細ペインの [Public IP] フィールドで確認します。

パブリック IP アドレス機能は起動時にのみ使用できます。ただし、起動時にパブリック IP アドレスをインスタンスに割り当てるかどうかにかかわらず、起動後に Elastic IP アドレスをインスタンスに関連付けることができます。詳細については、「[Elastic IP アドレス \( EIP \) \( p. 540 \)](#)」を参照してください。

## パブリック IP アドレッシング向けの API およびコマンドラインツール

パブリック IP アドレス機能を有効または無効にするには、下の表に示すいずれかのメソッドを使用します。これらのコマンドラインインターフェイスについての詳細は、[Amazon EC2 へのアクセス \( p. 3 \)](#) を参照してください。

メソッド	パラメータ
AWS CLI	<a href="#">run-instances</a> コマンドで <code>--associate-public-ip-address</code> または <code>--no-associate-public-ip-address</code> オプションを使用します。
Amazon EC2 CLI	<a href="#">ec2-run-instances</a> コマンドで <code>--associate-public-ip-address</code> オプションを使用します。
AWS Tools for Windows PowerShell	<a href="#">New-EC2Instance</a> コマンドで <code>-AssociatePublicIp</code> パラメータを使用します。
クエリ API	<a href="#">RunInstances</a> リクエストで <code>NetworkInterface.n.AssociatePublicIpAddress</code> パラメータを使用します。

## 複数のプライベート IP アドレス

EC2-VPC では、インスタンスに複数のプライベート IP アドレスを指定できます。インスタンスに指定できるネットワークインターフェイスとプライベート IP アドレスの数は、インスタンスタイプによっ

て異なります。詳細については、「[インスタンスタイプあたりの ENI ごとのプライベート IP アドレス \(p. 546\)](#)」を参照してください。

次のような場合、複数のプライベート IP アドレスを VPC 内のインスタンスに割り当てると便利です。

- 1つのサーバーで複数の SSL 証明書を使用し、各インターフェイスに各 IP アドレスに割り当てることで、1つのサーバーで複数のウェブサイトをホストする。
- 各ネットワークインターフェイス用に複数のプライベート IP アドレスを持つネットワークアプライアンス (ファイアウォールやロードバランサーなど) を運用する。
- インスタンスでエラーが発生した場合に、セカンダリプライベート IP アドレスをスタンバイインスタンスに再割り当てすることによって、内部トラフィックをスタンバイインスタンスにリダイレクトする。

#### Topics

- [複数の IP アドレスを使用する方法 \(p. 535\)](#)
- [セカンダリプライベート IP アドレスを割り当てる \(p. 536\)](#)
- [セカンダリプライベート IP アドレスを認識するようにインスタンスのオペレーティングシステムを設定する \(p. 538\)](#)
- [Elastic IP アドレスをセカンダリプライベート IP アドレスに割り当てる \(p. 538\)](#)
- [セカンダリプライベート IP アドレスを表示する \(p. 538\)](#)
- [セカンダリプライベート IP アドレスの割り当てを解除する \(p. 539\)](#)

## 複数の IP アドレスを使用する方法

次の一覧は、ネットワークインターフェイスで複数の IP アドレスを使用する方法の説明です。

- セカンダリプライベート IP アドレスをネットワークインターフェイスに割り当てることができます。ネットワークインターフェイスは、インスタンスにアタッチしたり、インスタンスからデタッチすることができます。
- ネットワークインターフェイスの場合、サブネットの CIDR ブロックの範囲に含まれるセカンダリプライベート IP アドレスを選択する必要があります。
- セキュリティグループは、IP アドレスではなく、ネットワークインターフェイスに適用されます。そのため、IP アドレスは、そのアドレスが指定されているネットワークインターフェイスのセキュリティグループの対象です。
- セカンダリプライベート IP アドレスは、実行中または停止したインスタンスにアタッチされた Elastic ネットワークインターフェイスに、割り当てたり、割り当て解除することができます。
- 1つのインターフェイスに割り当てられるセカンダリプライベート IP アドレスは、明示的に許可されていれば、別のインターフェイスに割り当て直すことができます。
- コマンドラインツールまたは API を使用して複数のセカンダリプライベート IP アドレスをネットワークインターフェイスに割り当てるときに、いずれかのセカンダリプライベート IP アドレスを割り当てることができない場合、処理全体が失敗します。
- プライマリプライベート IP アドレス、セカンダリプライベート IP アドレス、および関連付けられた Elastic IP アドレスは、インターフェイスをインスタンスからデタッチしたり、別のインスタンスにアタッチしても、ネットワークインターフェイスへの割り当ては維持します。
- プライマリネットワークインターフェイスをインスタンスから移動することはできませんが、プライマリネットワークインターフェイスのセカンダリプライベート IP アドレスを別のネットワークインターフェイスに再割り当てすることはできます。
- 追加のネットワークインターフェイスは、あるインスタンスから別のインスタンスに移動できます。

次の一覧は、Elastic IP アドレスで複数の IP アドレスを使用する方法の説明です。

- 各プライベート IP アドレスが関連付けられるのは 1 つの Elastic IP アドレスであり、反対に 1 つの Elastic IP アドレスに関連付けられるのは 1 つの IP アドレスです。
- セカンダリプライベート IP アドレスを別のインターフェイスに再割り当てした場合、セカンダリプライベート IP アドレスと Elastic IP アドレスの関連付けは維持されます。
- セカンダリプライベート IP アドレスとインターフェイスの割り当てを解除すると、関連付けられた Elastic IP アドレスとセカンダリプライベート IP アドレスとの関連付けは自動的に解除されます。

## セカンダリプライベート IP アドレスを割り当てる

セカンダリプライベート IP アドレスは、インスタンスの起動時、またはインスタンスの実行後に、インスタンスのネットワークインターフェイスに割り当てることができます。

EC2-VPC のインスタンスを起動するときにセカンダリプライベート IP アドレスを割り当てるには

1. Amazon EC2 コンソールを開きます。
2. [Launch Instance] ボタンをクリックします。
3. AMI を選択してその [Select] ボタンをクリックし、インスタンスタイプを選択して [Next: Configure Instance Details] をクリックします。
4. [Configure Instance Details] ページで、[Network] リストから VPC を選択し、[Subnet] リストからサブネットを選択します。
5. [Network Interfaces] セクションで、次の手順を実行し、[Next: Add Storage] をクリックします。
  - a. [Add Device] をクリックして、別のネットワークインターフェイスを追加します。コンソールでは、インスタンス起動時のネットワークインターフェイスを最大 2 つ指定できます。インスタンスを起動したら、ナビゲーションペインで [Network Interfaces] をクリックし、ネットワークインターフェイスを追加します。アタッチできるネットワークインターフェイスの合計数はインスタンスタイプによって異なります。詳細については、「[インスタンスタイプあたりの ENI ごとのプライベート IP アドレス \(p. 546\)](#)」を参照してください。
  - b. 各ネットワークインターフェイスについて、1 つのプライマリプライベート IP アドレスと、1 つ以上のセカンダリプライベート IP アドレスを指定できます。ただし、この例では、自動的に割り当てられた IP アドレスを使用します。
  - c. [Secondary IP Addresses] の下にある [Add] をクリックし、サブネットの範囲に含まれるプライベート IP アドレスを入力するか、デフォルトの Auto-assign のままにしてアドレスを自動的に選択します。



### Important

セカンダリプライベート IP アドレスをネットワークインターフェイスに追加した後、インスタンスに接続して、インスタンス自体でセカンダリプライベート IP アドレスを設定する必要があります。詳細については、「[セカンダリプライベート IP アドレスを認識するようにインスタンスのオペレーティングシステムを設定する \(p. 538\)](#)」を参照してください。

6. 次の [Add Storage] ページで、AMI によって指定されるボリューム ( ルートデバイスボリュームなど ) 以外にインスタンスにアタッチするボリュームを指定し、[Next: Tag Instance] をクリックします。
7. [Tag Instance] ページで、ユーザーフレンドリーな名前などを使ってインスタンスのタグを指定し、[Next: Configure Security Group] をクリックします。
8. [Configure Security Group] ページで、既存のセキュリティグループを選択するか、新しいグループを作成します。[Review and Launch] をクリックします。

9. [Review Instance Launch] ページで、設定内容を確認します。[Launch] をクリックして、キーペアを選択し、インスタンスを起動します。Amazon EC2 を初めて使用する場合、これまでにキーペアを作成したことがなければ、ウィザードによってキーペアを作成するよう求めるメッセージが表示されます。

起動中にコマンドラインを使用してセカンダリ IP アドレスを割り当てるには

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

- `run-instances` コマンド ( AWS CLI ) の `--secondary-private-ip-addresses` オプション
- `ec2-run-instances` コマンド ( Amazon EC2 CLI ) の `--secondary-private-ip-address` オプション

既存のインスタンスにセカンダリプライベート IP アドレスを割り当てるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Network Interfaces] をクリックし、インスタンスにアタッチされているネットワークインターフェイスを右クリックします。
3. [Manage Private IP Addresses] を選択します。
4. [Manage Private IP Addresses] ダイアログボックスで、次の操作を行います。
  - a. [Assign new IP] をクリックします。
  - b. インスタンスのサブネットの範囲に含まれる特定の IP アドレスを入力するか、フィールドを空のままにして IP アドレスを自動的に選択することができます。
  - c. ( 省略可能 ) セカンダリプライベート IP アドレスがすでに別のネットワークインターフェイスに割り当てられている場合、[Allow reassignment] を選択して、セカンダリプライベート IP アドレスを割り当て直すことができます。
  - d. [Yes, Update] をクリックし、[Close] をクリックします。



#### Note

また、ナビゲーションペインの [Instances] をクリックして、インスタンスを右クリックし、[Manage Private IP Addresses] をクリックすると、セカンダリプライベート IP アドレスをインスタンスに割り当てることもできます。上記のステップの場合と同様に、同じ情報をダイアログで設定することもできます。

コマンドラインを使用して既存のインスタンスにセカンダリプライベート IP を割り当てるには

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

- `assign-private-ip-addresses` ( AWS CLI )
- `ec2-assign-private-ip-addresses` ( Amazon EC2 CLI )
- `Register-EC2PrivateIPAddress` ( AWS Tools for Windows PowerShell )

## セカンダリプライベート IP アドレスを認識するようにインスタンスのオペレーティングシステムを設定する

セカンダリプライベート IP アドレスをインスタンスに割り当てたら、セカンダリプライベート IP アドレスを認識するようにインスタンスのオペレーティングシステムを設定する必要があります。

Amazon Linux を使用している場合、ec2-net-utils パッケージがこの処理を自動実行します。このパッケージは、インスタンスの実行中にアタッチされる追加のネットワークインターフェイスを設定し、DHCP リースの更新中にセカンダリ IP アドレスを更新し、関連するルーティングルールを更新します。ネットワーク構成を手動で構成する必要がある場合、ec2-net-utils パッケージを削除できます。詳細については、[ec2-net-utils を使用したネットワークインターフェイスの設定 \(p. 549\)](#) を参照してください。

別の Linux ディストリビューションを使用している場合、Linux ディストリビューションのドキュメントを参照してください。追加のネットワークインターフェイスとセカンダリ IP アドレスの設定に関する情報が記載されています。同じサブネットのインスタンスに複数のインターフェイスがある場合、非同期のルーティングに対処する方法については、ルーティングルールの使用に関する情報を検索してください。

Windows インスタンスの設定については、「[Amazon Elastic Compute Cloud Microsoft Windows Guide](#)」の「[Configuring a Secondary Private IP Address for Your Windows Instance](#)」を参照してください。

## Elastic IP アドレスをセカンダリプライベート IP アドレスに割り当てる

EIP を EC2-VPC 内のセカンダリプライベート IP アドレスに関連付けるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Elastic IPs] をクリックします。
3. IP アドレスを右クリックし、[Associate] をクリックします。
4. [Associate Address] ダイアログボックスの [Network Interface] ドロップダウンリストからネットワークインターフェイスを選択し、[Private IP address] ドロップダウンリストからセカンダリ IP アドレスを選択します。
5. [Associate] をクリックします。

コマンドラインを使用してセカンダリプライベート IP アドレスに EIP を関連付けるには

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

- [associate-address](#) ( AWS CLI )
- [ec2-associate-address](#) ( Amazon EC2 CLI )
- [Register-EC2Address](#) ( AWS Tools for Windows PowerShell )

## セカンダリプライベート IP アドレスを表示する

EC2-VPC のネットワークインターフェイスに割り当てられたプライベート IP アドレスを表示するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. プライベート IP アドレスを表示するネットワークインターフェイスを選択します。

4. 詳細ペインの [Details] タブで、[Primary private IP] フィールドと [Secondary private IPs] フィールドに表示されている、ネットワークインターフェイスに割り当てられているプライマリプライベート IP アドレスとセカンダリプライベート IP アドレスを確認します。

インスタンスに割り当てられたプライベート IP アドレスを表示するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. プライベート IP アドレスを表示するインスタンスを選択します。
4. 詳細ペインの [Description] タブで、ネットワークインターフェイス経由でインスタンスに割り当てられているプライマリプライベート IP アドレスとセカンダリプライベート IP アドレスの [Private IPs] フィールドと [Secondary Private IPs] フィールドを確認します。

## セカンダリプライベート IP アドレスの割り当てを解除する

セカンダリプライベート IP アドレスが不要になった場合、インスタンスやネットワークインターフェイスから割り当てを解除することができます。セカンダリプライベート IP アドレスを Elastic ネットワークインターフェイスから割り当て解除した場合、Elastic IP アドレス ( 存在する場合 ) の関連付けも解除されます。

インスタンスからセカンダリプライベート IP アドレスの割り当てを解除するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. インスタンスを右クリックし、[Manage Private IP Addresses] をクリックします。
4. [Manage Private IP Addresses] ダイアログボックスで、割り当て解除するセカンダリプライベート IP アドレスの横にある [Unassign] をクリックします。
5. [Yes, Update] をクリックして、ダイアログボックスを閉じます。

ネットワークインターフェイスからセカンダリプライベート IP アドレスの割り当てを解除するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで [Network Interface] をクリックします。
3. インスタンスを右クリックし、[Manage Private IP Addresses] をクリックします。
4. [Manage Private IP Addresses] ダイアログボックスで、割り当て解除するセカンダリプライベート IP アドレスの横にある [Unassign] をクリックします。
5. [Yes, Update] をクリックし、[Close] をクリックします。

コマンドラインを使用してセカンダリプライベート IP アドレスの割り当てを解除するには

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

- [unassign-private-ip-addresses](#) ( AWS CLI )
- [ec2-unassign-private-ip-addresses](#) ( Amazon EC2 CLI )
- [Unregister-EC2PrivateIpAddress](#) ( AWS Tools for Windows PowerShell )

# Elastic IP アドレス ( EIP )

## Abstract

Elastic IP アドレスを 1 つのインスタンスから別のインスタンスに迅速に再マッピングして、インスタンスやソフトウェアの問題に対応します。

*Elastic IP アドレス ( EIP )* は、動的なクラウドコンピューティングのために設計された静的 IP アドレスです。EIP を使用すると、アドレスをアカウント内の別のインスタンスに迅速に再マッピングして、インスタンスやソフトウェアのエラーを隠すことができます。EIP は AWS アカウントと関連付けられ ( 特定のインスタンスとはなく )、アカウントとの関連付けはお客様が明示的に解除するまで続きます。

EC2-Classic プラットフォームで使用できる EIP のプールが 1 つあり、VPC で使用できる EIP のプールが 1 つあります。VPC で使用するために割り当てた EIP を EC2-Classic のインスタンスに関連付けることや、その逆を行うことはできません。EC2-Classic と EC2-VPC に関する詳細は、「[サポートされているプラットフォーム \(p. 527\)](#)」を参照してください。

## Topics

- [EC2-Classic の Elastic IP アドレス \(p. 540\)](#)
- [VPC の Elastic IP アドレス \(p. 541\)](#)
- [EC2-Classic と EC2-VPC の違い \(p. 541\)](#)
- [Elastic IP アドレスの割り当て \(p. 542\)](#)
- [Elastic IP アドレスの説明 \(p. 542\)](#)
- [Elastic IP アドレスを実行中のインスタンスに関連付ける \(p. 543\)](#)
- [Elastic IP アドレスを別の実行中のインスタンスに関連付ける \(p. 543\)](#)
- [Elastic IP アドレスを解放する \(p. 544\)](#)
- [電子メールアプリケーションでの逆引き DNS の使用 \(p. 544\)](#)
- [Elastic IP アドレスの制限 \(p. 545\)](#)

## EC2-Classic の Elastic IP アドレス

デフォルトでは、起動時に EC2-Classic の各インスタンスに 2 つの IP アドレスが割り当てられます。プライベート IP アドレスと、ネットワークアドレス変換 ( NAT ) によりプライベート IP アドレスにマッピングされるパブリック IP アドレスです。パブリック IP アドレスは EC2-Classic パブリック IP アドレスプールから割り当てられ、AWS アカウントでなくインスタンスに関連付けられます。インスタンスから関連付けを解除したパブリック IP アドレスは再利用できません。

動的 DNS を使用して既存の DNS 名を新しいインスタンスのパブリック IP アドレスにマッピングした場合、その IP アドレスがインターネット内に伝達されるまでに最大 24 時間かかることがあります。その結果、新しいインスタンスはトラフィックを受信せず、終了したインスタンスがリクエストの受信を継続することがあります。この問題を解決するには、EIP を使用します。

EIP をインスタンスに関連付けると、そのインスタンスの現在のパブリック IP アドレスは EC2-Classic パブリック IP アドレスプールに解放されます。インスタンスから EIP の関連付けを解除すると、数分以内に自動的に新しいパブリック IP アドレスがインスタンスに割り当てられます。また、インスタンスを停止した場合も、EIP アドレスの関連付けが解除されます。

EIP を効率的に使用するために、EIP に実行中インスタンスが関連付けられていない場合には少額の時間単位料金が発生します。詳細については、[Amazon EC2 料金表](#)を参照してください。

## VPC の Elastic IP アドレス

起動時にデフォルト VPC の各インスタンスに 2 つの IP アドレスが割り当てられます。プライベート IP アドレスと、ネットワークアドレス変換 (NAT) によりプライベート IP アドレスにマッピングされるパブリック IP アドレスです。パブリック IP アドレスは EC2-VPC パブリック IP アドレスプールから割り当てられ、AWS アカウントでなくインスタンスに関連付けられます。インスタンスから関連付けを解除したパブリック IP アドレスは再利用できません。

デフォルトではない VPC 内の各インスタンスには、プライベート IP アドレスのみを割り当てます。ただし、起動時に特別にパブリック IP アドレスをリクエストした場合を除きます。プライベート IP アドレスが割り当てられていないデフォルトではない VPC 内のインスタンスがインターネット通信できるようにするには、Elastic IP アドレスを VPC で使用できるように割り当て、そのインスタンスに接続されている Elastic ネットワークインターフェイス (ENI) にその EIP を関連付ける必要があります。

起動時に EIP をデフォルト VPC のインスタンスに関連付けた場合、またはパブリック IP を割り当てたインスタンスを eth0 ネットワークインターフェイスに割り当てた場合、現在のパブリック IP アドレスは EC2-VPC パブリック IP アドレスプールに解放されます。インスタンスから EIP の関連付けを解除すると、数分以内に自動的に新しいパブリック IP アドレスがインスタンスに割り当てられます。ただし、2 つ目のネットワークインターフェイスをそのインスタンスに関連付けた場合、そのインスタンスには自動的に新しいパブリック IP アドレスは割り当てられません。手動で EIP をインスタンスに割り当てる必要があります。インスタンスを停止しても、EIP はインスタンスに関連付けられた状態を維持します。

EIP を効率的に使用するために、EIP に実行中インスタンスが関連付けられていない場合や、停止しているインスタンスまたはアタッチされていないネットワークインターフェイスが関連付けられている場合、小額の時間単位料金が発生します。インスタンスを実行しているときは、インスタンスに関連付けられた 1 つの EIP については課金されませんが、インスタンスに関連付けられた追加の EIP がある場合、それに課金されます。詳細については、[Amazon EC2 料金表](#)を参照してください。

EIP と VPC のインスタンスの使用に関する詳細は、*Amazon Virtual Private Cloud ユーザーガイド* の「[Elastic IP アドレス](#)」を参照してください。

## EC2-Classic と EC2-VPC の違い

次の表は、EC2-Classic の EIP と EC2-VPC の EIP の違いをまとめたものです。

特徴	EC2-Classic	EC2-VPC
割り当て	割り当てた EIP は EC2-Classic 内のみで使用されます。	割り当てた EIP は VPC 内のみで使用されます。
関連付け	EIP とインスタンスを関連付けます。	EIP は Elastic ネットワークインターフェイス (ENI) のプロパティです。EIP とインスタンスを関連付けるには、インスタンスにアタッチされている ENI を更新します。詳細については、「 <a href="#">Elastic Network Interface (ENI) (p. 545)</a> 」を参照してください。
再関連付け	すでに別のインスタンスに関連付けられている EIP を関連付けようとすると、アドレスは自動的に新しいインスタンスに関連付けられます。	すでに別のインスタンスに関連付けられている EIP の関連付けは、再関連付けを許可している場合のみ成功します。

特徴	EC2-Classic	EC2-VPC
インスタンスの停止	インスタンスを停止すると、その EIP の関連付けが解除されます。インスタンスを再開するときに EIP を再関連付けする必要があります。	インスタンスを停止すると、その EIP は関連付けを維持します。
複数 IP	インスタンスは単一のプライベート IP アドレスと対応する EIP のみをサポートします。	インスタンスは複数の IP アドレスをサポートし、それぞれの IP アドレスに対応する EIP を設定できます。詳細については、「 <a href="#">複数のプライベート IP アドレス (p. 534)</a> 」を参照してください。

## Elastic IP アドレスの割り当て

Elastic IP アドレスは、AWS Management Console または コマンドラインを使用して割り当てることができます。

EC2-Classic で使用するために、コンソールを使用して Elastic IP アドレスを割り当てるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Elastic IPs] をクリックします。
3. [Allocate New Address] をクリックします。
4. [EC2EIP] リストで [] を選択し、[Yes, Allocate] をクリックします。確認ダイアログボックスを閉じます。

コマンドラインを使用して Elastic IP アドレスを割り当てるには

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

- `allocate-address` ( AWS CLI )
- `ec2-allocate-address` ( Amazon EC2 CLI )
- `New-EC2Address` ( AWS Tools for Windows PowerShell )

## Elastic IP アドレスの説明

Elastic IP アドレスは、AWS Management Console または コマンドラインを使用して記述することができます。

コンソールを使用して Elastic IP アドレスを記述するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Elastic IPs] をクリックします。
3. 表示されているリストをフィルタするには、インスタンスに割り当てられている EIP または ID の一部を検索ボックスに入力します。

コマンドラインを使用して Elastic IP アドレスを記述するには

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

- [describe-addresses](#) ( AWS CLI )
- [ec2-describe-addresses](#) ( Amazon EC2 CLI )
- [Get-EC2Address](#) ( AWS Tools for Windows PowerShell )

## Elastic IP アドレスを実行中のインスタンスに関連付ける

Elastic IP アドレスは、AWS Management Console または コマンドラインを使用してインスタンスに関連付けることができます。

コンソールを使用して Elastic IP アドレスをインスタンスに関連付けるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Elastic IPs] をクリックします。
3. EIP を選択して [Associate Address] をクリックします。
4. [Associate Address] ダイアログボックスで、[Instance] リストボックスからインスタンスを選択し、[Associate] をクリックします。

コマンドラインを使用して Elastic IP アドレスを関連付けるには

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

- [associate-address](#) ( AWS CLI )
- [ec2-associate-address](#) ( Amazon EC2 CLI )
- [Register-EC2Address](#) ( AWS Tools for Windows PowerShell )

## Elastic IP アドレスを別の実行中のインスタンスに関連付ける

Elastic IP アドレスは、AWS Management Console または コマンドラインを使用して別のインスタンスに関連付けることができます。

コンソールを使用して Elastic IP アドレスを別のインスタンスに関連付けるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Elastic IPs] をクリックします。
3. EIP を選択し、[Disassociate] ボタンをクリックします。
4. プロンプトが表示されたら、[Yes, Disassociate] をクリックします。
5. EIP を選択し、[Associate] をクリックします。
6. [Associate Address] ダイアログボックスで、[Instance] リストボックスから新しいインスタンスを選択し、[Associate] をクリックします。

コマンドラインを使用して Elastic IP アドレスを別のインスタンスに関連付けるには

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

- [disassociate-address](#) ( AWS CLI )

- [ec2-disassociate-address](#) ( Amazon EC2 CLI )
- [Unregister-EC2Address](#) ( AWS Tools for Windows PowerShell )

コマンドラインを使用して Elastic IP アドレスを関連付けるには

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

- [associate-address](#) ( AWS CLI )
- [ec2-associate-address](#) ( Amazon EC2 CLI )
- [Register-EC2Address](#) ( AWS Tools for Windows PowerShell )

## Elastic IP アドレスを解放する

不要になった EIP は解放することをお勧めします ( このアドレスをインスタンスに関連付けることはできません )。EC2-Classic で使用するために割り当てられ、インスタンスに関連付けられていない EIP には料金が発生します。

AWS Management Console またはコマンドラインを使用して、Elastic IP アドレスを解放することができます。

コンソールを使用して Elastic IP アドレスを解放するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Elastic IPs] をクリックします。
3. Elastic IP アドレスを選択し、[Release Address] ボタンをクリックして、メッセージが表示されたら [Yes, Release] をクリックします。

コマンドラインを使用して Elastic IP アドレスを解放するには

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

- [release-address](#) ( AWS CLI )
- [ec2-release-address](#) ( Amazon EC2 CLI )
- [Remove-EC2Address](#) ( AWS Tools for Windows PowerShell )

## 電子メールアプリケーションでの逆引き DNS の使用

インスタンスから第三者に電子メールを送信する場合、1 つ以上の Elastic IP アドレスをプロビジョニングし、[電子メール送信制限解除申請フォーム](#)で当社にそれらのアドレスを提供していただくようお願いいたします。AWS は、ISP およびインターネットアンチスパム組織 ( Spamhaus など ) と協力して、これらのアドレスから送信された E メールにスパムのフラグが付く可能性を減らしています。

また、E メール送信に使用される Elastic IP アドレスに静的逆引き DNS レコードを割り当てると、アンチスパム組織により E メールにスパムのフラグが付くことを避けられることがあります。前述のフォームを使用して、アドレスに関連付ける逆引き DNS レコードを当社に送信できます。逆引き DNS レコードを作成できるようになる前に、その Elastic IP アドレスを参照する、対応するフォワード DNS レコード ( A レコード ) が存在している必要があることにご注意ください。

## Elastic IP アドレスの制限

デフォルトでは、すべての AWS アカウントで 5 つの EIP までに制限されています。これは、パブリック (IPv4) インターネットアドレスが数に限りのあるパブリックリソースであるためです。EIP は主に負荷分散のユースケースに使用し、DNS ホスト名は他のすべてのノード間通信に使用することを強くお勧めします。

ご利用のアーキテクチャーで追加の EIP が必要な場合、[Amazon EC2 Elastic IP アドレスリクエストフォーム](#)に記入してください。追加アドレスの必要性を理解できるように、当社はお客様にユースケースの説明をお願いします。

## Elastic Network Interface ( ENI )

### Abstract

インスタンスに Elastic ネットワークインターフェイスをアタッチすると、プライベート IP アドレス、Elastic IP アドレス、およびセキュリティグループを指定できます。

Elastic ネットワークインターフェイス ( ENI ) は、VPC のインスタンスにアタッチできる仮想ネットワークインターフェイスです。ENI には次の属性を含めることができます。

- プライマリプライベート IP アドレス
- 1 つ以上のセカンダリプライベート IP アドレス
- プライベート IP アドレスごとに 1 つの Elastic IP アドレス
- パブリック IP アドレス。eth0 のネットワークインターフェイスについては、インスタンスを起動したときに自動的に割り当てることができます。ただし既存のネットワークインターフェイスを使用するのではなく、eth0 のネットワークインターフェイスを作成する場合のみです。
- 1 つまたは複数のセキュリティグループ
- MAC アドレス
- 送信元/送信先チェックフラグ
- 説明

ネットワークインターフェイスを作成したり、インスタンスにアタッチしたり、インスタンスからデタッチしたり、別のインスタンスにアタッチしたりできます。ネットワークインターフェイスをインスタンスにアタッチしたり、インスタンスからデタッチしたり、別のインスタンスに再アタッチしたりするとき、ネットワークインターフェイスの属性はネットワークインターフェイスについて行きます。インスタンス間でネットワークインターフェイスを移動すると、ネットワークトラフィックは新しいインスタンスにリダイレクトされます。

VPC の各インスタンスにデフォルトのネットワークインターフェイスが与えられます。デフォルトのネットワークインターフェイスには、その VPC の IP アドレス範囲でプライマリプライベート IP アドレスが与えられます。追加のネットワークインターフェイスを作成し、アタッチできます。使用できるネットワークインターフェイスの最大数はインスタンスタイプによって異なります。詳細については、「[インスタンスタイプあたりの ENI ごとのプライベート IP アドレス \(p. 546\)](#)」を参照してください。

次の作業を行う場合、複数のネットワークインターフェイスをインスタンスにアタッチすると便利です。

- 管理用ネットワークを作成する。
- VPC 内でネットワークアプライアンスやセキュリティアプライアンスを使用する。
- 別個のサブネット上のワークロード/ロールを使用するデュアルホーム接続インスタンスを作成する。

- 低予算で可能性の高いソリューションを作成する。

#### Topics

- [インスタンスタイプあたりの ENI ごとのプライベート IP アドレス \(p. 546\)](#)
- [管理用ネットワークの作成 \(p. 547\)](#)
- [VPC 内でネットワークアプライアンスとセキュリティアプライアンスを使用する \(p. 548\)](#)
- [別個のサブネット上のワークロード/ロールを使用するデュアルホーム接続インスタンスを作成する \(p. 548\)](#)
- [低予算で可用性の高いソリューションを構築する \(p. 549\)](#)
- [ネットワークインターフェイスの設定に関するベストプラクティス \(p. 549\)](#)
- [ec2-net-utils を使用したネットワークインターフェイスの設定 \(p. 549\)](#)
- [ネットワークインターフェイスを作成する \(p. 550\)](#)
- [ネットワークインターフェイスの削除 \(p. 551\)](#)
- [ネットワークインターフェイスに関する詳細の表示 \(p. 551\)](#)
- [インスタンスの起動時にネットワークインターフェイスをアタッチする \(p. 552\)](#)
- [停止したインスタンスまたは実行中のインスタンスにネットワークインターフェイスをアタッチする \(p. 553\)](#)
- [ネットワークインターフェイスをインスタンスからデタッチする \(p. 554\)](#)
- [ネットワークインターフェイスのセキュリティグループを変更する \(p. 555\)](#)
- [ネットワークインターフェイスの送信元/送信先チェックを変更する \(p. 555\)](#)
- [Elastic IP アドレスとネットワークインターフェイスを関連付ける \(p. 556\)](#)
- [Elastic IP アドレスとネットワークインターフェイスの関連付けを解除する \(p. 557\)](#)
- [ネットワークインターフェイスの終了動作を変更する \(p. 557\)](#)
- [ネットワークインターフェイスの説明を追加または編集する \(p. 558\)](#)
- [ネットワークインターフェイスのタグを追加または編集する \(p. 558\)](#)

## インスタンスタイプあたりの ENI ごとのプライベート IP アドレス

インスタンスタイプあたりの Elastic Network Interface ( ENI ) の最大数、および ENI あたりのプライベート IP アドレスの最大数を次の表に示します。ENI および複数のプライベート IP アドレスは、VPC で実行されているインスタンスのみに使用できます。詳細については、[複数のプライベート IP アドレス \(p. 534\)](#) を参照してください。

インスタンスタイプ	最大インターフェイス	インターフェイスあたりの IP アドレス
c1.medium	2	6
c1.xlarge	4	15
c3.large	3	10
c3.xlarge	4	15
c3.2xlarge	4	15
c3.4xlarge	8	30
c3.8xlarge	8	30

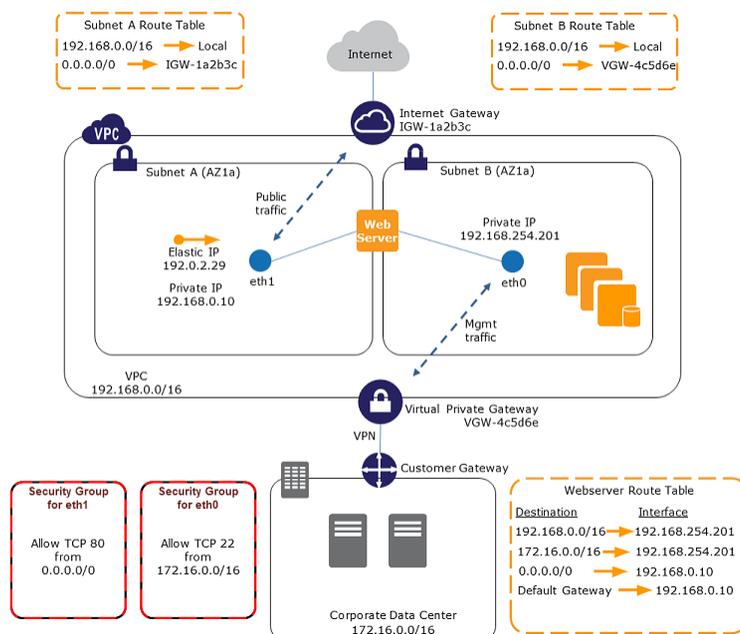
インスタンスタイプ	最大インターフェイス	インターフェイスあたりの IP アドレス
cc2.8xlarge	8	30
cgl.4xlarge	8	30
cr1.8xlarge	8	30
g2.2xlarge	4	15
hi1.4xlarge	8	30
hs1.8xlarge	8	30
i2.xlarge	4	15
i2.2xlarge	4	15
i2.4xlarge	8	30
i2.8xlarge	8	30
m1.small	2	4
m1.medium	2	6
m1.large	3	10
m1.xlarge	4	15
m2.xlarge	4	15
m2.2xlarge	4	30
m2.4xlarge	8	30
m3.medium	2	6
m3.large	3	10
m3.xlarge	4	15
m3.2xlarge	4	30
r3.large	3	10
r3.xlarge	4	15
r3.2xlarge	4	15
r3.4xlarge	8	30
r3.8xlarge	8	30
t1.micro	2	2

## 管理用ネットワークの作成

ネットワークインターフェイスを利用して管理用ネットワークを作成できます。このシナリオでは、インスタンスのセカンダリネットワークインターフェイスでパブリック側のトラフィックを処理し、プライマリネットワークインターフェイスでバックエンドの管理用トラフィックを処理します。プライマリ

ネットワークインターフェイスはアクセス制御を強化したVPC内の個別のサブネットに接続されます。パブリック側のインターフェイス（ロードバランサーの背後に置かれる場合とそうでない場合があります）には、インターネットからサーバーへのアクセスを許可するセキュリティグループが関連付けられます（例えば、0.0.0.0/0またはロードバランサーからのTCPポート80および443を許可します）。プライベート側のインターフェイスには、VPC内またはインターネットからの許容範囲のIPアドレス、VPC内のプライベートサブネット、または仮想プライベートゲートウェイからのSSHアクセスのみを許可するセキュリティグループが関連付けられます。

フェイルオーバー機能が確実に動作するように、ネットワークインターフェイスの受信トラフィックに対してセカンダリプライベートIPを使用することをお勧めします。インスタンスに障害が発生した場合は、インターフェイスまたはセカンダリプライベートIPアドレスあるいはその両方をスタンバイ用のインスタンスに移行できます。



## VPC内でネットワークアプライアンスとセキュリティアプライアンスを使用する

ロードバランサー、ネットワークアドレス変換 (NAT) サーバー、プロキシサーバーなど、ネットワークアプライアンスやセキュリティアプライアンスの中には、複数のネットワークインターフェイスを使用した構成が優先されるものがあります。セカンダリネットワークインターフェイスを作成して、これらのタイプのアプリケーションを実行するVPC内のインスタンスにアタッチし、専用のパブリックIPアドレスとプライベートIPアドレス、セキュリティグループ、および送信元/送信先チェックを使用して追加のインターフェイスを構成することができます。

## 別個のサブネット上のワークロード/ロールを使用するデュアルホーム接続インスタンスを作成する

アプリケーションサーバーが存在するミッドティアネットワークに接続するWebサーバーのそれぞれにネットワークインターフェイスを置くことができます。アプリケーションサーバーは、データベースサーバーが存在するバックエンドネットワーク（サブネット）にデュアルホーム接続することもできます。デュアルホーム接続されたインスタンスを介してネットワークパケットをルーティングする代わりに、デュアルホーム接続された各インスタンスは、フロントエンドでリクエストを受信して処理し、バックエンドとの接続を開始し、バックエンドネットワーク上のサーバーにリクエストを送信します。

## 低予算で可用性の高いソリューションを構築する

特定の機能にサービスを提供しているインスタンスのいずれかが機能しなくなった場合は、そのネットワークインターフェイスを同じ役割で構成された交換用またはホットスタンバイ用のインスタンスにアタッチすることで、サービスを迅速に回復できます。例えば、データベースインスタンスや NAT インスタンスなどの重要なサービスに対するプライマリまたはセカンダリのネットワークインターフェイスとして ENI を使用することができます。そのインスタンスが機能しなくなった場合、ユーザーは (通常はユーザーの代わりに実行されるコードで) ENI をホットスタンバイ用のインスタンスにアタッチすることができます。インターフェイスでは、プライベート IP アドレス、Elastic IP アドレス、および MAC アドレスがそのまま維持されるため、交換用のインスタンスに ENI を接続するとすぐに、ネットワークトラフィックはスタンバイ用のインスタンスに流れ始めます。インスタンスに障害が発生してから、ENI がスタンバイ用のインスタンスにアタッチされるまで、ユーザーは少しの間接続を失った状態になりますが、VPC ルートテーブルや DNS サーバーに変更を加える必要はありません。

## ネットワークインターフェイスの設定に関するベストプラクティス

- ネットワークインターフェイスは、インスタンスの実行中、インスタンスの停止中、インスタンスの起動中にインスタンスにアタッチできます (それぞれ、ホットアタッチ、ウォームアタッチ、コールドアタッチと呼ばれています)。
- セカンダリ (ethN) ネットワークインターフェイスは、インスタンスの実行中または停止中にデタッチできます。ただし、プライマリ (eth0) インターフェイスをデタッチすることはできません。
- あるサブネットのネットワークインターフェイスを同じ VPC 内にある別のサブネットのインスタンスにアタッチできますが、ネットワークインターフェイスとインスタンスの両方が同じ利用可能ゾーン内に存在している必要があります。
- CLI または API からインスタンスを起動する際、プライマリ (eth0) ネットワークインターフェイスと追加のネットワークインターフェイスの両方に対して、インスタンスにアタッチするネットワークインターフェイスを指定できます。
- 複数のネットワークインターフェイスでインスタンスを起動すると、インスタンスのオペレーティングシステム上でインターフェイス、プライベート IP アドレス、ルートテーブルが自動的に設定されます。追加ネットワークインターフェイスをウォームアタッチまたはホットアタッチするとき、場合によっては、手動で 2 つ目のインターフェイスを起動し、プライベート IP アドレスを設定し、ルートテーブルを適宜変更する必要があります。(Microsoft Windows Server または Amazon Linux を実行するインスタンスはウォームアタッチまたはホットアタッチを自動的に認識し、自動的に設定します。)
- インスタンスに別のネットワークインターフェイスをアタッチする機能は、デュアルホーム接続インスタンスに対するネットワーク帯域幅を増加または倍増するためのものではありません。

## ec2-net-utils を使用したネットワークインターフェイスの設定

Amazon Linux AMI には、ec2-net-utils と呼ばれる追加のスク립トが Amazon Web Services によってインストールされていることがあります。これらのスク립トはオプションで、Elastic Network Interface (ENI) の設定を自動化します。

パッケージをまだインストールしていない場合は、以下のコマンドを使用して Amazon Linux にインストールします。既にインストール済みで、更新が利用可能な場合は更新します。

```
yum install ec2-net-utils
```

ec2-net-utils には、以下のコンポーネントが含まれます。

#### udev ルール ( /etc/udev/rules.d )

実行中のインスタンスにネットワークインターフェイスがアタッチ、デタッチ、または再アタッチされたときに、そのネットワークインターフェイスを特定し、ホットプラグスクリプトが実行されることを確認します ( 53-ec2-network-interfaces.rules )。MAC アドレスをデバイス名にマッピングします ( 70-persistent-net.rules を生成する 75-persistent-net-generator.rules )。

#### ホットプラグスクリプト

DHCP での使用に適したインターフェイス設定ファイルを生成します ( /etc/sysconfig/network-scripts/ifcfg-ethN )。また、ルート設定ファイルも生成します ( /etc/sysconfig/network-scripts/route-ethN )。

#### DHCP スクリプト

ネットワークインターフェイスが新しい DHCP リースを受け取るたびに、このスクリプトがインスタンスメタデータに対し、Elastic IP アドレスを求めるクエリを実行します。これにより、各 Elastic IP アドレスごとに、そのアドレスからのアウトバンドトラフィックが正しいネットワークインターフェイスを使用するよう、ルーティングポリシーデータベースにルールが追加されます。また、各プライベート IP アドレスを、セカンダリアドレスとしてネットワークインターフェイスに追加します。

#### ec2ifup ethN

標準の ifup の機能を拡張します。このスクリプトが設定ファイル ifcfg-ethN および route-ethN を書き換えた後、ifup を実行します。

#### ec2ifdown ethN

標準の ifdown の機能が拡張されます。このスクリプトがルーティングポリシーデータベースからネットワークインターフェイスのルールをすべて削除した後、ifdown を実行します。

#### ec2ifscan

まだ設定されていないネットワークインターフェイスを探して、それらを設定します。

このスクリプトは、ec2-net-utils の初期リリースでは提供されていないことにご留意ください。

ec2-net-utils によって生成された設定ファイルをリストするには、以下のコマンドを使用します。

```
ls -l /etc/sysconfig/network-scripts/*-eth?
```

インスタンスごとの自動化を無効にするには、対応する ifcfg-ethN ファイルに EC2SYNC=no を追加します。例えば、eth1 インターフェイスの自動化を無効にするには、以下のコマンドを使用します。

```
sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

自動化を完全に無効にするには、以下のコマンドを使用してパッケージを削除します。

```
yum remove ec2-net-utils
```

## ネットワークインターフェイスを作成する

AWS Management Console またはコマンドラインを使用して、ネットワークインターフェイスを作成できます。

コンソールを使用してネットワークインターフェイスを作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. [Create Network Interface] をクリックします。

4. [Create Network Interface] ダイアログボックスで、ネットワークインターフェイスについての次の情報を入力し、[Yes, Create] をクリックします。
  - a. [Description] で、記述的な名前を入力します。
  - b. [Subnet] で、サブネットを選択します。作成後、ネットワークインターフェイスを別のサブネットに移動することはできません。同じサブネットのインスタンスにアタッチすることはできません。
  - c. [Private IP] で、プライベート IP アドレスを入力します。IP アドレスを指定しない場合、選択されているサブネット内で使用可能な IP アドレスが自動選択されます。
  - d. [Security groups] で、1 つまたは複数のセキュリティグループを選択します。

コマンドラインを使用してネットワークインターフェイスを作成するには

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

- [create-network-interface](#) ( AWS CLI )
- [ec2-create-network-interface](#) ( Amazon EC2 CLI )
- [New-EC2NetworkInterface](#) ( AWS Tools for Windows PowerShell )

## ネットワークインターフェイスの削除

ネットワークインターフェイスを削除する前に、そのネットワークインターフェイスをインスタンスからデタッチする必要があります。ネットワークインターフェイスを削除すると、そのネットワークインターフェイスに関連付けられているすべての属性が解放され、別のインスタンスで使用できるように、プライベート IP アドレスまたは Elastic IP アドレスが解放されます。

AWS Management Console またはコマンドラインを使用して、ネットワークインターフェイスを削除できます。

コンソールを使用してネットワークインターフェイスを削除するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを選択し、[Delete] ボタンをクリックします。
4. [Delete Network Interface] ダイアログボックスで、[Yes, Delete] をクリックします。

コマンドラインを使用してネットワークインターフェイスを削除するには

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

- [delete-network-interface](#) ( AWS CLI )
- [ec2-delete-network-interface](#) ( Amazon EC2 CLI )
- [Remove-EC2NetworkInterface](#) ( AWS Tools for Windows PowerShell )

## ネットワークインターフェイスに関する詳細の表示

AWS Management Console またはコマンドラインを使用して、ネットワークインターフェイスを記述できます。

コンソールを使用してネットワークインターフェイスを記述するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを選択します。
4. [Details] タブで詳細を確認します。

コマンドラインを使用してネットワークインターフェイスを記述するには

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

- [describe-network-interfaces](#) ( AWS CLI )
- [ec2-describe-network-interfaces](#) ( Amazon EC2 CLI )
- [Get-EC2NetworkInterface](#) ( AWS Tools for Windows PowerShell )

コマンドラインを使用してネットワークインターフェイス属性を記述するには

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

- [describe-network-interface-attribute](#) ( AWS CLI )
- [ec2-describe-network-interface-attribute](#) ( Amazon EC2 CLI )
- [Get-EC2NetworkInterfaceAttribute](#) ( AWS Tools for Windows PowerShell )

## インスタンスの起動時にネットワークインターフェイスをアタッチする

VPC にインスタンスを起動するとき、追加のネットワークインターフェイスをインスタンスにアタッチできます。



### Note

ネットワークインターフェイスをインスタンスにアタッチしたときにエラーが発生した場合、インスタンスは正しく起動されません。

AWS Management Console またはコマンドラインを使用して、ネットワークインターフェイスをインスタンスにアタッチできます。

コンソールを使用してインスタンスの起動時にネットワークインターフェイスをアタッチするには

1. Amazon EC2 コンソールを開きます。
2. [Launch Instance] をクリックします。
3. AMI を選択してその [Select] ボタンをクリックし、インスタンスタイプを選択して [Next: Configure Instance Details] をクリックします。
4. [Configure Instance Details] ページで、[Network] リストから VPC を選択し、[Subnet] リストからサブネットを選択します。

インスタンスにパブリック IP アドレスを割り当てるには、[Automatically assign a public IP address to your instances] チェックボックスをオンにします。デフォルトサブネットを選択した場合は、こ

のチェックボックスがデフォルトでオンになっています。プライマリネットワークインターフェイス (eth0) に既存のネットワークインターフェイスを指定する場合や、次の手順で複数のネットワークインターフェイスを指定する場合は、インスタンスにパブリック IP アドレスを割り当てることはできません。

5. コンソールの [Network Interfaces] セクションで、インスタンスを起動する際に最大 2 つのネットワークインターフェイス (新規、既存、またはその組み合わせ) を指定することができます。また、すべての新規ネットワークインターフェイスについて、1 つのプライマリ IP アドレスと、1 つまたは複数のセカンダリ IP アドレスを入力することができます。終了したら、[Next: Add Storage] をクリックします。

インスタンスの起動後に、追加のネットワークインターフェイスを追加できることにご留意ください。アタッチできるネットワークインターフェイスの合計数はインスタンスタイプによって異なります。詳細については、「[インスタンスタイプあたりの ENI ごとのプライベート IP アドレス \(p. 546\)](#)」を参照してください。

6. 次の [Add Storage] ページで、AMI によって指定されるボリューム (ルートデバイスボリュームなど) 以外にインスタンスにアタッチするボリュームを指定し、[Next: Tag Instance] をクリックします。
7. [Tag Instance] ページで、ユーザーフレンドリーな名前などを使ってインスタンスのタグを指定し、[Next: Configure Security Group] をクリックします。
8. [Configure Security Group] ページで、既存のセキュリティグループを選択するか、新しいグループを作成します。[Review and Launch] をクリックします。
9. [Review Instance Launch] ページに、プライマリおよび追加ネットワークインターフェイスの詳細情報が表示されます。設定を確認し、[Launch] をクリックして、キーペアを選択し、インスタンスを起動します。Amazon EC2 を初めて使用する場合、これまでにキーペアを作成したことがなければ、ウィザードによってキーペアを作成するよう求めるメッセージが表示されます。

コマンドラインを使用してインスタンスの起動時にネットワークインターフェイスをアタッチするには

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

- [run-instances](#) ( AWS CLI )
- [ec2-run-instances](#) ( Amazon EC2 CLI )
- [New-EC2Instance](#) ( AWS Tools for Windows PowerShell )

## 停止したインスタンスまたは実行中のインスタンスにネットワークインターフェイスをアタッチする

Amazon EC2 コンソールの [Instances] ページまたは [Network Interfaces] ページのどちらかから、あるいはコマンドラインインターフェイスを使用して、VPC 内で停止したインスタンスまたは実行中のインスタンスにネットワークインターフェイスをアタッチできます。

インスタンスページを使用してネットワークインターフェイスをインスタンスにアタッチするには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. インスタンスを右クリックし、[Attach Network Interface] を選択します。
4. [Attach Network Interface] ダイアログボックスで、ネットワークインターフェイスを選択し、[Attach] をクリックします。

ネットワークインターフェイスページを使用してネットワークインターフェイスをインスタンスにアタッチするには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを選択します。
4. [Attach] ボタンをクリックします。
5. [Attach Network Interface] ダイアログボックスでインスタンスを選択し、[Attach] をクリックします。

コマンドラインを使用してインスタンスにネットワークインターフェイスをアタッチするには

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

- `attach-network-interface` ( AWS CLI )
- `ec2-attach-network-interface` ( Amazon EC2 CLI )
- `Add-EC2NetworkInterface` ( AWS Tools for Windows PowerShell )

## ネットワークインターフェイスをインスタンスからデタッチする

セカンダリネットワークインターフェイスは、Amazon EC2 コンソールの [Instances] ページまたは [Network Interfaces] ページを使用して、またはコマンドラインインターフェイスを使用して、いつでもデタッチできます。

インスタンスページを使用してネットワークインターフェイスをインスタンスからデタッチするには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. インスタンスを右クリックし、[Detach Network Interface] を選択します。
4. [Detach Network Interface] ダイアログボックスで、ネットワークインターフェイスを選択し、[Detach] をクリックします。

ネットワークインターフェイスページを使用してネットワークインターフェイスをインスタンスからデタッチするには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを選択し、[Detach] ボタンをクリックします。
4. [Detach Network Interface] ダイアログボックスで [Yes, Detach] をクリックします。ネットワークインターフェイスをインスタンスからデタッチできなかった場合は、[Force detachment] を選択し、再試行します。

コマンドラインを使用してネットワークインターフェイスをデタッチするには

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

- `detach-network-interface` ( AWS CLI )

- [ec2-detach-network-interface](#) ( Amazon EC2 CLI )
- [Dismount-EC2NetworkInterface](#) ( AWS Tools for Windows PowerShell )

## ネットワークインターフェイスのセキュリティグループを変更する

ネットワークインターフェイスに関連付けられているセキュリティグループを変更できます。セキュリティグループを作成するとき、ネットワークインターフェイスのサブネットと同じ VPC を必ず指定します。

AWS Management Console または コマンドラインを使用して、ネットワークインターフェイスのセキュリティグループを変更することができます。



### Note

Elastic Load Balancing などの他のアマゾン ウェブ サービスが所有するインターフェイスのセキュリティグループメンバーシップを変更するには、そのサービスのコンソールまたはコマンドラインインターフェイスを使用します。

コンソールを使用してネットワークインターフェイスのセキュリティグループを変更するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを選択します。
4. ネットワークインターフェイスを右クリックし、[Change Security Groups] を選択します。
5. [Change Security Groups] ダイアログボックスで、使用するセキュリティグループを選択し、[Save] をクリックします。

コマンドラインを使用してネットワークインターフェイスのセキュリティグループを変更するには

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

- [modify-network-interface-attribute](#) ( AWS CLI )
- [ec2-modify-network-interface-attribute](#) ( Amazon EC2 CLI )
- [Edit-EC2NetworkInterfaceAttribute](#) ( AWS Tools for Windows PowerShell )

## ネットワークインターフェイスの送信元/送信先チェックを変更する

[Source/Destination Check] 属性により、送信元/送信先チェックがインスタンスで有効になっているかどうか制御されます。この属性を無効にすると、インスタンスで自身にアドレス指定されていないネットワークトラフィックを処理することが可能になります。例えば、ネットワークアドレス変換、ルーティング、ファイアウォールなどのサービスを実行するインスタンスではこの値を `disabled` に設定する必要があります。デフォルト値は `enabled` です。

AWS Management Console または コマンドラインを使用して、送信元/送信先チェックを変更できます。

コンソールを使用してネットワークインターフェイスの送信元/送信先チェックを変更するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを右クリックし、[Change Source/Dest Check] を選択します。
4. ダイアログボックスで [Enabled] (有効にする場合) または [Disabled] (無効にする場合) を選択して、[Save] をクリックします。

コマンドラインを使用してネットワークインターフェイスの送信元/送信先チェックを変更するには

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

- `modify-network-interface-attribute` ( AWS CLI )
- `ec2-modify-network-interface-attribute` ( Amazon EC2 CLI )
- `Edit-EC2NetworkInterfaceAttribute` ( AWS Tools for Windows PowerShell )

## Elastic IP アドレスとネットワークインターフェイスを関連付ける

Elastic IP アドレスが与えられている場合、ネットワークインターフェイスのプライベート IP アドレスの 1 つをそれと関連付けることができます。1 つの Elastic IP アドレスと各プライベート IP アドレスを関連付けることができます。

AWS Management Console またはコマンドラインを使用して、Elastic IP アドレスを関連付けることができます。

コンソールを使用して Elastic IP アドレスを関連付けるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを右クリックし、[Associate Address] を選択します。
4. [Associate Elastic IP Address] ダイアログボックスで、[Address] リストから Elastic IP アドレスを選択します。
5. [Associate to private IP address] で、Elastic IP アドレスに関連付けるプライベート IP アドレスを選択します。
6. [Allow reassociation] を選択して、指定したネットワークインターフェイスに Elastic IP アドレスを関連付け ( 現在、別のインスタンスまたはネットワークインターフェイスに関連付けられている場合 )、[Associate Address] をクリックします。

コマンドラインを使用して Elastic IP アドレスを関連付けるには

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

- `associate-address` ( AWS CLI )
- `ec2-associate-address` ( Amazon EC2 CLI )
- `Register-EC2Address` ( AWS Tools for Windows PowerShell )

## Elastic IP アドレスとネットワークインターフェイスの関連付けを解除する

ネットワークインターフェイスに Elastic IP アドレスが関連付けられている場合、アドレスの関連付けを解除し、別のネットワークインターフェイスに関連付けるか、解放してアドレスプールに戻すことができます。ネットワークインターフェイスは特定のサブネットに固有になるため、これはネットワークインターフェイスを利用し、異なるサブネットまたは VPC のインスタンスに Elastic IP アドレスを関連付ける唯一の方法になります。

AWS Management Console または コマンドラインを使用して、Elastic IP アドレスの関連付けを解除することができます。

コンソールを使用して Elastic IP アドレスの関連付けを解除するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを右クリックし、[Disassociate Address] を選択します。
4. [Disassociate IP Address] ダイアログボックスで [Yes, Disassociate] をクリックします。

コマンドラインを使用して Elastic IP アドレスを別のインスタンスに関連付けるには

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

- `disassociate-address` ( AWS CLI )
- `ec2-disassociate-address` ( Amazon EC2 CLI )
- `Unregister-EC2Address` ( AWS Tools for Windows PowerShell )

## ネットワークインターフェイスの終了動作を変更する

ネットワークインターフェイスがアタッチされているインスタンスが削除されたときにネットワークインターフェイスも自動的に削除するように、インスタンスにアタッチされているネットワークインターフェイスの終了動作を設定できます。



### Note

デフォルトでは、AWS Management Console によって自動的に作成されインスタンスにアタッチされたネットワークインターフェイスは、インスタンスの終了時に終了するように設定されています。ただし、コマンドラインインターフェイスを使用して作成されたネットワークインターフェイスは、インスタンスの終了時に終了するように設定されていません。

AWS Management Console または コマンドラインを使用して、ネットワークインターフェイスの終了時の動作を変更することができます。

コンソールを使用してネットワークインターフェイスの終了時の動作を変更するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを右クリックし、[Change Termination Behavior] を選択します。

4. インスタンスの終了時にネットワークインターフェイスを削除する場合は、[Change Termination Behavior] ダイアログボックスの [Delete on termination] チェックボックスを選択します。

コマンドラインを使用してネットワークインターフェイスの終了時の動作を変更するには

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

- [modify-network-interface-attribute](#) ( AWS CLI )
- [ec2-modify-network-interface-attribute](#) ( Amazon EC2 CLI )
- [Edit-EC2NetworkInterfaceAttribute](#) ( AWS Tools for Windows PowerShell )

## ネットワークインターフェイスの説明を追加または編集する

AWS Management Console またはコマンドラインを使用して、ネットワークインターフェイスの説明を変更することができます。

コンソールを使用してネットワークインターフェイスの説明を変更するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを右クリックし、[Change Description] を選択します。
4. [Change Description] ダイアログボックスでネットワークインターフェイスの説明を入力し、[Save] をクリックします。

コマンドラインを使用してネットワークインターフェイスの説明を変更するには

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

- [modify-network-interface-attribute](#) ( AWS CLI )
- [ec2-modify-network-interface-attribute](#) ( Amazon EC2 CLI )
- [Edit-EC2NetworkInterfaceAttribute](#) ( AWS Tools for Windows PowerShell )

## ネットワークインターフェイスのタグを追加または編集する

タグとは、ネットワークインターフェイスに追加できるメタデータです。タグはプライベートとして扱われ、アカウントでのみ表示できます。各タグはキーとオプションの値で構成されます。タグの詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 662\)](#)」を参照してください。

AWS Management Console またはコマンドラインを使用して、リソースにタグを付けることができます。

コンソールを使用してネットワークインターフェイスのタグを追加または編集するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Network Interfaces] をクリックします。
3. ネットワークインターフェイスを選択します。

4. 詳細ペインで [Tags] タブをクリックし、[Add/Edit Tags] をクリックします。
5. [Add/Edit Tags] ダイアログボックスで、作成するタグごとに [Create Tag] をクリックして、キーとオプションの値を入力します。終了したら [Save] をクリックします。

コマンドラインを使用してネットワークインターフェイスのタグを追加または編集するには次のコマンドの 1 つを使用できます。これらのコマンドラインインターフェイスについての詳細は、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

- [create-tags](#) ( AWS CLI )
- [ec2-create-tags](#) ( Amazon EC2 CLI )
- [New-EC2Tag](#) ( AWS Tools for Windows PowerShell )

## VPC 内の Linux インスタンスでの拡張ネットワーキングの有効化

### Abstract

サポート対象の Linux インスタンスで、拡張ネットワーキング機能を有効化します。

C3、R3、および I2 インスタンスでは、拡張ネットワーキング機能を有効にできます。AWS では、シングルルート I/O 仮想化 ( SR-IOV ) を利用した拡張ネットワーキング機能のみがサポートされます。インスタンスで拡張ネットワーキング機能を有効化すると、パフォーマンス ( 秒あたりのパケット数 ) が拡張され、レイテンシーが低くなり、お客様の手間が省略されます。

インスタンスで拡張ネットワーキング機能を有効化する前に、カーネルに必要なドライバがインストールされていることと、また、インスタンスの `sriovNetSupport` 属性が設定済みであることを確認してください。

### Topics

- [要件 \(p. 559\)](#)
- [Amazon Linux の使用 \(p. 560\)](#)
- [Amazon Linux 以外の Linux ディストリビューションの使用 \(p. 561\)](#)
- [拡張ネットワーキングが有効化されているかどうかのテスト \(p. 562\)](#)

Windows での手順については、*Amazon Elastic Compute Cloud Microsoft Windows Guide* の [VPC 内の Windows インスタンスで拡張ネットワーキングを有効化する](#) に説明があります。

## 要件

- Linux カーネルバージョン 2.6.32 以上で、HVM AMI からインスタンスを起動します。
- VPC 内でインスタンスを起動します。
- [AWS CLI](#) または [Amazon EC2 CLI ツール](#) をインストールします。詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

Amazon EC2 CLI ツールを選択する場合は、バージョン 1.6.12.0 以降をインストールします。CLI ツールのバージョンを確認するには、[ec2-version](#) コマンドを使用できます。

## Amazon Linux の使用

最新の Amazon Linux HVM AMI では、拡張ネットワークに必要なドライバがインストールされており、必要な属性も設定されています。したがって、C3、R3、I2 のいずれかの Amazon EBS-Backed インスタンスを最新の Amazon Linux HVM AMI を使用して起動した場合は、拡張ネットワークが有効化されています。詳細については、[拡張ネットワークが有効化されているかどうかのテスト \(p. 562\)](#) を参照してください。

以前のバージョンの Amazon Linux AMI を使用してインスタンスを起動した場合、Amazon EBS-Backed インスタンスで拡張ネットワークを有効化するには、以下の手順を実行します。

以前のバージョンの Amazon Linux で拡張ネットワークを有効化するには

1. インスタンスに接続します。
2. インスタンスから以下のコマンドを実行します。

```
sudo yum update
```

3. コンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを再起動します。[reboot-instances](#) (AWS CLI) または [ec2-reboot-instances](#) (Amazon EC2 CLI)。
4. コンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを停止します。[stop-instances](#) (AWS CLI) または [ec2-stop-instances](#) (Amazon EC2 CLI)。
5. コンピュータから、次のいずれかのコマンドを使用して拡張ネットワークの属性を有効化します。このネットワーク属性は、いったん有効化すると無効化できないことに注意してください。

- [modify-instance-attribute](#) (AWS CLI)

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support Value=simple
```

- [ec2-modify-instance-attribute](#) (Amazon EC2 CLI)

```
$ ec2-modify-instance-attribute instance_id --sriov simple
```

6. (オプション) 「[Amazon EBS-Backed Linux AMI の作成 \(p. 77\)](#)」の説明に従って、インスタンスから AMI を作成します。AMI は、インスタンスから拡張ネットワーク属性を継承します。このため、この AMI を使用することで、拡張ネットワークがデフォルトで有効化された状態で C3、R3、I2 のインスタンスを起動することができます。
7. コンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを開始します。[start-instances](#) (AWS CLI) または [ec2-start-instances](#) (Amazon EC2 CLI)。

[Instance Store-Backed Instances]

Instance store-Backed インスタンスを使用している場合は、前述のようにインスタンスを停止したり、属性を変更したり、イメージを作成したりすることはできません。代わりに、[Instance Store-Backed Linux AMI の作成 \(p. 81\)](#) で説明しているように、新しい AMI を作成します。AMI を登録するときに、必ず拡張ネットワーク属性を有効化してください。

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --sriov-net-support Value=simple ...
```

- [ec2-register](#) ( Amazon EC2 CLI )

```
$ ec2-register --sriov simple ...
```

## Amazon Linux 以外の Linux ディストリビューションの使用

以下に、Amazon Linux 以外の Linux ディストリビューションで拡張ネットワーキングを有効化する際の一般的な手順を示します。コマンドの構文など、詳細については、ご使用の Linux ディストリビューションのドキュメントを参照してください。

Linux で拡張ネットワーキングを有効化するには

1. インスタンスに接続します。
2. インスタンスで PCI Hot Plug 機能が有効化されていることを確認します。
3. ixgbevf ドライバをダウンロードし、インスタンスでソースを抽出します。必ず、インスタンスのカーネルと互換性があるドライバをインストールしてください。
4. インスタンスで ixgbevf ドライバをコンパイルして、インストールします。
5. 最適なパフォーマンスを得るには、以下のようにして、インストールしたドライバに動的割り込みスロットリングを有効化します。

```
$ echo "options ixgbevf InterruptThrottleRate=1,1,1,1,1,1,1,1" > /etc/modprobe.d/ixgbevf.conf
```

これは一般的な ixgbevf.conf の場所ですが、お使いの Linux ディストリビューションによっては異なる場合があります。

6. コンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを停止します。 [stop-instances](#) ( AWS CLI ) または [ec2-stop-instances](#) ( Amazon EC2 CLI ) 。
7. コンピュータから、次のいずれかのコマンドを使用して拡張ネットワーキングの属性を有効化します。このネットワーキング属性は、いったん有効化すると無効化できないことに注意してください。

- [modify-instance-attribute](#) ( AWS CLI )

```
$ aws ec2 modify-instance-attribute --instance-id instance_id --sriov-net-support Value=simple
```

- [ec2-modify-instance-attribute](#) ( Amazon EC2 CLI )

```
$ ec2-modify-instance-attribute instance_id --sriov simple
```

8. ( オプション ) 「[Amazon EBS-Backed Linux AMI の作成 \(p. 77\)](#)」の説明に従って、インスタンスから AMI を作成します。AMI は、インスタンスから拡張ネットワーキング属性を継承します。こ

のため、この AMI を使用することで、拡張ネットワークングがデフォルトで有効化された状態で C3、R3、I2 のインスタンスを起動することができます。

9. コンピュータから、Amazon EC2 コンソールまたは次のいずれかのコマンドを使用してインスタンスを開始します。 [start-instances](#) (AWS CLI) または [ec2-start-instances](#) (Amazon EC2 CLI)。
10. (オプション) インスタンスに接続して、ドライバがインストールされていることを確認してください。

#### [Instance store-Backed Instances]

Instance store-Backed インスタンスを使用している場合は、前述のようにインスタンスを停止したり、属性を変更したり、イメージを作成したりすることはできません。代わりに、[Instance Store-Backed Linux AMI の作成 \(p. 81\)](#) で説明しているように、新しい AMI を作成します。AMI を登録するときに、必ず拡張ネットワークング属性を有効化してください。

- [register-image](#) (AWS CLI)

```
$ aws ec2 register-image --sriov-net-support Value=simple ...
```

- [ec2-register](#) (Amazon EC2 CLI)

```
$ ec2-register --sriov simple ...
```

#### 接続性の問題のトラブルシューティング

拡張ネットワークングを有効化しているときに接続が失われると、ixgbevf モジュールとカーネルの互換性が保たれない可能性があります。この場合、インスタンスの Linux ディストリビューションに含まれる ixgbevf モジュールのバージョンをインストールしてみます。

## 拡張ネットワークングが有効化されているかどうかのテスト

拡張ネットワークングが有効化されているかどうかをテストするには、インスタンスにドライバがインストールされていること、また、sriovNetSupport 属性が設定されていることを確認します。

#### [Driver (ixgbevf)]

ixgbevf ドライバがインストールされていることを確認するには、以下の要領で ethtool コマンドを使用します。

```
$ sudo ethtool -i eth0
driver: ixgbevf
version: 2.11.3
firmware-version: N/A
bus-info: 0000:00:03.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: no
supports-register-dump: yes
supports-priv-flags: no
```

#### [Attribute (sriovNetSupport)]

インスタンスに拡張ネットワーキング属性が設定されているかどうかを確認するには、次のいずれかのコマンドを使用します。

- [describe-instance-attribute](#) ( AWS CLI )

```
$ aws ec2 describe-instance-attribute --instance-id instance_id --attribute sriovNetSupport
```

拡張ネットワーキング属性が設定されていない場合、SriovNetSupport は空です。設定されている場合、SriovNetSupport は次のように設定されています。

```
"SriovNetSupport": {  
  "Value": "simple"  
},
```

- [ec2-modify-instance-attribute](#) ( :Amazon EC2 CLI )

```
$ ec2-describe-instance-attribute instance_id --sriov
```

拡張ネットワーキング属性が設定されていない場合、このコマンドに対する出力はありません。設定されている場合は、以下のような出力が得られます。

```
sriovNetSupport instance_id simple
```

AMIに拡張ネットワーキング属性がすでに設定されているかどうかを確認するには、次のいずれかのコマンドを使用します。

- [describe-image-attribute](#) ( AWS CLI )

```
$ aws ec2 describe-image-attribute --image-id ami_id --attribute sriovNetSupport
```

拡張ネットワーキング属性が設定されていない場合、SriovNetSupport は空です。設定されている場合、SriovNetSupport は次のように設定されています。

```
"SriovNetSupport": {  
  "Value": "simple"  
},
```

- [ec2-modify-image-attribute](#) ( Amazon EC2 CLI )

```
$ ec2-describe-image-attribute ami_id --sriov
```

拡張ネットワーキング属性が設定されていない場合、このコマンドに対する出力はありません。設定されている場合は、以下のような出力が得られます。

```
sriovNetSupport ami_id simple
```

# ストレージ

---

## Abstract

Amazon EC2 でサポートされているデータストレージオプションを使用して、ストレージ要件に対応する方法を学習します。

### Topics

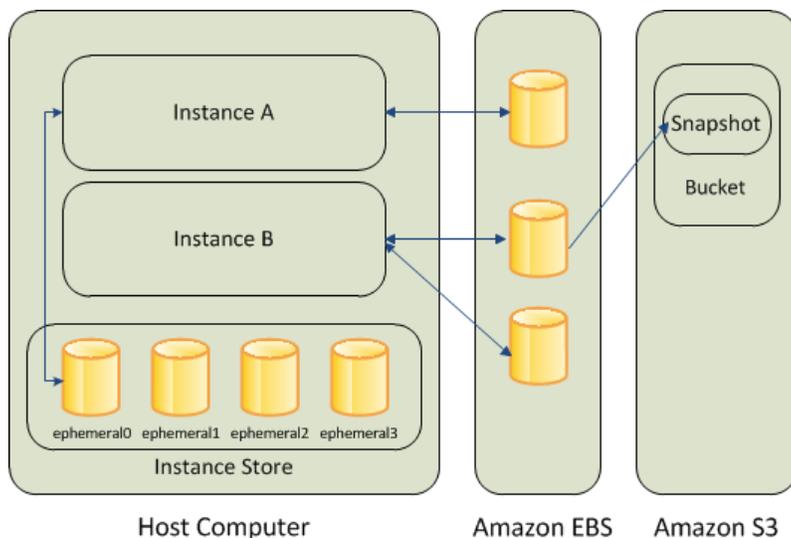
- [Amazon Elastic Block Store\(Amazon EBS\)](#) (p. 566)
- [Amazon EC2 インスタンスストア](#) (p. 631)
- [Amazon Simple Storage Service \( Amazon S3 \)](#) (p. 641)
- [ブロックデバイスマッピング](#) (p. 643)
- [パブリックデータセットの使用](#) (p. 655)

Amazon EC2 にはインスタンスを格納するための、柔軟で使いやすく、コスト効率の良いデータストレージオプションが用意されています。各オプションは独自のパフォーマンスと耐久性を備えています。これらのストレージオプションは、要件に応じて個別に使用することも、組み合わせて使用することもできます。

このセクションを読むことで、Amazon Elastic Compute Cloud がサポートするデータストレージオプションを使用して、特定の要件に対応する方法を十分に理解できるようになるはずです。例えば、次のようなストレージオプションがあります。

- [Amazon Elastic Block Store \( Amazon EBS \)](#)
- [Amazon EC2 インスタンスストア](#)
- [Amazon Simple Storage Service \( Amazon S3 \)](#)

各ストレージタイプの間関係を下の図に示します。



### Amazon EBS

Amazon EBS は、実行中の Amazon EC2 インスタンスにアタッチできる、堅牢なブロックレベルのストレージボリュームを提供します。この Amazon EBS は、細かな更新を頻繁に行う必要があるデータを対象とした主要ストレージデバイスとして使用できます。例えば、インスタンスでデータベースを実行するときに、Amazon EBS をストレージオプションとして使用することをお勧めします。

Amazon EBS ボリュームは、1つのインスタンスにアタッチできる、未加工、未フォーマットの外部ブロックデバイスのように動作します。これらのボリュームは、Amazon EC2 インスタンスの運用状況から独立した永続性を持ちます。インスタンスにアタッチした後の EBS ボリュームは、他の物理ハードドライブと同じように使用できます。前の図に示したように、複数のボリュームをインスタンスにアタッチできます。1つのインスタンスから EBS ボリュームをデタッチし、別のインスタンスにアタッチできます。

データのバックアップコピーを保持するには、EBS ボリュームのスナップショットを作成して Amazon S3 に保存します。スナップショットから新しい Amazon EBS ボリュームを作成して、別のインスタンスにアタッチすることもできます。詳細については、[Amazon Elastic Block Store \(Amazon EBS\) \(p. 566\)](#) を参照してください。

### Amazon EC2 インスタンスストア

多くの Amazon EC2 インスタンスは、ホストコンピュータに物理的にアタッチされたディスクからストレージにアクセスできます。このディスクストレージは、インスタンスストアと呼ばれます。インスタンスストアは、Amazon EC2 インスタンス用のブロックレベルの一時ストレージを提供します。インスタンスストアボリュームのデータは、関連する Amazon EC2 インスタンスの存続中のみ保持されます。インスタンスを停止または終了すると、インスタンスストアボリュームのすべてのデータが失われます。詳細については、「[Amazon EC2 インスタンスストア \(p. 631\)](#)」を参照してください。

### Amazon S3

Amazon S3 はインターネットデータ用のリポジトリです。Amazon S3 により、低コストで信頼性に優れたデータストレージインフラストラクチャが実現します。ウェブスケールのコンピューティングをさらに簡単に行えるように設計されており、Amazon EC2 内から、またはウェブ上のどこからでも、いつでも必要な量だけデータを格納および取得できます。例えば、データとアプリケーションのバックアップコピーを格納するときに S3 を使用できます。詳細については、「[Amazon Simple Storage Service \(Amazon S3\) \(p. 641\)](#)」を参照してください。

### ストレージの追加

AMIからインスタンスを起動するたびに、そのインスタンス用のルートストレージデバイスが作成されます。ルートストレージデバイスには、インスタンスの起動に必要な情報すべてが含まれます。AMIを作成するとき、またはブロックデバイスマッピングを使用してインスタンスを起動するときに、ルートデバイスボリュームの他にストレージボリュームを指定できます。詳細については、「[ブロックデバイスマッピング \(p. 643\)](#)」を参照してください。

実行中のインスタンスにEBS ボリュームをアタッチすることもできます。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#)」を参照してください。

## Amazon Elastic Block Store(Amazon EBS)

### Abstract

Amazon EC2 インスタンスで使用するためのブロックレベルのストレージボリュームを提供する、Amazon Elastic Block Store を使用します。

Amazon Elastic Block Store ( Amazon EBS ) は、Amazon EC2 インスタンスで使用するためのブロックレベルのストレージボリュームを提供します。Amazon EBS ボリュームは、同じアベイラビリティゾーンにある実行中のどのインスタンスにもアタッチできる、可用性と信頼性に優れたストレージボリュームです。Amazon EC2 にアタッチされた Amazon EBS ボリュームは、インスタンスの運用状況から、独立した永続性を保つストレージボリュームとして表示されます。Amazon EBS の場合は、使用した分のみをお支払いいただきます。Amazon EBS の価格設定の詳細については、[Amazon Elastic Block Store ページ](#)の概算費用のセクションを参照してください。

データが頻繁に変更され、さらに長期永続性がデータに必要な場合に、Amazon EBS の使用をお勧めします。Amazon EBS ボリュームは、ファイルシステムの主要ストレージやデータベースとしての使用に特に適しています。また、細かい更新が必要なアプリケーションや、ブロックレベルの未初期化のストレージを使用する必要があるアプリケーションにも適しています。Amazon EBS は、データセット全体でランダムな読み取り/書き込みが頻繁に発生するデータベーススタイルのアプリケーションで特に役立ちます。

AWS アカウントによって規定される制限内であれば、同一のインスタンスに複数のボリュームをアタッチできます。アカウントには、使用できる Amazon EBS ボリュームの数と、利用可能なストレージの総量に制限があります。これらの制限と、制限を緩和するよう要求する方法の詳細については、[Request to Increase the Amazon EBS Volume Limit](#) を参照してください。

### Topics

- [Amazon EBS の機能 \(p. 566\)](#)
- [Amazon EBS ボリューム \(p. 567\)](#)
- [Amazon EBS スナップショット \(p. 605\)](#)
- [Amazon EBS ボリュームのパフォーマンス \(p. 611\)](#)
- [Amazon EBS API およびコマンド概要 \(p. 629\)](#)

## Amazon EBS の機能

- 1 GB ~ 1 TB のサイズの Amazon EBS ストレージボリュームを作成し、デバイスとして Amazon EC2 インスタンスにマウントすることができます。複数のボリュームを同じインスタンスにマウントできますが、各ボリュームをアタッチできるインスタンスは 1 度に 1 つだけです。詳細については、[新しい Amazon EBS ボリュームの作成 \(p. 570\)](#) を参照してください。
- Amazon EBS Provisioned I/O Operations Per Second ボリュームでは、ボリュームあたり最大 4000 IOPS の特定レベルの I/O パフォーマンスをプロビジョニングできます。この機能を利用すると、Amazon EC2 インスタンスあたり数千 IOPS という規模に拡張したときも、期待どおりのパフォーマンスを得ることができます。詳細については、[Provisioned IOPS ボリューム \(p. 569\)](#) を参照してください。

- Amazon EBS ボリュームの動作は、未初期化のブロックデバイスに似ています。これらのボリュームにファイルシステムを構築したり、ブロックデバイスを使用する別の方法（ハードドライブとして使用など）で使用したりできます。ファイルシステムの作成およびボリュームのマウントの詳細については、[Amazon EBS ボリュームを使用できるようにする \(p. 576\)](#) を参照してください。
- Amazon EBS ボリュームの特定の時点におけるスナップショットを作成し、Amazon S3 に保管できます。長期的な耐久性を実現するために、スナップショットはデータを保護します。また、スナップショットは、新しい Amazon EBS ボリュームの開始点として使用できます。1 つのスナップショットからインスタンス化できるボリュームの数に制限はありません。これらのスナップショットは、複数の AWS リージョンにわたってコピーできます。詳細については、[Amazon EBS スナップショット \(p. 605\)](#) を参照してください。
- Amazon EBS ボリュームは、特定のアベイラビリティゾーンで作成され、そのアベイラビリティゾーン内のインスタンスにアタッチできます。アベイラビリティゾーンの外部でボリュームを使用できるようにするには、スナップショットを作成し、そのスナップショットをそのリージョン内の新しいボリュームに復元できます。スナップショットをその他のリージョンにコピーしてから、そのリージョン内の新しいボリュームに復元できるので、地理的な拡大やデータセンターの移行、災害復旧など、複数の AWS リージョンをより容易に活用することができます。詳細については、[Amazon EBS スナップショットの作成 \(p. 606\)](#)、[スナップショットからの Amazon EBS ボリュームの復元 \(p. 571\)](#)、および [Amazon EBS スナップショットのコピー \(p. 608\)](#) を参照してください。
- パブリックデータセットスナップショットの大きなリポジトリを Amazon EBS ボリュームに復元し、AWS クラウドベースのアプリケーションにシームレスに統合することができます。詳細については、[パブリックデータセットの使用 \(p. 655\)](#) を参照してください。
- AWS Management Console を介して、帯域幅、スループット、レイテンシー、平均キュー長などのパフォーマンスメトリックスを使用できます。Amazon CloudWatch のこれらのメトリックスを使用してボリュームのパフォーマンスをモニタリングすると、アプリケーションに対して十分なパフォーマンスを提供できているか、無駄なリソースにコストを費やしていないかを確認できます。詳細については、[Amazon EBS ボリュームのパフォーマンス \(p. 611\)](#) を参照してください。

## Amazon EBS ボリューム

### Abstract

Amazon EBS ボリュームは、頻繁に更新されるデータのプライマリストレージ用の 1 つの EC2 インスタンスにアタッチできるブロックレベルのストレージデバイスとして使用します。

Amazon EBS ボリュームは、1 つの EC2 インスタンスにアタッチできる、耐久性に優れたブロックレベルのストレージボリュームです。Amazon EBS ボリュームは、インスタンス用のシステムドライブ、データベースアプリケーションのストレージなど、頻繁に更新する必要があるデータのプライマリストレージとして使用できます。EBS ボリュームは、EC2 インスタンスの運用状況から独立した永続性を持ちます。インスタンスにアタッチした後のボリュームは、他の物理ハードドライブと同じように使用できます。Amazon EBS には、標準とプロビジョンド IOPS の 2 つのボリュームタイプがあります。この 2 つはパフォーマンス特性と料金が異なるため、アプリケーションのニーズに応じてストレージのパフォーマンスとコストを調整できます。詳細については、[Amazon EBS ボリュームの種類 \(p. 569\)](#) を参照してください。

### Topics

- [Amazon EBS ボリュームを使用するメリット \(p. 568\)](#)
- [Amazon EBS ボリュームの種類 \(p. 569\)](#)
- [新しい Amazon EBS ボリュームの作成 \(p. 570\)](#)
- [スナップショットからの Amazon EBS ボリュームの復元 \(p. 571\)](#)
- [インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#)
- [Amazon EBS ボリュームを使用できるようにする \(p. 576\)](#)
- [ボリューム情報を表示する \(p. 580\)](#)
- [ボリュームのステータスのモニタリング \(p. 580\)](#)

- [インスタンスからの Amazon EBS ボリュームのデタッチ \(p. 591\)](#)
- [Amazon EBS ボリュームの削除 \(p. 592\)](#)
- [ボリュームのストレージ領域を拡張する \(p. 593\)](#)
- [Linux パーティションを拡張する \(p. 600\)](#)

## Amazon EBS ボリュームを使用するメリット

### データの可用性

アベイラビリティゾーン内に Amazon EBS ボリュームを作成すると、そのボリュームは同じゾーン内で自動的にレプリケートされます。これは、1つのハードウェアコンポーネントの障害が原因でデータが失われることを防ぐためです。作成したボリュームは、同じアベイラビリティゾーン内の任意の Amazon EC2 インスタンスにアタッチできます。アタッチしたボリュームは、ハードドライブや他の物理デバイスと同じようなネイティブブロックとして表示されます。この時点で、インスタンスからボリュームへのインタラクティブな操作は、ローカルドライブと同じように行うことができます。また、ext3 (Linux) や NTFS (Windows) などのファイルシステムを使って Amazon EBS ボリュームをフォーマットしたり、アプリケーションをインストールしたりできます。

同じアベイラビリティゾーン内で Amazon EBS ボリュームをアタッチできるインスタンスは 1 度に 1 つだけです。ただし、1つのインスタンスに複数のボリュームをアタッチすることはできません。指定したデバイスに複数のボリュームをアタッチする場合は、ボリュームにまたがってデータをストライプすることで I/O とスループットのパフォーマンスを向上させることができます。

Amazon EBS ボリュームのモニタリングデータは無料で取得できます (Amazon EBS-Backed インスタンスのルートデバイスボリュームのデータも含まれます)。詳細については、「[CloudWatch によるボリュームのモニタリング \(p. 581\)](#)」を参照してください。

### データの永続性

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

デフォルトでは、実行中のインスタンスにアタッチされている Amazon EBS ボリュームは、インスタンスが終了すると、データはそのままの状態です。自動的にインスタンスからのアタッチが解除されます。アタッチが解除されたボリュームは、新しいインスタンスに再アタッチできるので、迅速な復旧が可能です。Amazon EBS-Backed インスタンスを使用している場合は、アタッチしたボリュームに格納されているデータに影響を与えずに、インスタンスを停止および再起動できます。ボリュームは停止/起動のサイクルを通じてアタッチされたままです。これにより、データセットの処理と格納を無限に行うことができ、必要なときだけ処理リソースとストレージリソースが使用されるようになります。データセットは、ボリュームを明示的に削除するまでボリューム内に維持されます。削除したボリュームは、どのインスタンスにもアタッチできなくなります。

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

### スナップショット

Amazon EBS は、Amazon EC2 ボリュームのスナップショット (バックアップ) を作成し、ボリューム内のデータのコピーを Amazon S3 に書き込む機能を備えています。そこで、データは複数のアベイラビリティゾーンに冗長的に保存されます。スナップショットを作成するために、対象のボリュームが実行中のインスタンスにアタッチされている必要はありません。ボリュームにデータを書き込み続けながら、そのボリュームのスナップショットを定期的に作成して、新しいボリュームのベースラインとして使用できます。このスナップショットは、新しい Amazon EBS ボリュームを複数作成したり、ボリュームサイズを拡張したり、アベイラビリティゾーン間でボリュームを移動したりするときに使用できます。

スナップショットから新規ポリユームを作成する場合、このポリユームはスナップショット作成時における元のポリユームの正確なコピーになります。オプションで別のポリユームサイズや別のアベイラビリティゾーンを指定することもできます。この機能は、既存のポリユームサイズを拡大したり、重複するポリユームを新しいアベイラビリティゾーンに作成したりする場合に便利です。スナップショットは特定のAWSアカウントと共有できます。また、一般公開も可能です。スナップショットを作成すると、Amazon S3でポリユームの合計サイズに基づいて、料金がかかります。ポリユームのその後のスナップショットについては、ポリユームの元のサイズを超える追加データ分にはのみ料金がかかります。

Amazon EBS のスナップショットは差分バックアップです。つまり、最後にスナップショットを作成した時点から、ポリユーム上で変更のあるブロックだけが保存されます。例えば、100 GiB のデータが格納されているポリユームがあるとします。最後にスナップショットを作成してから、そのうちの 5 GiB 分のデータしか変更されていない場合は、その変更された 5 GiB のデータだけが Amazon S3 に書き込まれます。スナップショットの保存は差分ベースで行われるものの、最新のスナップショットさえあればポリユームを復元できるようにスナップショット削除プロセスは設計されています。

ポリユームとスナップショットを分類および管理しやすくするため、任意のメタデータでタグ付けることができます。詳細については、「[Amazon EC2 リソースにタグを付ける \(p. 662\)](#)」を参照してください。

## Amazon EBS ポリユームの種類

### Abstract

標準とプロビジョンド IOPS という 2 種類のポリユームのいずれかを使用しているアプリケーションの必要性に合わせて、ストレージのパフォーマンスとコストを調整します。

Amazon EBS には、標準とプロビジョンド IOPS ( 入力または出力操作/秒 ) の 2 種類のポリユームがあります。この 2 つはパフォーマンス特性と料金が異なるため、アプリケーションのニーズに応じてストレージのパフォーマンスとコストを調整できます。

#### 標準ポリユーム

標準ポリユームは、コスト効果に優れたストレージであり、アプリケーションの I/O 要件が少ないかまたはバースト性である場合に適しています。これらのポリユームは、平均約 100 IOPS を実現し、バースト能力は最大約数百 IOPS です。ポリユームのサイズは 1 GB ~ 1 TB です。サイズの拡張とパフォーマンスの向上をご希望の場合は、複数の標準ポリユームを 1 つの RAID 構成内でまとめてストライプ構成することができます。次の使用シナリオ例は、Amazon EBS 標準ポリユームに適しています。

- ファイルサーバー
- ログ処理
- 低トラフィックのウェブサイト
- 分析
- システムブートポリユーム

#### Provisioned IOPS ポリユーム

Provisioned IOPS ポリユームは、ランダムアクセス I/O スループットにおけるストレージパフォーマンスと整合性が重要な、I/O 集約型ワークロード ( 特にデータベースワークロード ) のニーズを満たすように設計されています。ポリユーム作成時に IOPS レートを指定します。Amazon EBS は 1 年間のうちの 99.9%、プロビジョンド IOPS をパフォーマンスの 10% 内で提供します。

Provisioned IOPS ポリユームのサイズは 10 GB ~ 1 TB です。1 ポリユームあたり、最大 4000 IOPS をプロビジョニングできます。要求されたポリユームサイズに対してプロビジョニングされる IOPS の比率は最大 30 です。例えば、3000 IOPS のポリユームには少なくとも 100 GB のサイズが必要です。サイズの拡張とパフォーマンスの向上をご希望の場合、複数のポリユームを 1 つにまとめてストライプ構

成にすることができます。次の使用シナリオ例は、Amazon EBS Provisioned IOPS ボリュームに適しています。

- ビジネスアプリケーション
- MongoDB
- Microsoft SQL Server
- MySQL
- PostgreSQL
- Oracle

インスタンスの構成、I/O 特性、ワークロードのデマンドなど、Amazon EBS ボリュームのパフォーマンスに影響を与える可能性がある要因は複数存在します。Provisioned IOPS ボリュームを最大限活用するための詳細は、[Amazon EBS ボリュームのパフォーマンス \(p. 611\)](#) を参照してください。

## 新しい Amazon EBS ボリュームの作成

### Abstract

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

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

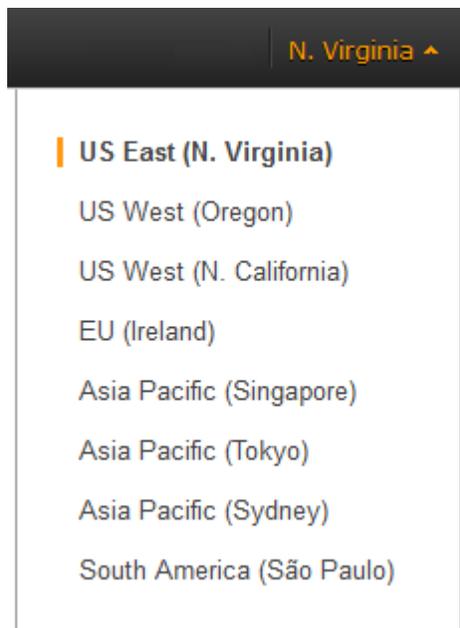
また、ブロックデバイスマッピングを指定すると、インスタンスの起動時に、Amazon EBS ボリュームを作成してアタッチすることもできます。詳細については、[インスタンスの起動 \(p. 325\)](#) および [ブロックデバイスマッピング \(p. 643\)](#) を参照してください。以前作成したスナップショットからボリュームを復元できます。詳細については、[スナップショットからの Amazon EBS ボリュームの復元 \(p. 571\)](#) を参照してください。

高性能のストレージシナリオ用にボリュームを作成する場合、Provisioned IOPS ボリュームを使用して、アプリケーションをサポートするために十分な帯域幅を持つインスタンス (EBS 最適化インスタンス、10 ギガビットネットワーク接続を備えたインスタンスなど) にアタッチしてください。詳細については、[Amazon EC2 インスタンスの構成 \(p. 612\)](#) を参照してください。

新しく作成した Amazon EBS ボリュームにデータブロックを初めて書き込む場合、通常よりも長いレイテンシーが発生する可能性があります。本番ワークロードで書き込みレイテンシーが増加する可能性を回避するには、最初にボリュームのすべてのブロックに書き込みを行い、最適なパフォーマンスを確保する必要があります。この方法を「ボリュームの事前ウォーミング」と呼びます。詳細については、[Amazon EBS ボリュームの事前ウォーミング \(p. 615\)](#) を参照してください。

新しい Amazon EBS ボリュームを作成するには コンソールを使用する

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. ナビゲーションバーから、ボリュームを作成するリージョンを選択します。この選択は、リージョン間で共有できるのが一部の Amazon EC2 リソースのみであるため、重要です。詳細については、「[リソースの場所 \(p. 658\)](#)」を参照してください。



3. ナビゲーションペインで [Volumes] をクリックします。
4. 上段のペインの上で、[Create Volume] をクリックします。
5. [Create Volume] ダイアログボックスで、[Volume Type] リストの [Standard] または [Provisioned IOPS] を選択します。詳細については、[Amazon EBS ボリュームの種類 \(p. 569\)](#) を参照してください。
6. [Size] ボックスに、ボリュームのサイズを GiB で入力します。
7. プロビジョンド IOPS ボリュームの場合、[IOPS] ボックスに、ボリュームがサポートできる IOPS (1 秒あたりの入力/出力オペレーションの数) の最大数を入力します。
8. [Availability Zone] リストで、ボリュームを作成するアベイラビリティゾーンを選択します。詳細については、[リージョンとアベイラビリティゾーン \(p. 8\)](#) を参照してください。
9. [Yes, Create] をクリックします。

コマンドラインを使用して、新しい Amazon EBS ボリュームを作成するには

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

- [create-volume](#) ( AWS CLI )
- [ec2-create-volume](#) ( Amazon EC2 CLI )
- [New-EC2Volume](#) ( AWS Tools for Windows PowerShell )

## スナップショットからの Amazon EBS ボリュームの復元

### Abstract

Amazon S3 に格納されているスナップショットのデータを使用して、Amazon EBS ボリュームを復元します。

Amazon S3 に格納されたスナップショットから、データが存在する Amazon EBS ボリュームを復元できます。ボリュームの復元元となるスナップショットの ID が必要です。また、スナップショットへのアクセス許可も必要です。スナップショットの詳細については、[Amazon EBS スナップショット \(p. 605\)](#) を参照してください。

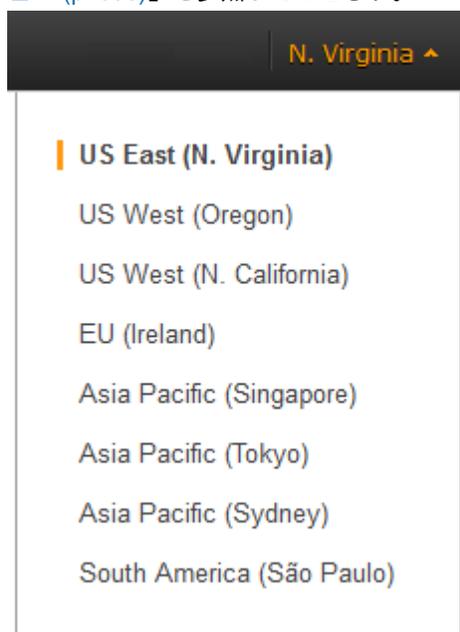
既存の Amazon S3 スナップショットから作成された新しいボリュームは、バックグラウンドで時間をかけて読み込まれます。つまり、スナップショットを基にボリュームを作成した後は、Amazon S3 から Amazon EBS ボリュームにすべてのデータが転送されるのを待たなくても、アタッチしたインスタンスからボリュームとそのすべてのデータへのアクセスを開始できます。まだ読み込まれていないデータに対してインスタンスからのアクセスがあった場合、ボリュームは要求されたデータを Amazon S3 から即座にダウンロードして、引き続き残りのデータをバックグラウンドで読み込みます。

新しく復元された Amazon EBS ボリュームのデータブロックが初めてアクセスされた場合、通常より長いレイテンシーが発生する可能性があります。実稼働時に読み取りおよび書き込みレイテンシーが増加する可能性を回避するには、最初にボリュームのすべてのブロックの読み取りを行い、最適なパフォーマンスを確保する必要があります。この方法を「ボリュームの事前ウォーミング」と呼びます。詳細については、[Amazon EBS ボリュームの事前ウォーミング \(p. 615\)](#) を参照してください。

スナップショットから Amazon EBS ボリュームを復元するには コンソールを使用する

次のようにコンソールを使って、スナップショットから Amazon EBS ボリュームを復元することができます

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. ナビゲーションバーから、スナップショットのリージョンを選択します。リージョン間で共有できるのが一部の Amazon EC2 リソースのみであるため、ここでの選択は重要です。詳細については、[リソースの場所 \(p. 658\)](#) を参照してください。スナップショットを別のリージョンのボリュームに復元する必要がある場合、スナップショットを新しいリージョンにコピーし、そのリージョンのボリュームに復元することができます。詳細については、「[Amazon EBS スナップショットのコピー \(p. 608\)](#)」を参照してください。



3. ナビゲーションペインで [Volumes] をクリックします。
4. 上段のペインの上で、[Create Volume] をクリックします。
5. [Create Volume] ダイアログボックスで、[Volume Type] リストの [Standard] または [Provisioned IOPS] を選択します。詳細については、[Amazon EBS ボリュームの種類 \(p. 569\)](#) を参照してください。
6. [Snapshot] フィールドに、ボリュームの復元元となるスナップショットの ID または説明を入力し、表示されたオプションリストから選択します。
7. [Size] ボックスに、ボリュームのサイズ (GiB) を入力するか、またはスナップショットのデフォルトのサイズが適切であるかを確認します。



#### Note

ボリュームサイズとスナップショット ID の両方を指定した場合は、スナップショットサイズ以上のサイズにする必要があります。ボリュームの種類とスナップショット ID を選択すると、ボリュームの最小サイズと最大サイズが [Size] リストの横に表示されます。そのスナップショットからの AWS Marketplace 製品コードがある場合は、その製品コードがボリュームに反映されます。

8. プロビジョンド IOPS の場合、[IOPS] ボックスに、ボリュームがサポートできる IOPS ( 1 秒あたりの入力/出力オペレーションの数 ) の最大数を入力します。
9. [Availability Zone] リストで、ボリュームを作成するアベイラビリティゾーンを選択します。詳細については、[リージョンとアベイラビリティゾーン \(p. 8\)](#) を参照してください。
10. [Yes, Create] をクリックします。



#### Important

スナップショットを、そのスナップショットのデフォルトよりも大きなボリュームに復元する場合、追加容量の利点を活用できるように、ボリュームのファイルシステムを拡張する必要があります。詳細については、[ボリュームのストレージ領域を拡張する \(p. 593\)](#) を参照してください。

コマンドラインを使用して、新しい Amazon EBS ボリュームを復元するには

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

- [create-volume](#) ( AWS CLI )
- [ec2-create-volume](#) ( Amazon EC2 CLI )
- [New-EC2Volume](#) ( AWS Tools for Windows PowerShell )

## インスタンスへの Amazon EBS ボリュームのアタッチ

### Abstract

作成した Amazon EBS ボリュームを、Amazon EC2 インスタンスにアタッチします。

このセクションでは、作成した Amazon EBS ボリュームをインスタンスにアタッチする方法について説明します。

次の表は、Amazon EC2 で使用できるデバイス名を表示しています。これらの名前は、実行中のインスタンスにボリュームをアタッチするとき、またはブロックデバイスマッピングによるインスタンスの起動時にボリュームをアタッチするときに指定できます。インスタンスのブロックデバイスドライバは、ボリュームのマウント時に実際のボリューム名を割り当てますが、これらの名前は、Amazon EC2 が推奨する名前とは異なる可能性があります。インスタンスストアの詳細については、[Amazon EC2 インスタンスストア \(p. 631\)](#) を参照してください。ルートデバイスストレージの詳細については、「[Amazon EC2 ルートデバイスボリューム \(p. 15\)](#)」を参照してください。

インスタンスタイプ	接続先として可能	ルート用に予約済み	インスタンスストアボリューム	EBS 接続に推奨されるもの
Linux/Unix ( 準仮想 )	/dev/sd[a-z] /dev/sd[a-z][1-15] /dev/hd[a-z] /dev/hd[a-z][1-15]	/dev/sda1	/dev/sd[b-e]	/dev/sd[f-p] /dev/sd[f-p][1-6]
Linux/Unix ( HVM )	/dev/sd[a-z] /dev/xvd[b-c][a-z]	AMI による違い /dev/sda1 or /dev/xvda	/dev/sd[b-e] /dev/sd[b-y] ( hs1.8xlarge インスタンスの場合 )	/dev/sd[f-p]
Windows ( HVM )	xvd[a-z] xvd[b-c][a-z] /dev/sda[1-2] /dev/sd[b-e]	/dev/sda1	xvd[a-e] xvdc[a-x] ( hs1.8xlarge インスタンスの場合 )	xvd[f-p] ( Red Hat PV ドライバを含む ) xvd[f-z] ( Citrix PV ドライバを含む )

Linux/Unix のインスタンスタイプの場合は、一部のカスタムカーネルの制約によって使用が /dev/sd[f-p] または /dev/sd[f-p][1-6] に制限される可能性があることが報告されています。/dev/sd[q-z] または /dev/sd[q-z][1-6] の使用に関して問題がある場合は、/dev/sd[f-p] または /dev/sd[f-p][1-6] に切り替えてみてください。



#### Important

末尾の番号の有無にかかわらず、同じデバイス文字を共有するボリュームをアタッチすることはできません。例えば、あるボリュームを /dev/sdc としてアタッチし、別のボリュームを /dev/sdc1 としてアタッチした場合、インスタンスは /dev/sdc だけを認識します。デバイス名で末尾の番号を使用する場合、同じ基本文字を共有するすべてのドライブ名で末尾の番号を使用する必要があります。

インスタンスのカーネルのブロックデバイスドライバに応じて、デバイスは指定した名前とは異なる名前でもアタッチされる可能性があります。例えば、/dev/sdh のデバイス名を指定すると、デバイスはカーネルによって /dev/xvdh または /dev/hdh に名前が変更されます。一部のケースでは、末尾の番号が変更される可能性もあります ( /dev/sda が /dev/xvde になる可能性があります )。Amazon Linux AMI は、名前が変更されたデバイスパスから指定した名前までのシンボリックリンクを作成しますが、その他の AMI の動作は異なることがあります。

ハードウェア仮想マシン ( HVM ) AMI ( ベース Windows およびクラスターコンピューティングイメージなど ) は、デバイス名で末尾の番号の使用をサポートしていません ( xvd[a-p][1-15] )。インスタンス詳細ページの [Virtualization] フィールドで、インスタンスの仮想化の種類を表示できます。

Amazon EC2 では、インスタンスのサイズに応じて、sd[b-e] ( Linux/UNIX ) または xvd[a-e] でインスタンスストアを提供します。これらのデバイス名を使用して Amazon EBS ボリュームに接続できませんが、動作を予測できない可能性があるため、この操作は行わないことを強くお勧めします。

Amazon EC2 Windows AMI には、[Ec2Config Service] と呼ばれる追加サービスがインストールされています。Ec2Config Service はローカルシステムとして実行され、インスタンスの初回起動時に備えてさまざまな機能を実行します。デバイスがドライブにマッピングされると、Ec2Config Service はドライブを初期化およびマウントします。ルートドライブは c:\ として初期化およびマウントされます。インスタンスにアタッチされるインスタンスストアは、d:\、e:\ などとして初期化およびマウントさ

れます。デフォルトでは、Amazon EBS ポリリュームが Windows インスタンスにアタッチされているときは、インスタンス上でいずれかのドライブ文字で表示されます。Ec2Config Service の設定を変更して、独自の指定によって Amazon EBS ポリリュームのドライブ文字を設定できます。詳細については、*Amazon Elastic Compute Cloud Microsoft Windows Guide* の [Using Ec2Config](#) を参照してください。

コンソールを使用して、Amazon EBS ポリリュームをインスタンスにアタッチするには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインで [Volumes] をクリックします。

コンソールに現在のポリリュームのリストが表示されます。

3. ポリリュームを選択し、Attach Volume をクリックします。

[Attach Volume] ダイアログボックスが表示されます。

4. [Instance] ボックスで、ポリリュームにアタッチするインスタンスの名前または ID を入力して、提案されたオプションリストからインスタンスを選択します ( ポリリュームと同じアベイラビリティゾーンに存在するインスタンスのみ表示されます )。
5. [Device] ボックスにデバイス名を入力するか、提案されたデバイス名が適切であることを確認します。
6. [Attach] をクリックし、ポリリュームをインスタンスにアタッチします。ポリリュームとそのアタッチ先インスタンスは同じアベイラビリティゾーンに存在している必要があります。



#### Note

- Windows インスタンスは Red Hat または Citrix 準仮想 ( PV ) ドライバを使用します。Windows インスタンスが Citrix PV ドライバを使用している場合、Amazon EC2 CLI を使用して、合計 25 までの Amazon EBS ポリリュームをアタッチできます。Red Hat PV ドライバを使用する Windows インスタンスは 16 ポリリュームに制限されます。使用しているドライバに関係なく、Amazon EC2 コンソールを使用して、最大 16 の Amazon EBS ポリリュームをアタッチできます。Red Hat から Citrix 準仮想ドライバに Windows インスタンスをアップグレードするには、「[Upgrading Your PV Drivers on Your Windows AMI](#)」を参照してください。
- Citrix PV ドライバを使用する Windows インスタンスに 25 以上のポリリュームをアタッチすることは技術的に可能ですが、パフォーマンスの問題が発生する可能性があるため、推奨されません。
- Amazon S3 オブジェクトを Amazon EBS ポリリュームにプッシュする場合、上記の手順に従って、ポリリュームをアタッチしてから、実行中のインスタンスでデータ転送アプリケーション ( FTP/SFTP や SCP など ) を使用して、データを転送します。

ポリリュームに AWS Marketplace 製品コードがある場合は、次のようになります。

- そのポリリュームのアタッチ先は、停止状態のインスタンスのルートデバイスのみとなります。
- ポリリューム上に存在する AWS Marketplace コードをサブスクライブしている必要があります。
- インスタンスの構成 ( インスタンスタイプ、オペレーティングシステム ) は、その特定の AWS Marketplace コードをサポートするものでなければなりません。例えば、Windows インスタンスからのポリリュームを Linux インスタンスにアタッチすることはできません。
- AWS Marketplace 製品コードがポリリュームからインスタンスにコピーされます。

コマンドラインを使用して、Amazon EBS ポリリュームをインスタンスにアタッチするには

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

- [attach-volume](#) ( AWS CLI )
- [ec2-attach-volume](#) ( Amazon EC2 CLI )
- [Add-EC2Volume](#) ( AWS Tools for Windows PowerShell )

## Amazon EBS ポリユームを使用できるようにする

### Abstract

Amazon EBS ポリユームを使用可能にして、他のポリユームと同じようにアクセスできるようにします。

Amazon EBS ポリユームをインスタンスにアタッチすると、それはブロックデバイスとして表示されます。任意のファイルシステムでポリユームをフォーマットし、マウントできます。Amazon EBS ポリユームを使用できるようにすると、他のポリユームと同じようにアクセスできます。バックアップ目的で、Amazon EBS ポリユームのスナップショットを作成したり、別のポリユームを作成する際のペー斯拉インとして使用したりすることもできます。

### Topics

- [Linux でポリユームを使用できるようにする \(p. 576\)](#)
- [Windows でポリユームを使用できるようにする \(p. 578\)](#)

## Linux でポリユームを使用できるようにする

Linux で Amazon EBS ポリユームを使用できるようにするには

1. SSH を使用してインスタンスに接続します。詳細については、[インスタンスへの接続 \(p. 29\)](#) を参照してください。
2. インスタンスのカーネルのブロックデバイスドライバに応じて、デバイスは指定した名前とは異なる名前でもアタッチされる可能性があります。例えば、`/dev/sdh` のデバイス名を指定すると、デバイスはカーネルによって `/dev/xvdf` または `/dev/hdh` に名前が変更されます。一部のケースでは、末尾の番号が変更される可能性もあります ( `/dev/sda` が `/dev/xvde` になる可能性があります )。Amazon Linux AMI は、名前が変更されたデバイスパスから指定した名前までのシンボリックリンクを作成しますが、その他の AMI の動作は異なることがあります。

`lsblk` コマンドを使用して、使用可能なディスクデバイスとマウントポイント ( 該当する場合 ) を表示し、使用する正しいデバイス名を決定します。

```
[ec2-user ~]$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf  202:80  0 100G  0 disk
xvda1 202:1   0    8G  0 disk /
```

`lsblk` の出力は、フルデバイスパスから `/dev/` を削除します。この例では、`/dev/xvda1` はルートデバイスとしてマウントされ ( `MOUNTPOINT` は、Linux ファイルシステム階層のルートである `/` として表示されている点に注意してください )、`/dev/xvdf` がアタッチされますが、マウントはまだ実行されていません。

3. ポリユームにファイルシステムを作成する必要があるかどうかを決定します。新しいポリユームは未加工のブロックデバイスで、マウントして使用する前に、ポリユームにファイルシステムを作成する必要があります。スナップショットから復元されたポリユームは、ファイルシステムを備えている可能性が高くなります。既存のファイルシステムの上に新しいファイルシステムを作成すると、データが上書きされます。特殊な情報 ( ファイルシステムの種類など ) を一覧表示するには、`sudo file -s device` コマンドを使用します。

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

前のコマンドの出力にデバイスの `data` だけが表示された場合、デバイスにはファイルシステムが存在していないので、ファイルシステムを作成する必要があります。Step 4 (p. 577) に移動できます。ファイルシステムを搭載しているデバイスでこのコマンドを実行すると、出力が異なります。

```
[ec2-user ~]$ sudo file -s /dev/xvda1
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-a14c-8c64d6819362 (needs journal recovery) (extents) (large files) (huge files)
```

上の例では、デバイスに `Linux rev 1.0 ext4 filesystem data` が含まれているため、このポリユームに作成したファイルシステムは不要です (出力にファイルシステムデータが表示される場合、Step 4 (p. 577) をスキップできます)。

4. (オプション) 次のコマンドを使って、ポリユームで `ext4` ファイルシステムを作成します。`device_name` をデバイス名 (`/dev/xvdf` など) に置き換えます。アプリケーションの要件またはオペレーティングシステムの制限に応じて、`ext3`、`XFS` など、異なるファイルシステムの種類を選択できます。



#### Caution

この手順では、空のポリユームをマウントすることを前提としています。既にデータが含まれるポリユーム (スナップショットから復元したポリユームなど) をマウントする場合は、ポリユームのマウント前に `mkfs` を使用しないでください (代わりに、次の手順に移動します)。ステップ 1 を実行した場合、ポリユームがフォーマットされ、既存のデータが削除されます。

```
[ec2-user ~]$ sudo mkfs -t ext4 device_name
```

5. 次のコマンドを使用して、ポリユームのマウントポイントディレクトリを作成します。マウントポイントとは、ポリユームをマウントした後、ファイルシステムツリー内でポリユームが配置され、ファイルの読み書きが実行される場所です。`mount_point` を場所 (`/data` など) に置き換えます。

```
[ec2-user ~]$ sudo mkdir mount_point
```

6. 次のコマンドを使用して、作成した場所にポリユームをマウントします。

```
[ec2-user ~]$ sudo mount device_name mount_point
```

7. (オプション) システムブート時に常に、この Amazon EBS ポリユームをマウントするには、`/etc/fstab` ファイルにデバイス用のエントリを追加します。
  - a. `/etc/fstab` ファイルのバックアップコピーを作成すると、編集中に誤って破壊/削除してしまった場合にこのコピーを使用できます。

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

- b. お好みのテキストエディタ (`nano`、`vim` など) を使って `/etc/fstab` ファイルを開きます。
  - c. 次のフォーマットを使って、ポリユームのファイルの最後に新しい行を追加します。

```
device_name mount_point file_system_type fs_mntops fs_freq fs_passno
```

この行の最後の3つのフィールドは、ファイルシステムのマウントオプション、ファイルシステムのダンプ頻度、ブート時に実行されるファイルシステムチェックの順番です。これらの値がわからない場合、以下の例の値を使用してください ( defaults 0 2 )。 /etc/fstab エントリの詳細については、fstab マニュアルページを参照してください ( コマンドラインで man fstab と入力します )。例えば、マウントポイント /data の /dev/xvdf デバイ스에 ext4 ファイルシステムをマウントするには、 /etc/fstab に次のエントリを追加します。

```
/dev/xvdf /data ext4 defaults 0 2
```

- d. 新しいエントリを /etc/fstab に追加した後、エントリが正しく動作するかを確認する必要があります。 sudo mount -a コマンドを実行すると、 /etc/fstab にすべてのファイルシステムがマウントされます。

```
[ec2-user ~]$ sudo mount -a
```

上記のコマンドを実行してもエラーが発生しない場合、 /etc/fstab ファイルに問題はありません。次回ブート時にシステムファイルは自動マウントを実行します。このコマンドを実行してエラーが発生した場合、エラーを調べて、 /etc/fstab を修正してください。



#### Warning

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

- e. ( オプション ) /etc/fstab のエラーの修正方法が不明な場合、次のコマンドを使って、いつでもバックアップの /etc/fstab ファイルを復元することができます。

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

## Windows でボリュームを使用できるようにする

Amazon EBS ボリュームを使用するには

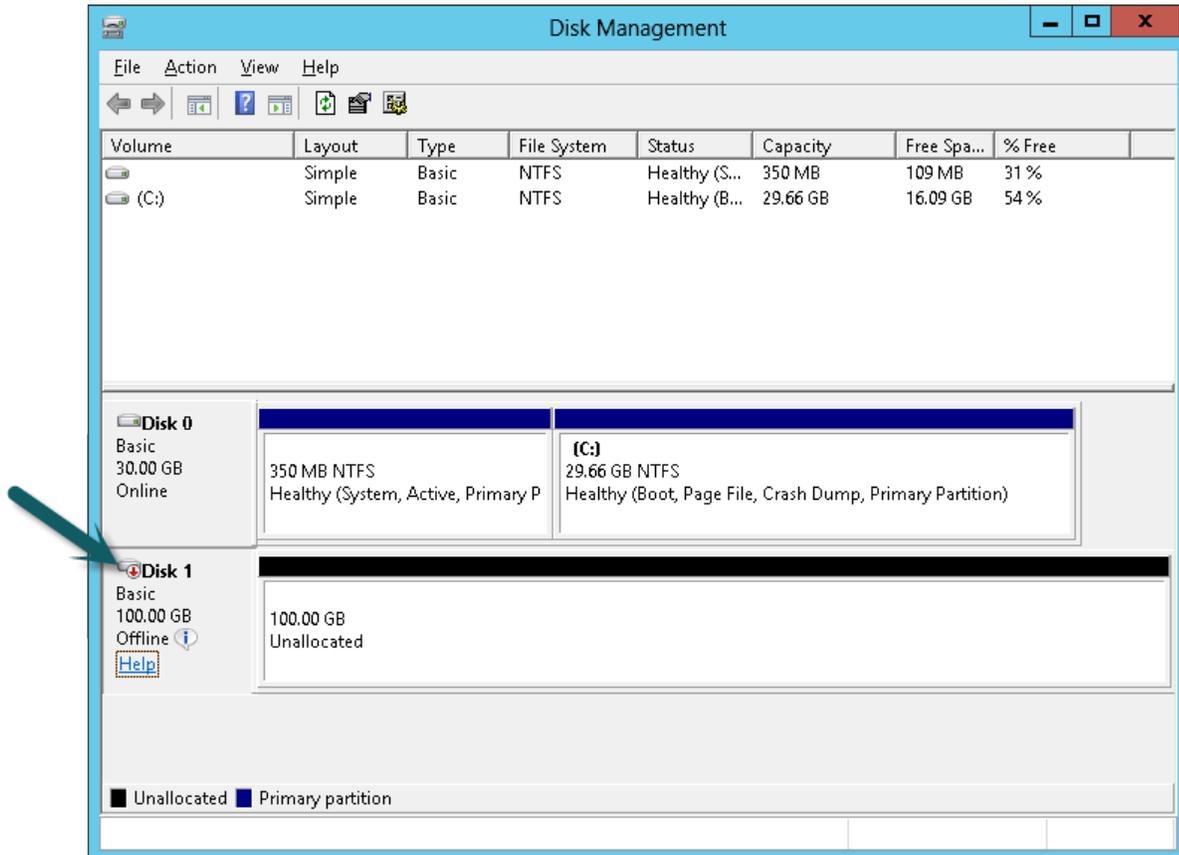
1. リモートデスクトップを使用してインスタンスにログインします。
2. • Windows Server 2012: [スタート] 画面に移動します。  
• Windows Server 2008: タスクバーの [Start] をクリックし、 [Run] をクリックします。
3. diskmgmt.msc と入力して、 Enter を押します。 Disk Managementユーティリティが表示されます。



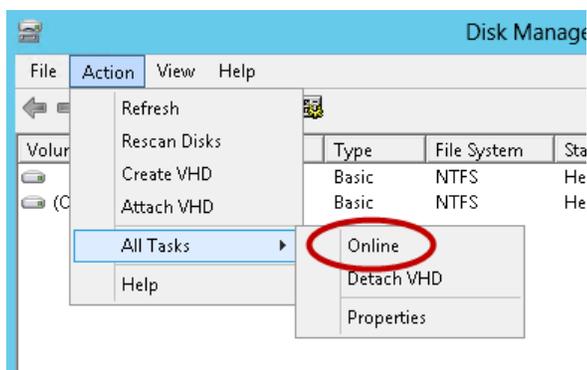
#### Caution

すでにデータが含まれるボリューム ( パブリックデータセットなど ) をマウントする場合は、ボリュームを再フォーマットしないように注意してください。再フォーマットすると、既存のデータが削除されます。

4. 新しい Amazon EBS ボリュームを表すディスクを選択します。



5. [Disk Management] メニューで、[Action] – [All Tasks] – [Online] を選択します。



6. 新しいディスクを使えるようにするには、初期化する必要があります。ディスクを初期化するには
- ディスクの管理ユーティリティで、新しい Amazon EBS ボリュームディスクを選択します。
  - [Disk Management] メニューで、[Action] – [All Tasks] – [Initialize Disk] を選択します。
  - [Initialize Disk] ダイアログで、初期化するディスクを選択し、目的のパーティションスタイルを選択して、[OK] をクリックします。

新しい Amazon EBS ボリュームが使用可能な状態になります。このファイルシステムに書き込まれるデータはすべて Amazon EBS ボリュームに書き込まれますが、デバイスを使用するアプリケーションには透過的になります。

## ボリューム情報を表示する

### Abstract

Amazon EBS ボリュームに関する情報 ( ボリュームのアタッチ先である特定の Amazon EC2 インスタンスなど ) を一覧表示します。

AWS Management Console に、選択したリージョン内のすべてのボリュームの詳細情報を一度に表示することができます。単一ボリュームの詳細 ( サイズ、ボリュームタイプ、ボリュームがアタッチしている特定のインスタンスなど ) について表示することもできます。

コンソールを使用して、Amazon EBS ボリュームについての情報を表示するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインで [Volumes] をクリックします。

現在のボリュームおよびボリュームのアタッチ先インスタンスのリストがコンソールに表示されます。

3. ボリュームの詳細情報を表示するには、そのボリュームを選択します。

ボリュームの情報が下のペインに表示されます。

コマンドラインを使用して、Amazon EBS ボリュームについての情報を表示するには

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

- [describe-volumes](#) ( AWS CLI )
- [ec2-describe-volumes](#) ( Amazon EC2 CLI )
- [Get-EC2Volume](#) ( AWS Tools for Windows PowerShell )

## ボリュームのステータスのモニタリング

### Abstract

Amazon CloudWatch メトリックスおよびボリュームステータスチェックを使用して、Amazon Elastic Block Store ボリュームをモニタリングします。

アマゾン ウェブ サービス ( AWS ) では、CloudWatch メトリックスやボリュームステータスチェックなどのデータが自動的に提供されます。これらのデータを使用して、Amazon Elastic Block Store ( Amazon EBS ) ボリュームをモニタリングすることができます。

### Topics

- [CloudWatch によるボリュームのモニタリング \(p. 581\)](#)
- [ステータスチェックによるボリュームのモニタリング \(p. 583\)](#)
- [ボリュームイベントのモニタリング \(p. 585\)](#)
- [障害のあるボリュームの操作 \(p. 587\)](#)
- [AutoEnableIO ボリューム属性の操作 \(p. 589\)](#)

## CloudWatch によるボリュームのモニタリング

CloudWatch メトリックスは、ボリュームの実行動作の表示、分析、それらの実行動作についてのアラームの設定に使用できる統計データです。

次の表は、Amazon EBS ボリュームで利用可能なモニタリングデータのタイプをまとめたものです。

タイプ	説明
基本	5分間のデータを自動的に取得できます。無料。このデータには、Amazon EBS-backed インスタンスのルートデバイスボリュームのデータが含まれます。
詳細	プロビジョニングされた Provisioned IOPS ボリュームは、1 分間のメトリックスを CloudWatch に自動的に送信します。

CloudWatch からデータを取得するときに、`Period` リクエストパラメータを含めて、返されるデータの詳細程度を指定できます。これは、データの収集に使用する期間 (5 分間) とは異なります。有効なデータが確実に返されるように、リクエストには収集期間以上の期間を指定することをお勧めします。

データは、Amazon CloudWatch API または Amazon EC2 コンソールのいずれかを使用して取得できます。コンソールは Amazon CloudWatch API から未加工データを取得し、そのデータに基づいて一連のグラフを表示します。必要に応じて、API のデータまたはコンソールのグラフのいずれかを使用できます。

### Amazon EBS のメトリックス

Amazon CloudWatch `GetMetricStatistics` API を使用して、次の表にある任意の Amazon EBS ボリュームメトリックスを取得できます。類似したメトリックスはグループにまとめてあります。最初の 2 行に挙げたメトリックスは、Amazon EC2 インスタンスのローカルストアにも使用できます。

メトリックス	説明
VolumeReadBytes VolumeWriteBytes	指定期間内に転送された総バイト数。データは、ボリュームがアクティブな場合にのみ Amazon CloudWatch に報告されます。ボリュームがアイドル状態の場合は、Amazon CloudWatch にデータは報告されません。  単位: バイト
VolumeReadOps VolumeWriteOps	指定期間内の I/O 操作の総数。   <b>Note</b>  その期間の 1 秒あたりの I/O 操作回数 ( IOPS ) の平均を算出するには、その期間の操作回数の合計をその期間の秒数で割ります。  単位: Count
VolumeTotalReadTime VolumeTotalWriteTime	指定期間内に完了した操作すべての、操作に要した時間 ( 秒 ) の合計。複数のリクエストが同時に送信された場合は、この合計が期間の長さを超えることがあります。例えば、期間が 5 分間 ( 300 秒 ) で、その期間内に完了した操作の数が 700 あり、1 つの操作に 1 秒かかるとすれば、この値は 700 秒となります。  単位: Seconds

メトリックス	説明
VolumeIdleTime	指定期間内に、読み取りと書き込みのどちらの操作も行われなかった時間 ( 秒 ) の合計。  単位: Seconds
VolumeQueueLength	指定期間内に完了を待っていた読み取りおよび書き込みの操作リクエストの数。  単位: Count
VolumeThroughputPercentage	<p>プロビジョンド IOPS ボリュームにのみ使用します。Amazon EBS ボリュームにプロビジョニングされた合計 IOPS ( 1 秒間あたりの I/O 操作回数 ) に対する、提供された IOPS の割合 ( パーセント )。プロビジョンド IOPS ボリュームは、1 年の 99.9 パーセントにわたり、プロビジョニングされた IOPS の 10 パーセント以内のパフォーマンスを提供します。</p> <p> <b>Note</b></p> <p>書き込みの間、他に保留中の I/O リクエストが 1 分以内になれば、メトリック値は 100% となります。また、お客様が行った操作によって ( 例えば使用率ピーク時にボリュームのスナップショットを作成する、EBS-Optimized インスタンス以外でボリュームを実行する、そのボリュームのデータに初めてアクセスするなど ) EBS ボリュームの I/O 性能が一時的に低下する場合があります。</p> <p>単位: パーセント</p>
VolumeConsumedReadWriteOps	<p>プロビジョンド IOPS ボリュームにのみ使用します。指定された期間内に消費された読み書き操作の合計数 ( 16K キャパシティーユニットに標準化 )。</p> <p>それぞれ 16K より小さい I/O 操作は、1 消費 IOPS とカウントされます。16K より大きい I/O 操作は、16K キャパシティーユニットでカウントされます。たとえば、64K I/O は 4 消費 IOPS としてカウントされます。</p> <p>単位: Count</p>

## Amazon EC2 コンソールのグラフ

ボリュームを作成したら、Amazon EC2 コンソールにアクセスし、ボリュームのモニタリンググラフを表示できます。これらのグラフは、EC2 コンソールの Volumes ページでボリュームを選択すると表示されます。ボリュームの Description タブの隣りに、Monitoring タブが表示されます。次の表は、表示されるグラフをまとめたものです。右側の欄は、各グラフを作成するために Amazon CloudWatch API の未加工データメトリックスがどのように使用されるかを示しています。すべてのグラフの期間は 5 分です。

グラフ名	未加工メトリックスの使用に関する説明
読み込み帯域幅 ( KiB/s )	合計 ( VolumeReadBytes ) / 期間 / 1024
書き込み帯域幅 ( KiB/s )	合計 ( VolumeWriteBytes ) / 期間 / 1024
読み込みスループット ( Ops/s )	合計 ( VolumeReadOps ) / 期間
書き込みスループット ( Ops/s )	合計 ( VolumeWriteOps ) / 期間

グラフ名	未加工メトリックスの使用に関する説明
平均キュー長 ( ops )	平均 ( VolumeQueueLength )
% アイドル時間	合計 ( VolumeldleTime ) / 期間 * 100
平均読み込みサイズ ( KiB/op )	平均 ( VolumeReadBytes ) / 1024
平均書き込みサイズ ( KiB/op )	平均 ( VolumeWriteBytes ) / 1024
平均読み込み待ち時間 ( ms/op )	平均 ( VolumeTotalReadTime ) * 1000
平均書き込み待ち時間 ( ms/op )	平均 ( VolumeTotalWriteTime ) * 1000

平均待ち時間グラフおよび平均サイズグラフでは、期間中に計算されたオペレーション ( 読み込みまたは書き込みのうち、いずれかグラフに該当する方 ) の合計数に基づいて平均が計算されます。

AWS マネジメントコンソールの中に、Amazon CloudWatch のコンソールがあります。この Amazon CloudWatch コンソールでは、すべての AWS リソースメトリックスの検索と一覧表示ができるほか、問題のトラブルシューティングや傾向の発見のためのグラフ表示や、問題を知らせるアラームの作成と編集も行うことができ、アラームと AWS のリソースの概要をひと目で確認できるようになっています。詳細は、*Amazon CloudWatch 開発者ガイド* の「[AWS Management Console](#)」を参照してください。

## ステータスチェックによるボリュームのモニタリング

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

ボリュームステータスチェックは自動的に試行され、成功または失敗のステータスを返します。すべてのチェックが成功した場合、ボリュームのステータスは `ok` です。チェックが失敗した場合、ボリュームのステータスは `impaired` です。ステータスが `insufficient-data` の場合、ボリュームのチェックがまだ実行中である可能性があります。ボリュームステータスチェックの結果を表示して、障害のあるボリュームを特定し、必要なアクションを行うことができます。

ボリュームのデータが潜在的に不整合であると Amazon EBS が判断した場合、デフォルトでは、アタッチされたすべての EC2 インスタンスからそのボリュームへの I/O が無効になります。これにより、データの破損を防ぐことができます。I/O が無効になると、次のボリュームステータスチェックが失敗し、ボリュームステータスは `impaired` になります。さらに、I/O が無効になったこと、およびボリュームへの I/O を有効にすることによってボリュームの障害ステータスを解決できることを伝えるイベントが表示されます。I/O を有効にするまで、インスタンスによるボリュームの使用を継続させるか、その前に `fsck` ( Linux/Unix ) または `chkdsk` ( Windows ) などのコマンドを使用して整合性チェックを実行するかを判断することはできません。

### Note

ボリュームステータスはボリュームステータスチェックに基づいており、ボリュームの状態を反映していません。従って、ボリュームステータスではボリュームが `error` 状態 ( 例えば、I/O を受け付けできない ) であることは判りません。

あるボリュームの整合性について心配しているわけではなく、そのボリュームに障害が発生した際にそのボリュームをすぐに利用できるようにしたい場合は、デフォルトの動作を上書きして、I/O を自動的に有効にするようにボリュームを設定することができます。AutoEnableIO ボリューム属性を有効にした場合、ボリュームステータスチェックは継続して成功します。また、ボリュームに潜在的な障害が

あると判断されたが、そのボリュームの I/O が自動的に有効になったことを伝えるイベントも表示されます。これにより、ボリュームの整合性を確認したり、後でボリュームを交換したりすることが可能になります。

次の表に、標準およびプロビジョニングされた IOPS ボリュームのステータスを示します。

全体的なボリュームステータス	I/O 有効ステータス	I/O パフォーマンスステータス (プロビジョニングされた IOPS ボリュームのみ)
ok	Enabled ( I/O Enabled または I/O Auto-Enabled )	Normal ( ボリュームパフォーマンスは想定どおり )
warning	Enabled ( I/O Enabled または I/O Auto-Enabled )	Degraded ( ボリュームのパフォーマンスが想定を下回っている )  Severely Degraded ( ボリュームのパフォーマンスが想定をかなり下回っている )
impaired	Enabled ( I/O Enabled または I/O Auto-Enabled )  Disabled ( ボリュームがオフラインで復旧の保留中、またはユーザーによる I/O の有効化待ち )	Stalled ( ボリュームのパフォーマンスは致命的な影響を受けている )  Not Available ( I/O が無効なため、I/O パフォーマンスを判定不能 )
insufficient-data	Enabled ( I/O Enabled または I/O Auto-Enabled )  Insufficient Data	Insufficient Data

ステータスチェックを表示または操作するには、Amazon EC2 コンソール、API、またはコマンドラインインターフェイスを使用します。

コンソールでステータスチェックを表示するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインの [Volumes] をクリックします。
3. [EBS Volumes] ページの [Volume Status] 列に、各ボリュームの動作状況が表示されます。
4. 各ボリュームのステータスを表示するには、ボリュームを選択して、[Status Checks] タブをクリックします。

Volumes: | vol-d882c69b

IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies may exist. [Troubleshoot](#) [Enable Volume IO](#)

Description Status Checks Monitoring Tags

Volume Status	impaired	Availability Zone	us-east-1d
IO Status	Disabled	IO Performance	Not Applicable
Since	December 23, 2013 7:06:41 PM UTC+2	Since	
Description	Awaiting Action: Enable IO	Description	This feature only applies to attach
Auto-Enabled IO	Disabled <a href="#">Edit</a>		

[Find out more](#) about working with volume status checks and events.  
If you need technical assistance with your volume, post your issue to the [Developer Forums](#) or visit our [Support Center](#).

- ステータスチェックが失敗したボリュームがある (ステータスが `impaired` である) 場合は、「[障害のあるボリュームの操作 \(p. 587\)](#)」を参照してください。

[Events] ペインを使用してインスタンスとボリュームのすべてのイベントを1つのペインに表示することもできます。詳細については、「[ボリュームイベントのモニタリング \(p. 585\)](#)」を参照してください。

コマンドラインを使用してボリュームステータスに関する情報を表示するには

Amazon EBS ボリュームのステータスを表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [describe-volume-status](#) (AWS CLI)
- [ec2-describe-volume-status](#) (Amazon EC2 CLI)
- [Get-EC2VolumeStatus](#) (AWS Tools for Windows PowerShell)

## ボリュームイベントのモニタリング

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

データが潜在的に不整合であるボリュームで I/O を自動的に有効にするには、`AutoEnableIO` ボリューム属性の設定を変更します。この属性の変更の詳細については、「[障害のあるボリュームの操作 \(p. 587\)](#)」を参照してください。

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

ボリュームステータスイベントには、次の説明のいずれかが含まれています。

### Awaiting Action: Enable IO

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

### IO Enabled

このボリュームに対する I/O 操作が明示的に有効にされました。

#### IO Auto-Enabled

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

#### Normal

プロビジョニングされた IOPS ボリュームのみ。ボリュームのパフォーマンスは想定どおりです。

#### Degraded

プロビジョニングされた IOPS ボリュームのみ。ボリュームのパフォーマンスは想定を下回っています。

#### Severely Degraded

プロビジョニングされた IOPS ボリュームのみ。ボリュームのパフォーマンスは想定をはるかに下回っています。

#### Stalled

プロビジョニングされた IOPS ボリュームのみ。ボリュームのパフォーマンスは致命的な影響を受けています。

Amazon EC2 コンソール、API、またはコマンドラインインターフェイスを使用して、ボリュームのイベントを表示できます。

コンソールでボリュームのイベントを表示するには

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

Event: vol-3682c675

⚠ IO operations have been disabled since 30 days, 15 hours and 22 minutes ago. Data inconsistencies may exist. [Enable Volume IO](#)

Availability Zone	us-east-1d
Event Type	potential-data-inconsistency
Event Status	Awaiting Action: Enable IO
IO status	IO Disabled
Attached to	<a href="#">i-93aae4ea</a>
Start Time	December 23, 2013 7:09:20 PM UTC+2
End time	

[Find out more about monitoring volume events.](#)

I/O が無効になっているボリュームがある場合は、「[障害のあるボリュームの操作 \(p. 587\)](#)」を参照してください。I/O パフォーマンスが通常の状態を下回っているボリュームがある場合、実行したアクションを原因とする一時的な状態である可能性があります ( ピーク使用時にボリュームのスナップショットを作成した、必要な I/O 帯域幅をサポートできないインスタンスでボリュームを実行した、ボリュームのデータに初めてアクセスした、など )。

コマンドラインを使用してボリュームのイベントを表示するには

Amazon EBS ボリュームのイベント情報を表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [describe-volume-status](#) ( AWS CLI )
- [ec2-describe-volume-status](#) ( Amazon EC2 CLI )
- [Get-EC2VolumeStatus](#) ( AWS Tools for Windows PowerShell )

## 障害のあるボリュームの操作

このセクションでは、ボリュームのデータに整合性がないなどの原因でボリュームに障害がある場合に使用可能なオプションについて説明します。

### Topics

- [オプション 1: インスタンスにアタッチされたボリュームで整合性チェックを実行する \(p. 587\)](#)
- [オプション 2: 別のインスタンスを使用してボリュームで整合性チェックを実行する \(p. 588\)](#)
- [オプション 3: 不要なボリュームを削除する \(p. 589\)](#)

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

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

アタッチされたボリュームで整合性チェックを実行するには

1. アプリケーションによるボリュームの使用を停止します。
2. ボリュームの I/O を有効にします。
  - a. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
  - b. ナビゲーションペインの [Volumes] をクリックします。
  - c. I/O 操作を有効にするボリュームを選択します。
  - d. 詳細ペインで、[Enable Volume I/O] をクリックします。

Volumes: | vol-d882c69b

The screenshot shows a warning banner at the top of the volume details page. The banner contains a yellow triangle icon, the text "IO operations have been disabled since 16 hours and 58 minutes ago ago. Data inconsistencies may exist.", and two buttons: "Troubleshoot" and "Enable Volume IO". Below the banner is a table with tabs for "Description", "Status Checks", "Monitoring", and "Tags". The "Description" tab is active, showing the following details:

Volume ID	vol-d882c69b	Alarm status	None
Capacity	100 GiB	Snapshot	-
Created	November 21, 2013 3:42:01 PM UTC+2	Availability Zone	us-east-1d
State	available	Attachment information	
Volume type	io1	IOPS	500
Product codes	-		

- e. [Enable Volume IO] で、[Yes, Enable] をクリックします。
3. ボリュームのデータを確認します。
    - a. `fsck` ( Linux ) または `chkdsk` ( Windows ) コマンドを実行します。

- b. (オプション) 関連するエラーメッセージがないか、使用可能なアプリケーションログまたはシステムログを確認します。
- c. ボリュームの障害が 20 分以上続く場合は、サポートに連絡してください。[Troubleshoot] をクリックし、[Troubleshoot Status Checks] ダイアログボックスの [Contact Support] をクリックして、サポートケースを送信します。

コマンドラインインターフェイスを使用してボリュームの I/O を有効にする方法については、「Amazon Elastic Compute Cloud コマンドラインリファレンス」の「[ec2-enable-volume-io](#)」を参照してください。API を使用してボリュームの I/O を有効にする方法については、「Amazon Elastic Compute Cloud API Reference」の「[EnableVolumeIO](#)」を参照してください。

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

実動環境外部のボリュームをチェックするには、次の手順に従います。



### Important

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

分離されたボリュームで整合性チェックを実行するには

1. アプリケーションによるボリュームの使用を停止します。
2. ボリュームをインスタンスからデタッチします。
  - a. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
  - b. ナビゲーションペインの [Volumes] をクリックします。
  - c. デタッチするボリュームを選択します。
  - d. [Actions]、[Force Detach Volume] の順にクリックします。確認のためのメッセージが表示されます。
3. ボリュームの I/O を有効にします。
  - a. ナビゲーションペインの [Volumes] をクリックします。
  - b. 前の手順でデタッチしたボリュームを選択します。
  - c. 詳細ペインで、[Enable Volume I/O] をクリックします。

Volumes: | vol-d882c69b

IO operations have been disabled since 16 hours and 58 minutes ago. Data inconsistencies may exist. Troubleshoot Enable Volume IO

Description	Status Checks	Monitoring	Tags
Volume ID	vol-d882c69b	Alarm status	None
Capacity	100 GiB	Snapshot	-
Created	November 21, 2013 3:42:01 PM UTC+2	Availability Zone	us-east-1d
State	available	Attachment information	
Volume type	io1	IOPS	500
Product codes	-		

- d. [Enable Volume IO] ダイアログボックスで、[Yes, Enable] をクリックします。

4. ボリュームを別のインスタンスにアタッチします。詳細については、「[インスタンスの起動 \(p. 324\)](#)」および「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#)」を参照してください。
5. ボリュームのデータを確認します。
  - a. fsck ( Linux ) または chkdsk ( Windows ) コマンドを実行します。
  - b. ( オプション ) 関連するエラーメッセージがないか、使用可能なアプリケーションログまたはシステムログを確認します。
  - c. ボリュームの障害が 20 分以上続く場合は、サポートに連絡してください。[Troubleshoot] をクリックし、トラブルシューティングのダイアログボックスで [Contact Support] をクリックして、サポートケースを送信します。

コマンドラインインターフェイスを使用してボリュームの I/O を有効にする方法については、「[Amazon Elastic Compute Cloud コマンドラインリファレンス](#)」の「[ec2-enable-volume-io](#)」を参照してください。API を使用してボリュームの I/O を有効にする方法については、「[Amazon Elastic Compute Cloud API Reference](#)」の「[EnableVolumeIO](#)」を参照してください。

### オプション 3: 不要なボリュームを削除する

環境からボリュームを削除するには、単にそれを削除します。ボリュームの削除の詳細については、「[Amazon EBS ボリュームの削除 \(p. 592\)](#)」を参照してください。

ボリュームのデータをバックアップするスナップショットを最近作成した場合、そのスナップショットから新しいボリュームを作成できます。スナップショットからのボリュームの作成の詳細については、「[スナップショットからの Amazon EBS ボリュームの復元 \(p. 571\)](#)」を参照してください。

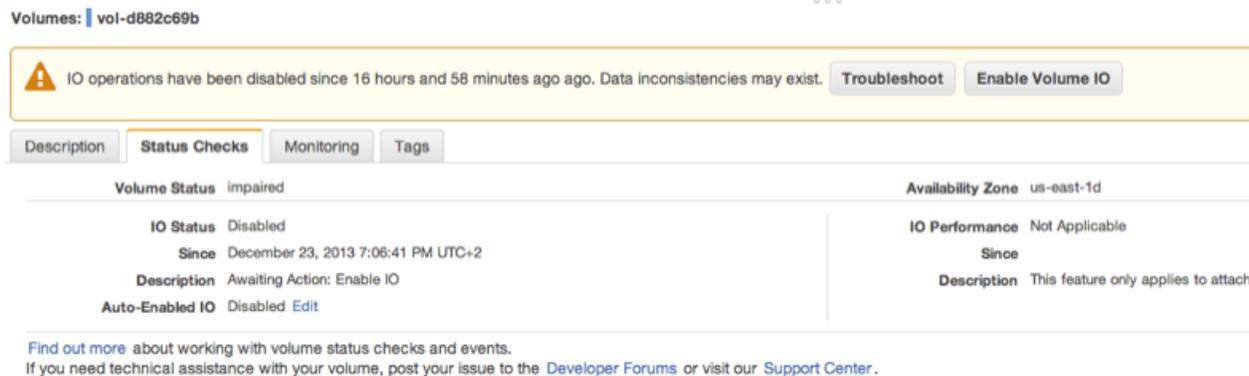
### AutoEnableIO ボリューム属性の操作

ボリュームのデータが潜在的に不整合であると Amazon EBS によって判断された場合、デフォルトでは、アタッチされているすべての EC2 インスタンスからそのボリュームへの I/O が無効になります。これにより、ボリュームステータスチェックが失敗し、障害の原因を示すボリュームステータスイベントが作成されます。特定のボリュームの整合性が問題でない場合、ボリュームが `impaired` であるときにそれを即時に使用可能にするには、I/O を自動的に有効にするようにボリュームを構成して、デフォルト動作をオーバーライドします。AutoEnableIO ボリューム属性を有効にすると、ボリュームとインスタンス間の I/O が再び自動的に有効になり、ボリュームのステータスチェックに合格します。また、ボリュームが潜在的に不整合な状態であること、ただしそのボリュームの I/O が自動的に有効になったことを伝えるイベントも表示されます。このイベントが発生した場合は、ボリュームの整合性をチェックし、必要に応じて置き換えます。詳細については、「[ボリュームイベントのモニタリング \(p. 585\)](#)」を参照してください。

このセクションでは、Amazon EC2 コンソール、コマンドラインインターフェイス、または API を使用して、ボリュームの AutoEnableIO 属性を表示および変更する方法について説明します。

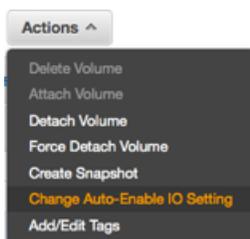
コンソールでボリュームの AutoEnableIO 属性を表示するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインの [Volumes] をクリックします。
3. ボリュームを選択します。
4. 下部のペインで、[Status Checks] タブをクリックします。
5. [Status Checks] タブで、[Auto-Enable IO] にボリュームの現在の設定が `Enabled` または `Disabled` と表示されます。

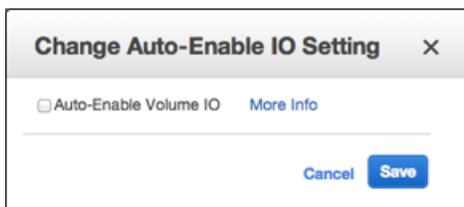


ボリュームの AutoEnableIO 属性を変更するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインの [Volumes] をクリックします。
3. ボリュームを選択します。
4. [Volumes] ページの一番上で、[Actions] をクリックします。
5. [Change Auto-Enable IO Setting] をクリックします。



6. [Change Auto-Enable IO Setting] ダイアログボックスで [Auto-Enable Volume IO] オプションを選択して、障害のあるボリュームの I/O を自動的に有効にします。この機能を無効にするには、オプションをオフにします。



7. [Save] をクリックします。

または、前の手順のステップ 4~6 を実行する代わりに、[Status Checks] タブに移動し、[Edit] をクリックします。

コマンドラインを使ってボリュームの AutoEnableIO 属性を表示または変更するには

Amazon EBS ボリュームの AutoEnableIO 属性を表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- [describe-volume-attribute](#) ( AWS CLI )
- [ec2-describe-volume-attribute](#) ( Amazon EC2 CLI )
- [Get-EC2VolumeAttribute](#) ( AWS Tools for Windows PowerShell )

ボリュームの `AutoEnableIO` 属性を変更するには、次のコマンドのいずれかを使用できます。

- [modify-volume-attribute](#) ( AWS CLI )
- [ec2-modify-volume-attribute](#) ( Amazon EC2 CLI )
- [Edit-EC2VolumeAttribute](#) ( AWS Tools for Windows PowerShell )

## インスタンスからの Amazon EBS ボリュームのデタッチ

### Abstract

インスタンスから Amazon EBS ボリュームのアタッチを解除するには、明示的に解除するか、インスタンスを終了します。

インスタンスから Amazon EBS ボリュームをデタッチするには、明示的にデタッチするか、インスタンスを終了します。ただし、ボリュームはデタッチする前に、インスタンス内部でアンマウントする必要があります。アンマウントせずにデタッチすると、ボリュームがビジー状態のままになり、そのファイルシステムまたは格納されているデータが破損する可能性があります。デタッチした ( アンマウントせずに ) ボリュームを再アタッチすることはできませんが、同じマウントポイントを取得できないことがあり、デタッチされたときにボリュームへの書き込みが処理中だった場合、ボリューム上のデータが同期しない可能性があります。



#### Note

インスタンスのルートデバイスである Amazon EBS ボリュームは、インスタンスを停止状態にししないとデタッチできません。

AWS Marketplace 製品コード付きのインスタンスからルートボリュームをデタッチすると、そのボリュームの AWS Marketplace 製品コードとそのインスタンスとの関連付けが失われます。



#### Important

ボリュームのアタッチを解除した後でも、ストレージ量が無料利用枠の上限を超えている限り、料金が発生します。不要な料金の発生を防ぐために、ボリュームを削除する必要があります。詳細については、[Amazon EBS ボリュームの削除 \(p. 592\)](#) を参照してください。

この例では、ボリュームをアンマウントし、インスタンスから明示的にデタッチします。この操作は、インスタンスを終了するときや、ボリュームを別のインスタンスにアタッチするときに便利です。ボリュームがインスタンスにアタッチされていないことを確認する方法については、「[ボリューム情報を表示する \(p. 580\)](#)」を参照してください。

コンソールを使用して、Amazon EBS ボリュームをデタッチするには

1. まず、ボリュームをアンマウントします。

Linux/Unix の場合は、次のコマンドを使用して、`/dev/sdh` デバイスのマウントを解除します。

```
# umount -d /dev/sdh
```

Windows の場合、Disk Managementを開き、アンマウントするボリュームを右クリックして、[Change Drive Letter and Path] を選択します。次に、削除するマウントポイントを選択し、[Remove] をクリックします。

2. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
3. ナビゲーションペインで [Volumes] をクリックします。

コンソールに現在のボリュームのリストが表示されます。

4. ボリュームを選択し、Detach Volume をクリックします。

確認のダイアログボックスが表示されます。

5. Yes, Detach をクリックします。

ボリュームがインスタンスからデタッチされます。



#### Caution

ボリュームの状態がデタッチ状態のまま変わらない場合は、[Force Detach] をクリックして、強制的にアタッチ解除することもできます。デタッチを強制的に行うと、データの損失やファイルシステムの破損につながる可能性があります。障害が発生したインスタンスからボリュームをアタッチ解除するための最後の手段として、またはボリュームを削除するためにデタッチする場合のみ、このオプションを使用してください。インスタンスは、ファイルシステムキャッシュやファイルシステムメタデータをフラッシュする機会を失います。このオプションを使用する場合は、ファイルシステムのチェックと修復の手順を手動で実行する必要があります。

数分間で複数回、ボリュームのデタッチを強制実行しても、デタッチ状態のままの場合、[Amazon EC2 forum](#) にヘルプの要請を投稿できます。迅速に解決できるようにするため、ボリューム ID と、これまでに実行した手順を記述してください。

コマンドラインを使用してインスタンスから Amazon EBS ボリュームをデタッチするには

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

- `detach-volume` ( AWS CLI )
- `ec2-detach-volume` ( Amazon EC2 CLI )
- `Dismount-EC2Volume` ( AWS Tools for Windows PowerShell )

## Amazon EBS ボリュームの削除

### Abstract

ボリュームが不要になった場合は、Amazon EBS ボリュームを削除します。

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

このセクションでは、ボリュームを削除する方法を説明します。

コンソールを使用してボリュームを削除するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインで [Volumes] をクリックします。

コンソールに現在のボリュームのリストが表示されます。

3. ボリュームを選択し、[Delete Volume] をクリックします。

確認のダイアログボックスが表示されます。

4. [Yes, Delete] をクリックします。

ボリュームが削除されます。

コマンドラインを使用して、Amazon EBS ボリュームを削除するには

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

- `delete-volume` ( AWS CLI )
- `ec2-delete-volume` ( Amazon EC2 CLI )
- `Remove-EC2Volume` ( AWS Tools for Windows PowerShell )

## ボリュームのストレージ領域を拡張する

### Abstract

ボリュームのデータを失うことなく、既存のボリュームのストレージスペースを拡大します。

ボリューム上のデータを失わずに、既存のボリュームのストレージ領域を拡大する必要がある場合があります。このトピックでは、Amazon EBS ボリュームのストレージスペースを拡張する方法について説明します。拡大したボリュームにデータを移行してから、ボリュームのファイルシステムを拡張すると、新たに使用可能になったスペースが認識されます。新しいボリュームが正常に動作していることを確認したら、古いボリュームを削除できます。

### Topics

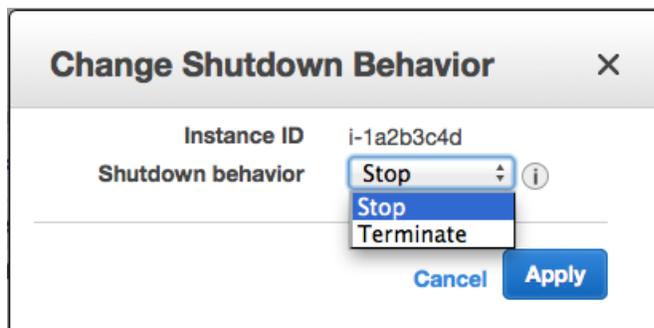
- [拡大したボリュームにデータを移行する \(p. 593\)](#)
- [Linux ファイルシステムを拡張する \(p. 595\)](#)
- [Windows ファイルシステムを拡張する \(p. 597\)](#)
- [古いボリュームを削除する \(p. 600\)](#)

## 拡大したボリュームにデータを移行する

拡大した Amazon EBS ボリュームにデータを移行するには

1. AWS マネジメントコンソールにサインインして Amazon EC2 コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. インスタンスの [Shutdown Behavior] 値が [Terminate] ではなく、[Stop] に設定されていることを確認します。既に [Stop] に設定されている場合は、ステップ 3 に進みます。
  - a. ナビゲーションペインで、[Instances] をクリックし、チェックするインスタンスを右クリックして、[Change Shutdown Behavior] を選択します。
  - b. [Shutdown behavior] が [Terminate] に設定されている場合は、リストから [Stop] を選択し、[Apply] をクリックします。

[Shutdown Behavior] が [Stop] に設定されている場合は、[Cancel] をクリックします。



3. インスタンスを停止します。インスタンスの停止方法の詳細については、[インスタンスの停止と起動 \(p. 347\)](#) を参照してください。
4. 拡張するボリュームのスナップショットを作成します。
  - a. ナビゲーションペインで、[Volumes] をクリックし、拡張するボリュームを右クリックして、[Create Snapshot] を選択します。
  - b. スナップショットの [Name] と [Description] を入力し、[Yes, Create] をクリックします。

5. スナップショットから新しいボリュームを作成します。
  - a. ナビゲーションペインで、[Snapshots] をクリックします。
  - b. 作成したスナップショットのステータスが [completed] の場合は、スナップショットを選択し、[Create Volume] をクリックします。



#### Note

スナップショットが完了するまで、数分かかる場合があります。

- c. [Create Volume] ダイアログボックスでご希望のボリュームタイプを選択し、ボリュームの新しいサイズを入力しアベイラビリティゾーンをインスタンスと同じアベイラビリティゾーンに設定し、[Yes, Create] をクリックします。
6. 古いボリュームをデタッチします。
    - a. ナビゲーションペインで [Volumes] をクリックし、ボリュームのリストから古いボリュームを選択して、[Attachment information] の device name の値を書き留めます。[Attachment Information] の値は次の形式になります。

```
instance information:device name
```

- b. 古いボリュームを右クリックし、[Detach Volume] を選択します。
  - c. [Detach Volume] ダイアログボックスで、[Yes, Force] をクリックします。ボリュームのデタッチには数分かかる場合があります。
7. 新たに拡張したボリュームをアタッチします。
    - a. ナビゲーションペインの [Volumes] をクリックし、ボリュームのリストから新しいボリュームを選択して、新しいボリュームを右クリックし、[Attach Volume] を選択します。

- b. [Instance] フィールドにインスタンス名または ID を入力して、インスタンスを選択し、[Step 6.a \(p. 594\)](#) で取得した同じデバイス名を入力して、[Yes, Attach] をクリックします。
8. インスタンスを再起動します。
    - a. ナビゲーションペインで、[Instances] をクリックし、インスタンスを右クリックして、[Start] をクリックします。
    - b. [Start Instances] ダイアログボックスで、[Yes, Start] を選択します。インスタンスの起動に失敗し、拡張したポリユームがルートポリユームの場合は、元のポリユームと同じデバイス名を使用して、拡張したポリユームをアタッチしていることを確認してください (ルートポリユームは `/dev/sda1` としてアタッチする必要があります)。



#### Important

停止時にパブリック IP アドレスを保持するインスタンスは、Elastic IP アドレスを持つ EC2-VPC インスタンスに限られます。インスタンスが EC2-Classic で実行されている場合は、インスタンスを停止すると EIP アドレスの関連付けが解除されるため、インスタンスを再起動してから EIP を再度関連付ける必要があります。詳細については、[Elastic IP アドレス \(EIP\) \(p. 540\)](#) を参照してください。インスタンスが EIP を使用していない場合、インスタンスに接続するには、Amazon EC2 コンソールのインスタンスページからインスタンスの新しいパブリック DNS 名を取得する必要があります。

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

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

[Size] 列の値が、新たに拡張したポリユームのサイズを示していない場合は、インスタンスが新しいスペースを使用できるように、デバイスのファイルシステムを拡張する必要があります。この手順は、インスタンスを実行しているオペレーティングシステムによって異なります。さまざまなオペレーティングシステムでこの手順を実行する方法については、次のトピックを参照してください。

- [Linux ファイルシステムを拡張する \(p. 595\)](#)
- [Windows ファイルシステムを拡張する \(p. 597\)](#)

## Linux ファイルシステムを拡張する

Linux では、ファイルシステム固有のコマンドを使用して、ファイルシステムのサイズを、新しいポリユームの拡大したサイズに変更します。このコマンドは、拡張するポリユームがルートポリユームである場合にも機能します。ext2、ext3、および ext4 ファイルシステムの場合、このコマンドは `resize2fs` です。XFS ファイルシステムの場合、このコマンドは `xfs_growfs` です。その他のファイルシステムの場合、ファイルシステムの拡張については、これらのファイルシステムに関する特定のドキュメントを参照してください。

使用しているファイルシステムが不明の場合は、`file -s` コマンドを使用して、デバイスのファイルシステムデータをリストできます。次の例は、Linux `ext4` ファイルシステムと SGI XFS ファイルシステムを示しています。

```
[ec2-user ~]$ sudo file -s /dev/xvd*
/dev/xvda1: Linux rev 1.0 ext4 filesystem data ...
/dev/xvdf:  SGI XFS filesystem data ...
```



#### Note

拡張するボリュームがパーティション分割されている場合は、ファイルシステムのサイズを変更する前に、パーティションのサイズを拡大する必要があります。詳細については、[Linux パーティションを拡張する \(p. 600\)](#) を参照してください。

ボリュームのパーティションのサイズ変更が必要かどうかを確認するには

- インスタンスにアタッチされたブロックデバイスを一覧表示するには、`lsblk` コマンドを使用します。以下の例は、`/dev/xvda`、`/dev/xvdb`、`/dev/xvdf` という 3 つのボリュームを示しています。

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda        202:0    0 30G  0 disk
  xvda1     202:1    0 30G  0 part /
xvdb        202:16   0 30G  0 disk /mnt
xvdf        202:80   0 35G  0 disk
  xvdf1    202:81   0  8G  0 part
```

ルートボリュームである `/dev/xvda1` は、`/dev/xvda` のパーティションです。どちらもサイズが 30 GB であることに注意してください。この場合、パーティションはデバイスの領域をすべて占有するので、サイズを変更する必要はありません。

ボリューム `/dev/xvdb` は、パーティション分割されていないので、サイズを変更する必要はありません。

ただし、`/dev/xvdf1` は、35 GB のデバイスに含まれる 8 GB のパーティションであり、このボリュームには他にパーティションがありません。この場合、ボリュームの残りの領域を使用するには、パーティションをサイズ変更する必要があります。詳細については、「[Linux パーティションを拡張する \(p. 600\)](#)」を参照してください。パーティションのサイズを変更した後、次の手順に従って、パーティションのすべての領域を占有するようにファイルシステムを拡張できます。

Linux ファイルシステムを拡張するには

1. SSH クライアントを使用して Linux インスタンスにログインします。Linux インスタンスへの接続の詳細については、「[SSH を使用した Linux/Unix インスタンスへの接続 \(p. 334\)](#)」を参照してください。
2. `df -h` コマンドを使用して、既存のファイルシステムディスク領域の使用状況をレポートします。この例では、`/dev/xvda1` デバイスは 70 GB に既に拡張されていますが、`ext4` ファイルシステムは単に元の 8 GB のサイズとして表示されています。また、`/dev/xvdf` デバイスは 100 GB に拡張されていますが、XFS ファイルシステムは単に元の 1 GB のサイズとして表示されています。

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

```
tmpfs          1.9G    0 1.9G   0% /dev/shm
/dev/xvdf      1014M   33M  982M   4% /mnt
```

3. ファイルシステム固有のコマンドを使用して、ファイルシステムのサイズをボリュームの新しいサイズに変更します。Linux ext2、ext3、または ext4 ファイルシステムの場合は、次のコマンドを使用して、拡張するデバイスの名前に置き換えます。

```
[ec2-user ~]$ sudo resize2fs /dev/xvda1
resize2fs 1.42.3 (14-May-2012)
Filesystem at /dev/xvda1 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 5
Performing an on-line resize of /dev/xvda1 to 18350080 (4k) blocks.
The filesystem on /dev/xvda1 is now 18350080 blocks long.
```

XFS ファイルシステムの場合は、次のコマンドを使用して、ファイルシステムのマウントポイントに置き換える必要があります (XFS ファイルシステムのサイズを変更するには、マウントする必要があります)。

```
[ec2-user ~]$ sudo xfs_growfs -d /mnt
meta-data=/dev/xvdf          isize=256    agcount=4, agsize=65536 blks
=                               sectsz=512   attr=2
data      =                   bsize=4096  blocks=262144, imaxpct=25
=                               sunit=0     swidth=0 blks
naming    =version 2          bsize=4096  ascii-ci=0
log       =internal          bsize=4096  blocks=2560, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096  blocks=0, rtextents=0
data blocks changed from 262144 to 26214400
```



#### Note

失敗した `xfsctl` を `xfsctl failed: Cannot allocate memory` エラーを受け取った場合、インスタンスの Linux カーネルの更新が必要なことがあります。詳細については、特定のオペレーティングシステムのドキュメントを参照してください。

4. `df -h` コマンドを使用して、既存のファイルシステムディスク領域の使用状況をレポートします。この時点で、ext4 ファイルシステムについては 70 GB、XFS ファイルシステムについては 100 GB の完全な容量が示されます。

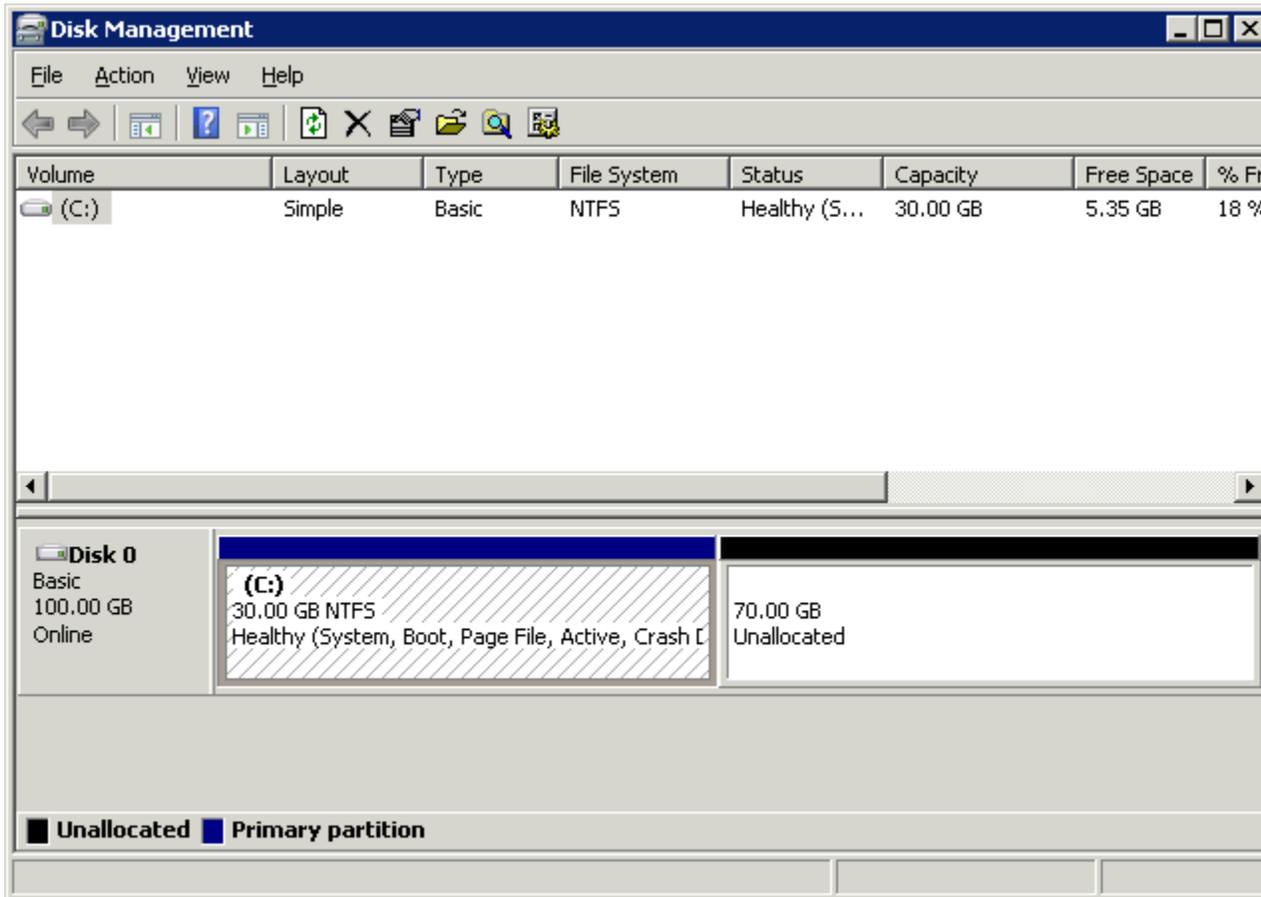
```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      69G   951M   68G   2% /
tmpfs           1.9G    0 1.9G   0% /dev/shm
/dev/xvdf       100G   45M  100G   1% /mnt
```

## Windows ファイルシステムを拡張する

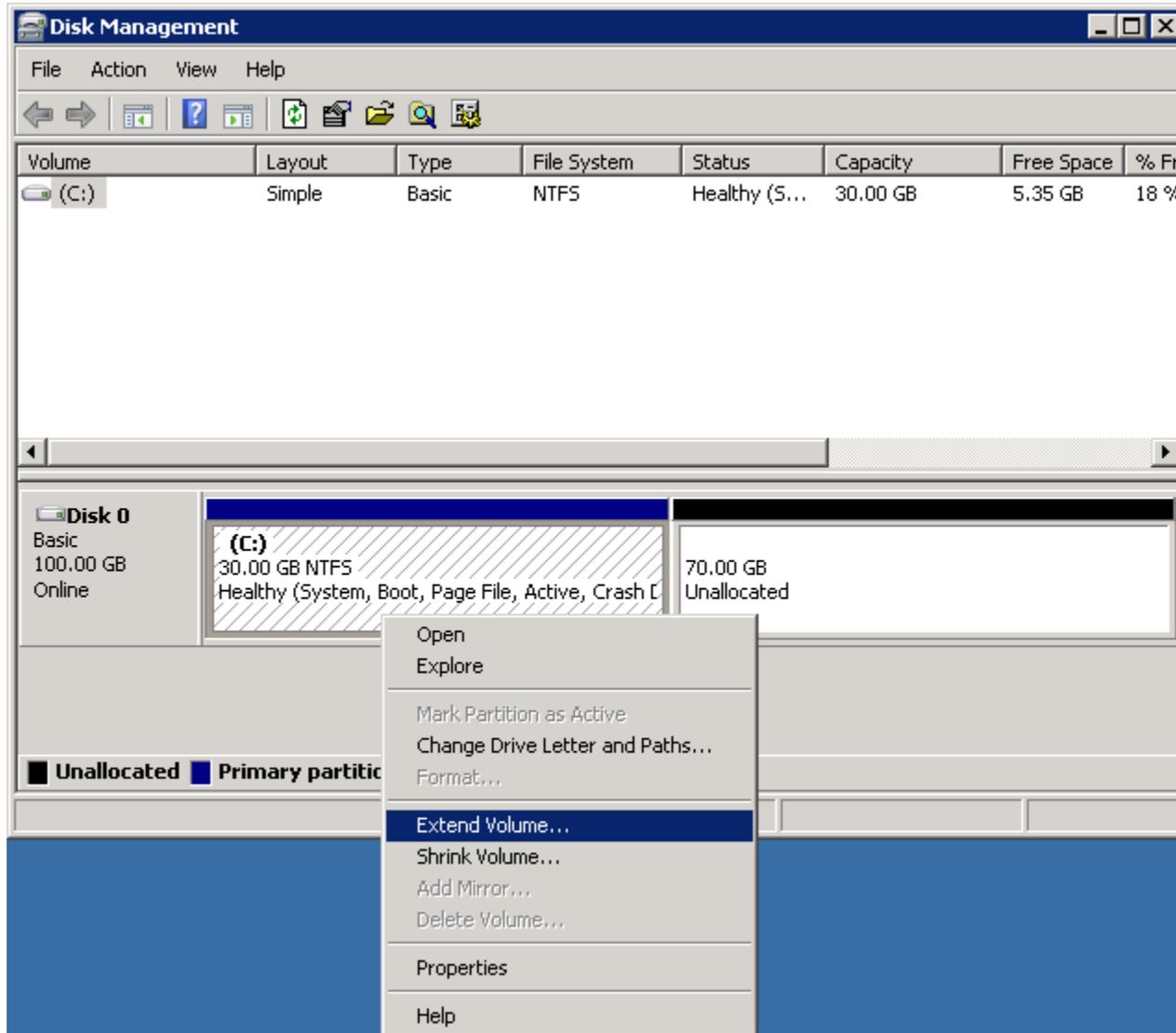
Windows では、ディスクの管理ユーティリティを使用して、ディスクサイズをボリュームの新しいサイズに拡大できます。

Windows ファイルシステムを拡張するには

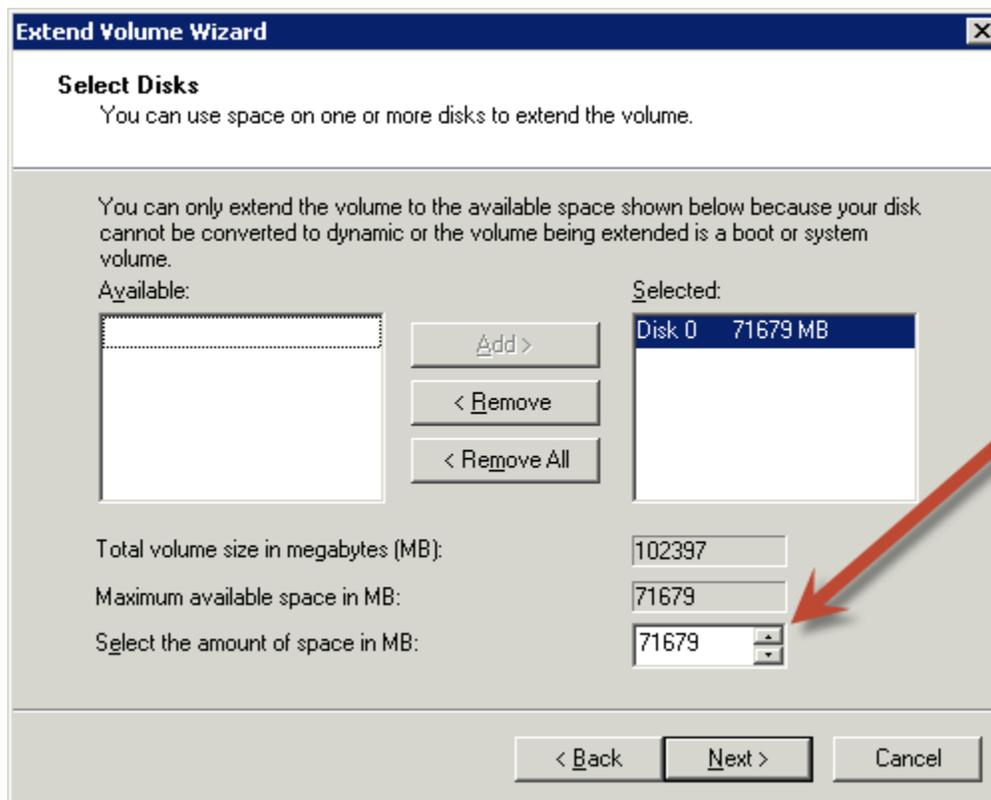
1. リモートデスクトップを使用して Windows インスタンスにログインします。
2.
  - Windows Server 2012: [スタート] 画面に移動します。
  - Windows Server 2008: タスクバーの [Start] をクリックし、[Run] をクリックします。
3. diskmgmt.msc と入力して、Enter を押します。Disk Managementユーティリティが表示されます。



4. 拡張したドライブを右クリックし、[Extend Volume] を選択します。



5. ボリュームの拡張ウィザードで [Next] をクリックし、[Select the amount of space in MB] フィールドに、ボリュームを拡大するメガバイト数を設定します。通常、これは使用可能な最大領域に設定します。ウィザードを終了します。



## 古いボリュームを削除する

インスタンスで新しいボリュームをアタッチし、拡張したら、必要なくなった古いボリュームを削除できます。

古いボリュームを削除するには

1. Amazon EC2 コンソール内で、ナビゲーションペインの [Volumes] をクリックします。
2. 古いボリュームを右クリックし、[Delete Volume] を選択します。
3. [Delete Volume] ダイアログボックスで、[Yes, Delete] をクリックします。

## Linux パーティションを拡張する

### Abstract

ボリュームのデータを失うことなく、既存のボリュームの Linux パーティションを拡張します。

一部の Amazon EC2 ルートボリュームやスナップショットから復元されたボリュームには、ファイルシステムおよびデータを実際に保持するパーティションが含まれています。ボリュームをコンテナと考えれば、パーティションはボリューム内のもう1つのコンテナであり、データはパーティション上に存在します。ボリュームのサイズを拡大しても、パーティションは拡大されません。拡大したボリュームを活用するには、パーティションを新しいサイズに拡張する必要があります。



### Note

スナップショットから復元したすべてのボリュームがパーティション分割されるわけではないので、この手順は、お使いのボリュームに適用されないことがあります。ボリュームのファイ

ルシステムのサイズを使用できる最大の容量に変更する必要があります。サイズを変更する必要があるパーティションがポリリューム内にあるかどうか不明な場合は、「[ポリリュームのパーティションのサイズ変更が必要かどうかを確認するには \(p. 596\)](#)」で詳細を参照してください。

拡張するパーティションがルートパーティションではない場合、単にこのパーティションをアンマウントし、インスタンス自体からパーティションのサイズを変更することができます。サイズを変更する必要があるパーティションがルートパーティションである場合、実行中のインスタンスのルートパーティションをアンマウントすることはできないため、プロセスはより複雑になります。セカンダリインスタンスと呼ばれる別のインスタンスに対して、以下の手順を実行する必要があります。



#### Important

以下の手順は Amazon Linux に関する説明であり、Amazon Linux でテスト済みです。その他のツールセットやツールバージョンによるディストリビューションは、異なる動作をする場合があります。

#### 拡張するために Linux ルートパーティションを準備するには

インスタンスのルートパーティションを拡張するために、次のいくつかのステップを実行する必要があります。拡張する必要があるパーティションがルートパーティションではない場合、この手順を行う必要はありません。

1. プライマリインスタンスを実行している場合は停止します。実行中のインスタンスでは、これ以降の手順を実行できません。詳細については、[インスタンスの停止と起動 \(p. 346\)](#)を参照してください。
2. ポリリュームのスナップショットを取得します。以下の手順では、データが容易に破損または消失する可能性があります。新しいスナップショットがある場合、失敗してもいつでもやり直すことができ、データは安全に保たれます。詳細については、[Amazon EBS スナップショットの作成 \(p. 606\)](#)を参照してください。
3. ポリリュームがアタッチされたデバイス名を記録します。この値は、インスタンスの詳細ペインの [Root device] フィールドで確認できます。通常、この値は `/dev/sda1` または `/dev/xvda` です。
4. ポリリュームをプライマリインスタンスからデタッチします。詳細については、[インスタンスからの Amazon EBS ポリリュームのデタッチ \(p. 591\)](#)を参照してください。
5. ポリリュームを、同じアベイラビリティゾーン内の別の (セカンダリ) インスタンスにアタッチします。詳細については、「[インスタンスへの Amazon EBS ポリリュームのアタッチ \(p. 573\)](#)」を参照してください。セカンダリインスタンスを既に持っている場合、それを起動する必要があります。詳細については、[インスタンスの起動 \(p. 325\)](#)を参照してください。
6. SSH を使用してセカンダリインスタンスにログインします。詳細については、[インスタンスへの接続 \(p. 334\)](#)を参照してください。次の手順に進みます。

#### Linux パーティションを拡張するには

拡張する必要があるパーティションがルートパーティションの場合、まずは前の手順を必ず行ってください。

1. 拡張するパーティションを含むデバイスを特定します。lsblk コマンドを使用して、インスタンスにアタッチされたすべてのデバイスおよびパーティションをリストします。

```
[ec2-user ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvdf         202:80   0 100G  0 disk
   xvdf1     202:81   0   8G  0 part /mnt
xvda1       202:1    0  30G  0 disk /
```

この例では、`xvdf` デバイスに 100 GB の使用可能なストレージがあり、8 GB のパーティションが含まれています。

- パーティションがマウントされている場合はアンマウントします。lsblk コマンドから MOUNTPOINT の値を指定して、`umount` コマンドを実行します。この例では、パーティションの MOUNTPOINT 値は `/mnt` です。

```
[ec2-user ~]$ sudo umount /mnt
```

- ボリュームのスナップショットを取得します ( 前の手順で取得しなかった場合 )。以下の手順では、データが容易に破損または消失する可能性があります。新しいスナップショットがある場合、失敗してもいつでもやり直すことができ、データは安全に保たれます。詳細については、[Amazon EBS スナップショットの作成 \(p. 606\)](#) を参照してください。
- デバイス ( デバイス上のパーティションではなく ) に対して `parted` コマンドを実行します。 `/dev/プレフィックス` を、`lsblk` が出力する名前に必ず追加してください。

```
[ec2-user ~]$ sudo parted /dev/xvdf
GNU Parted 2.1
Using /dev/xvdf
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

- `parted` 測定単位をセクターに変更します。

```
(parted) unit s
```

- `print` コマンドを実行して、デバイス上のパーティションをリストします。特定のパーティションテーブルタイプでは、拡大したボリュームのサイズに合わせてパーティションテーブルを修正するように要求される場合があります。

```
(parted) print
```

- ( オプション ) 以下のメッセージを受け取った場合は、「Fix」を入力して、バックアップ GPT テーブルをディスクの最後に移動します。

```
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another operating system believes the disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
Fix/Ignore/Cancel? Fix
```

- ( オプション ) 以下のメッセージを受け取った場合は、「Fix」を再度入力して、ドライブのスペースを使用できるようにします。

```
Warning: Not all of the space available to /dev/xvdf appears to be used, you can fix the GPT to use all of the space (an extra 46137344 blocks) or continue with the current setting?
Fix/Ignore? Fix
```

- ディスクの合計サイズ、パーティションテーブルタイプ、パーティションの番号、パーティションの開始点、およびパーティションの名前の出力を調べます。 `gpt` パーティションテーブルの場合は、パーティションの名前に留意します。 `msdos` パーティションテーブルの場合は、`Type` フィー

ルド ( `primary` または `extended` ) に留意します。これらの値は、この後のステップで使用します。

以下に、`gpt` パーティションテーブルの例を示します。

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt

Number  Start   End       Size      File system  Name              Flags
128     2048s   4095s     2048s                    BIOS Boot Partition
bios_grub
  1      4096s  16777182s 16773087s ext4          Linux
```

以下に、`msdos` パーティションテーブルの例を示します。

```
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdg: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End       Size      Type        File system  Flags
  1      2048s  35649535s 35647488s primary    ext3
```

8. 前のステップでの番号 ( 1 ) を使用して、パーティションのパーティションエントリを削除します。

```
(parted) rm 1
```

9. ボリュームの最後まで拡張された新しいパーティションを作成します。

( `gpt` パーティションテーブルの例では ) 上記のパーティション 1 の開始点と名前に留意します。`gpt` の例では、開始点は `4096s`、名前は `Linux` です。パーティション 1 の開始点、名前、および ( 使用可能なすべてのスペースを使用するために ) `100%` を指定して、`mkpart` コマンドを実行します。

```
(parted) mkpart Linux 4096s 100%
```

( `msdos` パーティションテーブルの例では ) 上記のパーティション 1 の開始点とパーティションタイプに留意します。`msdos` の例では、開始点は `2048s`、パーティションタイプは `primary` です。プライマリパーティションタイプ、パーティション 1 の開始点、および ( 使用可能なすべてのスペースを使用するために ) `100%` を指定して、`mkpart` コマンドを実行します。

```
(parted) mkpart primary 2048s 100%
```

10. `print` コマンドを再度実行して、パーティションを確認します。

( `gpt` パーティションテーブルの例では )

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 209715200s
Sector size (logical/physical): 512B/512B
```

```
Partition Table: gpt

Number  Start   End       Size      File system  Name
Flags
128     2048s   4095s     2048s                    BIOS Boot Partition
bios_grub
1       4096s   209713151s 209709056s ext4          Linux
```

( msdos パーティションテーブルの例では )

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdg: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End       Size      Type        File system  Flags
1       2048s   104857599s 104855552s primary    ext3
```

- quit コマンドを実行して、parted を終了します。

```
(parted) quit
```



#### Note

パーティションを移動し、パーティションを追加したので、parted は、`/etc/fstab` を更新する必要があることを警告する場合があります。この更新は、パーティション番号が変更された場合にのみ必要です。

- ファイルシステムを確認して、エラーがないことを確認します (これは、ファイルシステムを拡張する前に行う必要があります)。

```
[ec2-user ~]$ sudo e2fsck -f /dev/xvdf1
e2fsck 1.42.3 (14-May-2012)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/: 31568/524288 files (0.4% non-contiguous), 266685/2096635 blocks
```

- 次のステップは、拡張したパーティションが現在のインスタンスに属するかどうか、またはこのパーティションが別のインスタンスのルートパーティションであるかどうかによって異なります。
  - このパーティションが現在のインスタンスに属する場合は、[Step 2 \(p. 602\)](#) で特定した MOUNTPOINT でパーティションを再マウントします。

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

パーティションをマウントしたら、[Linux ファイルシステムを拡張する \(p. 595\)](#) の手順を行って、新たに使用可能になったスペースを使用するようにファイルシステムを拡張します。

- このボリュームが別のインスタンスのルートパーティションである場合は、次の手順を実行しません。

拡張したルートパーティションをその元のインスタンスに戻すには

1. 拡張したパーティションを、そのセカンダリインスタンスからデタッチします。詳細については、[インスタンスからの Amazon EBS ボリュームのデタッチ \(p. 591\)](#) を参照してください。
2. [preparation procedure \(p. 601\)](#) の [Step 3 \(p. 601\)](#) で特定したデバイス名を使用して、ボリュームをプライマリインスタンスに再アタッチします。詳細については、[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#) を参照してください。
3. プライマリインスタンスを起動します。詳細については、[インスタンスの停止と起動 \(p. 346\)](#) を参照してください。
4. ( オプション ) パーティションを拡張するためだけにセカンダリインスタンスを起動した場合は、インスタンスを終了して課金を停止できます。詳細については、[インスタンスの終了 \(p. 352\)](#) を参照してください。
5. プライマリインスタンスに接続し、[Linux ファイルシステムを拡張する \(p. 595\)](#) の手順を行って、新たに使用可能になったスペースを使用するようにファイルシステムを拡張します。

このファイルシステムの拡張が終了したら、インスタンスから、目的のパーティションサイズを持つ新しいインスタンスを起動するために使用できる AMI を作成できます。詳細については、[Amazon マシンイメージ \(AMI\) \(p. 55\)](#) を参照してください。

## Amazon EBS スナップショット

### Abstract

Amazon S3 に保存された、Amazon EBS ボリュームのポイントインタイム増分バックアップコピーとして、スナップショットを使用します。

Amazon EBS スナップショットとは、Amazon S3 に保存された Amazon EBS ボリュームのポイントインタイム増分バックアップコピーです。スナップショットは差分バックアップです。つまり、最後にスナップショットを作成した時点から、ボリューム上で変更のあるブロックだけが保存されます。スナップショットを削除すると、そのスナップショットに含まれないデータだけが削除されます。アクティブなスナップショットには、新しい Amazon EBS ボリュームから ( スナップショットの作成時から ) データを復元するために必要な情報がすべて含まれています。

### Topics

- [Amazon EBS スナップショットの作成 \(p. 606\)](#)
- [Amazon EBS スナップショットの削除 \(p. 607\)](#)
- [Amazon EBS スナップショットのコピー \(p. 608\)](#)
- [スナップショットに関する情報を表示する \(p. 609\)](#)
- [スナップショットの共有 \(p. 610\)](#)

新規 Amazon EBS ボリュームを作成するときに、既存のスナップショットに基づいてボリュームを作成できます。新規ボリュームは、スナップショットの作成に使われた元のボリュームの正確なコピーとして機能を開始します。既存の Amazon S3 スナップショットから作成された新規ボリュームは、バックグラウンドで時間をかけて読み込まれるため、直ちに使用を開始できます。まだ読み込まれていないデータに対してインスタンスからのアクセスがあった場合、ボリュームは要求されたデータを Amazon S3 から即座にダウンロードし、引き続きボリュームの残りのデータをバックグラウンドで読み込みます。スナップショット作成についての詳細は、[Amazon EBS スナップショットの作成 \(p. 606\)](#) を参照してください。

スナップショットを、特定のユーザーと共有することも、AWS コミュニティ全体で共有するために公開することもできます。スナップショットへのアクセス権限を持つユーザーは、スナップショットから独自の Amazon EBS ボリュームを作成することができますが、スナップショットは完全に元のまま保持されます。スナップショットを共有する方法については、[スナップショットの共有 \(p. 610\)](#) を参照してください。

Amazon EBS スナップショットは、スナップショットが作成されたリージョンに制限されます。Amazon EBS ボリュームのスナップショットを作成した後、そのスナップショットを使って、同じリージョンで新規ボリュームを作成できます。詳細については、[スナップショットからの Amazon EBS ボリュームの復元 \(p. 571\)](#) を参照してください。スナップショットは AWS リージョン間でコピーできるので、地理的な拡大、データセンターへの移行、災害復旧など、複数の AWS リージョンを活用することが簡単にできます。お客様は、アクセス権のあるスナップショットのうち、ステータスが「available」であるものはどれでもコピーできます。詳細については、[Amazon EBS スナップショットのコピー \(p. 608\)](#) を参照してください。

## Amazon EBS スナップショットの作成

### Abstract

ボリュームの Amazon EBS スナップショットを作成して、新規ボリュームやデータバックアップ用のベースラインとして使用できます。

Amazon EBS ボリュームにデータを書き込んだ後は、ボリュームのスナップショットを定期的に作成して、新しいボリュームまたはデータバックアップのためのベースラインとして使用できます。ボリュームのスナップショットを定期的に作成する場合は、差分だけが更新されます。つまり、最後にスナップショットを作成した時点から変更されたデバイス上のブロックだけが新しいスナップショットに保存されます。スナップショットの保存は差分ベースで行われるものの、最新のスナップショットさえあればボリュームを復元できるようにスナップショット削除プロセスは設計されています。

スナップショットの作成は非同期的に行われ、処理が完了するまでそのステータスは `pending` になります。

デフォルトでは、自分が所有するスナップショットからのみボリュームを起動できます。スナップショットについては、特定の AWS アカウントと共有するか、AWS コミュニティ全体で共有するかを選択することができます。詳細については、「[スナップショットの共有 \(p. 610\)](#)」を参照してください。

スナップショットが作成されるときに、ボリュームからの AWS Marketplace 製品コードがあれば、そのコードがスナップショットに反映されます。

使用中のアタッチ済みボリュームのスナップショットを取ることができます。ただし、スナップショットでは、スナップショットコマンドを実行した時点で Amazon EBS ボリュームに書き込まれているデータのみがキャプチャされます。これには、アプリケーションやオペレーティングシステムによってキャッシュされたデータは含まれない可能性があります。スナップショットを取る間、ボリュームへのすべてのファイルの書き込みを停止できれば、完全なスナップショットを取ることができます。ただし、ボリュームへのすべてのファイルの書き込みを停止できない場合は、一貫した完全なスナップショットを取ることができるように、インスタンス内からボリュームをアンマウントし、スナップショットコマンドを実行して、ボリュームを再マウントします。スナップショットのステータスが `available` の間は、ボリュームを再マウントして使用できます。

ルートデバイスとして機能する Amazon EBS ボリュームのスナップショットを作成するには、スナップショットを取る前にインスタンスを停止します。

Linux/Unix でボリュームをアンマウントするには、次のコマンドを使用します。

```
umount -d device_name
```

*device\_name* にはデバイス名 (例: `/dev/sdh`) を挿入します。

Windows でボリュームをアンマウントするには、ディスクの管理を開き、アンマウントするボリュームを右クリックして、[Change Drive Letter and Path] を選択します。次に、削除するマウントポイントを選択し、[Remove] をクリックします。

コンソールを使用してスナップショットを作成するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインで [Snapshots] をクリックします。

コンソールに現在のスナップショットのリストが表示されます。

3. Create Snapshot をクリックします。

[Create Snapshot] ダイアログボックスが表示されます。

4. スナップショットを作成するボリュームを選択し、Create をクリックします。

Amazon EC2 によってスナップショットの作成が開始されます。

コマンドラインを使用してスナップショットを作成するには

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

- `create-snapshot` ( AWS CLI )
- `ec2-create-snapshot` ( Amazon EC2 CLI )
- `New-EC2Snapshot` ( AWS Tools for Windows PowerShell )

## Amazon EBS スナップショットの削除

### Abstract

AWS マネジメントコンソール、CLI、または API を使って、Amazon EBS スナップショットを削除します。

このセクションでは、スナップショットを削除する方法を説明します。



### Note

- ボリュームのスナップショットを定期的に作成する場合は、差分だけが更新されます。つまり、最後にスナップショットを作成した時点から変更されたデバイス上のブロックだけが新しいスナップショットに保存されます。スナップショットの保存は差分ベースで行われるものの、最新のスナップショットさえあればボリュームを復元できるようにスナップショット削除プロセスは設計されています。
- 登録された AMI によって使用される EBS; ボリュームのルートデバイスのスナップショットを削除することはできません。スナップショットを削除するには、まず AMI の登録を解除する必要があります。

コンソールを使用してスナップショットを削除するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインで [Snapshots] をクリックします。

コンソールに現在のスナップショットのリストが表示されます。

3. スナップショットを選択して [Actions] リストから [Delete] を選択します。
4. [Yes, Delete] をクリックします。

スナップショットが削除されます。

コマンドラインを使用してスナップショットを削除するには

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

- [delete-snapshot](#) ( AWS CLI )
- [ec2-delete-snapshot](#) ( Amazon EC2 CLI )
- [Remove-EC2Snapshot](#) ( AWS Tools for Windows PowerShell )

## Amazon EBS スナップショットのコピー

### Abstract

Amazon EBS スナップショットは AWS リージョン間でコピーできるので、地理的な拡大、データセンターの移行、災害復旧など、複数の AWS リージョンを活用するのに役立ちます。

Amazon Elastic Block Store ( Amazon EBS ) は、Amazon Elastic Compute Cloud ( Amazon EC2 ) インスタンスで使用するためのブロックレベルのストレージボリュームです。Amazon EBS を使用すると、ボリュームの特定の時点におけるスナップショットを作成し、Amazon Simple Storage Service ( Amazon S3 ) に格納できます。スナップショットを Amazon S3 に格納した後、Amazon EC2 コンソール、Amazon EC2 CLI、または API を使用して、それらのスナップショットを 1 つの AWS リージョンから別のリージョンにコピーしたり、同じリージョン内でコピーしたりすることができます。スナップショットのコピーを使用して、データのバックアップを作成したり、新しい Amazon EBS ボリュームを作成したり、Amazon マシンイメージ ( AMI ) を作成したりすることができます。

アカウントあたり 1 つのコピー先に対して、最大 5 つのスナップショットコピーリクエストを実行できます。共有スナップショットや作成したスナップショットを含む "完了" ステータスのアクセス可能な任意の Amazon EBS スナップショットをコピーできます。また、AWS Marketplace、VM Import/Export、および AWS Storage Gateway のスナップショットをコピーすることもできます。ただし、そのスナップショットがコピー先のリージョンでサポートされていることを確認する必要があります。

ボリュームの最初のスナップショットコピーは常に完全コピーです。その後の各スナップショットコピーは差分です。つまり、同じコピー先に対する最後のスナップショットコピーから変更されたボリューム上のブロックのみが転送されます。差分スナップショットにより、コピー処理が速くなります。差分スナップショットのサポートは、リージョンペアに固有です。例えば、米国東部 ( バージニア北部 ) リージョンから米国西部 ( オレゴン ) リージョンにスナップショットをコピーする場合、ボリュームの最初のスナップショットコピーは完全なコピーになります。ただし、同じリージョン間で転送されるその後の同じボリュームのスナップショットコピーは、差分になります。スナップショットコピーは、コピー先のリージョンで同じボリュームの完全コピーが 1 つある限り、差分でのみ実行できます。



### Note

Amazon Relational Database Service ( Amazon RDS ) をコピーするには、『Amazon Relational Database Service ユーザーガイド』の「[DB スナップショットの作成](#)」を参照してください。

スナップショットをコピーした場合、課金の対象となるのは、リージョン間でスナップショットデータをコピーし、コピーされたスナップショットをコピー先リージョンに格納するために使用されたデータ転送およびストレージのみです。スナップショットコピーが失敗した場合は、課金されません。ただし、まだ完了していないスナップショットコピーをキャンセルしたか、コピーの実行中にコピー元のスナップショットを削除した場合、転送されたデータの帯域幅に対して課金されます。スナップショットは、安全な Amazon S3 Copy を使用してリージョン間でコピーされ、スナップショットコピーは元のスナップショットの ID と異なるスナップショット ID を受け取ります。

Amazon EBS スナップショットのコピーは次の方法で使用できます。

- 地理的拡張: アプリケーションを新しいリージョンで起動できます。
- 移行: アプリケーションを新しいリージョンに移行して、可用性を向上させ、コストを最小化することができます。
- 災害復旧: 異なる地理的場所にまたがって定期的にデータをバックアップし、ログを記録することができます。災害が発生した場合、二次的なリージョンに格納された特定の時点でのバックアップを使用してアプリケーションを復元できます。これにより、データ損失と復旧時間を最小限に抑えることができます。

Amazon EC2 コンソール、Amazon EC2 CLI、および API は直感的な操作性を実現するように設計されています。先述の Amazon EBS スナップショットの使用事例で、ユーザーのクリック数を最小にするために、コンソール設計でプッシュモデルを使用しています。コピー元のリージョンから開始することによって、コンソールから簡単にコピーを開始できます。Amazon では、お客様の自動化の使用状況におけるこれらの経験係数のため、Amazon EC2 CLI と API でプルモデルを使用しています。ユーザーは、Amazon EC2 CLI または API を使用してコピーを開始するコピー元のスナップショット ID とコピー元のリージョンを知っているだけで済みます。

Amazon EC2 コンソールを使用してスナップショットをコピーするには

Amazon EC2 コンソールを使用して Amazon EBS スナップショットのコピーを作成することができます。

1. Amazon EC2 コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインで、[Snapshots] をクリックします。
3. コピーするスナップショットを選択し、[Actions] リストから [Copy] を選択します。
4. [Copy Snapshot] ダイアログボックスで、必要に応じて次のものを更新します。
  - [Destination region]: スナップショットのコピーを書き込むリージョンを選択します。
  - [Description]: デフォルトでは、スナップショットとコピーを見分けられるよう、元のスナップショットに関する情報が説明に含まれています。この説明は、必要に応じて変更できます。
5. [Yes, Copy] をクリックします。
6. [Copy Snapshot] 確認ダイアログボックスで、[Snapshots] をクリックして指定したリージョンの [Snapshots] ページに移動するか、[Close] をクリックします。

コピー処理の進行状況を後で確認するには、Amazon EC2 コンソールを目的のリージョンに切り替え、[Snapshots] ページを更新します。進行中のコピーがページの上部に一覧表示されます。

コマンドラインを使用してスナップショットをコピーするには

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

- `copy-snapshot` ( AWS CLI )
- `ec2-copy-snapshot` ( Amazon EC2 CLI )
- `Copy-EC2Snapshot` ( AWS Tools for Windows PowerShell )

## スナップショットに関する情報を表示する

### Abstract

AWS マネジメントコンソール、CLI、または API を使って、作成したスナップショットの説明を入力します。

このセクションでは、作成したスナップショットに関する詳細情報を表示する方法を説明します。

コンソールでスナップショットに関する詳細情報を表示するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインで [Snapshots] をクリックします。

お客様がアクセスできるすべてのスナップショットのリストと、そのステータスがコンソールに表示されます。

3. リストを縮小するには、[Filter] リストからオプションを選択します。例えば、ご自分のスナップショットだけを表示するには、[Owned by Me] を選択します。

スナップショットの新しいリストがコンソールに表示されます。

4. スナップショットの詳細情報を表示するには、そのスナップショットを選択します。

スナップショットの情報が下のペインに表示されます。

コマンドラインを使用してスナップショットに関する詳細情報を表示するには

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

- [describe-snapshots](#) ( AWS CLI )
- [ec2-describe-snapshots](#) ( Amazon EC2 CLI )
- [Get-EC2Snapshot](#) ( AWS Tools for Windows PowerShell )

## スナップショットの共有

### Abstract

AWS マネジメントコンソール、CLI、または API を使って、Amazon EBS スナップショットを共有します。

このセクションでは、スナップショットの許可を変更することで、Amazon EBS スナップショットを同僚や AWS コミュニティ内の他のユーザーと共有する方法について説明します。

スナップショットの所有者から許可を受けたユーザーは、Amazon EBS の共有スナップショットをベースに自分の Amazon EBS ボリュームを簡単に作成できます。お客様が所有するデータを、すべての AWS ユーザーに一般公開することもできます。アクセスを許可されたユーザーは、スナップショットを基に独自の EBS ボリュームを作成できます。元のスナップショットが変更されることはありません。



#### Note

スナップショットは、スナップショットが作成されたリージョンに制限されます。別のリージョンとスナップショットを共有する場合は、スナップショットをそのリージョンにコピーする必要があります。スナップショットのコピーの詳細については、[Amazon EBS スナップショットのコピー \(p. 608\)](#) を参照してください。



#### Important

スナップショットを共有するときは ( 別の AWS アカウントと共有するか、すべてのユーザーに対して公開するかどうかに関係なく )、所有するスナップショットのすべてのデータに対するアクセス許可を他のユーザーに付与することになります。スナップショットの共有は、自分のスナップショットデータすべてを共有したい人とだけ行ってください。

コンソールでスナップショットのアクセス許可を変更するには

この手順により、スナップショットを共有することができます。

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションペインで [Snapshots] をクリックします。

現在のスナップショットのリストとそのステータスがコンソールに表示されます。

3. スナップショットを選択し、[Actions] リストから [Modify Snapshot Permissions] を選択します。
4. スナップショットを公開するか、特定の AWS アカウントとだけ共有するかを選択します。



#### Important

スナップショットを公開すると、すべてのスナップショットデータが全員と共有されません。AWS Marketplace の製品コードを持つスナップショットを公開することはできません。

- スナップショットを公開するには、[Public] を選択します。
- スナップショットを特定の AWS アカウントにのみ公開するには、[Private] を選択し、AWS アカウントの ID をハイフンなしで [AWS Account Number] フィールドに入力して、[Add Permission] をクリックします。必要なすべての AWS アカウントを追加するまで繰り返します。

完了したら [Save] をクリックします。

コマンドラインを使用して、スナップショットのアクセス許可を変更および表示するには

スナップショットの `createVolumePermission` 属性を表示するには、次のコマンドのいずれかを使用できます。これらのコマンドラインインターフェイスの詳細については、「[Amazon EC2 へのアクセス \(p. 3\)](#)」を参照してください。

- `describe-snapshot-attribute` ( AWS CLI )
- `ec2-describe-snapshot-attribute` ( Amazon EC2 CLI )
- `Get-EC2SnapshotAttribute` ( AWS Tools for Windows PowerShell )

スナップショットの `createVolumePermission` 属性を変更するには、次のコマンドのいずれかを使用できます。

- `modify-snapshot-attribute` ( AWS CLI )
- `ec2-modify-snapshot-attribute` ( Amazon EC2 CLI )
- `Edit-EC2SnapshotAttribute` ( AWS Tools for Windows PowerShell )

## Amazon EBS ボリュームのパフォーマンス

### Abstract

Amazon EBS ボリュームパフォーマンスを最適化するためのベストプラクティスを一覧表示します。

インスタンスの構成、I/O 特性、ワークロードのデマンド、ストレージの構成など、複数の要因が Amazon EBS ボリュームのパフォーマンスに影響を与える可能性があります。EBS ボリュームの基本操作について理解したら、必要とする I/O パフォーマンスと、EBS パフォーマンスを向上させるオプションを確認し、そのパフォーマンス要件に対応できるようにすることをお勧めします。

Topics

- [Amazon EBS パフォーマンスのヒント \(p. 612\)](#)
- [Amazon EC2 インスタンスの構成 \(p. 612\)](#)
- [I/O 特性 \(p. 614\)](#)
- [ワークロードのデマンド \(p. 615\)](#)
- [Amazon EBS ボリュームの事前ウォーミング \(p. 615\)](#)
- [RAID 構成 \(p. 620\)](#)
- [ボリュームの評価 \(p. 626\)](#)

## Amazon EBS パフォーマンスのヒント

- EBS ストレージアプリケーションのパフォーマンス要件を検討する場合、EBS 用に最適化されていて、お使いのアプリケーションストレージシステムが必要とする帯域幅を処理できる EC2 構成を使って始めることが重要です。詳細については、[Amazon EC2 インスタンスの構成 \(p. 612\)](#) を参照してください。
- EBS ボリュームのパフォーマンスを測定する場合 (特に Provisioned IOPS ボリュームの場合)、関連する測定単位と、パフォーマンスの計算方法を理解することが重要です。詳細については、[I/O 特性 \(p. 614\)](#) を参照してください。
- EBS ボリュームの最大パフォーマンス、EBS ボリュームが処理する I/O の量、および各トランザクションが完了するまでにかかる時間の間には関係があります。これらの各要因 (パフォーマンス、I/O、時間) は相互に影響を与えます。また、アプリケーションが異なると、影響を受ける要因もさまざまに異なります。詳細については、[ワークロードのデマンド \(p. 615\)](#) を参照してください。
- 新しく作成した、または復元した EBS ボリュームの各データブロックに初めてアクセスした場合、IOPS は 5~50% 減少します。あらかじめ各ブロックにアクセスすることで、このパフォーマンスの低下を回避できます。詳細については、[Amazon EBS ボリュームの事前ウォーミング \(p. 615\)](#) を参照してください。
- 一部のインスタンスタイプは、単一ボリュームに対してプロビジョニングできる IOPS よりも、多くの IOPS を処理できます。RAID 0 構成内で複数の Provisioned IOPS ボリュームをまとめて結合し、これらのインスタンスに対して利用できる帯域幅を使用できます。RAID 1 (ミラーリング) 構成のボリュームに対して、冗長性を持たせることもできます。詳細については、[RAID 構成 \(p. 620\)](#) を参照してください。
- アプリケーションを有効にする前に、予想されるパフォーマンスレベルを確実に実現するため、ストレージを評価して、構成を計算することができます。詳細については、[ボリュームの評価 \(p. 626\)](#) を参照してください。
- Amazon Web Services は、Amazon CloudWatch を使って分析と表示が可能な EBS に対して、パフォーマンスメトリックと、ボリュームの健全性の監視に使用できるステータスチェックを提供します。詳細については、[ボリュームのステータスのモニタリング \(p. 580\)](#) を参照してください。
- スナップショットを頻繁に作成することで高いレベルのデータ堅牢性を実現できますが、スナップショットの作成中はアプリケーションのパフォーマンスがわずかに低下する可能性があります。このトレードオフは、急速に変化するデータがある場合に重要になります。可能な場合は常に、ワークロードの影響を最小限に抑えるため、オフピーク時にスナップショットを実行するように計画してください。詳細については、[Amazon EBS スナップショット \(p. 605\)](#) を参照してください。

## Amazon EC2 インスタンスの構成

### Abstract

Amazon EBS ボリュームの計画と構成時に、Amazon EC2 インスタンスを構成してください。

アプリケーション用に EBS の計画と構成を行う場合、重要な点は、ボリュームにアタッチするインスタンスの構成を考慮することです。EBS ボリュームのパフォーマンスを最大限引き出すには、ボリュームをサポートするために十分な帯域幅があるインスタンスに EBS ボリュームをアタッチしてください (EBS 最適化ボリュームや、10 ギガビットのネットワーク接続を備えたインスタンスなど)。Provisioned

IOPS ボリュームを使用する場合、または 1 つの RAID 構成で複数のボリュームをストライプ構成する場合、これは特に重要です。

#### EBS 最適化インスタンス、または 10 ギガビットネットワークを使用する

最小限の変動性と EBS トラフィックへの専用 EC2 を必要とする、パフォーマンスの影響を受けやすいワークロード（本番データベースやビジネスアプリケーションなど）は、EBS に最適化されたインスタンスや 10 ギガビットのネットワーク接続を備えたインスタンスにアタッチされた Provisioned IOPS ボリュームを使用する必要があります。この基準を満たしていない EC2 インスタンスは、ネットワークリソースが保証されません。EC2 インスタンスと EBS ボリューム間で信頼できるネットワーク帯域幅を確実に維持する唯一の方法は、EBS 最適化として EC2 インスタンスを起動するか、10 ギガビットのネットワーク接続を備えたインスタンスタイプを選択することです。どのインスタンスタイプが 10 ギガビットのネットワーク接続を備えているかを確認するには、[Instance Type Details](#) を参照してください。

#### 十分な帯域幅がある EC2 インスタンスの選択

EBS 最適化されたインスタンスを起動すると、EC2 インスタンスと EBS ボリューム間で専用の接続が確立します。ただし、特に複数のボリュームをまとめて 1 つの RAID 構成にストライプ構成する場合、特定のインスタンスタイプに対して、使用可能な帯域幅を超える EBS ボリュームをプロビジョニングすることができます。次の表は、EBS 最適化として起動できるインスタンスタイプ、また、そのインスタンスがサポート可能な最大 IOPS 量を示しています。アプリケーションが必要とする EBS 帯域幅を超える専用 EBS スループットを提供する EBS 最適化インスタンスを必ず選択してください。そうでない場合、EBS が EC2 に接続する際に、パフォーマンスの障害となります。

インスタンスタイプ	専用 EBS スループット ( Mbps ) *	最大 16K IOPS**
c1.xlarge	1,000	8,000
c3.xlarge	500	4,000
c3.2xlarge	1,000	8,000
c3.4xlarge	2,000	16,000
g2.2xlarge	1,000	8,000
i2.xlarge	500	4,000
i2.2xlarge	1,000	8,000
i2.4xlarge	2,000	16,000
m1.large	500	4,000
m1.xlarge	1,000	8,000
m2.2xlarge	500	4,000
m2.4xlarge	1,000	8,000
m3.xlarge	500	4,000
m3.2xlarge	1,000	8,000
r3.xlarge	500	4,000
r3.2xlarge	1,000	8,000
r3.4xlarge	2,000	16,000

m1.large インスタンスは、最大 4,000 の 16 KB IOPS 値を備えています。このインスタンスタイプを EBS 最適化として起動しなければ、この値は絶対的な最良のシナリオとなり、保証はされません。4,000 16 KB IOPS を一貫して実現するには、このインスタンスを EBS 最適化として起動する必要があります。ただし、2 つの 4,000 Provisioned IOPS EBS ボリュームが、RAID 構成の EBS 最適化 m1.large インスタンスにアタッチされている場合、EC2 から EBS への接続帯域幅制限により、これらのボリュームはプロビジョニングされている最大 IOPS が提供できません。この場合は、m1.xlarge インスタンスタイプなど、8,000 16 KB IOPS をサポートする、EBS 最適化 EC2 インスタンスを使用する必要があります。

10 ギガビットイーサネット接続を備えるインスタンスタイプは、をサポートします。EBS ボリュームの最大 Provisioned IOPS 値は 4,000 であるため、複数の EBS ボリュームを同時に使って、これらのインスタンスタイプで使用可能な I/O パフォーマンスレベルに達する必要があります。どのインスタンスタイプが 10 ギガビットのネットワーク接続を備えているかを確認するには、[Instance Type Details](#) を参照してください。

Amazon EBS Provisioned IOPS ボリュームのパフォーマンス上のメリットを最大限に活用できる場合には、EBS 最適化インスタンスを使用する必要があります。詳細については、[Amazon EBS 最適化インスタンス \(p. 127\)](#) を参照してください。

## I/O 特性

### Abstract

使用しているボリューム構成の I/O 特性を設定して、バックエンドのパフォーマンスを向上させます。

一定の構成では、特定の I/O 特性によりバックエンドのパフォーマンス動作が向上します。プロビジョニングされた IOPS ボリュームは、I/O 操作がランダムかシーケンシャルにかかわらず、また I/O 操作がデータの読み取りか書き込みにかかわらず、一定のパフォーマンスを提供します。ただし、I/O サイズは IOPS の測定方法が原因で、IOPS に影響を与えます。Provisioned IOPS ボリュームとアプリケーションでの実行方法を完全に理解するには IOPS についてと、その測定方法を知ることが重要です。

IOPS とは何か？

IOPS とは 1 秒あたりの入出力操作です。Amazon EBS はこのような I/O 操作を 16 KB のチャンクで測定します。4,000 IOPS のボリュームをプロビジョニングして、必要な帯域幅を提供でき、EBS 最適化インスタンスにそのボリュームをアタッチした場合、1 秒あたり、4,000 個の 16 KB チャンクを転送できます (帯域幅が約 62.5 MB または 500 Mbps の場合)。

また、この構成の場合、1 秒あたり 4,000 個の 16 KB I/O 操作に匹敵する帯域幅の上限に達するまでに、1 秒あたり 2,000 個の 32 KB チャンク、または 1,000 個の 64 KB チャンクを転送することもできます。I/O チャンクが 16 KB より大きい場合、プロビジョニングする IOPS よりも値が小さくなる可能性があります。実際の帯域幅は、プロビジョニングする 16 KB IOPS によって使われる値と同じです。

16 KB 以下の I/O 操作の場合、プロビジョニングした IOPS の値が表示されているはずですが、ドライブをビジー状態に維持するために十分な I/O が実行されているためです。これより小さな I/O 操作の場合、プロビジョニングした値よりも大きな IOPS 値が表示される可能性があります (クライアント側で測定した場合)。これは、クライアントが複数の小さな I/O 操作を、単一の 16 KB チャンクにまとめている可能性があるためです。

予想された IOPS やプロビジョニングしたスループットが得られない場合、EC2 帯域幅が障害の要因ではないことを確認してください。インスタンスは EBS に最適化されていて (または 10 ギガビットのネットワーク接続を備えていて)、インスタンスタイプの EBS 専用帯域幅がプロビジョニングする IOPS より大きい必要があります。詳細については、[Amazon EC2 インスタンスの構成 \(p. 612\)](#) を参照してください。予想された IOPS が得られない別の原因として、EBS ボリュームに対して十分な I/O を提供していないことが考えられます。詳細については、[ワークロードのデマンド \(p. 615\)](#) を参照してください。

## ワークロードのデマンド

### Abstract

ワークロードのデマンドとリクエストのレイテンシーを理解することで、プロビジョニングした IOPS ボリュームの性能を最大限発揮させることができます。

ワークロードのデマンドは、Provisioned IOPS ボリュームを最大限活用するために重要な役割を果たします。プロビジョニングした IOPS 値をボリュームで実現するには、ボリュームに対して十分な I/O リクエストが送信される必要があります。ボリュームに対するデマンド、ボリュームに対してプロビジョニングした IOPS の量、リクエストのレイテンシー ( I/O 操作が完了するまでにかかる時間 ) の間には関係があります。

#### 平均キュー長

キュー長は、デバイスへの保留中の I/O リクエストの数です。最適な平均キュー長はお客様ごとのワークロードによって異なり、この値は IOPS およびレイテンシーに対する特定のアプリケーションの感度によって変わります。ワークロードが、最適な平均キュー長を維持するだけの十分な I/O リクエストを提供していない場合、ボリュームはプロビジョニングした IOPS を一貫して提供することができません。ただし、最適な値よりも高い平均キュー長をワークロードで維持している場合、リクエストあたりの I/O レイテンシーは増えます。この場合、ボリュームに対してより多くの IOPS をプロビジョニングする必要があります。200 Provisioned IOPS ごとに最適な平均キュー長を設定し、アプリケーションの要件に応じてこの値を調整することをお勧めします。例えば、1000 Provisioned IOPS のボリュームでは、平均キュー長 5 を設定します。



#### Note

リクエストあたりの I/O レイテンシーは、より長い平均キュー長で増加する場合があります。

#### レイテンシー

レイテンシーとは、I/O 操作にかかる、エンドツーエンドの実際の時間です。つまり、クライアントが IO を送信した場合に、I/O の読み取りまたは書き込みが終了したという応答を、ストレージサブシステムから取得するまでにかかる時間を指します。I/O レイテンシーが必要な値よりも高い場合、平均キュー長をチェックして、アプリケーションがプロビジョニングした IOPS 以上の処理を実行しようとしていないことを確認します。平均キュー長を低く維持する ( これはボリュームに対してより多くの IOPS を提供することで実現します ) ことで、レイテンシーを抑えながら、高い IOPS を維持することができます。

## Amazon EBS ボリュームの事前ウォーミング

### Abstract

新規 EBS ボリュームを使用する前に、このボリュームのすべてのブロックを対象に、書き込みまたは読み取りを行う ( このプロセスを「事前ウォーミング」と呼びます ) ことで、パフォーマンスの低下を回避します。

EBS ボリュームの新規作成時、またはスナップショットからのボリュームの復元時、バックエンドストレージブロックの割り当ては直ちに実行されます。ただし、ストレージのブロックに初めてアクセスする場合、ブロックにアクセスする前に、そのブロックのワイプを実行して消去する ( 新規ボリュームの場合 ) が、スナップショットからインスタンス化する ( 復元されたボリュームの場合 ) 必要があります。この準備処理には時間がかかるため、初めて各ブロックにアクセスした場合に、ボリュームの IOPS が 5 ~ 50 パーセント低下します。ほとんどのアプリケーションにとって、ボリュームの存続期間全体でこのコストを割り当てることは、許容範囲内です。一度データにアクセスされると、パフォーマンスは元に戻ります。

ただし、ボリュームを使用する前に、そのボリュームのすべてのブロックを対象に、書き込みまたは読み取りを実行することで、パフォーマンスの低下を回避することができます。このプロセスを「事前

ウォーミング」と呼びます。1つのボリュームですべてのブロックに書き込みを行うことが望ましいといえますが、スナップショットから復元されたボリュームの場合、選択できません。復元されたデータが上書きされてしまうためです。最初から作成された完全に新しいボリュームの場合、ボリュームを使用する前に、すべてのブロックを書き込む必要があります。スナップショットから作成された新しいボリュームの場合は、ボリュームを使用する前に、データのあるすべてのブロックを読み取る必要があります。

#### Topics

- [Linux の Amazon EBS ボリュームの事前ウォーミング \(p. 616\)](#)
- [Windows の Amazon EBS ボリュームの事前ウォーミング \(p. 618\)](#)

## Linux の Amazon EBS ボリュームの事前ウォーミング

### Linux の新規ボリューム事前ウォーミングするには

新規ボリュームの場合、ddコマンドを使って、ボリュームですべてのブロックに書き込みを行います。この手順では、デバイスのすべてのブロックにゼロを書き込み、ボリュームを使用する準備を整えます。ボリュームの既存データはすべて失われます。

1. 新しく作成したボリュームを Linux インスタンスにアタッチします。
2. インスタンスのブロックデバイスを一覧表示するには、lsblk コマンドを使用します。

```
[ec2-user ~]$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf  202:80  0  30G  0 disk
xvda1 202:1   0   8G  0 disk /
```

ここでは、新規ボリューム `/dev/xvdf` がアタッチされていますが、マウントはされていません (MOUNTPOINT 列の下にパスが表示されていないため)。



#### Important

新規ボリュームがマウントされている場合は、アンマウントします。[Step 4 \(p. 616\)](#) は、マウントされたデバイスでは実行できません。

3. (オプション) ボリュームがマウントされている場合、次のコマンドを使用してマウント解除します。

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

4. デバイスのすべてのブロックに書き込みを行うには、dd コマンドを使用します。if (ファイルのインポート) パラメータをいずれかの Linux 仮想デバイス (`/dev/zero` など) に設定します。ウォーミングを行うドライブに対して、of (ファイルの出力) パラメータを設定します。bs パラメータは書き込み操作のブロックサイズを設定します。最適なパフォーマンスのためには、1 MB に設定します。

```
[ec2-user ~]$ sudo dd if=/dev/zero of=/dev/xvdf bs=1M
```



#### Note

このステップは、使用している EC2 インスタンスの帯域幅、ボリュームに対してプロビジョニングされている IOPS、そしてボリュームのサイズに応じて、数分から数時間かかることがあります。

操作が終了すると、書き込み操作のレポートが表示されます。

```
dd: writing `'/dev/xvdf': No space left on device
15361+0 records in
15360+0 records out
32212254720 bytes (32 GB) copied, 2449.12 s, 13.2 MB/s
```

これでボリュームを使用する準備ができました。詳細については、[Amazon EBS ボリュームを使用できるようにする \(p. 576\)](#) を参照してください。

スナップショットから復元された Linux のボリュームを事前ウォームするには

スナップショットから復元されたボリュームの場合、ボリュームのすべてのブロックから読み取り（および任意でブロックへの書き戻し）を行うには dd コマンドを使用します。ボリュームのすべてのデータが保持されます。

1. 新しく復元されたボリュームを Linux インスタンスにアタッチします。
2. インスタンスのブロックデバイスを一覧表示するには、lsblk コマンドを使用します。

```
[ec2-user ~]$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf 202:80 0 30G 0 disk
xvda1 202:1 0 8G 0 disk /
```

ここでは、新規ボリューム `/dev/xvdf` がアタッチされていますが、マウントはされていません（MOUNTPOINT 列の下にパスが表示されていないため）。



#### Important

新規ボリュームがマウントされている場合は、アンマウントします。[Step 4 \(p. 617\)](#) は、マウントされたデバイスでは実行できません。

3. (オプション) ボリュームがマウントされている場合、次のコマンドを使用してマウント解除します。

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

4. デバイスのすべてのブロックを読み取るには、dd コマンドを使用します。事前ウォーミングの対象を以前書き込みを行ったブロックだけにするか、ボリューム全体にするかに応じて、以下のいずれかの dd コマンドを選択します。
  - 読み取り専用: 以前書き込みを行ったブロックだけを事前ウォーミングするには。このコマンドは、既存のデータと、以前に完全事前ウォーミングが行われたボリュームの復元されたスナップショットを事前ウォーミングします。このコマンドは、増分スナップショットを維持します。ただし、この操作は読み取り専用であるため、元のボリュームで書き込みが行われたことがない未使用スペースに対して、事前ウォーミングは行われません。

if (入力ファイル) パラメータは、ウォーミングを行うドライブに対して設定します。of (出力パラメータ) は、Linux ヌル仮想ドライブ `/dev/null` に設定する必要があります。bs パラメータは、読み取り操作のブロックサイズに設定します。最適なパフォーマンスを得るには、この値を 1 MB に設定します。

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

- 読み取り後の書き込みを実行: 復元されたボリューム全体を事前ウォーミングするには、このコマンドはデバイスからすべてのブロックを読み取り、同じ場所に再書き込みを行います。これにより、以前に事前ウォーミングされていない未使用のスペースを含め、ボリューム全体が事前ウォーミングされます。このコマンドはデバイスのすべてのブロックに書き込みを行うため、このように事前ウォーミングが実行されたボリュームに関する次のスナップショットは完全スナップショットが必要です。

`if` (入力ファイル) パラメータは、ウォーミングを行うドライブに対して設定します。`of` (出力ファイル) パラメータも、ウォーミングするドライブに設定する必要があります。`conv=notrunc` パラメータを指定すると、`dd` を実行した際、各出力ファイルブロックは削除される代わりに上書きされます。`bs` パラメータは、読み取り操作のブロックサイズに設定します。最適なパフォーマンスを得るには、この値を 1 MB に設定します。

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/xvdf conv=notrunc bs=1M
```



#### Note

このステップは、使用している EC2 インスタンスの帯域幅、ボリュームに対してプロビジョニングされている IOPS、そしてボリュームのサイズに応じて、数分から数時間かかることがあります。

この操作が終了すると、読み取り操作のレポートが表示されます。

```
15360+0 records in
15360+0 records out
32212254720 bytes (32 GB) copied, 2480.02 s, 13.0 MB/s
```

これでボリュームを使用する準備ができました。詳細については、[Amazon EBS ボリュームを使用できるようにする \(p. 576\)](#) を参照してください。

## Windows の Amazon EBS ボリュームの事前ウォーミング

Windows の EBS ボリュームを事前ウォーミングするには、複数の方法があります。最も簡単なソリューションは、ボリュームを完全フォーマットすることです。次のコマンドを使って、新規ボリュームの完全フォーマットを実行します。



#### Warning

次のコマンドを実行すると、ボリュームの既存データが破棄されます。

```
C:\>format drive_letter: /p:1
```

あるいは、Windows Explorer ウィンドウでドライブを右クリックし、[Format] をクリックしても、完全フォーマットを実行できます。この操作はボリュームの全データを破棄するため、新しいボリュームに対してのみ実行することが適切です。スナップショットから復元されたボリュームも、既存データが保存されたボリュームの事前ウォーミングできる、最も強力な事前ウォーミングツールとして、Windows 用 `dd` を検討してください。

#### Windows 用 dd のインストール

Windows 用 `dd` プログラムは、Linux や Unix システムで共通して使用できる `dd` プログラムと同様の環境を提供します。このプログラムを使用すると、新規 EBS ボリュームと、スナップショットから復元されたボリュームの両方を事前ウォーミングすることができます。本書の作成時点では、最新のベータ

バージョンには、スナップショットから復元されたボリュームを事前ウォーミングするために必要な /dev/null 仮想デバイスが含まれています。プログラムの詳細なドキュメントは、<http://www.chrysocome.net/dd> から入手できます。

1. Windows 用の最新バイナリバージョンの dd を、<http://www.chrysocome.net/dd> からダウンロードします。復元されたボリュームを事前ウォーミングするには、0.6 ベータ 3 以降のバージョンを使用する必要があります。
2. ( オプション ) 簡単に覚えられる場所に、コマンドラインユーティリティ用のフォルダを作成します ( C:\bin など )。コマンドラインユーティリティ用にフォルダを既に指定した場合、次の手順を実行する代わりに、そのフォルダを使用できます。
3. バイナリパッケージを解凍し、dd.exe ファイルをコマンドラインユーティリティのフォルダ ( C:\bin など ) にコピーします。
4. どこからでもフォルダ内のプログラムを実行できるようにするため、Path 環境変数にコマンドラインユーティリティのフォルダを追加します。



#### Important

以下の手順を行っても、現在開いているコマンドプロンプトウィンドウの環境変数は更新されません。これらの手順を完了してからコマンドプロンプトウィンドウを開くと、更新分が反映されます。環境が適切に設定されたことを確認するために、新しいコマンドプロンプトウィンドウを開く必要があるのは、このためです。

- a. [Start] ボタンをクリックして [Computer] を右クリックし、[Properties] をクリックします。
- b. [Advanced system settings] をクリックします。
- c. [Environment Variables] をクリックします。
- d. [System Variables] の [Path] という名前の変数をクリックし、[Edit] をクリックします。
- e. [Variable value] で、既存の値の最後に、セミコロンとコマンドラインユーティリティフォルダの場所 ( ;C:\bin\ ) を追加します。
- f. [OK] をクリックして、[Edit System Variable ] ウィンドウを閉じます。

#### Windows 用 dd を使ってボリュームを事前ウォーミングするには

1. システムで使用可能なデバイスを一覧表示するには、dd --list コマンドを使用します。

```
C:\>dd --list
rawwrite dd for windows version 0.6beta3.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by terms of the GPL Version 2.

Win32 Available Volume Information
\\.\Volume{6c195fc1-0511-11e3-9220-806e6f6e6963}\
  link to \\?\Device\HarddiskVolume1
  fixed media
  Mounted on \\.\c:

\\.\Volume{c608dec6-4280-11e3-89dd-123141015292}\
  link to \\?\Device\HarddiskVolume2
  fixed media
  Mounted on \\.\z:
```

Volume 名、Device 名、またはドライブ文字 ( \\.\z: ) を使って、以降の手順で事前ウォーミングするデバイスを特定することができます。ドライブを完全消去する場合は、[Step 2 \(p. 620\)](#) に進んでください。ボリュームのデータを維持する場合は、[Step 3 \(p. 620\)](#) に進んでください。

2. ( オプション: 破壊的事前ウォーミングコマンド ) 次のコマンドを実行すると、( 仮想デバイス /dev/zero から ) [Step 1 \(p. 619\)](#) で識別された指定デバイスにゼロが書き込まれます。bs=1M フラグは、1メガバイトのブロックサイズ ( これにより書き込みプロセスの速度が向上します ) を使用します。また --progress フラグは、書き込み操作実行時のステータス更新を提供します。



#### Warning

次のコマンドを実行すると、ボリュームの既存データが破棄されます。既存データが存在するボリューム ( スナップショットから復元されたボリュームなど ) の場合、[Step 3 \(p. 620\)](#) のコマンドを使用します。

```
C:\>dd if=/dev/zero of=\\.\drive_letter: bs=1M --progress
```

3. ( オプション: 読み取り専用の事前ウォーミングコマンド ) 指定したデバイスのすべてのブロックの読み取り ( および /dev/null 仮想デバイスへの出力の送信 ) を実行するには、次のコマンドを実行します。このコマンドを実行すると、既存のデータ、および以前に完全事前ウォーミングしたボリュームの復元済みスナップショットに対して、安全に事前ウォーミングが実行されます。

```
C:\>dd if=\\.\drive_letter: of=/dev/null bs=1M --progress
```

4. この操作が終了すると、新規ボリュームを使用する準備が完了します。詳細については、[Amazon EBS ボリュームを使用できるようにする \(p. 576\)](#) を参照してください。

## RAID 構成

### Abstract

Amazon EBS ボリュームで、従来のペアメタルサーバーで使用できる標準的な RAID 設定のいずれかを使用します。

Amazon EBS では、従来のペアメタルサーバーで使用できる標準的な RAID 設定はすべて使用できません。ただしその RAID 設定が、お使いのインスタンスのオペレーティングシステムでサポートされている必要があります。これは、RAID がすべてソフトウェアレベルで実現されるためです。単一のボリュームで実現できる以上の I/O パフォーマンスを実現するため、RAID 0 では複数のボリュームをともにストライピングできます。インスタンスでの冗長性確保のため、RAID 1 では 2 つのボリュームを同時にミラーリングできます。

次の表では、これら 2 つの一般的なストライピングオプションを比較しています。

設定	使用	利点	欠点
RAID 0	( データアプリケーションが既に別にセットアップされている ) 使用頻度が高いデータベースなど、I/O パフォーマンスが耐障害性よりも重要な場合。	I/O がストライプ内のボリュームにわたって分散されます。ボリュームを追加すると、スループットも追加されます。	ストライプのパフォーマンスは、セット内で最もパフォーマンスが良くないボリュームに制限されます。単一ボリュームの損失により、アレイのデータが完全に失われます。

設定	使用	利点	欠点
RAID 1	クリティカルなアプリケーション内など、耐障害性が I/O パフォーマンスより重要な場合。	データ堅牢性の観点から見て、より安全です。	書き込みパフォーマンスの向上は得られません。データが同時に複数のボリュームに書き込まれるため、非 RAID 構成と比較して大きな帯域幅が必要となります。



#### Important

RAID 5 と RAID 6 は、最適なパフォーマンスを実現しないため、Amazon EBS には推奨されません。

RAID 0 アレイを作成すると、単一の Amazon EBS ボリュームでプロビジョニングする場合よりも、ファイルシステムで高レベルのパフォーマンスが実現されます。RAID 1 アレイは、冗長性を向上させるため、データのミラーリングを可能にします。この手順を実行する前に、RAID アレイで必要となるサイズと、プロビジョニングする IOPS の数を決定してください。

RAID 0 アレイの最終的なサイズは、アレイ内のボリュームサイズの合計です。帯域幅は、アレイ内のボリュームで利用可能な帯域幅の合計です。RAID 1 アレイの最終的なサイズと帯域幅は、アレイ内にあるボリュームのサイズと帯域幅に等しくなります。たとえば、500 GiB Amazon EBS ボリュームが 2 つあり、それぞれに 4,000 の IOPS がプロビジョニングされる場合、RAID 0 アレイは 1 TiB で、利用可能な帯域幅は 8,000 16K IOPS となります。また RAID 1 アレイは 500 GiB で、利用可能な帯域幅は 4,000 16K IOPS となります。

#### Topics

- [Linux での RAID アレイの作成 \(p. 621\)](#)
- [Windows での RAID アレイの作成 \(p. 622\)](#)

## Linux での RAID アレイの作成

Linux で RAID アレイを作成するには

1. アレイに Amazon EBS ボリュームを作成します。詳細については、[新しい Amazon EBS ボリュームの作成 \(p. 570\)](#) を参照してください。



#### Important

アレイに作成するボリュームのサイズと IOPS パフォーマンス値は同一にしてください。EC2 インスタンスで利用可能な帯域幅を超えるアレイを作成しないよう注意してください。詳細については、[Amazon EC2 インスタンスの構成 \(p. 612\)](#) を参照してください。

2. アレイをホストするインスタンスに Amazon EBS ボリュームをアタッチします。詳細については、[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#) を参照してください。
3. (オプション) ボリュームを事前ウォームします。Amazon EBS ボリュームでは、ボリュームのブロックに初めてアクセスしたときに、パフォーマンスが 5 ~ 50% 低下することがあります。事前ウォームによって、ボリュームの各ブロックへのアクセスが必要となる前にあらかじめアクセスしておくと、実稼働時に最大のパフォーマンスを実現することができます。詳細については、「[Amazon EBS ボリュームの事前ウォーミング \(p. 615\)](#)」を参照してください。
4. mdadm コマンドを使用して、新しくアタッチした Amazon EBS ボリュームから論理 RAID デバイスを作成します。`number_of_volumes` に、構成するアレイ内のボリュームの数を入れ、`device_name` に、アレイ内の各ボリュームのデバイス名 (`/dev/xvdf` など) を入れます。



#### Note

インスタンスのデバイス名を見つけるには、lsblk コマンドを使用してデバイスのリストを表示します。

( RAID 0 の場合のみ ) RAID 0 アレイを作成するには、以下のコマンドを実行します ( --level=stripe オプションでアレイをストライピングしている点に注意してください )。

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=stripe --raid-devices=number_of_volumes device_name1 device_name2
```

( RAID 1 の場合のみ ) RAID 1 アレイを作成するには、以下のコマンドを実行します ( --level=mirror オプションでアレイをミラーリングしている点に注意してください )。

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=mirror --raid-devices=number_of_volumes device_name1 device_name2
```

- RAID デバイスにファイルシステムを作成します。例えば、ext4 ファイルシステムを作成するには、以下のコマンドを実行します。

```
[ec2-user ~]$ sudo mkfs.ext4 /dev/md0
```

アプリケーションの要件またはオペレーティングシステムの制限に応じて、ext3、XFS など、異なるファイルシステムの種類を選択できます。

- RAID アレイのマウントポイントを作成します。

```
[ec2-user ~]$ sudo mkdir /mnt/md0
```

- 最後に、作成したマウントポイントに RAID デバイスをマウントします。

```
[ec2-user ~]$ sudo mount -t ext4 /dev/md0 /mnt/md0
```

これで RAID デバイスを使用できるようになりました。

## Windows での RAID アレイの作成

Windows で RAID アレイを作成するには

- アレイに Amazon EBS ボリュームを作成します。詳細については、[新しい Amazon EBS ボリュームの作成 \(p. 570\)](#) を参照してください。



#### Important

アレイに作成するボリュームのサイズと IOPS パフォーマンス値は同一にしてください。EC2 インスタンスで利用可能な帯域幅を超えるアレイを作成しないよう注意してください。詳細については、[Amazon EC2 インスタンスの構成 \(p. 612\)](#) を参照してください。

- アレイをホストするインスタンスに Amazon EBS ボリュームをアタッチします。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#)」を参照してください。

- Windows インスタンスに接続する。詳細については、「[RDP を使用した Windows インスタンスへの接続 \(p. 344\)](#)」を参照してください。
- コマンドプロンプトを開いて、diskpart コマンドを入力します。

```
PS C:\Users\Administrator> diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: WIN-BM6QPPL51CO
```

- DISKPART プロンプトで、次のコマンドを使用して、利用できるディスクの一覧を表示できます。

```
DISKPART> list disk

Disk ###  Status              Size          Free           Dyn  Gpt
-----  -
Disk 0    Online              30 GB         0 B
Disk 1    Online              8 GB          0 B
Disk 2    Online              8 GB          0 B
Disk 3    Online              8 GB          0 B
Disk 4    Online              8 GB          0 B
Disk 5    Online             419 GB         0 B
Disk 6    Online             419 GB         0 B
```

アレイで使用するディスクを確認し、ディスクの番号を書き留めます。

- アレイで使用する各ディスクは、既存のボリュームを含まない動的なディスクである必要があります。ベーシックディスクを動的なディスクに変換する手順と既存のボリュームを削除する手順は以下のとおりです。
  - 次のコマンドを使用して、アレイで利用するディスクを選択します。*n* を利用するディスクの番号に置き換えます。

```
DISKPART> select disk n

Disk n is now the selected disk.
```

- 選択したディスクにおいて、前述の list disk コマンドの出力の Dyn 列にアスタリスクが付いていない場合、動的なディスクに変換する必要があります。

```
DISKPART> convert dynamic
```

- detail disk コマンドを使用して、選択したディスクの既存のボリュームを確認します。

```
DISKPART> detail disk

XENSRC PVDISK SCSI Disk Device
Disk ID: 2D8BF659
Type   : SCSI
Status : Online
Path   : 0
Target : 1
LUN ID : 0
Location Path : PCIROOT(0)#PCI(0300)#SCSI(P00T01L00)
```

```

Current Read-only State : No
Read-only      : No
Boot Disk     : No
Pagefile Disk  : No
Hibernation File Disk : No
Crashdump Disk : No
Clustered Disk : No

   Volume ###  Ltr  Label           Fs      Type        Size      Status
Info
-----
Volume 2      D    NEW VOLUME     FAT32   Simple      8189 MB   Healthy

```

ディスクのボリュームの番号を書き留めます。この例では、ボリュームの番号は 2 です。ボリュームがない場合、次の手順は省略できます。

- d. ( 前の手順でボリュームの存在が確認された場合のみ ) 前の手順で確認したディスクで既存のボリュームを選択して削除します。



#### Warning

この手順により、ボリュームの既存データがすべて失われます。

- i. ボリュームを選択します。 *n* をボリュームの番号と置き換えてください。

```

DISKPART> select volume n
Volume n is the selected volume.

```

- ii. ボリュームを削除します。

```

DISKPART> delete volume

DiskPart successfully deleted the volume.

```

- iii. 選択したディスクで削除する必要があるボリュームごとに以下の手順を繰り返します。

- e. アレイで使用するディスクごとに [Step 6 \(p. 623\)](#) を繰り返します。

7. 使用するディスクが動的なディスクであるかどうかを確認します。

```

DISKPART> list disk

Disk ###  Status      Size      Free      Dyn  Gpt
-----
Disk 0    Online     30 GB     0 B
Disk 1    Online     8 GB      0 B      *
Disk 2    Online     8 GB      0 B      *
Disk 3    Online     8 GB      0 B      *
* Disk 4    Online     8 GB      0 B      *
Disk 5    Online    419 GB    0 B
Disk 6    Online    419 GB    0 B

```

8. RAID アレイを作成します。Windows では、RAID0 ボリュームはストライプボリュームと呼ばれ、RAID 1 ボリュームはミラーボリュームと呼ばれます。

(ストライプボリュームのみ) ディスク 1 とディスク 2 にストライプボリュームアレイを作成するには、次のコマンドを使用します ( `stripe` オプションでアレイをストライピングしている点に注意してください)。

```
DISKPART> create volume stripe disk=1,2

DiskPart successfully created the volume.
```

(ミラーボリュームのみ) ディスク 3 とディスク 4 にミラーボリュームアレイを作成するには、次のコマンドを使用します ( `mirror` オプションでアレイをミラーリングしている点に注意してください)。

```
DISKPART> create volume mirror disk=3,4

DiskPart successfully created the volume.
```

9. 新しいボリュームを確認します。

```
DISKPART> list volume

Volume ###  Ltr  Label           Fs          Type          Size      Status       Info
-----  ---  -----  -
Volume 0          C                NTFS       Partition    29 GB      Healthy      System
* Volume 1                RAW        Mirror       8190 MB     Healthy
Volume 2                RAW        Stripe       15 GB      Healthy
Volume 5          Z  Temporary S    NTFS       Partition    419 GB     Healthy
Volume 6          Y  Temporary S    NTFS       Partition    419 GB     Healthy
```

この例では、Type 列に、 ボリュームと ボリュームが一覧表示されていることに注意してください。

10. ボリュームを選択してフォーマットし、ボリュームの使用を開始できるようにします。
- フォーマットするボリュームを選択します。 `n` をボリュームの番号に置き換えます。

```
DISKPART> select volume n

Volume n is the selected volume.
```

- ボリュームをフォーマットします。



#### Note

完全フォーマットを実行するには、`quick` オプションを省略します。

```
DISKPART> format quick recommended label="My new volume"

100 percent completed
```

```
DiskPart successfully formatted the volume.
```

- c. ボリュームに使用可能な任意のドライブ文字を割り当てます。

```
DISKPART> assign letter f  
  
DiskPart successfully assigned the drive letter or mount point.
```

新しいボリュームを使用する準備ができました。

## ボリュームの評価

### Abstract

ベンチマークテストを使ってワークロードをシミュレーションすることで、プロビジョニングされている IOPS ボリュームのパフォーマンスをテストします。

このセクションでは、データベースアプリケーションのワークロードと似たワークロードをシミュレートして、Provisioned IOPS ボリュームのパフォーマンスをテストする方法を示します。手順は次のとおりです。

1. EBS 最適化インスタンスを起動する
2. 新しい Provisioned IOPS ボリュームを作成する
3. EBS 最適化インスタンスにボリュームをアタッチする
4. ボリュームから RAID アレイを作成し、フォーマットしてマウントする
5. ツールをインストールし、I/O パフォーマンスを評価する
6. ボリュームの I/O パフォーマンスを評価する
7. ボリュームを削除し、料金が発生しないようにインスタンスを停止する

## インスタンスのセットアップ

Provisioned IOPS ボリュームで最適なパフォーマンスを実現するために、EBS 最適化インスタンスを使用することをお勧めします。EBS 最適化インスタンスは、Amazon EC2 と Amazon EBS の間で所定のスループットを実現するものであり、インスタンスタイプに応じて 500 ~ 2,000 Mbps の範囲内で選択できます。

EBS 最適化インスタンスを作成するには、EC2 コンソールを使用してインスタンスを起動するときに [Launch as an EBS-Optimized instance] を選択するか、コマンドラインを使用して `--ebs-optimized` を指定します。必ずこのオプションをサポートするインスタンスタイプのいずれかを起動してください。このトピックのテスト例では、`m1.xlarge` インスタンスを起動することをお勧めします。詳細については、「[Amazon EBS 最適化インスタンス \(p. 127\)](#)」を参照してください。

Provisioned IOPS ボリュームを作成するには、EC2 コンソールを使用してボリュームを作成するときに [Provisioned IOPS (io1)] を選択するか、コマンドラインを使用して `--type io1 --iopsiops` を指定します。インスタンスへのこれらのボリュームのアタッチについては、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#)」を参照してください。

テストの例では、高いレベルのパフォーマンスを提供する、6 ボリュームを持つ RAID アレイを作成することをお勧めします。使用したギガバイトとプロビジョニングされた IOPS 数 (ボリューム数ではありません) に対して料金が発生するので、複数の小さなボリュームを作成し、そのボリュームを使用し

でストライプセットを使用するための追加コストはありません。Oracle ORION を使用してポリュームを評価する場合は、Oracle ASM と同じ方法でストライピングをシミュレートできます。したがって、ORION でストライピングを行えるようにすることをお勧めします。別のベンチマークツールを使用する場合は、ポリュームを自身でストライプする必要があります。

Windows でのストライプポリュームの作成については、[Create a Striped Volume in Windows](#) を参照してください。

6 ポリュームのストライプセットを Linux に作成するには、次のようなコマンドを使用します。

```
$ sudo mdadm --create /dev/md0 --level=0 --chunk=64 --raid-devices=6 /dev/sdf
/dev/sdg /dev/sdh /dev/sdi /dev/sdj /dev/sdk
```

この例では、XFS ファイルシステムを使用します。要件を満たすファイルシステムを使用する必要があります。Amazon Linux では、次のコマンドを使用して XFS ファイルシステムをインストールします。

```
$ sudo yum install -y xfsprogs
```

そして、次のコマンドを使用して、XFS ファイルシステムを作成してマウントします。

```
$ sudo mkdir -p /media/p_iops_vol0 && sudo mkfs.xfs /dev/md0 &&
sudo mount -t xfs /dev/md0 /media/p_iops_vol0 && sudo chown ec2-user:ec2-user
/media/p_iops_vol0/
```

最後に、これらのテストの実行中、最も正確な結果を取得するには、ポリュームを事前ウォームしておく必要があります。最初から作成された完全に新しいポリュームの場合、ポリュームを使用する前に、すべてのブロックを書き込む必要があります。スナップショットから作成された新しいポリュームの場合は、ポリュームを使用する前に、データのあるすべてのブロックを読み取る必要があります。例えば、Linux では、次のコマンドを使用して、ポリューム上の各ブロックを読み取ることができます。

```
$ dd if=/dev/md0 of=/dev/null
```

Windows では、ポリュームの完全フォーマットによって、事前ウォームします。format <drive letter> /p:1 コマンドを使用して、ディスク全体にゼロを書き込みます。



#### Important

ポリュームを事前ウォームしないと、ポリュームへの初回アクセス時の IOPS が 5~50% 低下する可能性があります。

## ベンチマークツールをインストールする

Provisioned IOPS ポリュームのパフォーマンスを評価するツールとして使用できる、インストール可能なツールを次に示します。

ツール	プラットフォーム	説明
<a href="#">fio</a>	Linux、Windows	I/O パフォーマンスを評価します ( fio は libaio-devel に依存することに注意してください )。
Oracle Orion Calibration ツール	Linux、Windows	Oracle データベースで使用するストレージシステムの I/O パフォーマンスを調整します。

ツール	プラットフォーム	説明
SQLIO	Windows	Microsoft SQL Server で使用するストレージシステムの I/O パフォーマンスを調整します。  Microsoft SQL Server データベースのパフォーマンスを向上させる方法については、MSDN ウェブサイトの「 <a href="#">データベースの最適化</a> 」を参照してください。

## ベンチマークコマンドの例

これらのベンチマークツールは、さまざまなテストパラメータをサポートしています。使用するのは、ボリュームがサポートするワークロードを見積もるためのコマンドです。これらのコマンドを例として使用し、評価を開始するようにしてください。

事前ウォーム済みの Provisioned IOPS ボリュームがアタッチされている EBS 最適化インスタンスで、次のコマンドを実行します。

ボリュームのテストが完了したら、クリーンアップに関するトピック「[Amazon EBS ボリュームの削除 \(p. 592\)](#)」および「[インスタンスの終了 \(p. 352\)](#)」を参照してください。

fio のコマンド

作成したストライプセットで fio を実行します。

デフォルトでは、fio は、ルートパスではない /usr/local/bin にインストールされます。パスを更新するか、chmod を使用して、fio が ec2-user として実行できるようにする必要があります。

次のコマンドは、16 KB のランダム書き込みオペレーションを実行します。

```
fio --directory=/media/p_iops_vol0
--name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G
--numjobs=16 --time_based --runtime=180 --group_reporting
```

次のコマンドは、16 KB のランダム読み取りオペレーションを実行します。

```
fio --directory=/media/p_iops_vol0
--name fio_test_file --direct=1 --rw=randread --bs=16k --size=1G
--numjobs=16 --time_based --runtime=180 --group_reporting
```

結果の解釈の詳細については、Linux のチュートリアル「[Inspecting disk IO performance with fio](#)」を参照してください。

Oracle ORION のコマンド

Provisioned IOPS ボリュームで ORION を実行し、Oracle ASM ストライピングをシミュレートするように指定します。Windows ストライピングを使用するストライプセットを提供するものではありません。

ORION をインストールしたディレクトリに、ファイル piops\_test.lun を作成し、ストライプセットのボリュームを指定します。次のファイル例では、ストライプ対象として 6 つの Provisioned IOPS ボリュームを指定しています。

```
\\.\D:
\\.\E:
```

```
\\.\F:  
\\.\G:  
\\.\H:  
\\.\I:
```

次のコマンドは、16 KB のランダム I/O オペレーション (読み取りが 80 パーセント、書き込みが 20 パーセント) を実行し、64 KB の RAID-0 ストライプをシミュレートします。

```
orion -run advanced -testname piops_test -size_small 16 -size_large 16  
-type rand -simulate raid0 -stripe 64 -write 80 -matrix detailed -num_disks 6
```

コマンドが完了すると、同じディレクトリにコマンド結果の出力ファイルが生成されます。ORION の詳細については、その [ドキュメント](#) を参照してください。

### SQLIO のコマンド

作成したストライプセットで SQLIO を実行します。

param.txt ファイルを作成し、ストライプセットを指定します。このファイルのコンテンツはこのようになります (d:\ はストライプセットに対応し、テストでは、6 つのスレッドと 10 GB のファイルを使用します)。

```
d:\bigtestfile.dat 6 0x0 10240
```

次のコマンドは、16 KB のランダムデータ書き込みを実行します。

```
sqlio -kW -s600 -frandom -t8 -o8 -b16 -LS -BH -Fparam.txt
```

次のコマンドは、16 KB のランダムデータ読み取りを実行します。

```
sqlio -kR -s600 -frandom -t8 -o8 -b16 -LS -BH -Fparam.txt
```

結果はコマンドプロンプトウィンドウに表示されます。SQLIO の詳細については、SQLIO インストールディレクトリにある readme.txt ファイルを参照してください。

## Amazon EBS API およびコマンド概要

### Abstract

Amazon EBS ボリュームを作成および使用するために、使用可能な Amazon EBS コマンドと対応する API アクションを一覧表示します。

以下の表に、Amazon EBS ボリュームを作成および使用するための Amazon EBS コマンドとそれに対応する API アクションをまとめています。

コマンドと API アクション	説明
<a href="#">ec2-attach-volume</a> <a href="#">AttachVolume</a>	<p>指定されたボリュームを、指定されたインスタンスにアタッチし、指定されたデバイス名を使ってボリュームを公開します。</p> <p>ボリュームをアタッチできるのは常に1つのインスタンスに限られます。ボリュームとそのアタッチ先インスタンスは同じアベイラビリティゾーンに存在している必要があります。インスタンスは <code>running</code> または <code>stopped</code> の状態である必要があります。</p>
<a href="#">ec2-copy-snapshot</a> <a href="#">CopySnapshot</a>	<p>Amazon Elastic Block Store ( Amazon EBS ) ボリュームの特定の時点のスナップショットをコピーし、Amazon Simple Storage Service ( Amazon S3 ) に保存します。同じリージョン内のスナップショットか、あるリージョンから別のリージョンにスナップショットをコピーできます。スナップショットを使用して、新しい Amazon EBS ボリュームや Amazon マシンイメージ ( AMI ) を作成することができます。</p>
<a href="#">ec2-create-snapshot</a> <a href="#">CreateSnapshot</a>	<p>指定されたボリュームのスナップショットを作成します。</p> <p>作成後のスナップショットを使って、元のボリュームとまったく同じデータを含むボリュームを作成できます。</p>
<a href="#">ec2-create-volume</a> <a href="#">CreateVolume</a>	<p>サイズとタイプを指定するか、以前作成されたスナップショットを使用して、新しい Amazon EBS ボリュームを作成します。</p>
<a href="#">ec2-delete-disk-image</a>	<p>Amazon S3 からの変換のため、一部または完全にアップロードされたディスクイメージを削除します。</p>
<a href="#">ec2-delete-snapshot</a> <a href="#">DeleteSnapshot</a>	<p>指定されたスナップショットを削除します。</p> <p>このアクションは現在実行中の Amazon EBS ボリュームには影響しません ( そのボリュームが、スナップショットの作成に使用されたか、スナップショットから作成されたかにかかわらず ) 。</p>
<a href="#">ec2-delete-volume</a> <a href="#">DeleteVolume</a>	<p>指定されたボリュームを削除します。このコマンドでは、ボリュームから作成されたスナップショットは削除されません。</p>
<a href="#">ec2-describe-snapshot-attribute</a> <a href="#">DescribeSnapshotAttribute</a>	<p>スナップショットの属性の詳細を表示します。</p>
<a href="#">ec2-describe-snapshots</a> <a href="#">DescribeSnapshots</a>	<p>指定されたスナップショットの詳細を表示します。</p> <p>すべてのスナップショットの詳細を表示します。スナップショット作成元のボリューム、スナップショット開始時刻、進捗状況 ( 完了率 )、ステータス ( <code>pending</code>、<code>completed</code> など ) が含まれます。</p>
<a href="#">ec2-describe-volumes</a> <a href="#">DescribeVolumes</a>	<p>サイズ、ボリュームタイプ、作成元のスナップショット、アベイラビリティゾーン、作成時刻、ステータス ( <code>available</code> または <code>in-use</code> ) など、ボリュームの詳細を表示します。ボリュームが <code>in-use</code> である場合、アタッチに関する行にはボリューム ID、ボリュームのアタッチ先インスタンスのインスタンス ID、インスタンスで表示されるデバイス名、ボリュームのステータス ( <code>attaching</code>、<code>attached</code>、<code>detaching</code>、<code>detached</code> )、およびアタッチした時刻が表示されます。</p>

コマンドと API アクション	説明
<a href="#">ec2-describe-volume-attribute</a> <a href="#">DescribeVolumeAttribute</a>	ボリュームの属性の詳細を表示します。
<a href="#">ec2-describe-volume-status</a> <a href="#">DescribeVolumeStatus</a>	1つまたは複数のボリュームのステータスの詳細を表示します。ボリュームのステータスは、ボリュームに対して実行されたチェックの結果を示し、ボリュームのパフォーマンスを低下させる可能性のあるイベントを判断できます。
<a href="#">ec2-detach-volume</a> <a href="#">DetachVolume</a>	指定されたボリュームを、そのアタッチ先のインスタンスからデタッチします。  このコマンドではボリュームは削除されません。デタッチされたボリュームは別のインスタンスにアタッチでき、デタッチされた時点と同じデータを保有しています。
<a href="#">ec2-enable-volume-io</a> <a href="#">EnableVolumeIO</a>	ボリュームのデータに整合性がない可能性があったため、I/O 操作が無効にされたボリュームの I/O 操作を有効にします。
<a href="#">ec2-import-volume</a> <a href="#">ImportVolume</a>	指定したディスクイメージからメタデータを使用して新しいボリュームのインポートタスクを作成し、そのイメージを Amazon EC2 にインポートします。
<a href="#">ec2-modify-snapshot-attribute</a> <a href="#">ModifySnapshotAttribute</a>	スナップショットに対するアクセス許可 (スナップショットからボリュームを作成する権限) を変更します。1 つ以上の AWS アカウントを指定したり、all を指定してスナップショットを公開したりできます。
<a href="#">ec2-modify-volume-attribute</a> <a href="#">ModifyVolumeAttribute</a>	ボリュームの属性を変更して、I/O 操作に対して、ボリュームを自動的に有効にするかどうかを決定します。
<a href="#">ec2-reset-snapshot-attribute</a> <a href="#">ResetSnapshotAttribute</a>	指定したスナップショットのアクセス許可設定をリセットします。

## Amazon EC2 インスタンスストア

### Abstract

インスタンスのホストコンピュータに物理的にアタッチされているディスクストレージを使用します。

t1.micro インスタンスではない限り、各 Amazon EC2 インスタンスは、ホストコンピュータに物理的にアタッチされたディスクからディスクストレージにアクセスできます。このディスクストレージは、インスタンスストアと呼ばれます。

### Topics

- [インスタンスストレージの概念 \(p. 632\)](#)
- [インスタンスタイプで使用可能なインスタンスストア \(p. 633\)](#)
- [インスタンスストアスワップボリューム \(p. 634\)](#)
- [インスタンスストアデバイス名 \(p. 635\)](#)
- [インスタンスストアの使用シナリオ \(p. 637\)](#)
- [AMI にインスタンスストアボリュームを追加する \(p. 639\)](#)

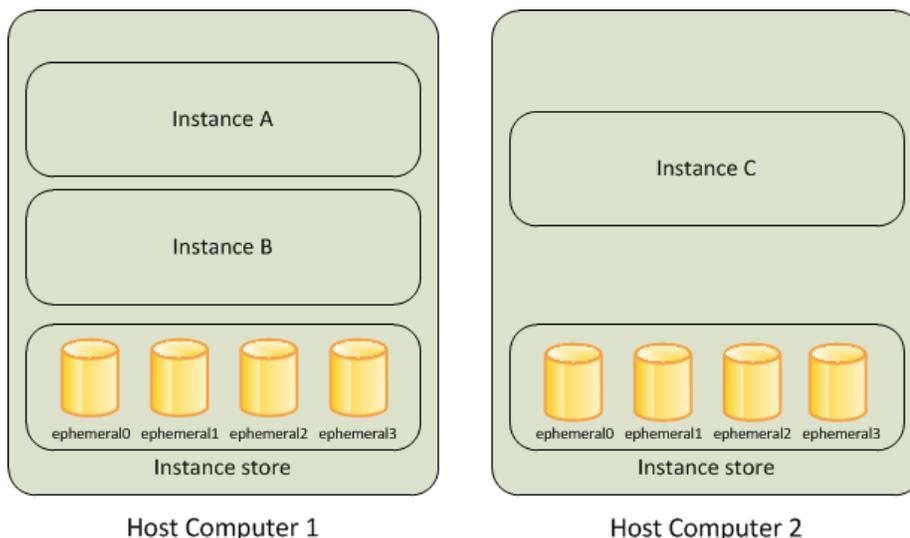
- [ディスクパフォーマンスの最適化 \(p. 640\)](#)

## インスタンスストレージの概念

インスタンスストアは、インスタンスで使用するためのブロックレベルの一時ストレージを提供します。インスタンスストアのサイズは 150 GiB から最大 48 TiB までで、インスタンスタイプによって異なります。大きなインスタンスタイプほどインスタンスストアが大きくなります。詳細については、「[インスタンスタイプで使用可能なインスタンスストア \(p. 633\)](#)」を参照してください。

インスタンスストアは1つ以上のインスタンスストアボリュームで構成されます。インスタンスストアボリュームは、起動時にブロックデバイスマッピングを使用して設定し、使用する前に実行中のインスタンスにマウントしておく必要があります。デフォルトでは、Amazon EBS-backed インスタンスから起動されたインスタンスには、マウントされたインスタンスストアボリュームはありません。instance store-backed AMI から起動されたインスタンスには、仮想マシンのルートデバイスボリューム用にマウントされたインスタンスストアボリュームがあります。また、インスタンスタイプに応じて、他のマウントされたインスタンスストアボリュームを持つこともできます。instance store-backed AMI と Amazon EBS-backed AMI の詳細については、「[ルートデバイスのストレージ \(p. 57\)](#)」を参照してください。

インスタンスストアボリュームは、インスタンスの運用期間中に1つのインスタンスからのみ使用できます。デタッチして他のインスタンスに再アタッチすることはできません。インスタンスから AMI を作成すると、そのインスタンスストアボリューム上のデータは保持されず、この AMI から起動するインスタンスのインスタンスストアボリュームに存在しません。インスタンスストアは特定のインスタンス専用割り当てられますが、ディスクサブシステムは、次の図に示すように、ホストコンピュータの複数のインスタンス間で共有されます。



インスタンスストア上のデータは、関連付けられたインスタンスの運用中のみ維持されます。インスタンスが再ブートされた場合、その再ブートが意図的なものでも、意図せずに行われたとしても、インスタンスストアのデータは維持されます。ただし、次の状況では、インスタンスストアボリュームのデータは失われます。

- 基盤となるドライブで障害が発生した
- Amazon EBS-backed インスタンスを停止した
- インスタンスを終了した

したがって、長期的に使用する重要なデータがある場合は、インスタンスストアボリュームに頼りすぎないようにしてください。代わりに、複数のインスタンス間でレプリケーション方法を使用するか、

データを Amazon S3 に保存するか、または Amazon EBS ボリュームを使用して、データの安全を維持します。詳細については、「[Amazon Elastic Block Store\(Amazon EBS\) \(p. 566\)](#)」を参照してください。

インスタンスを起動する場合は、Amazon EBS-backed AMI または instance store-backed AMI のどちらから起動しても、ブロックデバイスマッピングを使用して、インスタンスストアボリュームをインスタンスにアタッチできます。詳細については、「[AMI にインスタンスストアボリュームを追加する \(p. 639\)](#)」を参照してください。

## インスタンスタイプで使用可能なインスタンスストア

Amazon EC2 インスタンスは、さまざまなインスタンスタイプに分けられます。インスタンスのタイプによって、デフォルトで使用できるインスタンスストアのサイズが決まります。インスタンスタイプは、インスタンスを起動するときに指定できます。また、デフォルトのインスタンスタイプを使用することもできます。デフォルトのインスタンスタイプは、m1.small インスタンスです。

また、インスタンスタイプによって、インスタンスストアボリュームのハードウェアの種類も決まります。インスタンスタイプの中には、ソリッドステートドライブ (SSD) を使用して、非常に高いランダム I/O パフォーマンスを提供するものがあります。待機時間が非常に短いストレージが必要だが、インスタンスの終了時にストレージを保持する必要がない場合、または耐障害性を備えたアーキテクチャを使用できる場合は、このオプションを使用することをお勧めします。詳細については、[H1 インスタンス \(p. 119\)](#) を参照してください。

次の表は、インスタンスタイプと、各インスタンスタイプで使用可能なインスタンスストアボリュームのサイズおよび数量を示しています。これらのインスタンスストアボリュームは、インスタンスの時間単位のコストに含まれます。

インスタンスタイプ	インスタンスストアボリューム
c1.medium	1 x 350 GB†
c1.xlarge	4 x 420 GB ( 1680 GB )
c3.large	2 x 16 GB SSD ( 32 GB )
c3.xlarge	2 x 40 GB SSD ( 80 GB )
c3.2xlarge	2 x 80 GB SSD ( 160 GB )
c3.4xlarge	2 x 160 GB SSD ( 320 GB )
c3.8xlarge	2 x 320 GB SSD ( 640 GB )
cc2.8xlarge	4 x 840 GB ( 3360 GB )
cg1.4xlarge	2 x 840 GB ( 1680 GB )
cr1.8xlarge	2 x 120 GB SSD ( 240 GB )
hi1.4xlarge	2 x 1024 GB SSD ( 2048 GB )
hs1.8xlarge	24 x 2048 GB ( 49 TB )
i2.xlarge	1 x 800 GB SSD
i2.2xlarge	2 x 800 GB SSD ( 1600 GB )
i2.4xlarge	4 x 800 GB SSD ( 3200 GB )

インスタンスタイプ	インスタンスストアボリューム
i2.8xlarge	8 x 800 GB SSD ( 6400 GB )
m1.small	1 x 160 GB†
m1.medium	1 x 410 GB
m1.large	2 x 420 GB ( 840 GB )
m1.xlarge	4 x 420 GB ( 1680 GB )
m2.xlarge	1 x 420 GB
m2.2xlarge	1 x 850 GB
m2.4xlarge	2 x 840 GB ( 1680 GB )
m3.medium	1 x 4 GB SSD
m3.large	1 x 32 GB SSD
m3.xlarge	2 x 40 GB SSD ( 80 GB )
m3.2xlarge	2 x 80 GB SSD ( 160 GB )
r3.large	1 x 32 GB
r3.xlarge	1 x 80 GB
r3.2xlarge	1 x 160 GB
r3.4xlarge	1 x 320 GB
r3.8xlarge	2 x 320 GB ( 640 GB )
t1.micro	なし ( Amazon EBS ボリュームを使用 )

† c1.medium および m1.small インスタンスタイプには、900 MB のインスタンスストアスワップボリュームも含まれます。これは起動時に自動的に有効にされない場合があります。詳細については、「[インスタンスストアスワップボリューム \(p. 634\)](#)」を参照してください。

## インスタンスストアスワップボリューム

Linux のスワップ空間は、システムで物理的に割り当てられたよりも多くのメモリを必要とする場合に使用できます。スワップ空間を有効にすると、Linux システムは頻繁に使用されないメモリページを物理メモリからスワップ空間 ( 既存のファイルシステムの専用パーティションまたはスワップファイル ) にスワップし、高速なアクセスを必要とするメモリページのためにその空間を解放します。

c1.medium および m1.small インスタンスタイプで使用できる物理メモリの量には制限があり、起動時に Linux AMI の仮想メモリとして機能するために 900 MB のスワップボリュームが与えられます。Linux カーネルはこのスワップ空間をルートデバイス上のパーティションと見なしますが、実際にはルートデバイスタイプにかかわらず、別のインスタンスストアボリュームです。

Amazon Linux AMI は自動的にこのスワップ空間を有効にして使用しますが、AMI では、このスワップ空間を認識して使用するために、追加のステップが必要になる場合があります。インスタンスがスワップ空間を使用しているかどうかを確認するには、swapon -s コマンドを使用できます。

```
[ec2-user@ip-12-34-56-78 ~]$ swapon -s
Filename                                Type              Size    Used    Priority
/dev/xvda3                              partition         917500  0       -1
```

上記のインスタンスには、900 MB のスワップボリュームがアタッチされ、有効になっています。このコマンドでスワップボリュームが表示されない場合は、そのデバイスに対してスワップ空間を有効しなければならない可能性があります。利用可能なディスクは、lsblk コマンドを使用して確認します。

```
[ec2-user@ip-12-34-56-78 ~]$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1  0   8G  0 disk /
xvda3 202:3  0 896M  0 disk
```

ここで、スワップボリューム xvda3 はインスタンスで利用できますが、有効になっていません ( MOUNTPOINT フィールドが空です )。スワップボリュームは swapon コマンドを使って有効にできません。



#### Note

lsblk でリストされるデバイス名の先頭に /dev/ を付加する必要があります。デバイスは、sda3、sde3、xvde3 など、異なる名前になる場合があります。システムのデバイス名は、次のコマンドで使用します。

```
[ec2-user@ip-12-34-56-78 ~]$ sudo swapon /dev/xvda3
```

これで、スワップ空間が lsblk および swapon -s 出力に表示されます。

```
[ec2-user@ip-12-34-56-78 ~]$ lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda1 202:1  0   8G  0 disk /
xvda3 202:3  0 896M  0 disk [SWAP]
[ec2-user@ip-12-34-56-78 ~]$ swapon -s
Filename                                Type              Size    Used    Priority
/dev/xvda3                              partition         917500  0       -1
```

また、システムを起動するたびにこのスワップ空間が自動的に有効になるように、/etc/fstab ファイルを編集する必要もあります。

```
[ec2-user@ip-12-34-56-78 ~]$ sudo vim /etc/fstab
```

( システムの swap デバイス名を使用して ) 次の行を /etc/fstab ファイルに追加します。

```
/dev/xvda3          none      swap      sw      0          0
```

## インスタンスストアデバイス名

インスタンスストア内で、インスタンスストアボリュームはブロックデバイスとして表示されます。インスタンスストアボリュームの仮想デバイスは ephemeral[0-3] です。1 つのインスタンスストアボリュームをサポートするインスタンスタイプには、ephemeral0 があります。2 つのインスタンスストアボリュームをサポートするインスタンスタイプには、ephemeral0 と ephemeral1 があります。4

つのインスタンスストアボリュームをサポートするインスタンスタイプには、ephemeral0、ephemeral1、ephemeral2、および ephemeral3 があります。

多くのインスタンスストアボリュームは ext3 ファイルシステムを使用して事前にフォーマットされています。TRIM コマンドをサポートする SSD ベースのインスタンスストアボリュームは、ファイルシステムを使用して事前にフォーマットされていません。ただし、インスタンスを起動してから、選択したファイルシステムでボリュームをフォーマットすることもできます。Windows インスタンスの場合は、組み込みツールの EC2Config Service を使用して、インスタンスで使用できるすべてのインスタンスストアボリュームが NTFS ファイルシステムで再フォーマットします。

すべての instance store-backed AMI およびインスタンスには、インスタンスにアタッチされたインスタンスストアボリュームのマッピングが存在します。マッピングの各エントリが、デバイス名と、そのデバイス名がマッピングされたボリュームで構成されています。インスタンスストアボリュームはインスタンスで使用できますが、ボリュームがマウントされるまでアクセスすることはできません。Windows インスタンスの場合は、EC2Config Service を使用して、インスタンス用にインスタンスストアボリュームがマウントされます。Linux の場合は、インスタンスタイプによって、どのインスタンスストアボリュームがマウントされるのか、またご自分でマウントできるインスタンスが何かが決まります。インスタンスのブロックデバイスドライバは、ボリュームのマウント時に実際のボリューム名を割り当てますが、この割り当てられた名前は、Amazon EC2 が推奨する名前とは異なる可能性があります。

次の表は、インスタンスストアボリューム用にインスタンスタイプによって予約されているデバイス名と、instance store-backed インスタンス上でのストレージデバイスのデフォルトの状態 (formatted、mounted、available) を示しています。例えば、m1.small インスタンスには ephemeral0 ( ext3, 15 GiB ) と swap ( Linux swap、896 MB ) があります。

デバイス名	Linux/Unix Instance store-Backed インスタンス	Windows instance store-backed インスタンス
/dev/sda1, /dev/xvda1, /dev/xvde1	ルート (/) としてフォーマットおよびマウントされます。	C:\ としてフォーマットおよびマウントされます。
/dev/sda2, xvdb	m1.small インスタンスと c1.medium インスタンスで /mnt または /media/ephemeral0 としてフォーマットおよびマウントされます。	スモールインスタンスタイプでフォーマットおよびマウントされます。
/dev/sda3, /dev/xvda3, /dev/xvde3	m1.small および c1.medium インスタンスでスワップ空間として利用できます。詳細については、「 <a href="#">インスタンスストアスワップボリューム (p. 634)</a> 」を参照してください	利用不可。
/dev/sdb, xvdb	m1.medium、m1.large、m1.xlarge、c1.xlarge、cc2.8xlarge、cr1.8xlarge、m2.xlarge、m2.2xlarge、m2.4xlarge、および hi1.4xlarge で /mnt または /media/ephemeral0 としてフォーマットおよびマウントされます。	m1.medium、m1.large、m1.xlarge、c1.xlarge、m2.xlarge、および m2.2xlarge でフォーマットおよびマウントされます。
/dev/sdc, xvdc	m1.large、m1.xlarge、cc2.8xlarge、c1.xlarge、cr1.8xlarge、および hi1.4xlarge 上で利用できます。	m1.large、m1.xlarge、c1.xlarge、および m2.4xlarge でフォーマットおよびマウントされます。
/dev/sdd, xvdd	m1.xlarge、c1.xlarge、および cc2.8xlarge で利用できます。	m1.xlarge および c1.xlarge でフォーマットおよびマウントされます。

デバイス名	Linux/Unix Instance store-Backed インスタンス	Windows instance store-backed インスタンス
/dev/sde、 xvde	m1.xlarge および c1.xlarge で利用できます。	m1.xlarge および c1.xlarge でフォーマットおよびマウントされます。

インスタンスでは、複数のインスタンスストアボリュームをデバイスにマッピングできます。ただし、そのボリュームの数とサイズは、インスタンスタイプで使用できるインスタンスストアを超えてはいけません。詳細については、「[インスタンスタイプで使用可能なインスタンスストア \(p. 633\)](#)」を参照してください。

## インスタンスストアの使用シナリオ

インスタンスストアボリュームは、頻繁に変更される情報（バッファ、キャッシュ、スクラッチデータ、その他の一時コンテンツなど）の一時ストレージに最適です。また、インスタンスのフリート全体でレプリケートされるデータ（負荷分散されたウェブサーバープールなど）にも適しています。

### インスタンスでインスタンスストアを使用できるようにする

ルートデバイスに Amazon EBS を使用するインスタンスは、デフォルトでブート時にインスタンスストアを使用できるようにしません。また、インスタンスを起動したら、インスタンスストアボリュームをアタッチすることはできません。したがって、Amazon EBS-backed インスタンスでインスタンスストアボリュームを使用する必要がある場合は、AMI の作成時またはインスタンスの起動時に、ブロックデバイスマッピングを使用してそのボリュームを指定する必要があります。ブロックデバイスマッピングのエントリの例としては、`/dev/sdb=ephemeral0`、`/dev/sdc=ephemeral1` などがあります。ブロックデバイスマッピングの詳細については、「[ブロックデバイスマッピング \(p. 643\)](#)」を参照してください。

次の手順では、Amazon EBS-backed m1.large Windows インスタンスとインスタンスストアボリュームを起動する方法について説明します。

#### Amazon EBS-Backed Windows インスタンスとインスタンスストアボリュームを起動する

- Amazon EBS-Backed Windows AMI を特定します。
- ephemeral0 と ephemeral1 についてブロックデバイスマッピングのエントリを追加します。  
詳細については、「[インスタンスにボリュームを追加するには \(p. 650\)](#)」を参照してください。
- インスタンスに接続します。
- [スタート] メニューの [Computer] を選択します。
- デバイスが一覧表示されます。
  - ローカルディスク C:/ 9.98 GiB
  - ローカルディスク D:/ 419 GiB
  - ローカルディスク E:/ 419 GiB
- [Local Disk C:/] をダブルクリックします。インストールされているアプリケーションのリストが表示されます。これがルートデバイスです。
- [Local Disk D:/] をダブルクリックしてから、[Local Disk E:/] をダブルクリックします。これらのドライブは空です。これらは m1.large インスタンスを備えたインスタンスストアであり、ご利用のアプリケーションから使用できます。

次の手順では、Amazon EBS-backed m1.large Linux インスタンスとインスタンスストアボリュームを起動する方法について説明します。

## Amazon EBS-backed Linux インスタンス上のインスタンスストアへのアクセス

1. Amazon EBS-Backed Linux/Unix AMI を特定します。
2. `ephemeral0` と `ephemeral1` についてブロックデバイスマッピングのエントリを追加します。  
 詳細については、「[インスタンスにボリュームを追加するには \(p. 650\)](#)」を参照してください。
3. インスタンスに接続します。
4. 現在ディスクにマウントされているインスタンスストアを確認します。
5. `/` に 10 GiB のルートパーティション、`/media/ephemeral0` ボリュームに 420 GiB がそれぞれマウントされていることにご留意ください。`m1.large` インスタンスは 2 つの 420 GiB インスタンスストアボリュームを備えており、2 つ目のボリュームを使用できますが、そのボリュームを使用するには事前にマウントしておく必要があります。
6. もう一方の 420 GiB インスタンスストアボリュームをフォーマットおよびマウントするには
  - a. ご希望のファイルシステムをデバイス `/dev/sdc` に作成します ( `root` 権限が必要です )。
  - b. デバイスのマウント先となるディレクトリを作成します。
  - c. 新しく作成したディレクトリにデバイスをマウントします。
7. デバイスがマウントされていることを確認します。
8. オプションで、ルートデバイスのファイルを一覧表示します。

AMI を作成するときに、インスタンスストアボリュームをブロックデバイスにマッピングすることもできます。このような AMI から起動されたインスタンスでは、起動時にインスタンスストアボリュームが指定されます。AMI 作成中におけるブロックデバイスマッピングの追加については、「[Amazon EBS-Backed Linux AMI の作成 \(p. 77\)](#)」を参照してください。

次の手順では、Amazon EC2 instance store-backed `m1.large` Windows インスタンス内からインスタンスストアボリュームにアクセスする方法について説明します。

## Amazon EC2 instance store-backed Windows インスタンス上のインスタンスストアにアクセスするためのタスク

1	Amazon EC2 instance store-backed Windows AMI を特定します。
2	<code>m1.large</code> インスタンスを起動します。
3	インスタンスに接続します。
4	[Start] メニューの [Computer] を選択します。
5	デバイスが一覧表示されます。 <ul style="list-style-type: none"> <li>• ローカルディスク C:/ 9.98 GiB</li> <li>• ローカルディスク D:/ 419 GiB</li> <li>• ローカルディスク E:/ 419 GiB</li> </ul>
6	ローカルディスク C:/ をダブルクリックします。インストールされているすべてのアプリケーションのリストが表示されます。これがルートデバイスです。
7	ローカルディスク D:/ をダブルクリックし、次にローカルディスク E:/ をダブルクリックします。これらのドライブは空です。これらは <code>m1.large</code> インスタンスを備えたインスタンスストアボリュームであり、他の物理ドライブと同じようにアプリケーションから使用できます。

Amazon EC2 Instance store-Backed Linux および Unix インスタンスのインスタンスストアボリュームの中には、インスタンスタイプによっては、インスタンスの起動時にマウントされないものがあります。例えば、m1.large の Linux および Unix インスタンスのデバイス /dev/sdc については、フォーマット済みで使用可能ですが、使用する前にマウントする必要があります。

次の手順では、Amazon EC2 instance store-backed m1.large Linux インスタンス内からインスタンスストアにアクセスする方法について説明します。

#### Amazon EC2 instance store-backed Linux インスタンス上のインスタンスストアへのアクセス

1	Amazon EC2 instance store-backed Linux/UNIX AMI を特定します。
2	m1.large インスタンスを起動します。
3	インスタンスに接続します。
4	現在ディスク上にマウントされているファイルシステムを確認します。
5	10 GiB ephemeral0 420 GiB m1.large インスタンスは 2 つの 420 GiB インスタンスストアボリュームを備えており、2 つ目のボリュームを使用できますが、そのボリュームを使用するには事前にマウントしておく必要があります。
6	もう一方の420 GiB をマウントするには: a. デバイスのマウント先となるディレクトリを作成します。 b. 新しく作成したディレクトリにデバイスをマウントします。
7	デバイスがマウントされているかどうかを確認します。
8	必要に応じて、ルートデバイス上のファイルを一覧表示します。

## 起動時にインスタンスストアを抑制する

特定のインスタンスストレージボリュームをインスタンスにアタッチしないようにすることができます。この操作は、Amazon EC2 instance store-backed インスタンスと Amazon EBS-backed インスタンスの両方に対して行うことができます。例えば、インスタンスの起動時にマッピング /dev/sdc=none を指定すると、/dev/sdc がインスタンスにアタッチされなくなります。ブロックデバイスマッピングの詳細については、「[ブロックデバイスマッピング \(p. 643\)](#)」を参照してください。

## AMI にインスタンスストアボリュームを追加する

デフォルトでは、Amazon EBS-Backed AMI にはインスタンスストアは含まれません。しかし、Amazon EBS-Backed AMI から起動されるインスタンスに、インスタンスストアボリュームを含めたい場合もあります。

AMI にインスタンスストアボリュームを追加した後は、AMI から起動するすべてのインスタンスにそれらのインスタンスストアボリュームが含まれます。インスタンスストアデバイスが使用できるかどうかは、インスタンスメタデータを使用してインスタンスの内部から確認できます。詳細については、「[インスタンスストアボリュームのインスタンスブロックデバイスマッピングの表示 \(p. 653\)](#)」を参照してください。

M3 インスタンスの場合、インスタンスのブロックデバイスマッピングでインスタンスストアボリュームを指定する必要があります。M3 インスタンスを起動する際には、AMI のブロックデバイスマッピングで指定したインスタンスストアボリュームはすべて無視されます。

コンソールまたはコマンドラインを使用して、インスタンスストアボリュームを含む AMI を作成できます。

コンソールを使用して AMI にインスタンスストアボリュームを追加するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. インスタンスを選択し、[Actions] リストから [Create Image] を選択します。
4. [Create Image] ダイアログで、意味のある名前と説明をイメージに追加します。
5. 各インスタンスストアボリュームについて、[Add New Volume] をクリックし、[Type] リストからインスタンスストアボリューム、[Device] からデバイス名をそれぞれ選択します。
6. [Create Image] をクリックします。

コマンドラインを使用して AMI にインスタンスストアボリュームを追加するには

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

- `create-image` または `register-image` ( AWS CLI )
- `ec2-create-image ec2-register` ( Amazon EC2 CLI )

詳細については、「[ブロックデバイスマッピング \(p. 643\)](#)」を参照してください。

## ディスクパフォーマンスの最適化

Amazon EC2 でディスクを仮想化する方法により、スタンダードインスタンスストアボリュームに対する最初の書き込みは、書き込みの場所にかかわらず、それ以降の書き込みより速度が遅くなりますほとんどのアプリケーションでは、インスタンスの存続期間全体でこのコストを負担することは、許容範囲内です。ただし、高いディスクパフォーマンスを必要とする場合は、本稼働環境で使用する前に、ドライブのすべての場所に一度書き込みを行って、事前ウォームすることをお勧めします。



### Note

I2 ハイ I/O インスタンスタイプは、事前ウォームなしに、起動時に最大のパフォーマンスを提供する、直接アタッチされたソリッドステートドライブを使用します。

レイテンシーやスループットに関してさらに柔軟性が必要な場合は、Amazon EBS を使用することをお勧めします。

インスタンスストレージボリュームを事前ウォームするには、初期化するストア ( `/dev/sdb` など ) に応じて、次の Linux/Unix `dd` コマンドを使用します。



### Note

必ずドライブをアンマウントしてから、このコマンドを実行してください。

初期化には長い時間がかかる場合があります ( エクストララージのインスタンスで約 8 時間 )。

インスタンスストアボリュームを初期化するには、`m1.large`、`m1.xlarge`、`c1.xlarge`、`m2.xlarge`、`m2.2xlarge`、`m2.4xlarge` インスタンスタイプで次のコマンドを使用します。

```
dd if=/dev/zero of=/dev/sdb bs=1M
dd if=/dev/zero of=/dev/sdc bs=1M
dd if=/dev/zero of=/dev/sdd bs=1M
dd if=/dev/zero of=/dev/sde bs=1M
```

各インスタンスタイプで使用可能なインスタンスストレージについては、「[インスタンスタイプで使用可能なインスタンスストア \(p. 633\)](#)」を参照してください。

すべてのインスタンスストアボリュームに対して同時に初期化を実行するには、次のコマンドを使用します。

```
dd if=/dev/zero bs=1M|tee /dev/sdb|tee /dev/sdc|tee /dev/sde > /dev/sdd
```

ドライブをRAID用に構成すると、ドライブのすべての場所に書き込みを行うことで、ドライブが初期化されます。ソフトウェアベースのRAIDを構成するときは、再構築の最低速度を必ず変更してください。

```
echo $((30*1024)) > /proc/sys/dev/raid/speed_limit_min
```

## Amazon Simple Storage Service ( Amazon S3 )

### Abstract

Amazon S3 は、インターネットデータのリポジトリとして使用し、高速かつ低コストで信頼性に優れたデータストレージインフラストラクチャを実現します。

Amazon S3 はインターネットデータ用のリポジトリです。Amazon S3 により、信頼性に優れ、高速で、低コストのデータストレージインフラストラクチャが実現します。ウェブスケールのコンピューティングを簡単に行えるように設計されており、Amazon EC2 の内部やウェブ上のどこからでも、いつでも必要な量だけデータを格納および取得できます。Amazon S3 では、複数の施設にまたがる複数のデバイスにデータオブジェクトが冗長的に格納されるので、多数のさまざまなクライアントやアプリケーションスレッドからデータオブジェクトに対する読み込み/書き込みの同時アクセスが可能になります。Amazon S3 に格納された冗長データを使用すれば、インスタンスまたはアプリケーションの障害から迅速かつ確実に復旧できます。

Amazon EC2 は Amazon マシンイメージ ( AMI ) を格納するために Amazon S3 を使用します。AMI は EC2 インスタンスを起動するために使用します。インスタンスに障害が発生した場合は、格納済みの AMI を使用して別のインスタンスを即座に起動できるので、高速復旧と事業継続が可能になります。

Amazon EC2 はデータボリュームのスナップショット ( バックアップコピー ) の格納にも Amazon S3 を使用します。アプリケーションまたはシステムで障害が発生した場合、スナップショットを使用すれば、データをすばやく確実に回復できます。またスナップショットは、複数の新しいデータボリュームの作成、既存のデータボリュームのサイズ拡張、アベイラビリティゾーン間でのデータボリュームの移動を行うためのベースラインとして使用することもできます。このような機能により、データの使い方を大幅に拡張できます。データボリュームとスナップショットの使用については、「[Amazon Elastic Block Store \(p. 566\)](#)」を参照してください。

オブジェクトは、Amazon S3 に格納される基本エンティティです。Amazon S3 に格納されるすべてのオブジェクトは、バケットに保管されます。バケットは Amazon S3 ネームスペースの最上位レベルを構成し、個々のストレージを所有するアカウントを識別します。Amazon S3 のバケットはインターネットのドメイン名に似ています。バケットに格納されたオブジェクトは一意的なキー値を持ち、HTTP URL アドレスを使用して取得されます。例えば、キー値 `/photos/mygarden.jpg` を持つオブジェクトが `myawsbucket` バケットに格納されている場合、このオブジェクトは URL `http://myawsbucket.s3.amazonaws.com/photos/mygarden.jpg` を使用して、アドレス指定できます。

Amazon S3 の機能について詳しくは、[Amazon S3 の製品ページ](#)を参照してください。

## Amazon S3 と Amazon EC2

Amazon S3 にはストレージとしての利点があるので、このサービスを使用して、EC2 インスタンスで使用するためにファイルおよびデータセットを格納する場合があります。Amazon S3 とインスタンスとの間でデータを移動するには、いくつかの方法があります。以下に説明する例以外にも、コンピュータやインスタンスから Amazon S3 のデータにアクセスできるさまざまなツールが、他のユーザーによって作成されています。一般的な一部のツールについては、AWS フォーラムで取り上げられています。

アクセス許可がある場合は、以下の方法を使用して、Amazon S3 とインスタンスとの間でファイルをコピーできます。

GET または wget

wget コーティリティは、Amazon S3 からパブリックオブジェクトをダウンロードできる HTTP および FTP のクライアントです。これは、Amazon Linux やその他のほとんどのディストリビューションにデフォルトでインストールされており、Windows でも使用できます。Amazon S3 オブジェクトをダウンロードするには、次のコマンドを入力し、ダウンロードするオブジェクトの URL に置き換えます。

```
wget http://s3.amazonaws.com/my_bucket/my_folder/my_file.ext
```

この方法では、要求するオブジェクトがパブリックである必要があります。オブジェクトがパブリックではない場合、[ERROR 403: Forbidden] メッセージを受け取ります。このエラーを受け取った場合は、Amazon S3 コンソールを開き、オブジェクトのアクセス許可をパブリックに変更します。詳細については、[Amazon Simple Storage Service 開発者ガイド](#) を参照してください。

AWS コマンドラインインターフェイス

AWS コマンドラインインターフェイス (AWS CLI) は、AWS サービスを管理するための統合ツールです。ダウンロードおよび設定用の単一のツールのみを使用して、コマンドラインから複数の AWS サービスを制御し、スクリプトを使用してこれらを自動化することができます。AWS CLI を使用すると、ユーザーは自分自身を認証し、限定された項目を Amazon S3 からダウンロードしたり、項目をアップロードしたりできます。ツールのインストールおよび設定方法などの詳細については、[AWS コマンドラインインターフェイスの詳細ページ](#) を参照してください。

aws s3 cp コマンドは、Unix cp コマンドに似ています (構文は aws s3 cp *source destination*)。ファイルを Amazon S3 からインスタンスにコピーしたり、ファイルをインスタンスから Amazon S3 にコピーしたりできるほか、ファイルを Amazon S3 の 1 つの場所から別の場所にコピーすることもできます。

オブジェクトを Amazon S3 からインスタンスにコピーするには、次のコマンドを使用します。

```
aws s3 cp s3://my_bucket/my_folder/my_file.ext my_copied_file.ext
```

オブジェクトをインスタンスから Amazon S3 にコピーして戻すには、次のコマンドを使用します。

```
aws s3 cp my_copied_file.ext s3://my_bucket/my_folder/my_file.ext
```

オブジェクトを Amazon S3 の 1 つの場所から別の場所にコピーするには、次のコマンドを使用します。

```
aws s3 cp s3://my_bucket/my_folder/my_file.ext s3://my_bucket/my_folder/my_file2.ext
```

aws s3 sync コマンドは、Amazon S3 バケット全体をローカルディレクトリの場所に同期できます。この機能は、データセットをダウンロードし、リモートセットでローカルコピーを最新の状態に保つ際に役立ちます。コマンドの構文は aws s3 sync *source destination* です。Amazon S3 バケットに対して適切なアクセス許可がある場合は、コマンドで送信元と送信先の場所を入れ替えることで、終了時にローカルディレクトリバックアップをクラウドにプッシュできます。

Amazon S3 バケット全体をインスタンスのローカルディレクトリにダウンロードするには、次のコマンドを使用します。

```
aws s3 sync s3://remote_S3_bucket local_directory
```

#### AWS Tools for Windows PowerShell

Windows インスタンスには、Amazon S3 コンソールに直接アクセスするために使用できるグラフィカルブラウザであるという利点があります。ただし、Windows ユーザーは、スクリプティング目的で [AWS Tools for Windows PowerShell](#) を使用して、Amazon S3 との間でオブジェクトを移動することもできます。

Amazon S3 オブジェクトを Windows インスタンスにコピーするには、次のコマンドを使用します。

```
Copy-S3Object -BucketName my_bucket -Key my_folder/my_file.ext -LocalFile my_copied_file.ext
```

#### Amazon S3 API

開発者は API を使用して、Amazon S3 のデータにアクセスできます。詳細については、[Amazon Simple Storage Service 開発者ガイド](#) を参照してください。この API およびその例を使用すると、アプリケーションを開発し、boto Python インターフェイスなどのその他の API および SDK と統合するのに役立ちます。

## ブロックデバイスマッピング

### Abstract

ブロックデバイスマッピングを使用して、インスタンスの起動時にそのインスタンスに対して追加の EBS ボリュームまたはインスタンスストアボリュームを指定します。

起動する各 Amazon EC2 インスタンスには、Amazon Elastic Block Store ( EBS ) ボリュームまたはインスタンスストアボリュームという、どちらかのルートデバイスボリュームが関連付けられています。ブロックデバイスマッピングを使用すると、インスタンスの起動時にそのインスタンスにアタッチする追加の EBS ボリュームまたはインスタンスストアボリュームを指定できます。追加する EBS ボリュームは、実行中のインスタンスにアタッチすることもできます。「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p.573\)](#)」をご参照ください。ただし、インスタンスストアボリュームについては、ブロックデバイスマッピングを使用して、インスタンスの起動時にアタッチする以外方法はありません。

ルートデバイスボリュームの詳細については、「[永続的ルートデバイスボリュームへの変更 \(p. 18\)](#)」を参照してください。

### Topics

- [ブロックデバイスマッピングの概念 \(p. 644\)](#)
- [AMI ブロックデバイスマッピング \(p. 647\)](#)
- [インスタンスブロックデバイスマッピング \(p. 650\)](#)

## ブロックデバイスマッピングの概念

ブロックデバイスは、一連のバイトまたはビット（ブロック）でデータを移動するストレージデバイスです。これらのデバイスはランダムアクセスをサポートし、通常は、バッファされた I/O を使用します。例えば、ハードディスク、CD-ROM ドライブ、フラッシュドライブなどがブロックデバイスに含まれます。ブロックデバイスは物理的にコンピュータにアタッチできます。また、コンピュータに物理的にアタッチされているかのように、リモートでアクセスすることもできます。Amazon EC2 は、2 種類のブロックデバイスをサポートしています。

- インスタンスストアボリューム（基盤となるハードウェアがインスタンスのホストコンピュータに物理的にアタッチされている仮想デバイス）
- Amazon EBS ボリューム（リモートストレージデバイス）

ブロックデバイスマッピングは、Amazon EC2 インスタンスにアタッチするブロックデバイス、および使用するデバイス名を定義します。ブロックデバイスマッピングは、AMI 作成プロセスの一環として、AMI から起動されるすべてのインスタンスによって使用されるように指定できます。また、インスタンスの起動時にブロックデバイスマッピングを指定することもできます。起動したインスタンスの AMI ですでに指定されているマッピングは、このマッピングによって上書きされます。

Amazon EC2 で使用できる仮想化のタイプには、準仮想化とハードウェア仮想マシン（HVM）の 2 種類があります。仮想化タイプは、インスタンスの起動に使用される AMI によって決まります。インスタンスタイプには、準仮想化と HVM の両方をサポートしているものもあれば、どちらか一方のみをサポートしているものもあります。使用できる有効な推奨デバイス名はインスタンスの仮想化タイプによって異なるので、ブロックデバイスマッピングを作成するときは、AMI で使用される仮想化タイプに留意してください。

## ブロックデバイスマッピングの指定

ブロックデバイスマッピングを使用して、インスタンスストアボリュームと EBS ボリュームを Amazon EC2 インスタンスにアタッチします。

ブロックデバイスマッピングを作成するとき、インスタンスにアタッチする必要があるブロックデバイスごとにこの情報を指定します。

- この表に示されている Amazon EC2 内のデバイス名。インスタンスのブロックデバイスドライバは、ボリュームのマウント時に実際のボリューム名を割り当てますが、この割り当てられた名前は、Amazon EC2 が推奨する名前とは異なる可能性があります。

インスタンスタイプ	接続先として可能	ルート用に予約済み	インスタンスストアボリューム	EBS 接続に推奨されるもの
Linux/Unix（準仮想）	/dev/sd[a-z]  /dev/sd[a-z][1-15]  /dev/hd[a-z]  /dev/hd[a-z][1-15]	/dev/sda1	/dev/sd[b-e]	/dev/sd[f-p]  /dev/sd[f-p][1-6]
Linux/Unix（HVM）	/dev/sd[a-z]  /dev/xvd[b-c][a-z]	AMI による違い  /dev/sda1 or /dev/xvda	/dev/sd[b-e]  /dev/sd[b-y] (hs1.8xlarge インスタンスの場合)	/dev/sd[f-p]

インスタンスタイプ	接続先として可能	ルート用に予約済み	インスタンスストアボリューム	EBS 接続に推奨されるもの
Windows ( HVM )	xvd[a-z]  xvd[b-c][a-z]  /dev/sda[1-2]  /dev/sd[b-e]	/dev/sda1	xvd[a-e]  xvdc[a-x] ( hs1.8xlarge イ ンスタンスの場 合 )	xvd[f-p] ( Red Hat PV ドライバを含 む )  xvd[f-z] ( Citrix PV ドライバを含む )



### Important

デバイスによっては、ブロックデバイスマッピングで指定した名前が、デフォルトのブロックデバイス名と競合することがあります。この問題を回避するには、`/dev/sda[2-15]` の形式名を使用しないでください。

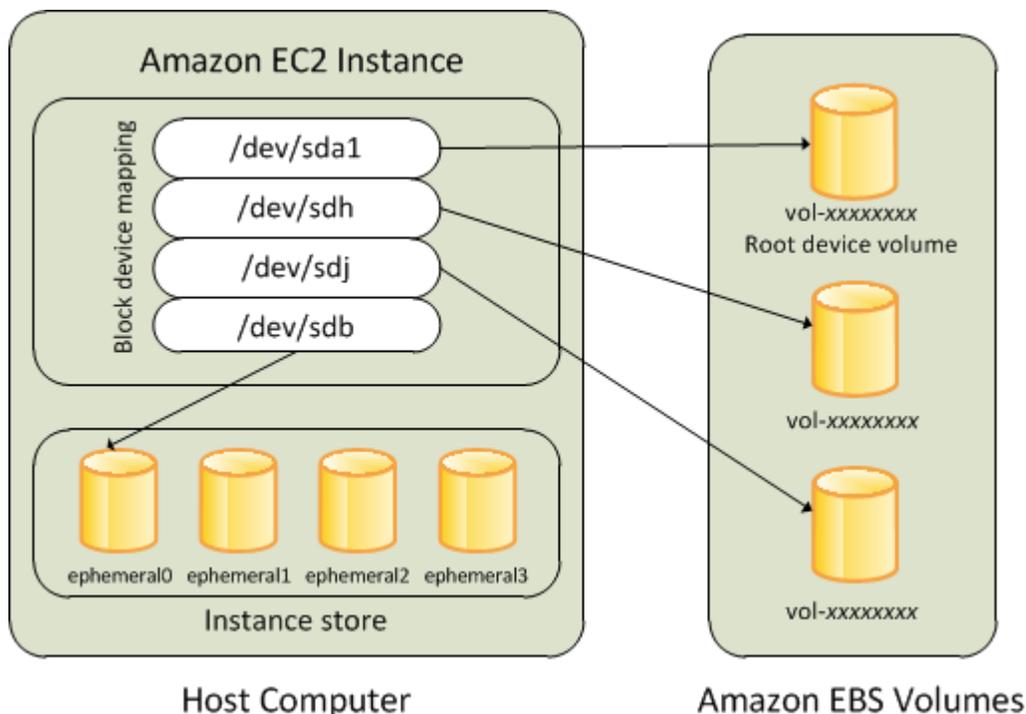
ハードウェア仮想マシン ( HVM ) AMI ( ベース Windows およびクラスターコンピューティングイメージなど ) は、デバイス名で末尾の番号の使用をサポートしていません ( `xvd[a-p][1-15]` )。インスタンス詳細ページの [Virtualization] フィールドで、インスタンスの仮想化の種類を表示できます。

インスタンスのカーネルのブロックデバイスドライバに応じて、デバイスは指定した名前とは異なる名前でもアタッチされる可能性があります。例えば、`/dev/sdh` のデバイス名を指定すると、デバイスはカーネルによって `/dev/xvdh` または `/dev/hdh` に名前が変更されます。一部のケースでは、末尾の番号が変更される可能性もあります ( `/dev/sda` が `/dev/xvde` になる可能性があります )。Amazon Linux AMI は、名前が変更されたデバイスパスから、指定した名前へのシンボリックリンクを作成しますが、その他の AMI では、異なる動作をする場合があります。

- [インスタンスストアボリュームのみ] 仮想デバイス: `ephemeral[0-3]`。
- [EBS ボリュームのみ] ブロックデバイスを作成するとき使用するスナップショットの ID ( `snap-xxxxxxx` )。ボリュームサイズを指定する場合、この値はオプションです。
- [EBS ボリュームのみ] ボリュームのサイズ ( GiB 単位 )。指定されたサイズは、指定されたスナップショットのサイズ以上である必要があります。
- [EBS ボリュームのみ] インスタンス終了時にボリュームを削除するかどうか ( `true` または `false` )。デフォルト値は `true` です。
- [EBS ボリュームのみ] ( 標準の EBS ボリューム ) と `io1` ( プロビジョンド IOPS ボリューム ) のどちらかのボリュームタイプ。デフォルト値は `standard` です。
- [EBS ボリュームのみ] ボリュームがサポートする 1 秒あたりの入力/出力オペレーションの数 ( IOPS ) ( `standard` ボリュームでは使用されません )。

## ブロックデバイスマッピングの例

この図は、Amazon EBS-Backed インスタンスのブロックデバイスマッピングの例を示しています。この例では、`/dev/sdb` を `ephemeral0` にマッピングし、2 つの EBS ボリュームを 1 つは `/dev/sdh` に、もう 1 つは `/dev/sdj` にマッピングします。また、ルートデバイスボリュームである EBS ボリューム、`/dev/sda1` も示しています。



このブロックデバイスマッピングの例は、このトピックのコマンドおよび API の例で使用されています。ブロックデバイスマッピングを作成するコマンドおよび API の例については、[ここ](#)を参照してください。

- [AMI 用のブロックデバイスマッピングの指定 \(p. 647\)](#)
- [インスタンスの起動時にブロックデバイスマッピングを更新する \(p. 650\)](#)

## オペレーティングシステムでデバイスを使用できるようにする方法

Amazon EC2 では、ブロックデバイスの記述に、`/dev/sdh` や `xvdh` などのデバイス名が使われます。また、Amazon EC2 では、Amazon EC2 インスタンスにアタッチするブロックデバイスを、ブロックデバイスマッピングで指定します。ストレージデバイスにアクセスするには、インスタンスにアタッチしたブロックデバイスが、オペレーティングシステムによって事前にマウントされていなければなりません。ブロックデバイスがインスタンスからデタッチされると、そのデバイスはオペレーティングシステムによってアンマウントされ、ストレージデバイスにアクセスできなくなります。

Linux インスタンスの場合、ブロックデバイスマッピングで指定されたデバイス名は、インスタンスの初回起動時に対応するブロックデバイスにマッピングされます。デフォルトでフォーマットおよびマウントされるインスタンスストアボリュームは、インスタンスタイプによって決まります。インスタンスタイプで利用できるインスタンスストアボリューム数を超えていない場合は、起動時に追加のインスタンスストアボリュームをマウントできます。詳細については、「[Amazon EC2 インスタンスストア \(p. 631\)](#)」を参照してください。ボリュームがフォーマットおよびマウントされるときに使用されるデバイスは、インスタンスのブロックデバイスドライバによって決まります。詳細については、「[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#)」を参照してください。

Windows インスタンスの場合、ブロックデバイスマッピングで指定されたデバイス名は、インスタンスの初回起動時に対応するブロックデバイスにマッピングされ、Ec2Config サービスによってドライブが初期化されマウントされます。ルートデバイスボリュームは、`C:\` としてマウントされます。インスタンスストアボリュームは、`D:\`、`E:\` などとしてマウントされます。EBS ボリュームについては、使用

可能な任意のドライブ文字を使用してマウントできます。ただし、Ec2Config がドライブ文字を EBS ボリュームに割り当てる方法を構成することができます。詳細については、[Using EC2Config Service](#) を参照してください。

## ブロックデバイスマッピングの表示

ブロックデバイスマッピングの各ブロックデバイスの情報を表示できます。詳細については、以下を参照してください。

- [AMI ブロックデバイスマッピングの EBS ボリュームの表示 \(p. 649\)](#)
- [インスタンスブロックデバイスマッピングの EBS ボリュームの表示 \(p. 652\)](#)
- [インスタンスストアボリュームのインスタンスブロックデバイスマッピングの表示 \(p. 653\)](#)

## AMI ブロックデバイスマッピング

各 AMI にブロックデバイスマッピングがあります。このブロックデバイスマッピングは、AMI からのインスタンスの起動時にそのインスタンスにアタッチするブロックデバイスを指定します。Amazon が提供する AMI には、ルートデバイスのみが含まれます。追加のブロックデバイスを AMI に追加するには、独自の AMI を作成する必要があります。

### Topics

- [AMI 用のブロックデバイスマッピングの指定 \(p. 647\)](#)
- [AMI ブロックデバイスマッピングの EBS ボリュームの表示 \(p. 649\)](#)

## AMI 用のブロックデバイスマッピングの指定

AMI を作成する場合に、ルートボリュームに加えて、ボリュームを指定するには、2つの方法があります。インスタンスから AMI を作成する前に、実行中のインスタンスにすでにボリュームをアタッチしている場合、AMI のブロックデバイスマッピングにそれらの同じボリュームが含まれます。EBS ボリュームの場合、既存のデータが新しいスナップショットに保存され、それがブロックデバイスマッピングで指定される新しいスナップショットになります。インスタンスストアボリュームの場合、データは維持されません。

EBS-backed AMI の場合、ブロックデバイスマッピングを使用して、EBS ボリュームとインスタンスストアボリュームを追加できます。instance store-backed AMI の場合、ブロックデバイスマッピングでは、インスタンスストアボリュームしか追加できません。

M3 インスタンスの場合、インスタンスのブロックデバイスマッピングでインスタンスストアボリュームを指定する必要があります。M3 インスタンスを起動する際には、AMI のブロックデバイスマッピングで指定したインスタンスストアボリュームはすべて無視されます。

コンソールを使用してボリュームを AMI に追加するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. インスタンスを選択し、[Actions] リストから [Create Image] を選択します。
4. [Create Image] ダイアログボックスで、[Add New Volume] をクリックします。
5. [Type] リストからボリュームタイプを選択し、[Device] リストからデバイス名を選択します。EBS ボリュームの場合、オプションで、スナップショット、ボリュームサイズ、および EBS ボリュームタイプを指定できます。
6. [Create Image] をクリックします。

AWS CLI を使用してボリュームを AMI に追加するには

Amazon EBS-Backed AMI のブロックデバイスマッピングを指定するには、[create-image](#) コマンドを使用します。Instance Store-Backed AMI のブロックデバイスマッピングを指定するには、[register-image](#) コマンドを使用します。

次のパラメータを使用してブロックデバイスマッピングを指定します。

```
--block-device-mappings [mapping, ...]
```

インスタンスストアボリュームを追加するには、次のマッピングを使用します。

```
{  
  "DeviceName": "/dev/sdf",  
  "VirtualName": "ephemeral0"  
}
```

空の 100 GiB EBS ボリュームを追加するには、次のマッピングを使用します。

```
{  
  "DeviceName": "/dev/sdg",  
  "Ebs": {  
    "VolumeSize": 100  
  }  
}
```

スナップショットに基づいた EBS ボリュームを追加するには、次のマッピングを使用します。

```
{  
  "DeviceName": "/dev/sdh",  
  "Ebs": {  
    "SnapshotId": "snap-xxxxxxxx"  
  }  
}
```

デバイスマッピングを省略するには、次のマッピングを使用します。

```
{  
  "DeviceName": "/dev/sdj",  
  "NoDevice": ""  
}
```

Amazon EC2 CLI を使用してボリュームを AMI に追加するには

Amazon EBS-Backed AMI のブロックデバイスマッピングを指定するには、[ec2-create-image](#) コマンドを使用します。Instance Store-Backed AMI のブロックデバイスマッピングを指定するには、[ec2-register](#) コマンドを使用します。

次のパラメータを使用してブロックデバイスマッピングを指定します。

```
-b "devicename=blockdevice"
```

*devicename*

Amazon EC2 内のデバイス名

#### *blockdevice*

AMI からデバイスマッピングを省略するには、`none` を指定します。

インスタンスストアボリュームを追加するには、`ephemeral[0..3]` を指定します。

EBS-Backed インスタンスに EBS ボリュームを追加するには、`[snapshot-id]:[size]:[delete-on-termination]:[type]:[iops]` を指定します。

- 空の EBS ボリュームを追加するには、スナップショット ID を省略し、代わりにボリュームサイズを指定します。
- 終了時に EBS ボリュームを削除するかどうかを指定するには、`true` または `false` を指定します。デフォルト値は `true` です。
- プロビジョニングされた IOPS ボリュームを作成するには、`io1` を指定します。デフォルトのタイプは `standard` です。タイプが `io1` の場合、ボリュームがサポートする IOPS 数もプロビジョニングできます。

1 つのコマンドで複数の `-b` パラメータを使用して、複数のブロックデバイスを指定できます。たとえば次のパラメータを使用すると、インスタンスストアボリュームが `/dev/sdb` として追加され、スナップショットに基づく EBS ボリュームが `/dev/sdh` として追加され、空の 100 GiB EBS ボリュームが `/dev/sdj` として追加されます。

```
-b "/dev/sdb=ephemeral0" -b "/dev/sdh=snap-d5eb27ab" -b "/dev/sdj=:100"
```

## AMI ブロックデバイスマッピングの EBS ボリュームの表示

AMI のブロックデバイスマッピングの EBS ボリュームを簡単に列挙できます。

コンソールを使用して AMI の EBS ボリュームを表示するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [AMIs] をクリックします。
3. [Filter] ドロップダウンリストから [EBS Images] を選択して、EBS-Backed AMI のリストを取得します。
4. ご希望の AMI を選択し、[Details] タブを確認します。少なくとも、ルートデバイスについては次の情報を使用できます。
  - ルートデバイスタイプ ( `ebs` )
  - [Root Device] ( `/dev/sda1` など )
  - [Block Devices] ( `/dev/sda1=snap-e1eb279f:8:true` など )

AMI がブロックデバイスマッピングを使用して追加の EBS ボリュームで作成された場合、[Block Devices] フィールドには、その追加の EBS ボリュームのマッピングも表示されます ( この画面には、インスタンスストアボリュームが表示されないことに注意してください )。

AWS CLI を使用して AMI の EBS ボリュームを表示するには

`describe-images` コマンドを使用して、AMI のブロックデバイスマッピング内の EBS ボリュームを列挙します。

Amazon EC2 CLI を使用して AMI の EBS ボリュームを表示するには

`ec2-describe-images` コマンドを使用して、AMI のブロックデバイスマッピング内の EBS ボリュームを列挙します。

## インスタンスブロックデバイスマッピング

デフォルトでは、起動するインスタンスには、そのインスタンスを起動した AMI のブロックデバイスマッピングで指定されたストレージデバイスが含まれます。インスタンスを起動するときに、インスタンスのブロックデバイスマッピングへの変更を指定できます。この変更は AMI のブロックデバイスマッピングを上書きするか、このブロックデバイスマッピングに統合されます。ただし、ルートデバイスボリュームのブロックデバイスマッピングエントリを変更することはできません。

### Topics

- [インスタンスの起動時にブロックデバイスマッピングを更新する \(p. 650\)](#)
- [インスタンスブロックデバイスマッピングの EBS ボリュームの表示 \(p. 652\)](#)
- [インスタンスストアボリュームのインスタンスブロックデバイスマッピングの表示 \(p. 653\)](#)

## インスタンスの起動時にブロックデバイスマッピングを更新する

インスタンスの起動時に、EBS ボリュームとインスタンスストアボリュームをインスタンスに追加できます。インスタンスのブロックデバイスマッピングを更新しても、そのインスタンスが起動された AMI のブロックデバイスマッピングは完全には変更されないことに注意してください。

コンソールを使用してボリュームをインスタンスに追加するには

1. Amazon EC2 コンソールを開きます。
2. Amazon EC2 コンソールダッシュボードで、[Launch Instance] をクリックします。
3. [Choose an Amazon Machine Image (AMI)] ページで、使用する AMI を選択し、[Select] をクリックします。
4. ウィザードにしたがって [Choose an Instance Type] ページと [Configure Instance Details] ページを設定します。
5. [Add Storage] ページで、以下のようにルートボリューム、EBS ボリューム、およびインスタンスストアボリュームを変更できます。
  - ルートボリュームのサイズを変更するには、[Type] 列で [Root] ボリュームを見つけて、[Size] フィールドを変更します。
  - インスタンスの起動に使用された AMI のブロックデバイスマッピングで指定された EBS ボリュームを削除するには、ボリュームを見つけて、[Delete] アイコンをクリックします。
  - EBS ボリュームを追加するには、[Add New Volume] をクリックし、[Type] リストから [EBS] を選択して、フィールド ([Device]、[Snapshot] など) に入力します。
  - インスタンスの起動に使用された AMI のブロックデバイスマッピングで指定されたインスタンスストアボリュームを削除するには、ボリュームを見つけて、[Delete] アイコンをクリックします。
  - インスタンスストアボリュームを追加するには、[Add New Volume] をクリックし、[Type] リストから [Instance Store] を選択して、[Device] からデバイス名を選択します。
6. ウィザードの残りのページを完了した後、[Launch] をクリックします。

AWS CLI を使用してボリュームをインスタンスに追加するには

`run-instances` コマンドを使用して、インスタンスのブロックデバイスマッピングを指定します。

次のパラメータを使用してブロックデバイスマッピングを指定します。

```
--block-device-mappings [mapping, ...]
```

例えば、EBS-backed Linux AMI が、次のブロックデバイスマッピングを指定するとします。

- /dev/sdb=ephemeral0
- /dev/sdh=snap-92d333fb
- /dev/sdj=:100

この AMI から起動したインスタンスに /dev/sdj がアタッチされないようにするには、次のマッピングを使用します。

```
{  
  "DeviceName": "/dev/sdj",  
  "NoDevice": ""  
}
```

/dev/sdh の サイズを 300 GiB に増やすには、次のマッピングを指定します。デバイス名を指定することでボリュームを特定できるので、/dev/sdh のスナップショット ID を指定する必要がないことに注意してください。

```
{  
  "DeviceName": "/dev/sdh",  
  "Ebs": {  
    "VolumeSize": 300  
  }  
}
```

追加インスタンスストアボリューム /dev/sdc をアタッチするには、次のマッピングを指定します。インスタンスタイプが複数のインスタンスストアボリュームをサポートしていない場合、このマッピングは効果がありません。

```
{  
  "DeviceName": "/dev/sdc",  
  "VirtualName": "ephemeral1"  
}
```

Amazon EC2 CLI を使用してボリュームをインスタンスに追加するには

[ec2-run-instances](#) コマンドを使用して、インスタンスのブロックデバイスマッピングを指定します。

次のパラメータを使用してブロックデバイスマッピングを指定します。

```
-b "devicename=blockdevice"
```

*devicename*

Amazon EC2 内のデバイス名

*blockdevice*

AMI からデバイスのマッピングを省略するには、none を指定します。

インスタンスストアボリュームを追加するには、ephemeral[0..3] を指定します。

EBS-Backed インスタンスに EBS ボリュームを追加するには、  
[*snapshot-id*]:[*size*]:[*delete-on-termination*]:[*type*:[*iops*]] を指定します。

- 空の EBS ボリュームを追加するには、スナップショット ID を省略し、代わりにボリュームサイズを指定します。
- 終了時に EBS ボリュームを削除するかどうかを指定するには、`true` または `false` を指定します。デフォルト値は `true` です。
- プロビジョニングされた IOPS ボリュームを作成するには、`io1` を指定します。デフォルトのタイプは `standard` です。タイプが `io1` の場合、ボリュームがサポートする IOPS 数もプロビジョニングできます。

例えば、EBS-backed Linux AMI が、次のブロックデバイスマッピングを指定するとします。

- `/dev/sdb=ephemeral0`
- `/dev/sdh=snap-92d333fb`
- `/dev/sdj=:100`

この AMI から起動したインスタンスに `/dev/sdj` がアタッチされないようにするには、次のオプションを使用します。

```
-b "/dev/sdj=none"
```

`/dev/sdh` のサイズを 300 GiB に増やすには、次のオプションを使用します。

```
-b "/dev/sdh=:300"
```

デバイス名を指定することでボリュームを特定できるので、`/dev/sdh` のスナップショット ID を指定する必要がないことに注意してください。

追加インスタンスストアボリューム `/dev/sdc` をアタッチするには、次のオプションを使用します。インスタンスタイプが複数のインスタンスストアボリュームをサポートしていない場合、このオプションは効果がありません。

```
-b "/dev/sdc=ephemeral1"
```

## インスタンスブロックデバイスマッピングの EBS ボリュームの表示

インスタンスにマッピングされた EBS ボリュームを簡単に列挙できます。



### Note

2009 年 10 月 31 日 API のリリース以前に起動されたインスタンスについては、AWS では、ブロックデバイスマッピングを表示できません。AWS がブロックデバイスマッピングを表示できるようにするには、ボリュームをデタッチしてから再アタッチする必要があります。

コンソールを使用してインスタンスの EBS ボリュームを表示するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックします。
3. [Filter] リストから、[EBS root device] を選択します。これにより、Amazon EBS-Backed インスタンスのリストが表示されます。
4. 目的のインスタンスを特定してクリックし、[Description] タブに表示された詳細を確認します。少なくとも、ルートデバイスでは次の情報を使用できます。

- ルートデバイスタイプ ( `ebs` )
- ルートデバイス ( `sda1` など )
- ブロックデバイス ( `sda1`、`sdh`、`sdj` など )

インスタンスがブロックデバイスマッピングを使用して追加の EBS ボリュームで起動された場合、[Block Devices] ボックスには、その追加の EBS ボリュームと、ルートデバイスが表示されます。(このダイアログボックスには、インスタンスストアボリュームが表示されないことに注意してください)。

<b>Root device type</b>	<code>ebs</code>
<b>Root device</b>	<code>/dev/sda1</code>
<b>Block devices</b>	<code>/dev/sda1</code> <code>/dev/sdf</code>

5. ブロックデバイスの追加情報を表示するには、[Block devices] の横でそのエントリをクリックします。これにより、ブロックデバイスに関する次の情報が表示されます。
  - [EBS ID] ( `vol-xxxxxxx` )
  - ルートデバイスタイプ ( `ebs` )
  - [Attachment time] ( `yyyy-mmTthh:mm:ss.TZD` )
  - ブロックデバイスステータス ( `attaching`、`attached`、`detaching`、`detached` )
  - 終了時に削除 ( `Yes`、`No` )

AWS CLI を使用してインスタンスの EBS ボリュームを表示するには

`describe-instances` コマンドを使用して、インスタンスのブロックデバイスマッピング内の EBS ボリュームを列挙します。

Amazon EC2 CLI を使用してインスタンスの EBS ボリュームを表示するには

`ec2-describe-instances` コマンドを使用して、インスタンスのブロックデバイスマッピング内の EBS ボリュームを列挙します。

## インスタンスストアボリュームのインスタンスブロックデバイスマッピングの表示

インスタンスのブロックデバイスマッピングを表示した場合、EBS ボリュームのみが表示され、インスタンスストアボリュームは表示されません。インスタンスメタデータを使用すると、完全なブロックデバイスマッピングに対してクエリを実行できます。インスタンスメタデータのすべてのリクエストの基本 URI は `http://169.254.169.254/latest/` です。

まず、実行中にインスタンスに接続します。インスタンスが Linux を実行している場合、GET コマンドはすでに使用できます。インスタンスが Windows を使用している場合、`wget` をインスタンスにインストールし、以下に示すように GET を `wget` に置き換えます。

実行中のインスタンスでこのクエリを使用し、そのブロックデバイスマッピングを取得します。

```
GET http://169.254.169.254/latest/meta-data/block-device-mapping/
```

レスポンスには、インスタンスのブロックデバイスの名前が含まれます。例えば、`instance store-Backed m1.small` インスタンスの出力は次のようになります。

```
ami
ephemeral0
root
swap
```

ami デバイスは、インスタンスによって判断されるルートデバイスです。インスタンスストアボリュームの名前は ephemeral[0-3] です。swap デバイスはページファイル用です。EBS ボリュームもマップした場合、そのボリュームは、ebs1、ebs2 のように表示されます。

ブロックデバイスマッピングの個別のブロックデバイスの詳細を確認するには、ここで示すように、前のクエリにブロックデバイスの名前を追加します。

```
GET http://169.254.169.254/latest/meta-data/block-device-mapping/ephemeral0
```

詳細については、「[インスタンスメタデータとユーザーデータ \(p. 289\)](#)」を参照してください。

# パブリックデータセットの使用

## Abstract

パブリックデータセットを AWS クラウドベースのアプリケーションにシームレスに統合します。

### Topics

- [パブリックデータセットの概念 \(p. 655\)](#)
- [パブリックデータセットの検索 \(p. 655\)](#)
- [スナップショットからパブリックデータセットボリュームを作成する \(p. 656\)](#)
- [パブリックデータセットボリュームをアタッチしてマウントする \(p. 657\)](#)

このセクションでは、EC2 でパブリックデータセットを使用する方法について説明します。

## パブリックデータセットの概念

Amazon Web Services は、AWS クラウドベースのアプリケーションとシームレスに統合可能な、パブリックデータセットのリポジトリを提供します。Amazon が保管するこのデータセットは、コミュニティが無料で利用できます。AWS のすべてのサービスがそうであるように、お支払いいただくのはお客様のアプリケーションのために使用したコンピューティングおよびストレージの分だけです。

ヒト遺伝子のマッピングや、米国情勢調査データなどの膨大なデータセットでは、その位置特定、ダウンロード、カスタマイズ、分析のために、従来は何時間も、また何日もかかっていました。現在では、Amazon EC2 インスタンスから誰もがこれらのデータセットにアクセスし、数分以内にデータの計算を開始することができます。また、AWS エコシステム全体を活用し、他の AWS ユーザーと簡単に共同作業を行うことができます。例えば、ツールやアプリケーションを用いて、あらかじめ構築されたサーバー画像を作成または使用し、データセットの分析を行うことができます。Amazon EC2 のような費用対効果の高いサービスで、こうした重要で有益なデータをホスティングすることにより、AWS は様々な分野や業界にまたがる研究者に、さらなる革新性と高速性を可能にするツールを提供したいと考えています。

詳細については、[Public Data Sets on AWS Page](#) ページを参照してください。

## 使用可能なパブリックデータセット

パブリックデータセットは現在、以下のカテゴリで使用できます。

- 生物学 – ヒトゲノム計画 (GenBank) およびその他のコンテンツが含まれています。
- 化学 – 複数バージョンの PubChem およびその他のコンテンツが含まれています。
- 経済学 – 国勢調査データ、労働統計、運輸統計、およびその他のコンテンツが含まれています。
- 百科事典 – 複数のソースからの Wikipedia コンテンツおよびその他のコンテンツが含まれています。

## パブリックデータセットの検索

パブリックデータセットを使用するには、まずデータセットを見つけて、データセットがホスティングされた形式を判断する必要があります。データセットは、Amazon EBS スナップショットまたは Amazon S3 バケットの 2 種類の有効な形式で使用できます。

パブリックデータセットを見つけてその形式を判断するには

1. [Public Data Sets](#) ページに移動して、使用可能なすべてのパブリックデータセットのリストを表示します。また、このページで検索フレーズを入力して、使用可能なパブリックデータセットのリストに対してクエリを実行することもできます。

2. データセットの名前をクリックして、その詳細ページを表示します。
3. データセットの詳細ページでスナップショット ID リストを探して、Amazon EBS フォーマットデータセットまたは Amazon S3 URL を特定します。

Amazon EBS スナップショット形式のデータセットは、Amazon EC2 インスタンスにアタッチする新しい Amazon EBS ボリュームを作成するために使用されます。詳細については、[スナップショットからパブリックデータセットボリュームを作成する \(p. 656\)](#) を参照してください。

Amazon S3 形式のデータセットの場合は、AWS SDK または HTTP クエリ API を使用して情報にアクセスするか、または AWS CLI を使用してインスタンスとの間でデータをコピーまたは同期することができます。詳細については、[Amazon S3 と Amazon EC2 \(p. 642\)](#) を参照してください。

また、Amazon Elastic MapReduce を使用して、パブリックデータセットを分析および使用することもできます。詳細については、[What is Amazon EMR?](#) を参照してください。

## スナップショットからパブリックデータセットボリュームを作成する

Amazon EBS スナップショット形式のパブリックデータセットを使用するには、新しいボリュームを作成し、パブリックデータセットのスナップショット ID を指定します。新しいボリュームは、AWS Management Console を使用して以下のようにして作成できます。お好みで、`ec2-create-volume` コマンドを使用することもできます。

スナップショットからパブリックデータセットボリュームを作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションバーで、データセットスナップショットが存在するリージョンを選択します。



### Important

スナップショット ID は単一のリージョンに制限されており、別のリージョンに存在するスナップショットからボリュームを作成することはできません。また、Amazon EBS ボリュームは、同じアベイラビリティゾーン内のインスタンスのみにアタッチできます。詳細については、[リソースの場所 \(p. 658\)](#) を参照してください。

別のリージョンでこのボリュームを作成する必要がある場合は、スナップショットをご希望のリージョンにコピーしてから、そのリージョン内のボリュームに復元します。詳細については、[Amazon EBS スナップショットのコピー \(p. 608\)](#) を参照してください。

3. ナビゲーションペインの [Volumes] をクリックします。
4. 上段のペインの上で、[Create Volume] をクリックします。
5. [Create Volume] ダイアログボックスの [Type] リストで、[Standard] または [Provisioned IOPS] を選択します。詳細については、[Amazon EBS ボリュームの種類 \(p. 569\)](#) を参照してください。
6. [Snapshot] フィールド、データセットに対してスナップショットの ID または説明を入力します。推奨されるオプションのリストから、スナップショットを選択します。



### Note

表示されるはずのスナップショット ID が表示されない場合は、Amazon EC2 コンソールで別のリージョンが選択されている可能性があります。[パブリックデータセットの検索 \(p. 655\)](#) で特定したデータセットが詳細ページでリージョンを指定していない場合は、us-east-1 (バージニア北部) リージョンに含まれていると考えられます。

- [Size] フィールドに、ボリュームのサイズ ( GiB または TiB ) を入力するか、またはスナップショットのデフォルトのサイズが適切であるかを確認します。



#### Note

ボリュームサイズとスナップショット ID の両方を指定した場合は、スナップショットサイズ以上のサイズにする必要があります。ボリュームの種類とスナップショット ID を選択すると、ボリュームの最小サイズと最大サイズが [Size] リストの横に表示されます。

- プロビジョンド IOPS の場合は、[IOPS] フィールドに、ボリュームがサポートできる IOPS ( 1 秒あたりの入力/出力オペレーション ) の最大数を入力します。
- [Availability Zone] リストで、インスタンスを起動するアベイラビリティゾーンを選択します。詳細については、[リージョンとアベイラビリティゾーン \(p. 8\)](#) を参照してください。



#### Important

Amazon EBS ボリュームをアタッチできるインスタンスは、同じアベイラビリティゾーンに存在するものに限られます。詳細については、[リージョンとアベイラビリティゾーン \(p. 8\)](#) を参照してください。

- [Yes, Create] をクリックします。



#### Important

( [Step 7 \(p. 657\)](#) でサイズを指定して ) そのスナップショットのデフォルトサイズよりも大きいボリュームを作成した場合、追加のスペースを活用するには、ボリュームのファイルシステムを拡張する必要があります。詳細については、[ボリュームのストレージ領域を拡張する \(p. 593\)](#) を参照してください。

## パブリックデータセットボリュームをアタッチしてマウントする

新しいデータセットボリュームを作成したら、データにアクセスするためにこのボリュームを Amazon EC2 インスタンスにアタッチする必要があります ( このインスタンスも新しいボリュームと同じアベイラビリティゾーンに存在する必要があります )。詳細については、[インスタンスへの Amazon EBS ボリュームのアタッチ \(p. 573\)](#) を参照してください。

ボリュームをインスタンスにアタッチしたら、ボリュームをインスタンスにマウントする必要があります。詳細については、[Amazon EBS ボリュームを使用できるようにする \(p. 576\)](#) を参照してください。

# リソースとタグ

---

## Abstract

AMI、インスタンス、Amazon EBS ボリューム、スナップショットなどの Amazon EC2 リソースを管理します。

次のトピックでは、Amazon EC2 リソース ( AMI、インスタンス、Amazon EBS ボリューム、スナップショットなど ) の管理に使用できる機能について説明します。

### Topics

- [リソースの場所 \(p. 658\)](#)
- [リソースのリスト表示とフィルタリング \(p. 660\)](#)
- [Amazon EC2 リソースにタグを付ける \(p. 662\)](#)
- [Amazon EC2 使用状況レポート \(p. 671\)](#)

## リソースの場所

### Abstract

グローバル、リージョン、アベイラビリティゾーンに基づくリソースのいずれであるかに応じて、Amazon EC2 リソースのタイプごとにリソースの場所を一覧で示します。

次の表は、どの Amazon EC2 リソースがグローバル、リージョン、またはアベイラビリティゾーンに基づくものかをまとめたものです。

リソース	タイプ	説明
AWS アカウント	グローバル	すべてのリージョンで同じ AWS アカウントを使用できます。
DevPay 製品コード	グローバル	すべてのリージョンで同じ DevPay 製品コードを使用できます。

リソース	タイプ	説明
SSH キーペア	グローバルまたはリージョン別	Amazon EC2 を使用して作成した SSH キーペアは、そのペアを作成したリージョンでのみ使用できます。すべてのリージョンで使用できる RSA キーペアを作成し、アップロードすることができます。 詳細については、「 <a href="#">Amazon EC2 のキーペア (p. 470)</a> 」を参照してください。
Amazon EC2 リソース識別子	リージョン別	各リソース識別子 (AMI ID、インスタンス ID、EBS ボリューム ID、EBS スナップショット ID など) はリージョンに固定され、そのリソースを作成したリージョンでのみ使用できます。
ユーザーが指定したリソース名	リージョン別	各リソース名 (セキュリティグループ名、SSH キーペア名など) はリージョンに固定され、そのリソースを作成したリージョンでのみ使用できます。複数の地域で同じ名前のリソースを作成することはできませんが、それぞれが相互に関連付けられることはありません。
AMI	リージョン別	AMI は、Amazon S3 内でそのファイルが置かれているリージョンに固定されます。AMI は、別のリージョンにコピーできます。詳細については、「 <a href="#">AMI のコピー (p. 90)</a> 」を参照してください。
Elastic IP アドレス	リージョン別	Elastic IP アドレスはリージョンに固定されており、同じリージョンのインスタンスにのみ関連付けることができます。
セキュリティグループ	リージョン別	セキュリティグループはリージョンに固定されており、同じリージョンのインスタンスにのみ割り当てることができます。インスタンスが、セキュリティグループルールを使用するリージョン以外のインスタンスと通信できるようにすることはできません。別のリージョン内のインスタンスからのトラフィックは、WAN 帯域幅とみなされます。
EBS スナップショット	リージョン別	EBS スナップショットはリージョンに固定されており、同じリージョン内のボリュームの作成にのみ使用できます。スナップショットは、別のリージョンにコピーできません。詳細については、「 <a href="#">Amazon EBS スナップショットのコピー (p. 608)</a> 」を参照してください。
EBS ボリューム	アベイラビリティゾーン	Amazon EBS ボリュームはアベイラビリティゾーンに固定されており、同じアベイラビリティゾーンのインスタンスにのみアタッチできます。
インスタンス	アベイラビリティゾーン	インスタンスは、そのインスタンスを起動したアベイラビリティゾーンに固定されています。ただし、インスタンス ID はリージョンに固定されています。

#### 関連トピック

- [リージョンとアベイラビリティゾーン \(p. 8\)](#)

# リソースのリスト表示とフィルタリング

## Abstract

使用可能な複数の AWS リソースを一覧表示して、フィルタリングします。

Amazon EC2 では、異なる複数のリソースを使用することができます。リソースにはイメージ、インスタンス、Amazon EBS ボリューム、スナップショットなどがあります。Amazon EC2 コンソールを使用して、一部のタイプのリソースのリストを取得できます。対応するコマンドまたは API アクションを使用して、各タイプのリソースのリストを取得できます。リソースが多い場合は、所定の条件に一致するリソースのみが表示されるように結果をフィルタリングすることができます。

### Topics

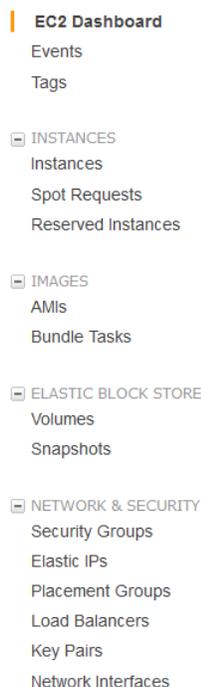
- [コンソールを使用してリソースをリスト表示する \(p. 660\)](#)
- [コンソールを使用してリソースをフィルタリングする \(p. 661\)](#)
- [CLI および API を使用した一覧表示とフィルタリング \(p. 661\)](#)

## コンソールを使用してリソースをリスト表示する

コンソールを使用して、最も一般的に使用される Amazon EC2 リソースタイプを表示できます。その他のリソースを表示するには、コマンドラインインターフェイスまたは API アクションを使用します。

コンソールを使用して EC2 リソースをリスト表示するには

1. 次の URL で Amazon EC2 コンソールを開きます: <https://console.aws.amazon.com/ec2/>。
2. ナビゲーションペインで、リソースに対応するオプションをクリックします ([AMIs] や [Instances] など)。



3. このページには、すべての利用可能なリソースが表示されます。

## コンソールを使用してリソースをフィルタリングする

Amazon EC2 コンソールを使用して、最も一般的に使用されるリソースタイプについて基本的なフィルタリングとソートを実行できます。コンソールのほとんどのページでは、特定のリソースを見つけやすくする厳選されたオプションを持つフィルタリングリストが少なくとも1個はあります。たとえば、インスタンスページでフィルタリストを使用して、実行中のインスタンスのみを表示できます。

また、各ページの検索フィールドを使用して、特定の属性または値を持つリソースを検索できます。文字列の一部または複数の文字列に基づいて検索するために、正規表現を使用できます。たとえば、MySG セキュリティグループを使用するすべてのインスタンスを検索するには、検索フィールドに「MySG」と入力します。その結果には、文字列の一部にMySGを含むすべての値が含まれます。たとえば、MySG2 や MySG3 が該当します。結果をMySGだけに絞り込むには、検索フィールドに「\bMySG\b」と入力します。タイプが m1.small または m1.large であるすべてのインスタンスを一覧表示するには、検索フィールドに「m1.small|m1.large」と入力します。

その他のフィルタリング機能を実行するには、コマンドラインインターフェイスまたは API アクションを使用します。

us-east-1b アベイラビリティゾーンでステータスが `available` のボリュームをリスト表示するには

1. ナビゲーションペインの [Volumes] をクリックします。
2. フィルタのリストから [Detached volumes] を選択します ( デタッチされたボリュームは、同じアベイラビリティゾーン内のインスタンスにアタッチすることができます )。
3. 検索フィールドに us-east-1b を入力します。
4. この基準に一致するボリュームが表示されます。

64 ビット Windows のパブリック Amazon EBS-Backed AMI を一覧表示するには

1. ナビゲーションペインの [AMIs] をクリックします。
2. [Filter] ペインのドロップダウンリストから、[Public Images]、[EBS Images]、および [Windows] を選択します。
3. 検索フィールドに x86\_64 を入力します。
4. この基準に一致する AMI が表示されます。

## CLI および API を使用した一覧表示とフィルタリング

リソースタイプごとに、そのタイプのリソースを一覧表示するための CLI コマンドまたは API リクエストが用意されています。たとえば、ec2-describe-images または DescribeImages を使用すると、Amazon Machine Image ( AMI ) を一覧表示できます。レスポンスには、すべてのリソースに関する情報が含まれます。

リソースのリストが長くなる場合には、特定の条件と一致するリソースのみが表示されるように結果をフィルタリングすることができます。フィルタ値は複数指定できます。また、複数のフィルタを指定することもできます。たとえば、タイプが m1.small または m1.large で、インスタンスの終了時に削除するように設定された EBS ボリュームがアタッチされているすべてのインスタンスを一覧表示できます。表示結果に含まれるには、インスタンスがすべてのフィルタと一致する必要があります。

フィルタの値には、ワイルドカードを使用することもできます。アスタリスク ( \* ) は0個以上の文字、クエスチョンマーク ( ? ) は1つの文字に相当します。例えば、フィルタの値として \*database\* を使用すると、説明に database を含む EBS のすべてのスナップショットを取得することができます。フィルタの値として database を指定した場合は、説明が database と等しいスナップショットのみが返されます。フィルタの値は大文字と小文字が区別されます。正確な文字列の一致、または部分文字列の一致 ( ワイルドカードを使用 ) のみをサポートしています。



#### Tip

検索には、ワイルドカード文字のリテラル値を含めることができます。ただ、文字の前にバックスラッシュを使用してエスケープする必要があります。例えば、 \\*amazon\?\\ という値では、リテラル文字列 \*amazon?\ が検索されます。

特定の EC2 リソースで使用可能なフィルタのリストを表示するには、該当するコマンドに移動してください。

- Amazon EC2 CLI については、[Amazon Elastic Compute Cloud コマンドラインリファレンス](#)にある、該当する ec2-describe コマンドを参照してください。



#### Tip

Windows Amazon EC2 でコマンドラインツールを使用する場合は、次に示すように引用符の使用が必要になる場合があります。

- AWS CLI については、[AWS Command Line Interface Reference](#)にある、該当する describe コマンドを参照してください。
- Windows PowerShell については、[AWS Tools for Windows PowerShell Reference](#)にある、該当する Get コマンドを参照してください。
- クエリ API については、[Amazon Elastic Compute Cloud API Reference](#)にある、該当する Describe API アクションを参照してください。

## Amazon EC2 リソースにタグを付ける

### Abstract

Amazon EC2 の各リソースに独自のメタデータタグを割り当てて、インスタンス、イメージ、その他のリソースを管理します。

インスタンス、イメージなどの Amazon Elastic Compute Cloud ( Amazon EC2 ) リソースの管理を支援するため、各リソースにはタグという形式で独自のメタデータを割り当てることができます。ここでは、タグとその作成方法について説明します。

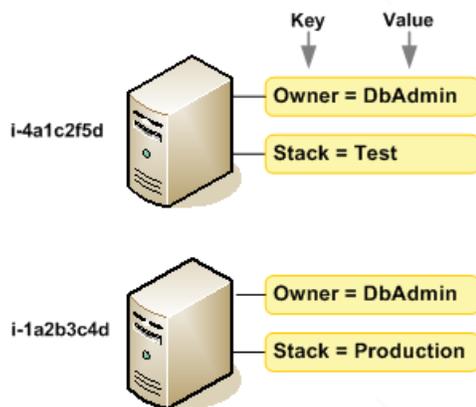
#### Topics

- [タグの基本 \(p. 663\)](#)
- [タグの制限 \(p. 663\)](#)
- [請求用のリソースにタグを付ける \(p. 665\)](#)
- [コンソールによるタグの操作 \(p. 665\)](#)
- [API と CLI の概要 \(p. 670\)](#)

## タグの基本

タグを使用すると、AWS リソースを目的、所有者、環境などさまざまな方法で分類することができます。タグはそれぞれ、1つのキーと1つの値で構成されており、どちらもお客様側が定義します。例えば、アカウントの各インスタンスの所有者とスタックレベルを追跡しやすくするため、Amazon EC2 インスタンスに対して一連のタグを定義できます。ニーズを満たす一連のタグキーをリソースタイプごとに考案されることをお勧めします。一貫性のあるタグキーセットを使用することで、リソースの管理が容易になります。追加したタグに基づいてリソースを検索およびフィルタリングできます。

次の図は、タグの機能を示しています。図の中では、インスタンスのそれぞれに2つのタグを割り当てています。1つは `Owner`、もう1つは `Stack` という名前です。各タグには値も関連付けられています。



タグには、Amazon EC2 に関連する意味はなく、完全に文字列として解釈されます。また、タグは自動的にリソースに割り当てられます。

タグの操作には、AWS Management Console、Amazon EC2 コマンドラインインターフェイス ( CLI )、および Amazon EC2 API を使用できます。

タグを割り当てることができるのは、既存のリソースに限られます。Amazon EC2 コンソールを使用すると、タグリストにアクセスしてインスタンスに追加することができます。タグリストは、インスタンスの作成直後に適用されます。特定のリソースについて既存のタグと同じキーを持つタグを追加した場合、古い値は新しい値によって上書きされます。タグのキーと値は編集でき、タグはリソースからいつでも削除できます。タグの値を空の文字列に設定することはできますが、null に設定することはできません。

AWS Identity and Access Management ( IAM ) を使用している場合は、AWS アカウント内のユーザーに対してタグを作成、編集、削除するアクセス許可を割り当てることができます。IAM の詳細については、「[Amazon EC2 のリソースに対するアクセスの制御 \(p. 485\)](#)」を参照してください。

## タグの制限

タグには以下のような基本制限があります。

- リソースあたりのタグの最大数 - 10
- キーの最大長 - 127 文字 ( Unicode )
- 値の最大長 - 255 文字 ( Unicode )
- タグのキーと値は大文字と小文字が区別されます。

- タグの名前または値に `aws:` プレフィックスは使用しないでください。このプレフィックスは AWS 用に予約されています。このプレフィックスが含まれるタグの名前または値は編集または削除できません。

タグのみに基づいてリソースを終了、停止、削除することはできません。リソース識別子を指定する必要があります。例えば、DeleteMe という名前のタグキーが付けられているスナップショットを削除するには、まず、タグを指定するフィルタを使用して DescribeSnapshots を呼び出して、そのようなスナップショットの一覧を取得する必要があります。次に、スナップショットの ID ( snap-1a2b3c4d など ) を指定して DeleteSnapshots を使用します。タグを指定したフィルタを使用して DeleteSnapshots を呼び出すことはできません。リソースをリスト表示するときのフィルタの使い方については、[リソースのリスト表示とフィルタリング \(p. 660\)](#) を参照してください。

パブリックリソースまたは共有リソースにタグを付けることはできますが、割り当てるタグはタグ付けを行った AWS アカウントだけが使用できるものであり、リソースを共有するその他のアカウントは使用できません。

すべてのリソースにタグを付けることができない場合、タグの指定には API アクションまたはコマンドラインを使用する必要があります。次の表に、すべての Amazon EC2 リソースと、適用されるタグ指定に制限がある場合はその制限を示します。タグ指定の制限がないリソースには、API アクション、CLI、およびコンソールを使用してタグを指定できます。

リソース	タグ指定のサポート	タグ指定の制限
AMI	○	なし
バンドルタスク	×	
カスタマーゲートウェイ	○	なし
DHCP オプション	○	なし
EBS ボリューム	○	なし
Elastic IP	×	
インスタンス	○	なし
インターネットゲートウェイ	○	なし
キーペア	×	
ロードバランサー	×	
ネットワーク ACL	○	なし
ネットワークインターフェイス	○	なし
配置グループ	×	
リザーブドインスタンス	○	なし
リザーブドインスタンスのリスト	いいえ	
ルートテーブル	○	なし
スポットインスタンスリクエスト	○	なし
セキュリティグループ - EC2 Classic	○	なし

リソース	タグ指定のサポート	タグ指定の制限
セキュリティグループ - VPC	○	なし
スナップショット	○	なし
サブネット	○	なし
仮想プライベートゲートウェイ	○	なし
VPC	○	なし
VPN 接続	○	なし

AWS コンソールを使用したタグ指定の詳細については、「[コンソールによるタグの操作 \(p. 665\)](#)」を参照してください。API または コマンドラインを使用したタグ指定の詳細については、「[API と CLI の概要 \(p. 670\)](#)」を参照してください。

## 請求用のリソースにタグを付ける

タグを使用して AWS 請求書を整理し、自分のコスト構造を反映することができます。そのためには、AWS アカウントにサインアップして、タグキー値が含まれた AWS アカウントの請求書を取得する必要があります。次に、リソースを組み合わせたコストを確認するために、同じタグキー値を持つリソースに基いて、請求情報を整理します。例えば、複数のリソースに特定のアプリケーション名のタグを付け、請求情報を整理することで、複数のサービスを利用しているアプリケーションの合計コストを確認することができます。詳細については、[About AWS Account Billing](#) の *Cost Allocation and Tagging* を参照してください。

## コンソールによるタグの操作

Amazon EC2 コンソールを使用して、同じリージョン内のすべての Amazon EC2 リソースで使用中のタグを表示できます。タグは、リソース別およびリソースタイプ別で表示し、指定したタグに関連付けられている各リソースタイプの項目数を表示することができます。また、Amazon EC2 コンソールを使用して、同時に 1 つまたは複数のリソースについてタグの適用またはタグの削除を行うことができます。

### Topics

- [タグを表示する \(p. 665\)](#)
- [個々のリソースでのタグの追加と削除 \(p. 666\)](#)
- [リソースグループへのタグの追加と削除 \(p. 667\)](#)
- [インスタンスを起動するときにタグを追加する \(p. 669\)](#)
- [タグを使用してリソースリストをフィルタリングする \(p. 669\)](#)

## タグを表示する

Amazon EC2 コンソールでは、2 種類の方法でタグを表示できます。個々のリソースまたはすべてのリソースについて、タグを表示できます。

個々のリソースのタグを表示するには

Amazon EC2 コンソールでリソース固有のページを選択すると、リソースリストが表示されます。例えば、ナビゲーションペインの [Instances] を選択すると、Amazon EC2 インスタンスリストが表示されます。このようなリスト ( インスタンスなど ) からリソースを選択し、リソースがタグをサポートしている場合、タグを表示し、管理することができます。ほとんどのリソースページで、詳細ペインの

[Tags] タブでタグを表示できます。次の図は、2 つのタグ ( Name = DNS Server および Purpose = Network Management ) を持つインスタンスの [Tags] タブを示しています。

リソースリストには、キーが同じタグのすべての値を表示する列を追加できます。この列で、タグを使用してリソースリストの並べ替えやフィルタリングを行うことができます。新しい列をリソースリストに追加し、タグを表示するには、2 つの方法があります。

- [Tags] タブでタグの [Show Column] をクリックするだけで、タグがリストに表示されます。
- [Show/Hide Columns] ( 歯車型のアイコン ) をクリックし、[Show/Hide Columns] ダイアログボックスの [Your Tag Keys] のタグキーを選択します。

すべてのリソースのタグを表示するには

Amazon EC2 コンソールのナビゲーションペインの [Tags] を選択して、すべてのリソースのタグを表示できます。次の図は、リソースタイプごとに使用中のすべてのタグが表示された [Tags] ペインです。

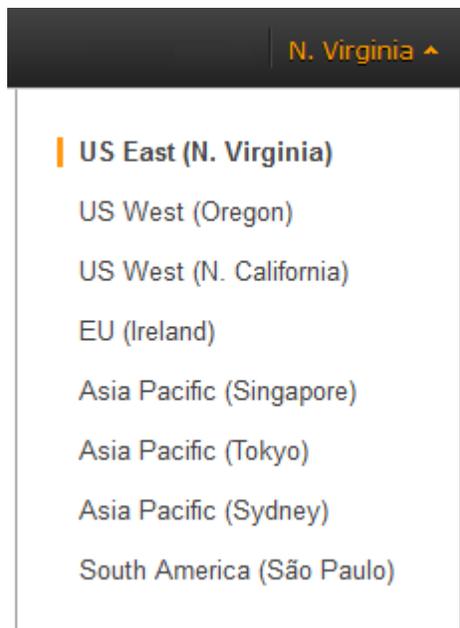
	Tag Key	Tag Value	Total	Instances	AMIs	Volumes
Manage Tag	Name	DNS Server	1	1	0	0
Manage Tag	Owner	TeamB	2	0	0	2
Manage Tag	Owner	TeamA	2	0	0	2
Manage Tag	Purpose	Project2	1	0	0	1
Manage Tag	Purpose	Logs	1	0	0	1
Manage Tag	Purpose	Network Management	1	1	0	0
Manage Tag	Purpose	Project1	2	0	0	2

## 個々のリソースでのタグの追加と削除

リソースのページから、個々のリソースのタグを直接管理できます。AMIのタグを管理している場合、他のリソースのタグとは手順が異なります。すべての手順について以下に説明します。

個々のリソースにタグを追加するには

1. Amazon EC2 コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションバーで、ニーズに合ったリージョンを選択します。この選択は、リージョン間で共有できるのが一部の Amazon EC2 リソースのみであるため、重要です。詳細については、「[リソースの場所 \(p. 658\)](#)」を参照してください。



3. ナビゲーションペインで、リソースタイプをクリックします ([Instances] など)。
4. リソースリストからリソースを選択します。
5. 詳細ペインの [Tags] タブをクリックします。
6. [Add/Edit Tags] ボタンをクリックします。
7. [Add/Edit Tags] ダイアログボックスで、各タグのキーと値を指定し、[Save] をクリックします。

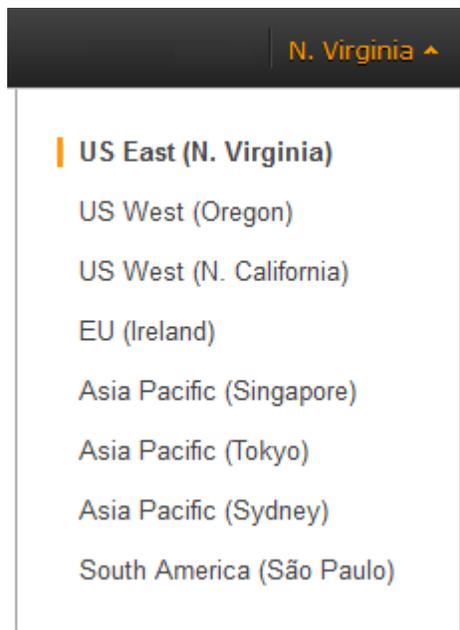
個々のリソースからタグを削除するには

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. ナビゲーションバーで、ニーズに合ったリージョンを選択します。この選択は、リージョン間で共有できるのが一部の Amazon EC2 リソースのみであるため、重要です。詳細については、「[リソースの場所 \(p. 658\)](#)」を参照してください。
3. ナビゲーションペインで、リソースタイプをクリックします ([Instances] など)。
4. リソースリストからリソースを選択します。
5. 詳細ペインの [Tags] タブをクリックします。
6. [Add/Edit Tags] をクリックし、タグの [Delete] アイコンをクリックして、[Save] をクリックします。

## リソースグループへのタグの追加と削除

リソースグループにタグを追加するには

1. Amazon EC コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. ナビゲーションバーで、ニーズに合ったリージョンを選択します。この選択は、リージョン間で共有できるのが一部の Amazon EC2 リソースのみであるため、重要です。詳細については、「[リソースの場所 \(p. 658\)](#)」を参照してください。



3. ナビゲーションペインの [Tags] をクリックします。
4. コンテンツペインの上部にある [Manage Tags] をクリックします。
5. [Filter] ドロップダウンリストから、タグを追加するリソースタイプ ( インスタンスなど ) を選択します。
6. リソースリストで、タグを追加する各リソースの横にあるチェックボックスをオンにします。
7. [Add Tag] の [Key] ボックスと [Value] ボックスに、使用するタグのキーと値を入力し、[Add Tag] をクリックします。



#### Note

既存のタグとタグキーが同じ新しいタグを追加すると、既存のタグは新しいタグで上書きされます。

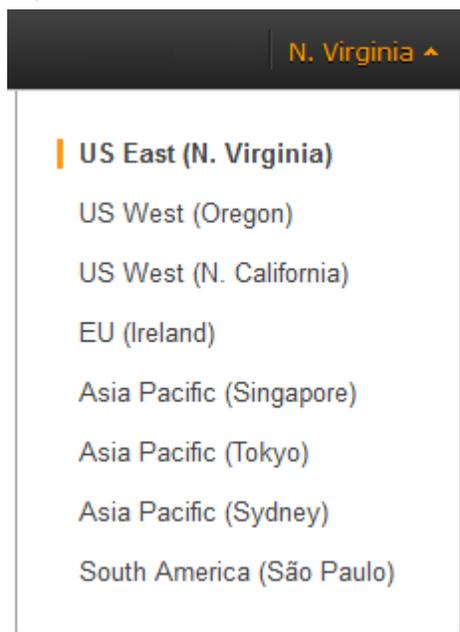
#### リソースグループからタグを削除するには

1. Amazon EC コンソール ( <https://console.aws.amazon.com/ec2/> ) を開きます。
2. ナビゲーションバーで、ニーズに合ったリージョンを選択します。この選択は、リージョン間で共有できるのが一部の Amazon EC2 リソースのみであるため、重要です。詳細については、「[リソースの場所 \(p. 658\)](#)」を参照してください。
3. ナビゲーションペインの [Tags] をクリックします。
4. コンテンツペインの上部にある [Manage Tags] をクリックします。
5. 使用中のタグを表示するには、[Show/Hide Columns] ( 歯車型のアイコン ) をクリックし、[Show/Hide Columns] ダイアログボックスで、表示するタグキーを選択し、[Close] をクリックします。
6. [Filter] ドロップダウンリストから、タグを削除するリソースタイプ ( インスタンスなど ) を選択します。
7. リソースリストで、タグを削除する各リソースの横にあるチェックボックスをオンにします。
8. [Remove Tag] で、[Key] ボックスをクリックしてキーを選択するか、キー名を入力して、[Remove Tag] をクリックします。

## インスタンスを起動するときにタグを追加する

[Launch Wizard] を使用してタグを追加するには

1. ナビゲーションバーで、インスタンスを起動するリージョンを選択します。この選択は、リージョン間で共有できるのが一部の Amazon EC2 リソースのみであるため、重要です。ニーズに合ったリージョンを選択してください。詳細については、「[リソースの場所 \(p. 658\)](#)」を参照してください。



2. EC2 ダッシュボードの [Launch Instance] ボタンをクリックします。
3. [Choose an Amazon Machine Image (AMI)] ページには、Amazon マシンイメージ (AMI) と呼ばれる基本設定リストが表示されます。使用する AMI を選択し、[Select] ボタンをクリックします。AMI の選択の詳細については、「[適切な AMI の検索 \(p. 60\)](#)」を参照してください。
4. [Configure Instance Details] ページで、必要に応じてインスタンスの設定を行い、[Next: Add Storage] をクリックします。
5. [Add Storage] ページで、インスタンスに追加のストレージボリュームを指定できます。完了したら、[Next: Tag Instance] をクリックします。
6. [Tag Instance] ページで、キーと値の組み合わせを指定することによって、インスタンスのタグを指定します。[Create Tag] をクリックして、複数のタグをインスタンスに追加します。完了したら、[Next: Configure Security Group] をクリックします。
7. [Configure Security Group] ページで、所有する既存のセキュリティグループから選択するか、ウィザードで新しいセキュリティグループを作成します。完了したら、[Review and Launch] をクリックします。
8. 設定を確認します。選択した内容でよければ、[Launch] をクリックします。既存のキーペアを選択するか、新しいキーペアを作成し、確認のチェックボックスを選択して、[Launch Instances] をクリックします。

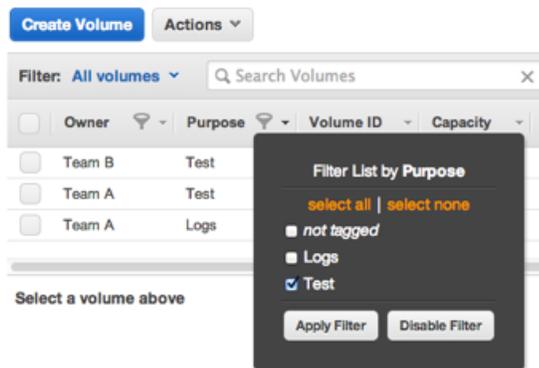
## タグを使用してリソースリストをフィルタリングする

1 つまたは複数のタグキーとタグ値に基づいて、リソースリストをフィルタリングできます。

タグを使用してリソースリストをフィルタリングするには

1. 次の手順でタグの列を表示します。

- a. リソースのいずれかを選択します。
  - b. 詳細ページの [Tags] タブをクリックします。
  - c. リスト内のタグを選択し、[Show Column] をクリックします。
2. フィルタリストを表示するタグの列の右上にあるフィルタアイコンをクリックします。
  3. タグ値を選択し、[Apply Filter] をクリックして結果リストをフィルタリングします。



## API と CLI の概要

リソースのタグの追加、更新、リスト表示、および削除には、次の API および CLI コマンドを使用します。各コマンドのドキュメントには、例が記載されています。これらのコマンドラインインターフェイスについての詳細は、[Amazon EC2 へのアクセス \(p. 3\)](#) を参照してください。

説明	Amazon EC2 CLI	AWS CLI	AWS Tools for Windows PowerShell	API アクション
指定したリソースの1つまたは複数のタグを追加または上書きします。	<a href="#">ec2-create-tags</a>	<a href="#">create-tags</a>	<a href="#">New-EC2Tag</a>	<a href="#">CreateTags</a>
指定したリソースから指定したタグを削除します。	<a href="#">ec2-delete-tags</a>	<a href="#">delete-tags</a>	<a href="#">Remove-EC2Tag</a>	<a href="#">DeleteTags</a>
リソースの1つまたは複数について説明します。	<a href="#">ec2-describe-tags</a>	<a href="#">describe-tags</a>	<a href="#">Get-EC2Tag</a>	<a href="#">DescribeTags</a>

また、タグに基づいてリソースのリストをフィルタリングすることもできます。構文例については、[コンソールを使用してリソースをフィルタリングする \(p. 661\)](#) を参照してください。各リソースでサポートされているフィルタについては、該当するコマンドを参照してください:

- Amazon EC2 CLI については、[Amazon Elastic Compute Cloud コマンドラインリファレンス](#) にある、該当する `ec2-describe` コマンドを参照してください。
- AWS CLI については、[AWS Command Line Interface Reference](#) にある、該当する `describe` コマンドを参照してください。

- Windows PowerShell については、[AWS Tools for Windows PowerShell Reference](#) にある、該当する Get コマンドを参照してください。
- クエリ API については、[Amazon Elastic Compute Cloud API Reference](#) にある、該当する Describe API アクションを参照してください。

## Amazon EC2 使用状況レポート

### Abstract

Amazon EC2 使用状況レポートを使用して、インスタンスの使用状況を分析します。

Amazon EC2 で提供されている使用状況レポートを使用すると、インスタンスの使用状況を詳細に分析できます。使用状況レポートのデータは、毎日複数回更新されます。レポートは、AWS アカウント、リージョン、アベイラビリティゾーン、オペレーティングシステム、インスタンスタイプ、購入オプション、テナンシー、およびタグでフィルタリングできます。

アカウントの使用状況およびコストデータを取得するには、そのアカウントの認証情報を持っており、アカウントのリソースおよびタグ付きの請求明細レポートを有効にする必要があります。一括請求を使用しており、支払いアカウントにログインしている場合は、支払いアカウントおよびそのリンクされているすべてのアカウントのデータを表示できます。一括請求を使用しており、リンクされているいずれかのアカウントにログインしている場合は、そのリンクされているアカウントのデータのみを表示できます。一括請求については、[Pay Bills for Multiple Accounts with Consolidated Billing](#) を参照してください。

### Topics

- [使用可能なレポート \(p. 671\)](#)
- [使用状況レポートをセットアップする \(p. 671\)](#)
- [IAM ユーザーに Amazon EC2 使用状況レポートへのアクセラ権を付与する \(p. 673\)](#)
- [インスタンスの使用状況レポート \(p. 674\)](#)
- [リザーブドインスタンスの使用率レポート \(p. 677\)](#)

## 使用可能なレポート

生成できるレポートは次のとおりです。

- [インスタンスの使用状況レポート \(p. 674\)](#)。このレポートは、オンデマンドインスタンス、スポットインスタンス、およびリザーブドインスタンスの使用状況を対象としています。
- [リザーブドインスタンスの使用率レポート \(p. 677\)](#)。このレポートは、予約済みの処理能力の使用状況を対象としています。

## 使用状況レポートをセットアップする

開始する前に、以下の手順に示すように、リソースおよびタグ付きの請求明細レポートを有効にします。この手順を完了したら、インスタンスの使用状況データの収集を開始します。請求明細レポートを既に有効にしている場合は、有効にしてから収集している使用状況データにアクセスできます。



### Important

これらの手順を完了するには、AWS アカウント認証情報を使用してログインする必要があります。IAM ユーザー認証情報を使用してログインした場合、これらの手順を完了することはできません。

### 請求明細レポートを有効にするには

1. 使用状況データを受け取るために既存の Amazon S3 バケットを選択します。請求書データが含まれているので、このバケットへのアクセス権を必ず管理してください (これらのファイルを保存しておく必要はありません。実際には、不要になったらすぐに削除してかまいません)。バケットがない場合は、以下のようにして作成します。
  - a. Amazon S3 コンソールを開きます。
  - b. [Create Bucket] をクリックします。
  - c. [Create a Bucket] ダイアログボックスにバケットの名前 (例えば、*username-ec2-usage-data*) を入力し、リージョンを選択して、[Create] をクリックします。バケット名の要件の詳細については、*Amazon Simple Storage Service* コンソールユーザーガイドの [Creating a Bucket](#) を参照してください。
2. 請求コンソールを <https://console.aws.amazon.com/billing/home?#> から開きます。
3. ナビゲーションペインの [Preferences] をクリックします。
4. [Receive Billing Reports] を選択します。
5. [Save to S3 Bucket] で Amazon S3 バケットの名前を指定し、[Verify] をクリックします。
6. Amazon S3 バケットに対して、使用状況データを公開するための AWS アクセス許可を付与します。
  - a. [Receive Billing Reports] で、[sample policy] をクリックします。サンプルポリシーをコピーします。サンプルポリシーは、指定したバケット名を使用しています。
  - b. 別のブラウザタブで、Amazon S3 コンソールを開きます。バケットを選択して、[Properties] をクリックし、[Permissions] を展開します。[Permissions] セクションで、[Add bucket policy] をクリックします。サンプルポリシーをテキストエリアに貼り付けて、[Save] をクリックします。[Permissions] セクションで、[Save] をクリックします。
  - c. サンプルポリシーのあるブラウザタブに戻り、[Done] をクリックします。
7. [Report] で、[Detailed billing report with resources and tags] を選択します。
8. [Save Preferences] をクリックします。



#### Note

レポートにデータが表示されるまで、最大 1 日かかる場合があります。

インスタンスは、タグを使用して分類できます。インスタンスにタグを付加してから、これらのタグに基づくレポートを有効にする必要があります。

### タグによる使用状況レポートを有効にするには

1. インスタンスにタグを付加します。最良の結果を得るために、レポートに使用する各タグをそれぞれのインスタンスに必ず追加してください。インスタンスにタグを付加する方法の詳細については、[Amazon EC2 リソースにタグを付ける \(p. 662\)](#) を参照してください。
2. 請求コンソールを <https://console.aws.amazon.com/billing/home?#> から開きます。
3. ナビゲーションペインの [Preferences] をクリックします。
4. [Report] で、[Manage report tags] をクリックします。
5. ページに作成したタグのリストが表示されます。インスタンスの使用状況データのフィルタリングまたはグループ化に使用するタグを選択し、[Save] をクリックします。選択しないタグはすべて、インスタンスの使用状況レポートから自動的に除外されます。



#### Note

これらの変更は、現在の月のデータのみにも適用されます。これらの変更が有効になるまで、最大 1 日かかる場合があります。

## IAM ユーザーに Amazon EC2 使用状況レポートへの アクセス権を付与する

デフォルトでは、IAMユーザーは Amazon EC2 使用状況レポートにアクセスできません。IAMユーザーにこれらのレポートへのアクセス許可を付与する IAM ポリシーを作成する必要があります。

次のポリシーでは、ユーザーは両方の Amazon EC2 使用状況レポートを表示できます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-reports:*",
    "Resource": "*"
  }]
}
```

次のポリシーでは、ユーザーはインスタンスの使用状況レポートを表示できます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewInstanceUsageReport",
    "Resource": "*"
  }]
}
```

次のポリシーでは、ユーザーはリザーブドインスタンスの使用率レポートを表示できます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "ec2-reports:ViewReservedInstanceUtilizationReport",
    "Resource": "*"
  }]
}
```

IAM ポリシーの詳細については、IAM を使用するの [Permissions and Policies](#) を参照してください。

# インスタンスの使用状況レポート

## Abstract

インスタンスの使用状況をレポートとして表示します。

インスタンスの使用状況レポートを使用して、インスタンスの使用状況およびコストの傾向を表示できます。使用状況データは、インスタンス時間またはコストで表示できます。時間単位、日次、および月次の使用状況データの集計を表示するように選択できます。レポートは、リージョン、アベイラビリティゾーン、インスタンスタイプ、AWS アカウント、プラットフォーム、テナンシー、購入オプション、またはタグでフィルタリングまたはグループ化できます。レポートを設定したら、後で簡単に戻れるようにブックマークしておくことができます。

以下に、インスタンスの使用状況レポートを作成すると回答を得られる、いくつかの質問の例を示します。

- 各インスタンスタイプのインスタンスには、どのくらいのコストがかかりますか？
- 特定の部門でどのくらいの実験時間を使用されていますか？
- 自分のインスタンスの使用は、複数のアベイラビリティゾーンにわたってどのように分散されていますか？
- 自分のインスタンスの使用は、複数の AWS アカウントにわたってどのように分散されていますか？

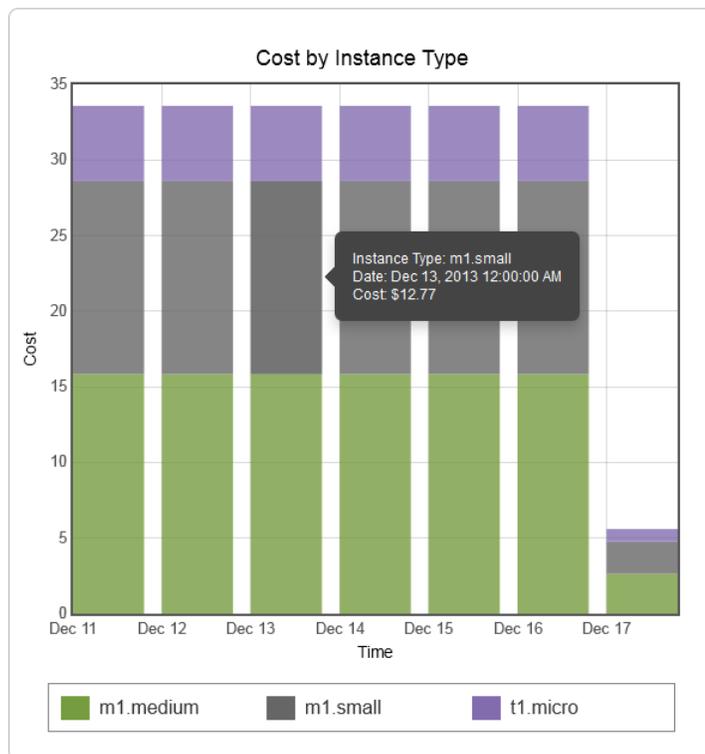
## Topics

- [レポートの形式 \(p. 674\)](#)
- [インスタンスの使用状況を表示する \(p. 675\)](#)
- [カスタムレポートをブックマークする \(p. 677\)](#)
- [使用状況データをエクスポートする \(p. 677\)](#)

## レポートの形式

要求した使用状況データは、グラフと表の両方で表示されます。

例えば、次のグラフには、インスタンスタイプ別のコストが表示されています。グラフの要点には、どの色がどのインスタンスタイプを表しているかが示されています。棒グラフのセグメントに関する詳細情報を表示するには、その上にマウスを移動します。



対応する表には、各インスタンスタイプについて1つの列が表示されています。列の一番上には、グラフ内のインスタンスタイプと同じ色を持つ色の帯が表示されていることに注意してください。

Time (UTC)	m1.medium	m1.small	t1.micro
12/11/13	\$15.84	\$12.77	\$4.97
12/12/13	\$15.84	\$12.77	\$4.97
12/13/13	\$15.84	\$12.77	\$4.97
12/14/13	\$15.84	\$12.77	\$4.97
12/15/13	\$15.84	\$12.77	\$4.97
12/16/13	\$15.84	\$12.77	\$4.97
12/17/13	\$2.64	\$2.13	\$0.83
Total	\$97.68	\$78.75	\$30.65

## インスタンスの使用状況を表示する

以下の手順は、提供されているいくつかの機能を使用して、使用状況レポートを生成する方法を示しています。

開始する前に、セットアップが終了している必要があります。詳細については、[使用状況レポートをセットアップする \(p. 671\)](#) を参照してください。

インスタンスの使用状況をインスタンスタイプでフィルタリングおよびグループ化するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Reports] をクリックし、[EC2 Instance Usage Report] をクリックします。
3. [Unit] のオプションを選択します。インスタンスが実行されている時間を時間単位で表示するには、Instance Hours を選択します。インスタンスの使用状況のコストを表示するには、Cost を選択します。
4. [Granularity] および [Time range] のオプションを選択します。
  - 時間範囲内で時間ごとに要約されたデータを表示するには、Hourly 単位を選択します。時間単位のデータを表示する場合は、最大 2 日間の時間範囲を選択できます。
  - 時間範囲内で日ごとに要約されたデータを表示するには、Daily 単位を選択します。日次データを表示する場合は、最大 2 か月間の時間範囲を選択できます。
  - 時間範囲内で月ごとに要約されたデータを表示するには、Monthly 単位を選択します。
5. [Filter] リストで、[Instance Type] を選択します。[Group by] リストで、[Instance Type] を選択します。
6. フィルタリングエリアで 1 つ以上のインスタンスタイプを選択し、[Update Report] をクリックします。[Applied Filters] に、指定したフィルタリングが表示されます。

[EC2 Management Console](#) > [Reports](#) > EC2 Instance Usage

Instance Usage Reports Download

Granularity: Daily Time range: Last 7 Days Unit: Cost

Filter: Select Group by: Instance Type

Applied Filters: Instance Type: t1.micro; m1.small; m1.medium; Clear All Filters Update Report

ページの一番上で [Reports] または [EC2 Management Console] をクリックすると、Amazon EC2 コンソールに戻ることができます。

タグに基づいてインスタンスの使用状況をグループ化するには

1. インスタンスの使用状況レポートページを開きます。
2. [Unit] のオプションを選択します。インスタンスが実行されている時間を時間単位で表示するには、Instance Hours を選択します。インスタンスの使用状況のコストを表示するには、Cost を選択します。
3. [Granularity] および [Time range] のオプションを選択します。
  - 時間範囲内で時間ごとに要約されたデータを表示するには、Hourly 単位を選択します。時間単位のデータを表示する場合は、最大 2 日間の時間範囲を選択できます。
  - 時間範囲内で日ごとに要約されたデータを表示するには、Daily 単位を選択します。日次データを表示する場合は、最大 2 か月間の時間範囲を選択できます。
  - 時間範囲内で月ごとに要約されたデータを表示するには、Monthly 単位を選択します。
4. [Group by] リストで、[Tag] を選択します。
5. [Key Name] ボックスをクリックし、リストから名前を選択して、[Update Report] をクリックします。このリストに項目がない場合は、タグによる使用状況レポートを有効にする必要があります。詳細については、[タグによる使用状況レポートを有効にするには \(p. 672\)](#) を参照してください。

Instance Usage Reports

Granularity: **Daily** Time range: **Last 14 Days** Unit: **Instance Hours**

Filter: **Select** Group by: **Tag** Key Name:

Applied Filters: None

[Clear All Filters](#) [Update Report](#)

## カスタムレポートをブックマークする

カスタムレポートを再度生成することができます。そのためには、レポートをブックマークしておきます。

カスタムレポートをブックマークするには

1. レポートのオプションとフィルタを選択します。項目を1つ選択すると、コンソール URL にパラメータが1つ追加されます。例えば、granularity=Hourly と Filters を選択すると、filter\_list となります。
2. ブラウザを使用して、コンソール URL をブックマークに追加します。
3. 今後同じレポートを生成するには、作成したブックマークを使用します。

## 使用状況データをエクスポートする

レポートのグラフまたは表を他のレポートに含めることができます。そのためには、データをエクスポートします。

使用状況データをエクスポートするには

1. レポートのオプションとフィルタを選択します。
2. 使用状況データを .csv ファイルとしてテーブルからエクスポートするには、[Download] をクリックして [CSV Only] を選択します。
3. グラフ形式の使用状況データを .png としてエクスポートするには、[Download] をクリックして [Graph Only] を選択します。

## リザーブインスタンスの使用率レポート

### Abstract

リザーブインスタンスの使用率レポートを使用して、購入した Amazon EC2 リザーブインスタンスの使用率を確認します。

リザーブインスタンスの使用状況レポートには、所有する Amazon EC2 リザーブインスタンスの各グループ（つまり、バケット）の一定期間における使用量が示されます。各バケットには、リージョン、アベイラビリティゾーン、インスタンスタイプ、テナンシー、提供タイプ、およびプラットフォームの一意の組み合わせが存在します。レポートの対象とする時間範囲を、1日間（昨日、今日）から数週間、1か月から数か月間、1年間、または3年間から指定できます。参照可能なデータは、アカウントの請求明細レポートをいつ有効にしたかによって異なります（[使用状況レポートをセットアップする \(p. 671\)](#) を参照）。たとえば、リザーブインスタンスの使用状況レポートでは、バケット内でのインスタンス利用に対して支払われたリザーブインスタンス料金とオンデマンド価格が比較され、レポートの対象時間範囲の節約額が示されます。



#### Note

リザーブインスタンスバケットでは、Amazon VPC と Amazon VPC 以外 (EC2 Classic) のネットワークプラットフォームタイプにおけるリザーブインスタンスが、請求書の計算と同じように集計されます。さらに、バケット内のリザーブインスタンスによって、前払い価格と時間単価が異なる可能性があります。

以下に、リザーブインスタンスの使用率レポートを使用して答えを示すことができる、いくつかの質問の例を示します。

- 自分はリザーブインスタンスをどのくらい適切に使用しているのでしょうか？
- リザーブインスタンスによってコストを削減できているのでしょうか？

リザーブインスタンスの詳細については、[リザーブインスタンス \(p. 218\)](#) を参照してください。

開始する前に、セットアップが終了している必要があります。詳細については、[使用状況レポートをセットアップする \(p. 671\)](#) を参照してください。

#### Topics

- [レポートについて理解する \(p. 678\)](#)
- [カスタムレポートをブックマークする \(p. 682\)](#)
- [使用状況データをエクスポートする \(p. 682\)](#)
- [オプションリファレンス \(p. 682\)](#)

## レポートについて理解する

リザーブインスタンスの使用率レポートには、要求した使用率データがグラフと表の形式で表示されます。



レポートでは、一定期間のリザーブインスタンスの利用データがバケット別に集計されます。レポートのテーブルの各行はバケットを表しており、次のメトリックスが示されます。

- Count - レポートの期間内に同時に所有されたリザーブインスタンスの最大数。
- Usage Cost - リザーブインスタンスバケットの対象となるインスタンス利用に適用される、リザーブインスタンス合計使用料。
- Total Cost - リザーブインスタンスバケットに関連付けられた利用期間における、使用料と前払い償却料金の合計金額。



#### Note

リザーブインスタンスマーケットプレイスで販売したリザーブインスタンスがバケットに含まれており、そのリザーブインスタンスがレポート期間のどの時点においてもアクティブであった場合、バケットの合計コストが増加し、節約額が低く計算される可能性があります。

- Savings - その期間の利用に対してかかるはずのオンデマンド価格と、リザーブインスタンスを使用して実際にかかるコスト（合計コスト）の差額。
- Average Utilization - 一定期間におけるリザーブインスタンスバケットの1時間あたりの平均使用率。
- [Maximum Utilization - レポート対象期間における任意の1時間の最大使用率。

テーブルの行（リザーブインスタンスバケット）ごとに、レポートに選択された [Time range] における、選択された [Show] メトリックに基づくデータがグラフで表されます。グラフの各ポイントは、ある時点でのメトリックスを表しています。レポートオプションについては、[オプションリファレンス \(p. 682\)](#) を参照してください。

表内で選択した各行の左端に表示される色の帯は、グラフ内のレポートの線に対応しています。行の左端にあるチェックボックスをオンにすると、グラフにその行を表示できます。

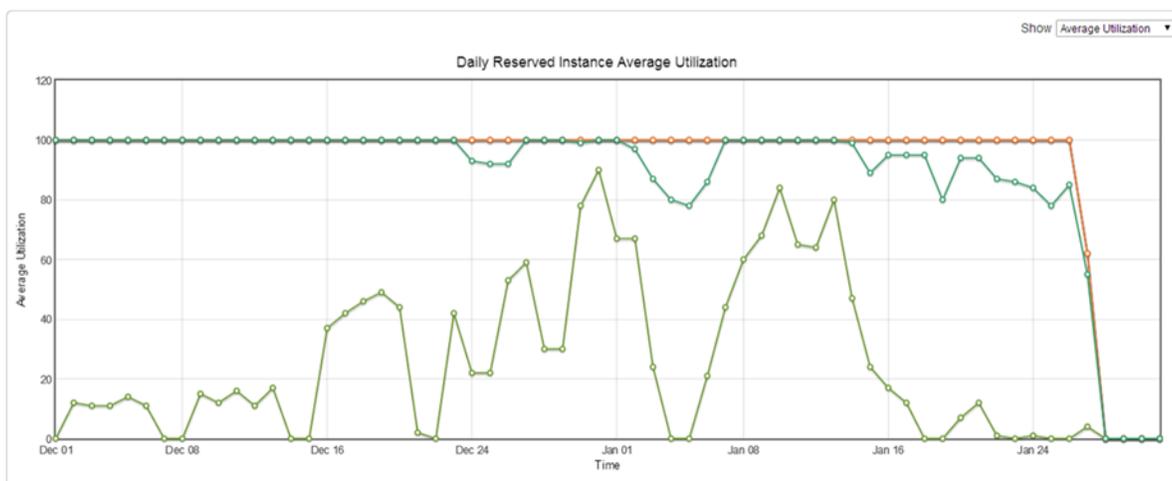
	Instance Type	Platform	Tenancy	Availability Zone	Offering Type	Count	Usage Cost	Total Cost	Savings	Average Utilization	Maximum Utilization
<input checked="" type="checkbox"/>	m1.large	Linux/Unix	Shared	eu-west-1a	Medium	12	\$438.30	\$491.04	\$648.54	27.00%	100.00%

デフォルトでは、リザーブインスタンスの使用率レポートは、過去 14 日間におけるすべてのリザーブインスタンスバケットのデータを返します。グラフには、表内の先頭の 5 つのバケットの平均使用率が表示されます。レポートグラフをカスタマイズして、1 日、数週間、数か月、または数年の期間におけるさまざまな使用率（平均使用率、最大使用率）やコスト（総コスト、使用コスト）データを表示することができます。

たとえば、次のレポートテーブルは 2 か月間のリザーブインスタンスの使用率を示しています。この場合の期間は、12 月 1 日～1 月 31 日です。このレポートは、eu-west-1a、eu-west-1b、および eu-west-1c 内のリザーブインスタンスバケットに関するデータのみを返すようにフィルタリングされています。

	Instance Type	Platform	Tenancy	Availability Zone	Offering Type	Count	Usage Cost	Total Cost	Savings	Average Utilization	Maximum Utilization
<input checked="" type="checkbox"/>	m1.large	Linux/Unix	Shared	eu-west-1a	Medium	12	\$438.30	\$491.04	\$648.54	27.00%	100.00%
<input checked="" type="checkbox"/>	m1.medium	Linux/Unix	Shared	eu-west-1a	Medium	2	\$137.90	\$164.27	\$194.27	100.00%	100.00%
<input checked="" type="checkbox"/>	m1.xlarge	Linux/Unix	Shared	eu-west-1b	Heavy	3	\$665.28	\$791.86	\$1,359.38	100.00%	100.00%
<input checked="" type="checkbox"/>	m1.small	Linux/Unix	Shared	eu-west-1a	Heavy	2	\$55.44	\$71.26	\$108.01	100.00%	100.00%
<input checked="" type="checkbox"/>	m1.large	Linux/Unix	Shared	eu-west-1b	Heavy	15	\$1,663.20	\$1,726.49	\$3,417.87	95.00%	100.00%
<input type="checkbox"/>	m1.small	Linux/Unix	Shared	eu-west-1a	Medium	10	\$253.70	\$293.25	\$366.37	74.00%	100.00%
<input type="checkbox"/>	m1.medium	Linux/Unix	Shared	eu-west-1a	Heavy	4	\$151.76	\$172.96	\$387.08	100.00%	100.00%
<input type="checkbox"/>	m1.large	Linux/Unix	Shared	eu-west-1a	Heavy	17	\$1,884.96	\$1,948.25	\$3,663.33	92.00%	100.00%

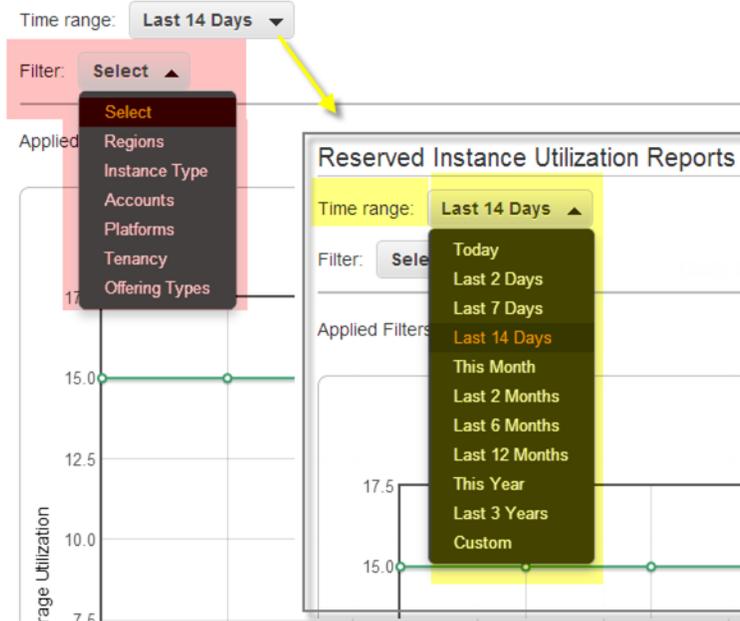
グラフには、選択したバケットのリザーブインスタンスの日次平均使用率が示されています。



## レポートをカスタマイズする

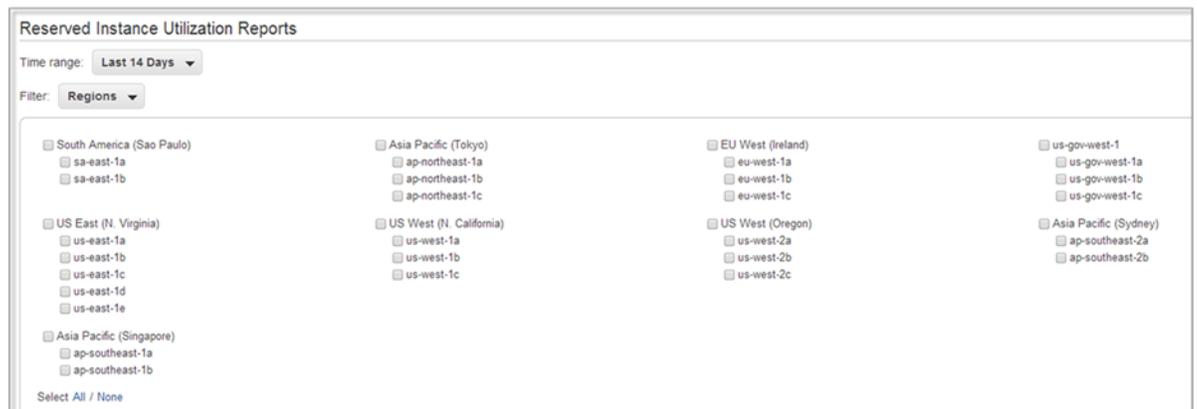
[Time range] および [Filter] オプションを使用して、リザーブインスタンスの使用率レポートをカスタマイズできます。

## Reserved Instance Utilization Reports



[Time range] では、一般的な相対時間範囲のリスト ( 今日 ~ 過去 3 年間 ) が示されます。ニーズに最も適した時間範囲を選択し、[Update Report] をクリックして変更を適用します。リストにない時間範囲を適用するには、[Custom] を選択し、レポート実行の開始日と終了日を入力します。

[Filter] を使用すると、リザーブインスタンスの使用率レポートを、リージョン、インスタンスタイプ、アカウント、プラットフォーム、テナンシー、および提供タイプのうち、1つ以上のリザーブインスタンスで絞り込むことができます。例えば、リージョン、リージョン内の特定のアベイラビリティゾーン、あるいはその両方でフィルタリングを行うことができます。リージョンでフィルタリングを行うには、[Regions] を選択し、レポートに含めるリージョンおよびアベイラビリティゾーンを選択して、[Update Report] をクリックします。



フィルタリングが何も適用されない場合、このレポートはすべての結果を返します。

レポートオプションについては、[オプションリファレンス \(p. 682\)](#) を参照してください。

## カスタムレポートをブックマークする

カスタムレポートを再度生成することができます。そのためには、レポートをブックマークしておきます。

カスタムレポートをブックマークするには

1. レポートのオプションとフィルタを選択します。項目を 1 つ選択すると、コンソール URL にパラメータが 1 つ追加されます。例えば、`granularity=Hourly` と `Filters` を選択すると、`filter_list` となります。
2. ブラウザを使用して、コンソール URL をブックマークに追加します。
3. 今後同じレポートを生成するには、作成したブックマークを使用します。

## 使用状況データをエクスポートする

レポートのグラフまたは表を他のレポートに含めることができます。そのためには、データをエクスポートします。

使用状況データをエクスポートするには

1. レポートのオプションとフィルタを選択します。
2. 使用状況データを `.csv` ファイルとしてテーブルからエクスポートするには、[Download] をクリックして [CSV Only] を選択します。
3. グラフ形式の使用状況データを `.png` としてエクスポートするには、[Download] をクリックして [Graph Only] を選択します。

## オプションリファレンス

レポートのグラフに表示するメトリックを指定するには、[Show] オプションを使用します。

- 平均使用率

選択した時間範囲における時間単位の使用率の平均が表示されます。1 時間のバケットの使用率は、その 1 時間に使用されたインスタンス時間数を、その 1 時間に所有されていたリザーブインスタンスの合計数で除算して計算されます。

- 最大使用率

選択した時間範囲における時間単位の使用率の最大値が表示されます。1 時間のバケットの使用率は、その 1 時間に使用されたインスタンス時間数を、その 1 時間に所有されていたリザーブインスタンスの合計数で除算して計算されます。

- 使用料

リザーブインスタンスの選択したバケットの時間単位料金に基づく合計コストが表示されます。重度使用リザーブインスタンスバケットの場合、期間内の時間数と、バケット内のリザーブインスタンスに関連付けられている時間単価を乗算して使用料が計算されます。中度使用および軽度使用リザーブインスタンスの場合、選択された期間に使用されるインスタンス時間数と、バケット内の各リザーブインスタンスの時間単価を乗算して使用料が計算されます。

- 合計コスト

レポート生成対象の期間における、使用料と、バケット内のリザーブインスタンスの前払い料金の償却部分の合計が表示されます。

レポートが基づく期間を指定するには、[Time range] を使用します。



## Note

すべての期間は UTC 時間で指定されます。

- 今日

現在の暦日の時間中に発生した使用状況のデータが表示されます。時間単位、日次単位、または月次単位で使用できます。

- 過去 2 日間

現在の暦日と直前の 1 暦日中に発生した使用状況のデータが表示されます。時間単位、日次単位、または月次単位で使用できます。

- 過去 7 日間

現在の暦日と直前の 6 暦日間に発生した使用状況のデータが表示されます。日次単位または月次単位で使用できます。

- 過去 14 日間

現在の暦日と直前の 13 暦日間に発生した使用状況のデータが表示されます。日次単位または月次単位で使用できます。

- 今月

現在の暦月中に発生した使用状況のデータが表示されます。日次単位または月次単位で使用できます。

- 過去 2 か月間

現在の暦月と直前の暦月中に発生した使用状況のデータが表示されます。日次単位または月次単位で使用できます。

- 過去 6 か月間

現在の暦月と直前の 5 暦月間に発生した使用状況のデータが表示されます。月次単位で使用できます。

- 過去 12 か月間

現在の暦月と直前の 11 暦月間に発生した使用状況のデータが表示されます。月次単位で使用できます。

- 今年

現在の暦年中に発生した使用状況のデータが表示されます。月次単位で使用できます。

- 過去 3 年間

現在の暦年と直前の 2 暦年間に発生した使用状況のデータが表示されます。月次単位で使用できます。

- カスタム

mm/dd/yyyy の形式で指定した開始日と終了日に基づく時間範囲のデータが表示されます。時間単位、日次単位、または月次単位で使用できますが、指定できる最大時間範囲は、時間単位のデータの場合は 2 日間、日次データの場合は 2 か月間、月次データの場合は 3 年間です。

レポートに表示するデータを絞り込むには、[Filter] を使用します。

- リージョン
- インスタンスタイプ
- アカウント
- プラットフォーム

- テナンシー
- 提供しているタイプ

# インスタンスのトラブルシューティング

---

## Abstract

インスタンスのトラブルシューティングを行います。

次のドキュメントは、インスタンスに関する問題を解決するために役立ちます。

### Topics

- インスタンスがすぐに終了する場合の対処方法 (p. 685)
- インスタンスへの接続に関するトラブルシューティング (p. 686)
- インスタンスの停止に関するトラブルシューティング (p. 691)
- インスタンスの終了 (シャットダウン) のトラブルシューティング (p. 693)
- ステータスチェックに失敗したインスタンスのトラブルシューティング (p. 693)
- インスタンスの容量に関するトラブルシューティング (p. 718)
- コンソール出力の取得とインスタンスの再起動 (p. 718)

Windows インスタンスの詳細なヘルプについては、『*Amazon Elastic Compute Cloud Microsoft Windows Guide*』の「[Troubleshooting Windows Instances](#)」を参照してください。

また、[Amazon EC2 forum](#) で答えを探し、質問を投稿することもできます。

## インスタンスがすぐに終了する場合の対処方法

### Abstract

インスタンスがすぐに終了する場合の問題を解決します。

インスタンスの起動後、そのステータスをチェックして、pending ステータスから terminated ステータスではなく running ステータスになったことを確認するようにお勧めします。

Amazon EBS-backed インスタンスがすぐに終了する理由を次にいくつか示します。

- ボリューム制限に達した。ボリューム制限に関する情報を確認し、ボリューム制限を緩和するリクエストを送信するには、「[Request to Increase the Amazon EBS Volume Limit](#)」を参照してください。
- AMI で必要な部分が欠けている。
- スナップショットが破損している。

## インスタンスの終了理由の取得

Amazon EC2 コンソール、CLI、または API を使用して、インスタンスが終了した理由に関する情報を取得できます。

インスタンスが終了した理由をコンソールを使用して調べるには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Instances] をクリックしてインスタンスの詳細を表示します。
3. インスタンスを選択します。
4. [Description] タブで、[State transition reason] ラベルの横にある理由を探します。インスタンスがまだ実行されている場合、通常は理由は表示されません。インスタンスを明示的に停止または終了した場合、理由は `User initiated shutdown` です。

インスタンスが終了した理由をコマンドラインを使用して調べるには

1. 次のように、`ec2-describe-instances` コマンドを詳細モードで使用します。

```
PROMPT> ec2-describe-instances instance_id -v
```

2. 表示された XML レスポンスで、`stateReason` 要素を見つけます。この要素は次の例のようになります。

```
<stateReason>  
  <code>Client.UserInitiatedShutdown</code>  
  <message>Client.UserInitiatedShutdown: User initiated shutdown</message>  
</stateReason>
```

この応答例は、実行中のインスタンスを停止または終了した後に表示される理由コードを示します。インスタンスがすぐに終了した場合、インスタンスが終了した理由 (例: `VolumeLimitExceeded`) を説明する `code` および `message` 要素が表示されます。

## インスタンスへの接続に関するトラブルシューティング

### Abstract

インスタンスへの接続に関するトラブルシューティングを行います。

以下では、発生する可能性のある問題、およびインスタンスへの接続時に表示される可能性のあるメッセージを示します。

### Topics

- [インスタンスへの接続エラー: 接続タイムアウト \(p. 687\)](#)

- エラー: ユーザーキーがサーバーによって認識されない (p. 688)
- エラー: Host key not found、Permission denied (publickey)、または Authentication failed, permission denied (ホストキーが見つかりません、権限の拒否 (publickey)、または認証失敗、権限の拒否) (p. 690)
- エラー: Unprotected Private Key File (保護されていないプライベートキーファイル) (p. 690)
- エラー: Server refused our key または No supported authentication methods available (サーバーはキーを拒否しましたまたは利用可能なサポートされる認証方法はありません) (p. 691)
- Safari ブラウザでの MindTerm 使用のエラー (p. 691)

Windows インスタンスの詳細なヘルプについては、『*Amazon Elastic Compute Cloud Microsoft Windows Guide*』の「[Troubleshooting Windows Instances](#)」を参照してください。

また、[Amazon EC2 forum](#) で答えを探し、質問を投稿することもできます。

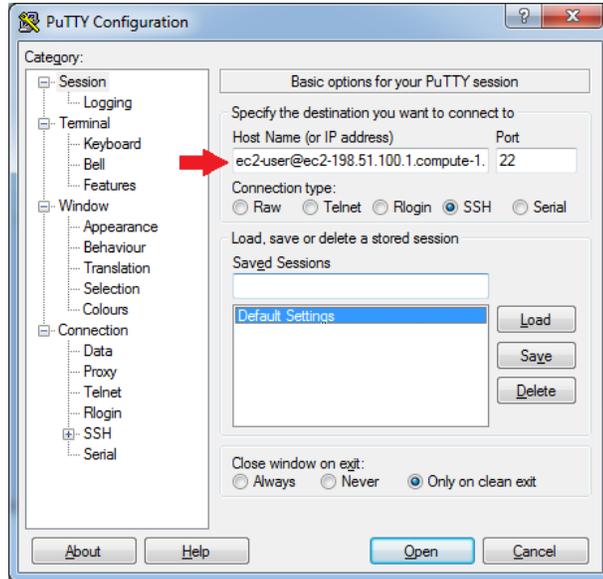
## インスタンスへの接続エラー: 接続タイムアウト

インスタンスへ接続しようとして、エラーメッセージ `Network error: Connection timed out` または `Error connecting to [instance], reason: -> Connection timed out: connect` が表示される場合、次を実行します。

- セキュリティグループルールを調べます。パブリック IP アドレスから適切なポートへのインバウンドトラフィックがセキュリティグループルールで許可されている必要があります。SSH を使用して Linux インスタンスに接続している場合、通常はポート 22 です。RDP を使用して Windows インスタンスに接続している場合、通常はポート 3389 です。詳細については、「[インスタンスへのネットワークアクセスの許可 \(p. 522\)](#)」を参照してください。
- [EC2-VPC] サブネットのルートテーブルを確認します。VPC (0.0.0.0/0) の外部へのすべてのトラフィックを VPC のインターネットゲートウェイに送信するには、ルートが必要です。
- サーバーが過負荷になっている可能性のあるインスタンスの CPU 負荷を確認します。AWS は自動的に Amazon CloudWatch メトリクスおよびインスタンスステータスなどのデータを提供します。これを使用してインスタンスの CPU 負荷を確認でき、必要に応じて、負荷の処理方法を調整できます。詳細については、「[CloudWatch によるインスタンスのモニタリング \(p. 395\)](#)」を参照してください。
- 負荷が変化する場合、[Auto Scaling](#) および [Elastic Load Balancing](#) を使用して、インスタンスの増減を自動的に縮小・拡張できます。
- 負荷が常に増加している場合、大きなインスタンスタイプに移動できます。詳細については、「[インスタンスのサイズ変更 \(p. 131\)](#)」を参照してください。
- インスタンスの起動時に選択されたキーペアに対応する秘密キーファイルを使用していることを確認します。詳細については、「[Amazon EC2 のキーペア \(p. 470\)](#)」を参照してください。
- AMI 用の適切なユーザー名で接続していることを確認します。
  - Amazon Linux AMI の場合は、ユーザー名は `ec2-user` です。
  - RHEL5 AMI の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。
  - Ubuntu AMI の場合は、ユーザー名は `ubuntu` です。
  - それ以外の場合は、AMI プロバイダーに確認してください。

MindTerm を使用して接続している場合、[Connect To Your Instance] ウィンドウの [User name] ボックスにユーザー名を入力します。

PuTTY を使用して接続している場合、[PuTTY Configuration] ウィンドウの [Host name] ボックスにユーザー名を入力します。



## エラー: ユーザーキーがサーバーによって認識されない

SSH を使用してインスタンスに接続している場合

- `ssh -vvv` を使用して、接続中に 3 倍詳細デバッグ情報を取得します。

```
#ssh -vvv -i [your key name].pem ec2-user@[public DNS address of your instance].compute-1.amazonaws.com
```

次のサンプル出力は、サーバーが認識しないキーを使用してインスタンスに接続しようとした場合に  
表示される可能性があります。

```
open/ANT/myusername/.ssh/known_hosts).
debug2: bits set: 504/1024
debug1: ssh_rsa_verify: signature correct
debug2: kex_derive_keys
debug2: set_newkeys: mode 1
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug2: set_newkeys: mode 0
debug1: SSH2_MSG_NEWKEYS received
debug1: Roaming not allowed by server
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug2: key: boguspem.pem ((nil))
debug1: Authentications that can continue: publickey
debug3: start over, passed a different list publickey
debug3: preferred gssapi-keyex,gssapi-with-mic,publickey,keyboard-interact
```

```
ive,password
debug3: authmethod_lookup publickey
debug3: remaining preferred: keyboard-interactive,password
debug3: authmethod_is_enabled publickey
debug1: Next authentication method: publickey
debug1: Trying private key: boguspem.pem
debug1: read PEM private key done: type RSA
debug3: sign_and_send_pubkey: RSA
9c:4c:bc:0c:d0:5c:c7:92:6c:8e:9b:16:e4:43:d8:b2
debug2: we sent a publickey packet, wait for reply
debug1: Authentications that can continue: publickey
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
Permission denied (publickey).
```

### SSH ( MindTerm ) を使用してインスタンスに接続している場合

- Java が有効になっていない場合、サーバーはユーザーキーを認識しません。Java を有効にするには、Java ドキュメントの [How do I enable Java in my web browser?](#) を参照してください。

### PuTTY を使用してインスタンスに接続している場合

- 秘密キー ( .pem ) ファイルが PuTTY によって認識される形式 ( .ppk ) に変換されていることを確認します。プライベートキーの変換の詳細については、「[PuTTY を使用した Windows から Linux/Unix インスタンスへの接続 \(p. 338\)](#)」を参照してください。



#### Note

PuTTYgen でプライベートキーファイルをロードし、[Generate] ではなく [Save Private Key] を選択します。

- AMI 用の適切なユーザー名で接続していることを確認します。[PuTTY Configuration] ウィンドウの [Host name] ボックスにユーザー名を入力します。
  - Amazon Linux AMI の場合は、ユーザー名は `ec2-user` です。
  - RHEL5 AMI の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。
  - Ubuntu AMI の場合は、ユーザー名は `ubuntu` です。
  - それ以外の場合は、AMI プロバイダーに確認してください。
- 適切なポートへのインバウンドトラフィックを許可しているインバウンドセキュリティグループがあることを確認します。詳細については、「[インスタンスへのネットワークアクセスの許可 \(p. 522\)](#)」を参照してください。



Amazon Elastic Compute Cloud ユーザーガイド  
エラー: Server refused our key または No supported authentication methods available ( サーバーはキーを拒否しましたまたは利用可能なサポートされる認証方法はありません )

## エラー: Server refused our key または No supported authentication methods available ( サーバーはキーを拒否しましたまたは利用可能なサポートされる認証方法はありません )

PuTTY を使用してインスタンスに接続し、Error: Server refused our key または Error: No supported authentication methods available のどちらかのエラーが発生した場合、AMI の適切なユーザー名で接続していることを確認します。[PuTTY Configuration] ウィンドウの [User name] ボックスにユーザー名を入力します。

適切なユーザー名は以下のとおりです。

- Amazon Linux AMI の場合は、ユーザー名は `ec2-user` です。
- RHEL5 AMI の場合は、ほとんどの場合ユーザー名は `root` ですが、`ec2-user` の場合もあります。
- Ubuntu AMI の場合は、ユーザー名は `ubuntu` です。
- それ以外の場合は、AMI プロバイダーに確認してください。

## Safari ブラウザでの MindTerm 使用のエラー

MindTerm を使用してインスタンスに接続し、Safari 6.1 または 7 ウェブブラウザを使用している場合、次のエラーを受け取ることがあります。

```
Error connecting to your_instance_ip, reason:  
-> Key exchange failed: Host authentication failed
```

ブラウザのセキュリティ設定を更新して、AWS Management Console で安全ではないモードでの Java プラグインの実行を許可する必要があります。

安全ではないモードでの Java プラグインの実行を有効にするには

1. Safari で、Amazon EC2 コンソールを開いたままにし、[Safari]、[Preferences]、[Security] の順に選択します。
2. [Manage Website Settings] をクリックします。
3. 左の Java プラグインを選択し、[Currently Open Websites] リストで AWS Management Console URL を見つけます。関連付けられているリストから [Run in Unsafe Mode] を選択します。
4. メッセージが表示されたら、警告ダイアログで [Trust] をクリックします。[Done] をクリックしてブラウザに戻ります。

## インスタンスの停止に関するトラブルシューティング

### Abstract

インスタンスが停止プロセスで「stuck」と表示された場合の問題を解決します。

Amazon EBS-Backed インスタンスを停止して `stopping` 状態に「stuck」が表示された場合、基になるホストコンピュータに問題がある可能性があります。

最初に、再度インスタンスの停止を試みます。[stop-instances](#) (AWS CLI) コマンドまたは [ec2-stop-instances](#) (Amazon EC2 CLI) コマンドを使用する場合は、必ず `--force` オプションを使用します。

インスタンスを強制停止できない場合は、インスタンスから AMI を作成し、代替のインスタンスを起動できます。

代替のインスタンスを作成するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[Instances] をクリックし、インスタンスを選択します。
3. [Actions] をクリックし、[Create Image] をクリックします。
4. [Create Image] ダイアログボックスで次のフィールドに入力し、[Create Image] をクリックします。
  - a. AMI の名前と説明を指定します。
  - b. [No reboot] を選択します。
5. AMI からインスタンスを起動し、インスタンスが動作していることを確認します。
6. 処理が止まってしまっているインスタンスを選択して、[Actions] をクリックし、次に [Terminate] をクリックします。インスタンスの終了処理も止まってしまう場合は、Amazon EC2 は数時間以内に自動的にそのインスタンスを強制終了します。

前の手順で説明されたように、インスタンスから AMI を作成できない場合は、次のようにして代替のインスタンスを設定できます。

代替のインスタンスを作成するには ( 前の手順が失敗する場合 )

1. インスタンスを選択し、[Description] タブを開いて、[Block devices] リストを表示します。各ボリュームを選択し、そのボリューム ID を書き留めます。必ずどのボリュームがルートボリュームであるかメモしておきます。
2. ナビゲーションペインの [Volumes] をクリックします。インスタンスの各ボリュームを選択し、[Actions] をクリックして、[Create Snapshot] をクリックします。
3. ナビゲーションペインで、[Snapshots] をクリックします。先ほど作成したスナップショットを選択し、[Actions] リストから [Create Volume] を選択します。
4. 処理が止まってしまったインスタンスと同じ種類のインスタンスを起動します ( Amazon Linux、Windows など )。そのルートボリュームのボリューム ID とデバイス名をメモしておきます。
5. ナビゲーションペインで [Instances] をクリックし、起動したインスタンスを選択して、[Stop] をクリックします。
6. ナビゲーションペインで [Volumes] をクリックし、停止したインスタンスのルートボリュームを選択し、[Actions] をクリックして、[Detach Volume] をクリックします。
7. 処理が停止してしまったインスタンスから作成したルートボリュームを選択し、[Attach Volume] をクリックして、そのルートボリュームとして新しいインスタンスにアタッチします ( 書き留めたデバイス名を使用 )。その他の非ルートボリュームをインスタンスにアタッチします。
8. ナビゲーションペインで、[Instances] をクリックし、代替のインスタンスを選択します。[Actions] をクリックし、次に [Start] をクリックします。インスタンスが動作していることを確認します。
9. 処理が止まってしまっているインスタンスを選択して、[Actions] をクリックし、次に [Terminate] をクリックします。インスタンスの終了処理も止まってしまう場合は、Amazon EC2 は数時間以内に自動的にそのインスタンスを強制終了します。

これらの手順を完了できない場合は、ヘルプリクエストを [Amazon EC2 forum](#) に投稿できます。迅速な解決のために、インスタンス ID を含めて、既に行った手順について説明してください。

# インスタンスの終了 ( シャットダウン ) のトラブルシューティング

## Abstract

インスタンスの終了またはシャットダウンに関する問題を解決します。

インスタンスが `running` 状態ではない場合、インスタンス時間に対しては課金されません。つまり、インスタンスを終了させると、そのステータスが `shutting-down` に変わるとすぐに、そのインスタンスへの課金は停止します。

## インスタンスの終了の遅延

インスタンスの `shutting-down` 状態が数分以上続く場合は、インスタンスによって実行されるシャットダウンスクリプトが原因で遅れている可能性があります。

もう 1 つ考えられる原因として、基盤となるホストコンピュータの問題があります。インスタンスの `shutting-down` 状態が数時間以上続く場合、Amazon EC2 はそれを停止したインスタンスとして扱い、強制終了します。

インスタンスの終了処理が停止していると考えられ、すでに数時間以上経過している場合は、[Amazon EC2 forum](#) にヘルプリクエストを投稿してください。迅速な解決のために、インスタンス ID を含めて、既に行った手順について説明してください。

## インスタンスを自動的に起動または終了する

すべてのインスタンスを終了すると、Amazon が新しいインスタンスを起動する場合があります。インスタンスを起動すると、Amazon がいずれかのインスタンスを終了する場合があります。通常このような動作は、お客様が `Auto Scaling` または `AWS Elastic Beanstalk` を使用して、お客様が定義した条件に基づいてコンピューティングリソースを自動的に拡大/縮小したために起こります。

詳細については、[Auto Scaling 開発者ガイド](#) または [AWS Elastic Beanstalk Developer Guide](#) を参照してください。

# ステータスチェックに失敗したインスタンスのトラブルシューティング

## Abstract

インスタンスがステータスチェックに失敗した場合のトラブルシューティングを行います。

### トピック

- [最初に行う手順 \(p. 694\)](#)
- [Linux ベースのインスタンスに対するインスタンスステータスチェックのトラブルシューティング \(p. 694\)](#)

## 最初に行う手順

インスタンスのステータスチェックが失敗した場合は、最初にアプリケーションで問題が発生しているかどうかを確認します。インスタンスでアプリケーションを正常に実行していないことを確認した場合、次の手順に従います。

AWS Management Console を使用して、問題のあるインスタンスを調査するには

1. Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [Instances] をクリックし、インスタンスを選択します。
3. 詳細ペインの [Status Checks] タブをクリックして、すべての [System Status Checks] と [Instance Status Checks] に関する個々の結果を表示します。

システムのステータスチェックに失敗した場合、次のいずれかの方法を試すことができます。

- Amazon EBS-Backed AMI を使用するインスタンスの場合、いったんインスタンスを停止してから再開します。
- Instance-store Backed AMI を使用するインスタンスの場合、インスタンスを終了し、代替りのインスタンスを起動します。
- Amazon EC2 が問題を解決するのを待ちます。
- 問題を [Amazon EC2 forum](#) に投稿します。

インスタンスのステータスチェックに失敗した場合、次の手順を行います。

1. インスタンスを右クリックしてから、[Reboot] をクリックします。システムが再起動するまでには数分かかることがあります。
2. 問題がまだ存在することを確認します。再起動によって、問題が解決することがあります。
3. インスタンスが `running` 状態になるまで待ちます。
4. インスタンスを右クリックし、[Get System Log] をクリックします。この情報が問題の特定に役立つことがあります。最近再起動してログの不必要な情報を消去したことを確認してください。
5. 画面に表示されるログを調べます。
6. 下記の既知のシステムログエラー文のリストを使用して、問題のトラブルシューティングを行います。
7. ステータスチェックの結果が実際とは異なる場合、またはステータスチェックでは検出されない問題がインスタンスに発生している場合、[Status Checks] タブの下部にある [Submit feedback] をクリックして、検出テストの改善にご協力ください。
8. 問題が解決されない場合は、問題を [Amazon EC2 forum](#) に投稿できます。

## Linux ベースのインスタンスに対するインスタンスステータスチェックのトラブルシューティング

インスタンスの接続性チェックなど、インスタンスのステータスチェックに失敗した Linux ベースのインスタンスの場合、上記の手順に従ってシステムログを取得したことを確認します。次のリストは、一般的なシステムログエラー、および各エラーの問題解決に対して推奨する対処を示しています。

### Memory Errors

- [メモリ不足: プロセスの終了 \(p. 695\)](#)
- [エラー: mmu\\_update failed \(メモリ管理の更新に失敗しました\) \(p. 696\)](#)

### Device Errors

- I/O エラー ( ブロックデバイス障害 ) ( p. 697 )
- I/O エラー: ローカルでもリモートディスクでもありません ( 破損した分散ブロックデバイス ) ( p. 698 )

### Kernel Errors

- request\_module: runaway loop modprobe ( 古い Linux バージョンでレガシーカーネル modprobe がループしている ) ( p. 699 )
- 「FATAL: kernel too old」および「fsck: No such file or directory while trying to open /dev」 ( カーネルと AMI の不一致 ) ( p. 700 )
- 「FATAL: Could not load /lib/modules」または「BusyBox」 ( カーネルモジュールの欠如 ) ( p. 701 )
- エラー: 無効のカーネル ( EC2 と互換性のないカーネル ) ( p. 702 )

### File System Errors

- request\_module: runaway loop modprobe ( 古い Linux バージョンでレガシーカーネル modprobe がループしている ) ( p. 704 )
- fsck: No such file or directory while trying to open... ( ファイルシステムが見つからない ) ( p. 705 )
- General error mounting filesystems ( マウント失敗 ) ( p. 706 )
- VFS: Unable to mount root fs on unknown-block ( ルートファイルシステム不一致 ) ( p. 708 )
- エラー: Unable to determine major/minor number of root device... ( ルートファイルシステム/デバイス不一致 ) ( p. 709 )
- XENBUS: Device with no driver... ( p. 710 )
- ... days without being checked, check forced ( ファイルシステムのチェックが必要です ) ( p. 711 )
- fsck died with exit status... ( デバイスが見つかりません ) ( p. 712 )

### Operating System Errors

- GRUB プロンプト ( grubdom> ) ( p. 713 )
- Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. ( ハードコードされた MAC アドレス ) ( p. 715 )
- SELinux ポリシーを読み込めません。Machine is in enforcing mode. Halting now. ( SELinux の誤設定 ) ( p. 716 )
- XENBUS: Timeout connecting to devices ( Xenbus タイムアウト ) ( p. 717 )

## メモリ不足: プロセスの終了

メモリ不足エラーは、下記のようなシステムログで示されます。

```
[115879.769795] Out of memory: kill process 20273 (httpd) score 1285879  
or a child  
[115879.769795] Killed process 1917 (php-cgi) vsz:467184kB, anon-  
rss:101196kB, file-rss:204kB
```

## 可能性のある原因

メモリの枯渇

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次のいずれかを行ってください。 <ul style="list-style-type: none"><li>インスタンスを停止し、異なるインスタンスタイプを使用するようにインスタンスを変更した後、インスタンスを再び起動します。例えば、大きいインスタンスタイプやメモリ最適化インスタンスタイプです。</li><li>インスタンスを再起動して、障害のないステータスに戻します。インスタンスタイプを変更しない限り、問題が再び発生する可能性があります。</li></ul>
Instance store-Backed	以下のいずれかを行ってください。 <ul style="list-style-type: none"><li>インスタンスを終了し、別のインスタンスタイプを指定して、新しいインスタンスを起動します。例えば、大きいインスタンスタイプやメモリ最適化インスタンスタイプです。</li><li>インスタンスを再起動して、障害のないステータスに戻します。インスタンスタイプを変更しない限り、問題が再び発生する可能性があります。</li></ul>

## エラー: mmu\_update failed (メモリ管理の更新に失敗しました)

メモリ管理更新失敗は、下記のようなシステムログで示されます。

```
...
Press `ESC' to enter the menu... 0 [H[J Booting 'Amazon Linux 2011.09
(2.6.35.14-95.38.amzn1.i686)']

root (hd0)

Filesystem type is ext2fs, using whole disk

kernel /boot/vmlinuz-2.6.35.14-95.38.amzn1.i686 root=LABEL=/ console=hvc0 LANG=en_US.UTF-8 KEYTABLE=us

initrd /boot/initramfs-2.6.35.14-95.38.amzn1.i686.img

ERROR: mmu_update failed with rc=-22
```

## 可能性のある原因

Amazon Linux に関する問題

## 推奨する対処

開発者フォーラムに問題を投稿するか [AWS サポート](#) に連絡して、支援を求めることができます。

## I/O エラー ( ブロックデバイス障害 )

入力/出力エラーは、次の例のようなシステムログで示されます。

```
[9943662.053217] end_request: I/O error, dev sde, sector 52428288
[9943664.191262] end_request: I/O error, dev sde, sector 52428168
[9943664.191285] Buffer I/O error on device md0, logical block 209713024
[9943664.191297] Buffer I/O error on device md0, logical block 209713025
[9943664.191304] Buffer I/O error on device md0, logical block 209713026
[9943664.191310] Buffer I/O error on device md0, logical block 209713027
[9943664.191317] Buffer I/O error on device md0, logical block 209713028
[9943664.191324] Buffer I/O error on device md0, logical block 209713029
[9943664.191332] Buffer I/O error on device md0, logical block 209713030
[9943664.191339] Buffer I/O error on device md0, logical block 209713031
[9943664.191581] end_request: I/O error, dev sde, sector 52428280
[9943664.191590] Buffer I/O error on device md0, logical block 209713136
[9943664.191597] Buffer I/O error on device md0, logical block 209713137
[9943664.191767] end_request: I/O error, dev sde, sector 52428288
[9943664.191970] end_request: I/O error, dev sde, sector 52428288
[9943664.192143] end_request: I/O error, dev sde, sector 52428288
[9943664.192949] end_request: I/O error, dev sde, sector 52428288
[9943664.193112] end_request: I/O error, dev sde, sector 52428288
[9943664.193266] end_request: I/O error, dev sde, sector 52428288
...
```

## 可能性のある原因

インスタンスタイプ	可能性のある原因
Amazon EBS-Backed	障害が発生した Amazon EBS ボリューム
Instance store-Backed	障害が発生した物理ドライブ

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	<p>次の手順を実行してください。</p> <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. ボリュームをデタッチします。</li><li>3. ボリュームの回復を試みます。</li></ol> <p> <b>Tip</b></p> <p>Amazon EBS ボリュームのスナップショットを頻繁に作成することをお勧めします。これによって、エラーのためにデータを損失する危険性が大幅に減少します。</p> <ol style="list-style-type: none"><li>4. ボリュームにインスタンスに再アタッチします。</li><li>5. ボリュームをデタッチします。</li></ol>
Instance store-Backed	<p>インスタンスを終了し、新しいインスタンスを起動します。</p> <p> <b>Note</b></p> <p>データを復旧できない。バックアップから復旧します。</p> <p> <b>Tip</b></p> <p>Amazon S3 または Amazon EBS をバックアップに使用することをお勧めします。インスタンスストアボリュームは、単一のホストと単一のディスクエラーに直接結びついています。</p>

## I/O エラー: ローカルでもリモートディスクでもありません ( 破損した分散ブロックデバイス )

デバイスでの入力/出力エラーは、次の例のようなシステムログで示されます。

```
...  
block drbd1: Local IO failed in request_timer_fn. Detaching...  
  
Aborting journal on device drbd1-8.
```

```
block drbd1: IO ERROR: neither local nor remote disk  
Buffer I/O error on device drbd1, logical block 557056  
lost page write due to I/O error on drbd1  
JBD2: I/O error detected when updating journal superblock for drbd1-8.
```

## 可能性のある原因

インスタンスタイプ	可能性のある原因
Amazon EBS-Backed	障害が発生した Amazon EBS ボリューム
Instance store-Backed	障害が発生した物理ドライブ

## 推奨する対処

インスタンスを終了し、新しいインスタンスを起動します。

Amazon EBS-Backed インスタンスの場合、最新スナップショットからイメージを作成して、データを回復できます。スナップショットを作成した後に追加されたデータは回復できません。

## request\_module: runaway loop modprobe (古い Linux バージョンでレガシーカーネル modprobe がループしている)

下記のようなシステムログで、この状態が示されます。不安定であるか古い Linux カーネル (例: 2.6.16-xenU) を使用すると、起動時に無限ループが発生することがあります。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1  
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007  
BIOS-provided physical RAM map:  
  
Xen: 0000000000000000 - 0000000026700000 (usable)  
  
OMB HIGHMEM available.  
...  
  
request_module: runaway loop modprobe binfmt-464c  
request_module: runaway loop modprobe binfmt-464c
```

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次のいずれかのオプションを使用して、Grubペー スまたは静的な新しいカーネルを使用します。  オプション 1: インスタンスを終了し、 <code>-kernel</code> および <code>-ramdisk</code> パラメータを指定して新しいイ ンスタンスを起動します。  オプション 2:  1. インスタンスを停止します。 2. 新しいカーネルを使用するようカーネルとラム ディスク属性を変更します。 3. インスタンスを起動します。
Instance store-Backed	インスタンスを終了し、 <code>-kernel</code> および <code>-ramdisk</code> パラメータを指定して新しいインス タンスを起動します。

## 「FATAL: kernel too old」 および 「fsck: No such file or directory while trying to open /dev」 (カーネルと AMI の不一致)

下記のようなシステムログで、この状態が示されます。

```
Linux version 2.6.16.33-xenU (root@dom0-0-50-45-1-a4-ee.z-2.aes0.internal)
(gcc version 4.1.1 20070105 (Red Hat 4.1.1-52)) #2 SMP Wed Aug 15 17:27:36 SAST
2007
...
FATAL: kernel too old
Kernel panic - not syncing: Attempted to kill init!
```

## 可能性のある原因

互換性のないカーネルとユーザーランド

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次の手順を実行してください。  1. インスタンスを停止します。 2. 新しいカーネルを使用するよう設定を変更しま す。 3. インスタンスを起動します。

インスタンスタイプ	処理
Instance store-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>より新しいカーネルを使用する AMI を作成します。</li><li>インスタンスを終了します。</li><li>作成した AMI から新しいインスタンスを起動します。</li></ol>

## 「FATAL: Could not load /lib/modules」または「BusyBox」 (カーネルモジュールの欠如)

下記のようなシステムログで、この状態が示されます。

```
[ 0.370415] Freeing unused kernel memory: 1716k freed
Loading, please wait...
WARNING: Couldn't open directory /lib/modules/2.6.34-4-virtual: No such file
or directory
FATAL: Could not open /lib/modules/2.6.34-4-virtual/modules.dep.temp for writing:
No such file or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
Couldn't get a file descriptor referring to the console
Begin: Loading essential drivers... ...
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
Done.
Begin: Running /scripts/init-premount ...
Done.
Begin: Mounting root file system... ...
Begin: Running /scripts/local-top ...
Done.
Begin: Waiting for root file system... ...
Done.
Gave up waiting for root device. Common problems:
- Boot args (cat /proc/cmdline)
  - Check rootdelay= (did the system wait long enough?)
  - Check root= (did the system wait for the right device?)
- Missing modules (cat /proc/modules; ls /dev)
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
FATAL: Could not load /lib/modules/2.6.34-4-virtual/modules.dep: No such file
or directory
ALERT! /dev/sdal does not exist. Dropping to a shell!

BusyBox v1.13.3 (Ubuntu 1:1.13.3-lubuntu5) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```
(initramfs)
```

## 可能性のある原因

次の 1 つ以上の条件によって、この問題が発生する可能性があります。

- ラムディスクが見つからない
- ラムディスクに正しいモジュールが見つからない
- Amazon EBS ルートボリュームが `/dev/sda1` として正しくアタッチされていない

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. Amazon EBS ボリュームに対して正しいラムディスクを選択します。</li><li>2. インスタンスを停止します。</li><li>3. ボリュームをデタッチし、修復します。</li><li>4. ボリュームをインスタンスにアタッチします。</li><li>5. インスタンスを起動します。</li><li>6. 正しいラムディスクを使用するよう AMI を変更します。</li></ol>
Instance store-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. インスタンスを終了してから、正しいラムディスクを使って新たなインスタンスを起動します。</li><li>2. 正しいラムディスクを使って新たな AMI を作成します。</li></ol>

## エラー: 無効のカーネル ( EC2 と互換性のないカーネル )

下記のようなシステムログで、この状態が示されます。

```
...  
root (hd0)  
  
Filesystem type is ext2fs, using whole disk
```

```
kernel /vmlinuz root=/dev/sda1 ro

initrd /initrd.img

ERROR Invalid kernel: elf_xen_note_check: ERROR: Will only load images
built for the generic loader or Linux images
xc_dom_parse_image returned -1

Error 9: Unknown boot failure

  Booting 'Fallback'

root (hd0)

  Filesystem type is ext2fs, using whole disk

kernel /vmlinuz.old root=/dev/sda1 ro

Error 15: File not found
```

## 可能性のある原因

次の一方または両方の条件によって、この問題が発生する可能性があります。

- 指定されたカーネルは Grub でサポートされていません
- フォールバックカーネルが存在しません

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. 機能するカーネルに変更します。</li><li>3. フォールバックカーネルをインストールします。</li><li>4. カーネルを訂正して AMI を変更します。</li></ol>
Instance store-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. インスタンスを終了してから、正しいカーネルを使って新たなインスタンスを起動します。</li><li>2. 正しいカーネルを使って AMI を作成します。</li><li>3. ( オプション ) <a href="#">AWS Support</a> を使用してデータ回復の技術サポートを求めます。</li></ol>

## request\_module: runaway loop modprobe (古い Linux バージョンでレガシーカーネル modprobe がループしている)

下記のようなシステムログで、この状態が示されます。不安定であるか古い Linux カーネル (例: 2.6.16-xenU) を使用すると、起動時に無限ループが発生することがあります。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1  
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
```

```
BIOS-provided physical RAM map:
```

```
Xen: 0000000000000000 - 0000000026700000 (usable)
```

```
OMB HIGHMEM available.
```

```
...
```

```
request_module: runaway loop modprobe binfmt-464c
```

### 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次のいずれかのオプションを使用して、Grub ベースまたは静的な新しいカーネルを使用します。  オプション 1: インスタンスを終了し、 <code>-kernel</code> および <code>-ramdisk</code> パラメータを指定して新しいインスタンスを起動します。  オプション 2: <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. 新しいカーネルを使用するようカーネルとラムディスク属性を変更します。</li><li>3. インスタンスを起動します。</li></ol>
Instance store-Backed	インスタンスを終了し、 <code>-kernel</code> および <code>-ramdisk</code> パラメータを指定して新しいインスタンスを起動します。

## fsck: No such file or directory while trying to open... ( ファイルシステムが見つからない )

下記のようなシステムログで、この状態が示されます。

```
Welcome to Fedora
Press 'I' to enter interactive startup.
Setting clock : Wed Oct 26 05:52:05 EDT 2011 [ OK ]

Starting udev: [ OK ]

Setting hostname localhost: [ OK ]

No devices found
Setting up Logical Volume Management: File descriptor 7 left open
  No volume groups found
[ OK ]

Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1: clean, 82081/1310720 files, 2141116/2621440 blocks
[/sbin/fsck.ext3 (1) -- /mnt/dbbackups] fsck.ext3 -a /dev/sdh
fsck.ext3: No such file or directory while trying to open /dev/sdh

/dev/sdh:
The superblock could not be read or does not describe a correct ext2
filesystem.  If the device is valid and it really contains an ext2
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>

[FAILED]

*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell.
Give root password for maintenance
(or type Control-D to continue):
```

### 可能性のある原因

- ラムディスクファイルシステム定義 /etc/fstab にバグがある
- /etc/fstab のファイルシステム定義の設定が不適切
- ドライブが見つからないかドライブにエラーがある

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	<p>次の手順を実行してください。</p> <ol style="list-style-type: none"><li>1. インスタンスを停止し、ルートボリュームをデタッチし、/etc/fstab を修復または変更し、ボリュームをインスタンスにアタッチし、インスタンスを起動します。</li><li>2. ラムディスクを修正して、変更した /etc/fstab を含めます ( 適用可能な場合 )。</li><li>3. 新しいラムディスクを使用するよう AMI を変更します。</li></ol> <p> Tip</p> <p>fstab の 6 番目のフィールドではマウントの可用性要件を定義します。0 以外の値を指定した場合、そのボリュームに対して fsck を実行して成功しなければならないことを意味します。Amazon EC2 でこのフィールドを使用すると、問題が発生することがあります。これは、実行に失敗すると通常は対話的なコンソールプロンプトが表示されますが、このコンソールプロンプトは現在 Amazon EC2 で使用できないためです。この機能は慎重に使用してください。また、fstab の Linux マニュアルページを参照してください。</p>
Instance store-Backed	<p>次の手順を実行してください。</p> <ol style="list-style-type: none"><li>1. インスタンスを終了し、新しいインスタンスを起動します。</li><li>2. 障害のある Amazon EBS ボリュームをデタッチし、インスタンスを再起動します。</li><li>3. ( オプション ) <a href="#">AWS Support</a> を使用してデータ回復の技術サポートを求めます。</li></ol>

## General error mounting filesystems ( マウント失敗 )

下記のようなシステムログで、この状態が示されます。

```
Loading xenblk.ko module
xen-vbd: registered block device major 8

Loading ehci-hcd.ko module
```

```
Loading ohci-hcd.ko module
Loading uhci-hcd.ko module
USB Universal Host Controller Interface driver v3.0

Loading mbcache.ko module
Loading jbd.ko module
Loading ext3.ko module
Creating root device.
Mounting root filesystem.
kjournald starting. Commit interval 5 seconds

EXT3-fs: mounted filesystem with ordered data mode.

Setting up other filesystems.
Setting up new root fs
no fstab.sys, mounting internal defaults
Switching to new root and running init.
unmounting old /dev
unmounting old /proc
unmounting old /sys
mountall:/proc: unable to mount: Device or resource busy
mountall:/proc/self/mountinfo: No such file or directory
mountall: root filesystem isn't mounted
init: mountall main process (221) terminated with status 1

General error mounting filesystems.
A maintenance shell will now be started.
CONTROL-D will terminate this shell and re-try.
Press enter for maintenance
(or type Control-D to continue):
```

## 可能性のある原因

インスタンスタイプ	可能性のある原因
Amazon EBS-Backed	<ul style="list-style-type: none"><li>• Amazon EBS ボリュームがデタッチされているか、Amazon EBS ボリュームにエラーがあります。</li><li>• 破損したファイルシステム。</li><li>• ラムディスクと AMI の組み合わせが一致していません ( 例: Debian ラムディスクと SUSE AMI )。</li></ul>
Instance store-Backed	<ul style="list-style-type: none"><li>• ドライブにエラーがあります。</li><li>• ファイルシステムが破損しています。</li><li>• ラムディスクと AMI の組み合わせが一致していません ( 例: Debian ラムディスクと SUSE AMI )。</li></ul>

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. ルートボリュームをデタッチします。</li><li>3. ルートボリュームを既知の動作しているインスタンスにアタッチします。</li><li>4. ファイルシステムチェックを実行します ( <code>fsck -a /dev/...</code> )。</li><li>5. エラーを修正します。</li><li>6. ボリュームを既知の動作しているインスタンスからデタッチします。</li><li>7. 停止したインスタンスにボリュームをアタッチします。</li><li>8. インスタンスを起動します。</li><li>9. インスタンスのステータスを再確認します。</li></ol>
Instance store-Backed	以下のいずれかを行ってください。 <ul style="list-style-type: none"><li>• 新しいインスタンスを起動します。</li><li>• ( オプション ) <a href="#">AWS Support</a> を使用してデータ回復の技術サポートを求めます。</li></ul>

## VFS: Unable to mount root fs on unknown-block ( ルートファイルシステム不一致 )

下記のようなシステムログで、この状態が示されます。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
Kernel command line: root=/dev/sdal ro 4
...
Registering block device major 8
...
Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(8,1)
```

## 可能性のある原因

インスタンスタイプ	可能性のある原因
Amazon EBS-Backed	<ul style="list-style-type: none"><li>• デバイスが正しくアタッチされていない。</li><li>• ルートデバイスが正しいデバイスポイントにアタッチされていない。</li><li>• ファイルシステムのフォーマットが正しくありません。</li><li>• レガシーカーネルを使用しています (例: 2.6.16-XenU)。</li></ul>
Instance store-Backed	ハードウェアデバイスのエラー。

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	<p>次のいずれかを行ってください。</p> <ul style="list-style-type: none"><li>• インスタンスを停止し、再起動します。</li><li>• 正しいデバイスポイントでアタッチするようルートボリュームを変更します。例えば、/dev/sda の代わりに /dev/sda1 を使用します。</li><li>• 停止し、新しいカーネルを使用するように変更します。</li></ul>
Instance store-Backed	インスタンスを終了し、新しいカーネルを使用して、新しいインスタンスを起動します。

## エラー: Unable to determine major/minor number of root device... ( ルートファイルシステム/デバイス不一致 )

下記のようなシステムログで、この状態が示されます。

```
...
XENBUS: Device with no driver: device/vif/0
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
:: Starting udevd...
done.
```

```
:: Running Hook [udev]
:: Triggering uevents...<30>udev[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

## 可能性のある原因

- 仮想ブロックデバイスドライバが見つからないか、設定が間違っている
- デバイス列挙が競合している ( sda と xvda または sda1 の代わりに sda )
- DomU カーネルが正しく選択されていない

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. ボリュームをデタッチします。</li><li>3. デバイスのマッピングの問題を解決します。</li><li>4. インスタンスを起動します。</li><li>5. AMI を変更して、デバイスのマッピングの問題に対処します。</li></ol>
Instance store-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. 適切な修正を使用して新しい AMI を作成します ( ブロックデバイスを正しくマッピングします )。</li><li>2. インスタンスを終了し、作成した AMI から新しいインスタンスを起動します。</li></ol>

## XENBUS: Device with no driver...

下記のようなシステムログで、この状態が示されます。

```
XENBUS: Device with no driver: device/vbd/2048
drivers/rtc/hctosys.c: unable to open rtc device (rtc0)
Initializing network drop monitor service
Freeing unused kernel memory: 508k freed
```

```
:: Starting udevd...
done.
:: Running Hook [udev]
:: Triggering uevents...<30>udev[65]: starting version 173
done.
Waiting 10 seconds for device /dev/xvda1 ...
Root device '/dev/xvda1' doesn't exist. Attempting to create it.
ERROR: Unable to determine major/minor number of root device '/dev/xvda1'.
You are being dropped to a recovery shell
    Type 'exit' to try and continue booting
sh: can't access tty; job control turned off
[ramfs /]#
```

## 可能性のある原因

- 仮想ブロックデバイスドライバが見つからないか、設定が間違っている
- デバイス列挙が競合している ( sda と xvda )。
- DomU カーネルが正しく選択されていない

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. ボリュームをデタッチします。</li><li>3. デバイスのマッピングの問題を解決します。</li><li>4. インスタンスを起動します。</li><li>5. AMI を変更して、デバイスのマッピングの問題に対処します。</li></ol>
Instance store-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. 適切な修正を使用して AMI を作成します ( ブロックデバイスを正しくマッピングします )。</li><li>2. インスタンスを終了し、作成した AMI を使用して新しいインスタンスを起動します。</li></ol>

## ... days without being checked, check forced ( ファイルシステムのチェックが必要です )

下記のようなシステムログで、この状態が示されます。

```
...
Checking filesystems
Checking all file systems.
[/sbin/fsck.ext3 (1) -- /] fsck.ext3 -a /dev/sda1
/dev/sda1 has gone 361 days without being checked, check forced
```

## 可能性のある原因

ファイルシステムのチェック期間が経過したため、ファイルシステムチェックが強制実行されている。

## 推奨する対処

- ファイルシステムチェックが完了するまで待ちます。ルートファイルシステムのサイズによっては、ファイルシステムチェックに時間がかかることがあります。
- tune2fs またはファイルシステムに適したツールを使用してファイルシステムを変更し、ファイルシステムチェック ( fsck ) の実行を削除します。

## fsck died with exit status... ( デバイスが見つかりません )

下記のようなシステムログで、この状態が示されます。

```
Cleaning up ifupdown....
Loading kernel modules...done.
...
Activating lvm and md swap...done.
Checking file systems...fsck from util-linux-ng 2.16.2
/sbin/fsck.xfs: /dev/sdh does not exist
fsck died with exit status 8
[31mfailed (code 8).[39;49m
```

## 可能性のある原因

- 存在しないドライブをラムディスクが検索している
- ファイルシステムの整合性チェックが強制実行されている
- ドライブにエラーがあるか、デタッチされている

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	<p>問題を解決するため、次のいずれかを試してください。</p> <ul style="list-style-type: none"><li>• インスタンスを停止し、ボリュームを既存の実行中インスタンスにアタッチします。</li><li>• 整合性チェックを手動で実行します。</li><li>• ラムディスクを修正して、関連するユーティリティを含めます。</li><li>• ファイルシステム調整パラメータを変更して、整合性要件を削除します ( お勧めしません ) 。</li></ul>
Instance store-Backed	<p>問題を解決するため、次のいずれかを試します。</p> <ul style="list-style-type: none"><li>• ラムディスクに正しいツールを再バンドリングします。</li><li>• ファイルシステム調整パラメータを変更して、整合性要件を削除します ( お勧めしません ) 。</li><li>• インスタンスを終了し、新しいインスタンスを起動します。</li><li>• ( オプション ) <a href="#">AWS Support</a> を使用してデータ回復の技術サポートを求めます。</li></ul>

## GRUB プロンプト ( grubdom> )

下記のようなシステムログで、この状態が示されます。

```
GNU GRUB  version 0.97  (629760K lower / 0K upper memory)

[ Minimal BASH-like line editing is supported.  For
the first word, TAB lists possible command
completions.  Anywhere else TAB lists the possible
completions of a device/filename. ]

grubdom>
```

## 可能性のある原因

インスタンスタイプ	可能性のある原因
Amazon EBS-Backed	<ul style="list-style-type: none"> <li>• grub.conf ファイルが見つかりません。</li> <li>• 間違った Grub イメージが使用されており、異なる場所にある grub.conf が必要です。</li> <li>• grub.conf の保存に、サポートされていないファイルシステムが使用されています。</li> </ul>
Instance store-Backed	<ul style="list-style-type: none"> <li>• grub.conf ファイルが見つかりません。</li> <li>• 間違った Grub イメージが使用されており、異なる場所にある grub.conf が必要です。</li> </ul>

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	<p>オプション 1: AMI を変更しインスタンスを再起動します</p> <ol style="list-style-type: none"> <li>1. ソース AMI を変更して、標準の場所に grub.conf を作成します ( /boot/grub/menu.lst )。</li> <li>2. 正しい Grub イメージを選択します ( hd0 – 第 1 ドライブまたは hd00 – 第 1 ドライブ、第 1 パーティション )。</li> <li>3. インスタンスを終了し、作成した AMI を使用して新しいインスタンスを起動します。</li> </ol> <p>オプション 2: 既存のインスタンスを修正します</p> <ol style="list-style-type: none"> <li>1. インスタンスを停止します。</li> <li>2. ルートファイルシステムをデタッチします。</li> <li>3. ルートファイルシステムを既知の動作しているインスタンスにアタッチします。</li> <li>4. ファイルシステムをマウントします。</li> <li>5. grub.conf を作成します。</li> <li>6. ファイルシステムをデタッチします。</li> <li>7. 元のインスタンスにアタッチします。</li> <li>8. カーネル属性を変更して、正しい Grub イメージを使用します ( 第 1 ディスクまたは第 1 ディスクの第 1 パーティション )。</li> <li>9. インスタンスを起動します。</li> </ol>

Amazon Elastic Compute Cloud ユーザーガイド  
Bringing up interface eth0: Device eth0 has different MAC  
address than expected, ignoring. (ハードコードされた MAC  
アドレス)

インスタンスタイプ	処理
Instance store-Backed	<p>オプション 1: AMI を変更しインスタンスを再起動します。</p> <ol style="list-style-type: none"><li>標準の場所 ( /boot/grub/menu.lst ) で、grub.conf とともに新しい AMI を作成します。</li><li>適切な Grub イメージを選択します ( hd0 – 第 1 ドライブまたは hd00 – 第 1 ドライブ、第 1 パーティション )。</li><li>インスタンスを終了し、作成した AMI を使用して新しいインスタンスを起動します。</li></ol> <p>オプション 2: インスタンスを終了し、正しいカーネルを指定して新しいインスタンスを起動します。</p> <p> <b>Note</b></p> <p>既存のインスタンスからデータを回復するには、AWS サポートにお問い合わせください。</p>

## Bringing up interface eth0: Device eth0 has different MAC address than expected, ignoring. (ハードコードされた MAC アドレス)

下記のようなシステムログで、この状態が示されます。

```
...  
Bringing up loopback interface: [ OK ]  
  
Bringing up interface eth0: Device eth0 has different MAC address than expected,  
ignoring.  
[FAILED]  
  
Starting auditd: [ OK ]
```

### 可能性のある原因

AMI 設定にハードコードされたインターフェイス MAC があります。

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	<p>次のいずれかを行ってください。</p> <ul style="list-style-type: none"><li>AMI を変更してハードコードを削除し、インスタンスを再起動します。</li><li>インスタンスを変更して、ハードコードされた MAC アドレスを削除します。</li></ul> <p>または</p> <p>次の手順に従ってください。</p> <ol style="list-style-type: none"><li>インスタンスを停止します。</li><li>ルートボリュームをデタッチします。</li><li>ボリュームを別のインスタンスにアタッチし、ボリュームを変更してハードコードされた MAC アドレスを削除します。</li><li>初期インスタンスにボリュームをアタッチします。</li><li>インスタンスを起動します。</li></ol>
Instance store-Backed	<p>次のいずれかを行ってください。</p> <ul style="list-style-type: none"><li>インスタンスを変更して、ハードコードされた MAC アドレスを削除します。</li><li>インスタンスを終了し、新しいインスタンスを起動します。</li></ul>

## SELinux ポリシーを読み込めません。Machine is in enforcing mode. Halting now. ( SELinux の誤設定 )

下記のようなシステムログで、この状態が示されます。

```
audit(1313445102.626:2): enforcing=1 old_enforcing=0 auid=4294967295
Unable to load SELinux Policy. Machine is in enforcing mode. Halting now.
Kernel panic - not syncing: Attempted to kill init!
```

## 可能性のある原因

SELinux が誤って有効にされた。

- Grub 指定されたカーネルではサポートされていません。
- フォールバックカーネルが存在しません。

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. インスタンスを停止します。</li><li>2. ボリュームをデタッチします。</li><li>3. SELinux を無効にします。</li><li>4. インスタンスを起動します。</li></ol>
Instance store-Backed	次の手順を実行してください。 <ol style="list-style-type: none"><li>1. インスタンスを終了し、新しいインスタンスを起動します。</li><li>2. ( オプション ) <a href="#">AWS Support</a> を使用してデータ回復の技術サポートを求めます。</li></ol>

## XENBUS: Timeout connecting to devices ( Xenbus タイムアウト )

下記のようなシステムログで、この状態が示されます。

```
Linux version 2.6.16-xenU (builder@xenbat.amazonsa) (gcc version 4.0.1
20050727 (Red Hat 4.0.1-5)) #1 SMP Mon May 28 03:41:49 SAST 2007
...
XENBUS: Timeout connecting to devices!
...
Kernel panic - not syncing: No init found. Try passing init= option to kernel.
```

## 可能性のある原因

- ブロックデバイスがインスタンスに接続されていない。
- このインスタンスは古い DomU カーネルを使用している。

## 推奨する対処

インスタンスタイプ	処理
Amazon EBS-Backed	次のいずれかを行ってください。 <ul style="list-style-type: none"><li>• AMI とインスタンスを変更して新しいカーネルを使用し、インスタンスを再起動します。</li><li>• インスタンスを再起動します。</li></ul>

インスタンスタイプ	処理
Instance store-Backed	次のいずれかを行ってください。 <ul style="list-style-type: none"><li>• インスタンスを終了します。</li><li>• AMI を変更して新しいカーネルを使用し、そのAMI を使用して新しいインスタンスを起動します。</li></ul>

## インスタンスの容量に関するトラブルシューティング

### Abstract

インスタンスの容量に関連する問題を解決します。

インスタンスの容量には次のエラーが関連します。

### エラー: `InsufficientInstanceCapacity`

インスタンスを起動しようとして `InsufficientInstanceCapacity` エラーが発生する場合は、現在 AWS にはリクエストに対応するだけの十分な容量がありません。大量のインスタンスをリクエストする場合、それらをホストするためのサーバー容量が十分でない場合があります。後で再試行するか、指定するインスタンスの数を少なくしてください。

### エラー: `InstanceLimitExceeded`

インスタンスを起動しようとして `InstanceLimitExceeded` エラーが発生する場合は、同時実行インスタンス制限に達しています。新規 AWS アカウントの場合、デフォルトの制限は 20 です。実行中のインスタンスを追加する必要がある場合は、「[Amazon EC2 インスタンス上限緩和申請](#)」フォームを使用してください。

## コンソール出力の取得とインスタンスの再起動

### Abstract

問題の診断に役立つコンソール出力を取得し、インスタンスを再起動する必要があるかどうか判断します。

コンソール出力は問題を診断する際に役立つツールで、特に、カーネルの問題やサービス設定の問題のトラブルシューティングを行うときに便利です。これらの問題が発生すると、SSH デーモンの開始前にインスタンスが停止したり、インスタンスに到達不能になったりする可能性があります。

同様に、到達できないインスタンスを再起動する機能も重要で、こうした機能はトラブルシューティングと一般的なインスタンス管理の両方で利用できます。

Amazon EC2 インスタンスには、コンソール出力を表示できる物理的なモニターがありません。また、インスタンスの起動、再起動、またはシャットダウンするための物理的なコントロール也没有ありません。これらのアクションを行うには、Amazon EC2 API およびコマンドラインインターフェイスツール (CLI) を通じて、Amazon がこれらのタスクをサポートします。

## インスタンスの再起動

リセットボタンを押してコンピュータをリセットするように、Amazon EC2 コンソール、CLI、または API を使用して Amazon EC2 インスタンスをリセットできます。詳細については、[インスタンスの再起動 \(p. 349\)](#) を参照してください。



### Caution

Windows インスタンスの場合、この動作によりハードウェアが再起動され、その結果データが破損する可能性があります。

## インスタンスコンソール出力

Linux/Unix インスタンスの場合、マシンに接続されている物理的なモニターに通常表示されるようなコンソール出力がインスタンスコンソール出力に表示されます。インスタンスで出力が生成されると、この出力はバッファに入れられ、ストアに送られます。インスタンスの所有者はこのストアから出力を取得できます。

Windows インスタンスの場合は、インスタンスコンソール出力に直近のシステムイベントログエラーが 3 つ表示されます。

表示される出力は、継続的には更新されず、更新する価値があると思われる場合にのみ更新されます。例えば、インスタンスの起動直後、再起動後、インスタンスの終了時に更新されます。



### Note

表示される出力のうち、保存されるのは最新の 64 KB のみです。この出力は、出力の送信から少なくとも 1 時間使用可能です。

インスタンスの所有者のみがコンソール出力にアクセスできます。コンソールまたはコマンドラインを使用して、インスタンスのコンソール出力を取得できます。

コンソールを使用してコンソール出力を取得するには

1. Amazon EC2 コンソールを開きます。
2. インスタンスを選択します。
3. [Actions]、[Get System Log] の順にクリックします。

コマンドラインを使用してコンソール出力を取得するには

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

- `get-console-output` ( AWS CLI )
- `ec2-get-console-output` ( Amazon EC2 CLI )

## ホストコンピュータに障害が発生した場合のインスタンスの復旧

基になるホストコンピュータのハードウェアで復旧不可能な問題が発生した場合、AWS はインスタンスの停止イベントをスケジュールすることがあります。このようなイベントは事前に E メールで通知されます。

障害が発生したホストコンピュータで Amazon EBS-Backed インスタンスが実行されている場合は、次の操作を行って復旧することができます。

1. インスタンスストアボリュームの重要なデータを Amazon EBS または Amazon S3 にバックアップします。
2. インスタンスを停止します。
3. インスタンスを起動します。
4. 重要なデータを復元します。
5. [EC2-Classic] インスタンスに関連付けられた Elastic IP アドレスがある場合は、次のようにして再び関連付ける必要があります。

詳細については、[インスタンスの停止と起動 \(p. 346\)](#) を参照してください。

Instance store-Backed インスタンスが、障害が発生したホストコンピュータで実行されている場合は、次の操作を行って復旧できます。

1. インスタンスから AMI を作成します。
2. イメージを Amazon S3 にアップロードします。
3. 重要なデータを Amazon EBS または Amazon S3 にバックアップします。
4. インスタンスを終了します。
5. AMI から新しいインスタンスを起動します。
6. 重要なデータを新しいインスタンスに復元します。
7. [EC2-Classic] 元のインスタンスに関連付けられた Elastic IP アドレスがある場合は、新しいインスタンスと関連付ける必要があります。

詳細については、[Instance Store-Backed Linux AMI の作成 \(p. 81\)](#) を参照してください。

# API リクエストを行う

---

## Abstract

アプリケーションを開発するときには、お好みのプログラミング言語を使用します。

アマゾン ウェブ サービス (AWS) 用のソフトウェア開発キット (SDK) に加えて Amazon EC2 用のクエリ API およびを使用すると、任意のプログラミング言語で Amazon EC2 を制御することができます。

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

## Topics

- [必要な知識 \(p. 721\)](#)
- [Amazon EC2 に対して使用可能な API \(p. 722\)](#)
- [クエリリクエスト \(p. 723\)](#)
- [API リクエストエラーのトラブルシューティング \(p. 725\)](#)
- [べき等性の確保 \(p. 728\)](#)
- [SOAP リクエスト \(p. 730\)](#)

## 必要な知識

API を介して Amazon EC2 にアクセスする場合、次の関する知識が必要です。

- XML
- Web サービス
- HTTP リクエスト
- Java、PHP、Perl、Python、Ruby、C#、C++ など、1 つまたは複数のプログラミング言語。

## Amazon EC2 に対して使用可能な API

Amazon EC2 Query API は、HTTP 動詞 ( GET または POST ) とクエリパラメータ `Action` で記述する HTTP または HTTPS リクエストです。このような API アクションの詳細については、[Amazon Elastic Compute Cloud API Reference](#) を参照してください。

AWS では、HTTP または HTTPS でリクエストを送信する代わりに言語固有の API を使用してアプリケーションを構築するソフトウェア開発者向けに、ライブラリ、サンプルコード、チュートリアルなどのリソースを提供しています。これらのライブラリには、リクエストの暗号化署名、リクエストの再試行、エラーレスポンスの処理などのタスクを自動処理する基本機能が用意されているので、開発を簡単に始められます。

AWS SDK のダウンロードの詳細については、[AWS SDKs and Tools](#) を参照してください。Amazon EC2 の言語固有の API の詳細については、次のドキュメントを参照してください。

### AWS SDK for .NET

- [Amazon.EC2](#)
- [Amazon.EC2.Model](#)
- [Amazon.EC2.Util](#)

### AWS SDK for Java

- [com.amazonaws.services.ec2](#)
- [com.amazonaws.services.ec2.model](#)
- [com.amazonaws.services.ec2.util](#)

### AWS SDK for JavaScript

- [AWS.EC2](#)

### AWS SDK for Python

- [boto.ec2](#)

### AWS SDK for Ruby

- [AWS::EC2](#)

### AWS SDK for PHP

- [Ec2Client](#)

### AWS SDK for IOS

- [AmazonEC2Client](#)

### AWS SDK for Android

- [com.amazonaws.services.ec2](#)
- [com.amazonaws.services.ec2.model](#)

- [com.amazonaws.services.ec2.util](#)

## クエリリクエスト

### Abstract

AWS から提供されている言語固有の API ライブラリ、サンプルコード、およびチュートリアルを使用してアプリケーションを構築します。

クエリリクエストは、HTTP 動詞 ( GET または POST ) とクエリパラメータ `Action` で記述する HTTP または HTTPS リクエストです。Amazon EC2 API アクションのリストについては、「[アクション](#)」を参照してください。

### Topics

- [GET リクエストの構造 \(p. 723\)](#)
- [エンドポイント \(p. 724\)](#)
- [クエリパラメータ \(p. 724\)](#)
- [クエリ API 認証 \(p. 725\)](#)
- [クエリレスポンス構造 \(p. 725\)](#)

## GET リクエストの構造

Amazon EC2 ドキュメントでは、GET リクエストが URL として記載されています。この URL はブラウザで直接使用できます。



### Tip

GET リクエストは URL なので、パラメータ値を URL エンコードする必要があります。Amazon EC2 ドキュメントでは、サンプル GET リクエストをエンコードせずに読みやすくしています。

リクエストには次が含まれます。

- **Endpoint:** ウェブサービスのエントリポイントとして機能する URL。
- **Action:** 実行するアクション ( 例えば、`RunInstance` を使用してインスタンスを実行します )。
- **Parameters:** アクションのパラメータ。各パラメータは、アンパサンド ( & ) で区切ります。AWS がリクエストの有効性と正当性を確保するために使用する、認可パラメータを含める必要があります。Amazon EC2 では、Signature Version 2 をサポートします。詳細については、[クエリ API 認証 \(p. 725\)](#) を参照してください。

インスタンスを起動するサンプルリクエストは、次のとおりです。

```
https://ec2.amazonaws.com/?Action=RunInstances&ImageId=ami-2bb65342&MaxCount=3&MinCount=1&Placement.AvailabilityZone=us-east-1b&Monitoring.Enabled=true&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Version=2014-02-01&Expires=2010-10-10T12:00:00Z&Signature=1BP67vCvG1DMBQ1dofZxg8E8SUEXAMPLE&SignatureVersion=2&SignatureMethod=HmacSHA256
```

これらのサンプルリクエストを読みやすくするために、Amazon EC2 ドキュメントにはリクエストが次の形式で記載されています。

```
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-2bb65342
&MaxCount=3
&MinCount=1
&Placement.AvailabilityZone=us-east-1b
&Monitoring.Enabled=true
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Version=2014-02-01
&Expires=2010-10-10T12:00:00Z
&Signature=lBP67vCvG1DMBQ1dofZxg8E8SUEXAMPLE
&SignatureVersion=2
&SignatureMethod=HmacSHA256
```

最初の行は、リクエストのエンドポイントを指定しています。エンドポイントの後ろにはクエスチョンマーク (?) が付いていますが、これはエンドポイントとパラメータを区別するものです。

Action パラメータは実行するアクションを示します。アクションの完全リストに関しては、*Amazon Elastic Compute Cloud API Reference* の「[アクション](#)」を参照してください。

残りの行はレポートの追加パラメータを示します。



#### Important

AWSAccessKeyId パラメータの値を指定する前に、[Best Practices for Managing AWS Access Keys](#) のガイダンスを確認し、それに従ってください。

## エンドポイント

エンドポイントは、ウェブサービスのエントリポイントとなる URL です。レイテンシーを少なくするリクエストを提出するとき、Amazon EC2 のリージョンエンドポイントを選択できます。リージョンの詳細については、「[リージョンとアベイラビリティゾーンに関する概念 \(p. 8\)](#)」を参照してください。Amazon EC2 のエンドポイントについては、*Amazon Web Services General Reference* の「[リージョンとエンドポイント](#)」を参照してください。

一般的なエンドポイント、ec2.amazonaws.com を指定する場合、us-east-1 のエンドポイントを使用します。別のリージョンを使用するには、その関連エンドポイントを指定します。例えば、エンドポイントとして ec2.us-west-2.amazonaws.com を指定する場合、us-west-2 エンドポイントにリクエストを送信します。

## クエリパラメータ

各クエリリクエストに、アクションの認証と選択を処理するための一般的な必須パラメータが含まれている必要があります。詳細は、*Amazon Elastic Compute Cloud API Reference* の「[共通クエリパラメータ](#)」を参照してください。

オペレーションの中にはパラメータのリストを取るものがあります。これらのリストは *param.n* の表記法で指定されます。n は 1 から始まる整数です。

次の例では、BlockDeviceMapping パラメータのリストを使用して、ブロックデバイスマッピングに複数のデバイスを追加しています。

```
http://ec2.amazonaws.com/?Action=RunInstances
&ImageId.1=ami-72aa081b
...
&BlockDeviceMapping.1.DeviceName=/dev/sdj
```



Amazon EC2 Query API では、エラーコードはクライアントエラーまたはサーバーエラーとして示されます。クライアントエラーは、通常、リクエストの構造、コンテンツ、または有効性に問題があることが原因で発生します。サーバーエラーは、通常、サーバー側に問題があることを示しています。

API エラーコードの詳細については、*Amazon Elastic Compute Cloud API Reference* の [エラーコード](#) を参照してください。

#### Topics

- [クエリ API リクエスト率 \(p. 726\)](#)
- [結果整合性 \(p. 727\)](#)
- [許可されていない操作 \(p. 728\)](#)

## クエリ API リクエスト率

Amazon EC2 API リクエストはパフォーマンスの低下を招かないよう、AWS アカウントごとに数が制限されます。Amazon EC2 API のすべての呼出 ( アプリケーション、Amazon EC2 コマンドラインインターフェイスの呼出、Amazon EC2 コンソールから発生するすべての呼出 ) は、許容される API リクエスト率を超過しないよう制御されます。IAM ユーザーが発行する API リクエストは、そのユーザーが所属する AWS アカウントから発行されたものとして扱われることに注意してください。

Amazon EC2 API アクションは次のカテゴリに分割されます。

- `DescribeInstances` や `DescribeVolumes` など、`Describe` アクション。これらのリクエストはキャッシュされたデータを取得するだけです。そのため、リクエスト制限が最高に設定されます。
- `RunInstances` や `CreateVolumes` など、`Modify` アクション。これらのリクエストはリソースを作成または変更します。そのため、`Describe` 呼出よりもリクエスト制限が低くなります。
- `CreateKeyPair`、`GetConsoleOutputAuthorizeSecurityGroupIngress`、`RevokeSecurityGroupIngress` アクション。これらのリクエストは完了に最も多くの時間とリソースを必要とするので、リクエスト制限が最低に設定されます。

API リクエストがそのカテゴリの API リクエスト率を超過する場合、`RequestLimitExceeded` エラーコードが返されます。このエラーを回避するには、アプリケーションが API リクエストを再試行する率を低くします。これは、ポーリングを慎重に実行し、再試行のタイミングを指数バックオフ方式で制御することで実現できます。

## ポーリング

場合によっては、アプリケーションで API を繰り返し呼び出し、ステータスの更新を確認する必要があります。ポーリングを開始する前に、リクエストの予想完了時間を指定します。ポーリングを開始するとき、連続するリクエストの間に適切なスリープ間隔を使用します。最良の結果を得るには、漸増スリープ間隔を使用します。

## 再試行またはバッチ処理

場合によっては、アプリケーションは API リクエストに失敗した後にそれを再試行するか、複数のリソース ( 例えば、すべてのボリューム ) を処理する必要があります。API リクエストの率を下げるには、連続するリクエストの間に適切なスリープ間隔を使用します。最良の結果を得るには、漸増または可変スリープ間隔を使用します。

## スリープ間隔の計算

API リクエストをポーリングまたは再試行する必要がある場合は、エクスポネンシャルバックオフアルゴリズムを使用して API コール間のスリープ間隔を計算することをお勧めします。エクスポネンシャルバックオフは、再試行間の待機時間を累進的に長くして、連続的なエラー応答を受信するという考え

に基づいています。詳細について、およびこのアルゴリズムの実装例については、「[Error Retries and Exponential Backoff in AWS](#)」を参照してください。

## 結果整合性

Amazon EC2 API は、API をサポートするシステムの分散特性に起因し、結果整合性モデルの後に続きます。つまり、Amazon EC2 リソースに影響を与える API コマンドを実行した結果が、リソースを実行する後続の一部のコマンドにはすぐに表示されない場合があります。前の API コマンドの直後に続く API コマンドを実行するときは、このことに注意してください。

結果整合性は、リソースの管理に影響を与える場合があります。例えば、コマンドを実行してリソースを作成する場合、リソースは他のコマンドに最終的に表示されます。つまり、コマンドを実行し、作成したリソースを変更または表現する場合、その ID がシステム全体に配信されておらず、リソースが存在しないというエラーを受け取ることがあります。

結果整合性を管理するには、次を実行できます。

- コマンドを実行してリソースを変更する前に、リソースのステータスを確認します。エクスポネンシャルバックオフアルゴリズムを使用して適切な Describe コマンドを実行し、前のコマンドがシステム全体に配信されるように十分な時間を与えます。これを行うには、Describe コマンドを繰り返し実行します。2 秒の待ち時間から始め、最大 5 分まで徐々に待ち時間を増やします。
- Describe コマンドが正確な応答を返す場合でも、後続のコマンド間の待ち時間を追加します。適用するエクスポネンシャルバックオフアルゴリズムの待ち時間は当初 2 秒とし、最大 5 分まで徐々に待ち時間を増やします。

### 結果整合性エラーの例

次は、結果整合性の結果として遭遇する可能性があるエラーコードの例です。

- `InvalidInstanceID.NotFound`

`RunInstances` コマンドを正常に実行し、`RunInstances` のレスポンスとして提供されたインスタンス ID を使用してすぐに別のコマンドを実行する場合、`InvalidInstanceID.NotFound` エラーが返される場合があります。これはインスタンスが存在しないという意味ではありません。

影響を受ける可能性がある特定のコマンドの一部:

- `DescribeInstances`: インスタンスの実際の状態を確認するには、エクスポネンシャルバックオフアルゴリズムを使用してこのコマンドを実行します。
- `TerminateInstances`: インスタンスの状態を確認するには、エクスポネンシャルバックオフアルゴリズムを使用して最初に `DescribeInstances` コマンドを実行します。



#### Important

`TerminateInstances` の実行後に `InvalidInstanceID.NotFound` エラーが表示された場合、これはそのインスタンスが強制終了される（または、強制終了されている）という意味ではありません。インスタンスは依然として実行されている可能性があります。そのため、`DescribeInstances` を使用してインスタンスの状態を最初に確認することが重要です。

- `InvalidGroup.NotFound`

`CreateSecurityGroup` コマンドを正常に実行し、`CreateSecurityGroup` のレスポンスとして提供されたインスタンス ID を使用してすぐに別のコマンドを実行する場合、`InvalidGroup.NotFound` エラーが返される場合があります。セキュリティグループを確認するには、エクスポネンシャルバックオフアルゴリズムを使用して `DescribeSecurityGroups` コマンドを実行します。

## 許可されていない操作

デフォルトでは、AWS Identity and Access Management ( IAM ) ユーザーが Amazon EC2 リソースを作成/変更したり、Amazon EC2 API を使用してタスクを実行したりすることは許可されていません。ただし、IAM ポリシーを介して許可が明示的に付与されている場合はその限りではありません。許可が付与されていないアクションを IAM ユーザーが実行しようとした場合、リクエストは `Client.UnauthorizedOperation` エラーを返します。

ポリシーが意図せずに制限的になると、このエラーが発生する場合があります。例えば、IAM ユーザーが特定のサブネット内にインスタンスを起動できるようにするには、インスタンス、ボリューム、AMI、特定のサブネット、ネットワークインターフェイス、キーペア、およびセキュリティグループといったリソースに許可を付与する必要があります。そのためには IAM ポリシー内に各リソースの ARN を指定します。例えば、ボリュームの許可を省略した場合、ユーザーは EBS-Backed インスタンスのルート EBS ボリュームを作成することを許可されないため、ユーザーは Instance store-Backed AMI からしかインスタンスを起動できません。

Amazon EC2 に対する IAM ポリシーの作成についての詳細は、[Amazon EC2 の IAM ポリシー \(p. 488\)](#) を参照してください。

現在、すべての API アクションがリソースレベルの許可をサポートしているわけではありません。2014 年にはサポートの幅が広がる予定です。Amazon EC2 API アクションで使用できる ARN の詳細については、[Amazon Elastic Compute Cloud API Reference](#) の [Granting IAM Users Required Permissions for Amazon EC2 Resources](#) を参照してください。

## べき等性の確保

### Abstract

リクエストが 1 回だけ完了するように、リクエストのべき等性を確保します。

べき等のオペレーションが完了するのは 1 回のみです。

インスタンスを起動した場合、オペレーションの完了前にリクエストが返されるのが一般的です。インスタンスの状態をモニタリングして、オペレーションが正常に終了したかどうかを判断します ( 状態は `pending` から `running` に移行します )。オペレーションがタイムアウトになった場合、または接続に問題がある場合、リクエストの再試行が必要かもしれません。ただし、元々のリクエストと再試行したリクエストが両方とも正常に終了した場合は、結果としてインスタンス数が起動予定の数より多くなります。

`ec2-run-instances` コマンドまたは `RunInstances` API アクションを使用してインスタンスを起動する場合は、必要に応じてクライアントトークンを指定して、リクエストのべき等性を確保することができます。リクエストを繰り返す場合は、繰り返されたリクエストごとに同じレスポンスが返されます。レスポンスで唯一異なる可能性がある情報は、インスタンスの状態です。

クライアントトークンは 64 文字以下の ASCII 文字で構成された固有の文字列で、大文字と小文字が区別されます。クライアントトークンは、インスタンスを記述するときにレスポンスに挿入されます。クライアントトークンは、インスタンスの終了後少なくとも 24 時間は有効です。後から別の呼び出しでクライアントトークンを再利用しないでください。

同じクライアントトークンでリクエストを繰り返す場合、リクエストパラメータを変えると、Amazon EC2 は `IdempotentParameterMismatch` エラーを返します。

さまざまなリージョンにわたる同じリクエストに対して、同じクライアントトークンを使用できます。例えば、べき等リクエストを送信して `us-east-1` リージョン内でインスタンスを起動し、その他の複数のリージョンで同じクライアントトークンをリクエストに使用した場合、各リージョンでインスタンスが起動されます。

次の表は、一般的なレスポンスコードと推奨される一連のアクションをまとめたものです。

コード	再試行	コメント
200 ( OK )	影響なし	リクエストは成功していますので、再試行を行っても効果はありません。
400 ( クライアントエラー )	非推奨	リクエストは成功しません ( 例えば、指定されたパラメータ値が有効ではありません )。リクエストに状態を変更するプロセスで使用されているリソースが含まれている場合は、リクエストを繰り返すことで成功する場合があります ( 例えば、available になろうとしている Amazon EBS ボリュームを使用してインスタンスを起動する )。
500 ( サーバー内部エラー )	推奨	このエラーは通常一時的なものです。適切なバックオフ戦略で再度リクエストを行います。
503 ( サーバーが利用不可 )	推奨	このエラーは極端に負荷がかかった場合に発生する可能性があります。適切なバックオフ戦略で再度リクエストを行います。

## べき等性のサポート

次に示すコマンドとアクションは、クライアントトークンを使用して、べき等オペレーションをサポートします。

- CopyImage
- CreateReservedInstancesListing
- ec2-copy-image
- ec2-create-reserved-instances-listing
- ec2-modify-reserved-instances
- ec2-run-instances
- ModifyReservedInstances
- RunInstances

次に示すコマンドとアクションは、べき等です。

- AssociateAddress
- DisassociateAddress
- ec2-associate-address
- ec2-disassociate-address
- ec2-terminate-instances
- TerminateInstances

## べき等コマンドの例

`ec2-run-instances` コマンドを次のように使用して、べき等リクエストを作成します。

```
PROMPT> ec2-run-instances ami-b232d0db -k my-key-pair --client-token 550e8400-  
e29b-41d4-a716-446655440000
```

--client-token オプションは、64 文字以下の ASCII 文字で構成された固有の文字列 ( 大文字と小文字の区別あり ) を必要とします。

## べき等クエリのサンプル

RunInstances アクションを次のように使用して、べき等リクエストを作成します。

```
https://ec2.amazonaws.com/?Action=RunInstances  
&ImageId=ami-3ac33653  
&MaxCount=1  
&MinCount=1  
&KeyName=my-key-pair  
&ClientToken=550e8400-e29b-41d4-a716-446655440000  
&AUTHPARAMS
```

ClientToken パラメータは、64 文字以下の ASCII 文字で構成された固有の文字列 ( 大文字と小文字の区別あり ) を必要とします。

## SOAP リクエスト

### Abstract

Amazon EC2 の SOAP リクエストは廃止になったので、それが存在する場合は置き換えます。

Amazon EC2 の SOAP API は廃止になりました。このサポートは 2014 年 6 月末で終了します。Amazon EC2 の Query API を使用するか、AWS のソフトウェア開発キット ( SDK ) を使用することをお勧めします。詳細については、[API リクエストを行う \(p. 721\)](#) を参照してください。

廃止日以前に AWS アカウントを作成し、プロセス内で SOAP API を既に使用している場合、限定期間中は引き続き使用できます。ただし、SOAP リクエストを使用しているプロセスについては、廃止日の 6 カ月以内にサポートされた形式のリクエストに移行することをお勧めします。廃止日後に AWS アカウントを作成した場合は、SOAP API は使用できません。

# ドキュメント履歴

## Abstract

Amazon EC2 ドキュメントの改訂日、関連リリース、重要な変更を検索します。

次の表は、Amazon EC2 ドキュメントへの重要な追加項目をまとめたものです。また、お客様からいただいたフィードバックに対応するために、ドキュメントを頻繁に更新しています。

Current API version: 2014-02-01.

機能	API バージョン	説明	リリース日
R3 インスタンス	2014-02-01	RAM の GiB 当たりの価格が最適に設定された高パフォーマンスの次世代型メモリ最適化インスタンス。これらのインスタンスは、R3 インスタンスの vCPU ごとの大容量メモリ、優れた処理能力、拡張ネットワーク機能によって恩恵を受けるリレーショナルおよび NoSQL データベース、インメモリ分析ソリューション、科学計算、およびその他のメモリ集約型アプリケーションに適しています。	2014 年 4 月 9 日
新しい Amazon Linux AMI リリース	2014-02-01	Amazon Linux AMI 2014.03 リリースについての情報を追加しました。詳細については、 <a href="#">Amazon Linux (p. 94)</a> を参照してください。	2014 年 3 月 27 日
VM Import/Export	2013-10-15	VM Import/Export のセクションを書き換えました。シナリオとアーキテクチャー図を追加しました。トラブルシューティングにエラーメッセージを追加しました。詳細については、「 <a href="#">インスタンスのインポートとエクスポート (p. 301)</a> 」を参照してください。	2014 年 3 月 21 日
Amazon EC2 使用状況レポート	2013-10-15	新しい Amazon EC2 使用状況レポート ( EC2 の使用状況のコストと使用状況データを示す一連のレポート ) についての情報を追加しました。詳細については、 <a href="#">Amazon EC2 使用状況レポート (p. 671)</a> を参照してください。	2014 年 1 月 28 日

機能	APIバージョン	説明	リリース日
新しい M3 インスタンス	2013-10-15	新しい M3 インスタンスのサイズ ( m3.medium と m3.large ) についての情報を追加しました。	2014 年 1 月 20 日
I2 インスタンス	2013-10-15	以下のインスタンスは、連続 SSD 書き込みパフォーマンスを向上させるため、非常に高い IOPS を提供し、TRIM をサポートします。また、I2 インスタンスは、インスタンス間のレイテンシーが低だけでなく、ネットワークのストレスが少なく、パケット毎秒 ( PPS ) が非常に大きい拡張ネットワークもサポートします。詳細については、 <a href="#">I2 インスタンス (p. 117)</a> を参照してください。	2013 年 12 月 19 日
更新された M3 インスタンス	2013-10-15	M3 インスタンスサイズ ( m3.xlarge と m3.2xlarge ) は、SSD ポリユームと合わせてインスタンスストアをサポートするようになりました。	2013 年 12 月 19 日
Linux 仮想マシンのインポートに関するサポートを追加しました。	2013-10-15	VM Import プロセスが、Linux インスタンスのインポートをサポートするようになりました。詳細については、 <a href="#">前提条件 (p. 302)</a> を参照してください。	2013 年 12 月 16 日
RunInstances に関するリソースレベルの許可	2013-10-15	Amazon EC2 RunInstances API アクションに対し、リソースレベルの許可を制御するため、AWS Identity and Access Management でのポリシーの作成についての情報を追加しました。詳細とポリシー例については、 <a href="#">Amazon EC2 のリソースに対するアクセスの制御 (p. 485)</a> を参照してください。	2013 年 11 月 20 日
C3 インスタンス	2013-10-15	計算能力が最適化されたインスタンスは、経済的な価格で、非常に高い CPU パフォーマンスを提供します。また、C3 インスタンスは、インスタンス間のレイテンシーが低だけでなく、ネットワークのストレスが少なくパケット毎秒 ( PPS ) が非常に大きい拡張ネットワークもサポートします。これらのインスタンスは、通信量の多いウェブアプリケーション、広告配信、バッチ処理、動画の暗号化、分散分析、高エネルギー物理学、ゲノム分析、計算流体力学に最適です。	2013 年 11 月 14 日
AWS Marketplace からのインスタンスの起動	2013-10-01	Amazon EC2 起動ウィザードを使った、AWS Marketplace からのインスタンスの起動についての情報を追加しました。詳細については、 <a href="#">AWS Marketplace インスタンスの起動 (p. 332)</a> を参照してください。	2013 年 11 月 11 日
G2 インスタンス	2013-10-01	これらのインスタンスはビデオ作成サービスや 3D 表示、グラフィックスを集中的に使用するアプリケーション、大規模なパラレル処理を必要とするサーバー側のワークロードに最適です。詳細については、 <a href="#">GPU インスタンス (p. 124)</a> を参照してください。	2013 年 11 月 4 日

機能	APIバージョン	説明	リリース日
新しい起動ウィザード	2013-10-01	再設計された EC2 起動ウィザードについての情報を追加しました。詳細については、 <a href="#">インスタンスの起動 (p. 325)</a> を参照してください。	2013 年 10 月 10 日
Amazon EC2 リザーブドインスタンスのインスタンスタイプの変更	2013-10-01	同じファミリ ( m1、m2、m3、c1 など ) 内の Linux/UNIX リザーブドインスタンスのインスタンスタイプを変更する新しいリザーブドインスタンス機能について、情報を追加しました。詳細については、 <a href="#">リザーブドインスタンスの変更 (p. 251)</a> を参照してください。	2013 年 10 月 09 日
新しい Amazon Linux AMI リリース	2013-09-30	Amazon Linux AMI 2013.09 リリースについての情報を追加しました。詳細については、 <a href="#">Amazon Linux (p. 94)</a> を参照してください。	2013 年 9 月 30 日
Amazon EC2 リザーブドインスタンスの変更	2013-08-15	あるリージョン内のリザーブドインスタンスを変更する新しい EC2 リザーブドインスタンス機能について、情報を追加しました。詳細については、 <a href="#">リザーブドインスタンスの変更 (p. 251)</a> を参照してください。	2013 年 9 月 11 日
パブリック ID アドレスの割り当て	2013-07-15	VPC でインスタンスを起動するための新しいパブリック IP アドレス機能に関する情報を追加しました。詳細については、「 <a href="#">パブリック ID アドレスの割り当て (p. 533)</a> 」を参照してください。	2013 年 8 月 20 日
リソースレベルのアクセス許可の付与	2013-06-15	新しい Amazon リソース名 ( ARN ) と Amazon EC2 条件キーに関する情報を追加しました。詳細については、「 <a href="#">Amazon EC2 の IAM ポリシー (p. 488)</a> 」を参照してください。	2013 年 7 月 8 日
インクリメンタルスナップショットコピー	2013-02-01	インクリメンタルスナップショットコピーに関する情報を追加しました。詳細については、「 <a href="#">Amazon EBS スナップショットのコピー (p. 608)</a> 」を参照してください。	2013 年 6 月 11 日
新しいタグペイン	2013-02-01	Amazon EC2 コンソールの新しいタグペインに関する情報を追加しました。詳細については、「 <a href="#">Amazon EC2 リソースにタグを付ける (p. 662)</a> 」を参照してください。	2013 年 04 月 4 日
新しい Amazon Linux AMI リリース	2013-02-01	Amazon Linux AMI 2013.03 のリリースについての情報を追加しました。詳細については、「 <a href="#">Amazon Linux (p. 94)</a> 」を参照してください。	2013 年 3 月 27 日
追加の EBS 最適化インスタンスタイプのサポート	2013-02-01	ハイ CPU エクストララージ ( c1.xlarge )、ハイメモリダブルエクストララージ ( m2.2xlarge )、M3 エクストララージ ( m3.xlarge )、M3 ダブルエクストララージ ( m3.2xlarge ) を、EBS 最適化インスタンスとして起動できるようになりました。  詳細については、「 <a href="#">Amazon EBS 最適化インスタンス (p. 127)</a> 」を参照してください。	2013 年 3 月 19 日

機能	APIバージョン	説明	リリース日
リージョン間でのAMIのコピー	2013-02-01	AMIをリージョン間でコピーして、整合性のあるインスタンスを複数のAWSリージョンですばやく簡単に起動できます。  詳細については、「 <a href="#">AMIのコピー (p. 90)</a> 」を参照してください。	2013年3月11日
デフォルトのVPCへのインスタンスの起動	2013-02-01	お客様のAWSアカウントは、リージョンに基づいてEC2-ClassicまたはEC2-VPCのプラットフォームのいずれか、あるいはEC2-VPCのプラットフォーム限定でジョブフローを起動できます。インスタンスを起動できるのがEC2-VPCだけである場合は、デフォルトのVPCが自動的に作成されます。お客様がインスタンスを起動するときは、デフォルトのVPCで起動されるようになります。ただし、お客様が非デフォルトVPCを作成してそのVPCでインスタンスを起動するよう指定した場合を除きます。  詳細については、「 <a href="#">サポートされているプラットフォーム (p. 527)</a> 」を参照してください。	2013年3月11日
ハイメモリクラスター (cr1.8xlarge) インスタンスタイプ	2012-12-01	大容量のメモリと、高パフォーマンスのCPUおよびネットワークの組み合わせです。このインスタンスは、インメモリアナリティクス、グラフ分析、科学計算アプリケーションに適しています。詳細については、「 <a href="#">インスタンスタイプ (p. 109)</a> 」を参照してください。	2013年1月21日
ハイストレージ (hs1.8xlarge) インスタンスタイプ	2012-12-01	ハイストレージインスタンスは、ストレージ密度がきわめて高く、インスタンスあたりの順次読み込み/書き込みのパフォーマンスの高さが特徴です。データウェアハウス、Hadoop/MapReduce、並列ファイルシステムに適しています。詳細については、「 <a href="#">HS1 インスタンス (p. 121)</a> 」を参照してください。	2012年12月20日
EBS スナップショットのコピー	2012-12-01	スナップショットのコピーを使用して、データのバックアップを作成したり、新しいAmazon EBS ボリュームを作成したり、Amazon マシンイメージ (AMI) を作成したりすることができます。詳細については、「 <a href="#">Amazon EBS スナップショットのコピー (p. 608)</a> 」を参照してください。	2012年12月17日
プロビジョニングされた IOPS ボリューム用の EBS メトリックスおよびステータスチェックの更新	2012-10-01	プロビジョニングされた IOPS ボリューム用の 2 つの新しいメトリックスを含むように EBS メトリックスを更新しました。詳細については、「 <a href="#">CloudWatch によるボリュームのモニタリング (p. 581)</a> 」を参照してください。また、プロビジョニングされた IOPS ボリューム用の新しいステータスチェックを追加しました。詳細については、「 <a href="#">ステータスチェックによるボリュームのモニタリング (p. 583)</a> 」を参照してください。	2012年11月20日

機能	APIバージョン	説明	リリース日
Microsoft Windows Server 2012 のサポート	2012-10-01	<p>Amazon EC2 は、複数の構成済み Windows Server 2012 AMI の提供を開始しました。これらの AMI は、あらゆるリージョンで、また、すべての 64 ビットインスタンスタイプですぐに利用できます。AMI は次の言語をサポートしています。</p> <ul style="list-style-type: none"> <li>• English</li> <li>• 簡体字中国語</li> <li>• 繁体字中国語</li> <li>• 繁体字中国語香港</li> <li>• 日本語</li> <li>• 韓国語</li> <li>• ポルトガル語</li> <li>• ブラジルポルトガル語</li> <li>• チェコ語</li> <li>• オランダ語</li> <li>• フランス語</li> <li>• ドイツ語</li> <li>• ハンガリー語</li> <li>• イタリア語</li> <li>• ポーランド語</li> <li>• ロシア語</li> <li>• スペイン語</li> <li>• スウェーデン語</li> <li>• トルコ語</li> </ul>	2012 年 11 月 19 日
Linux カーネル	2012-10-01	AKIID を更新し、ディストリビューションカーネルを再編成し、PVOps セクションを更新しました。	2012 年 11 月 13 日
Amazon Linux AMI	2012-10-01	「パッケージの追加」に情報を追加し、最新の Linux AMI の場所を追加しました。詳細については、「 <a href="#">パッケージの追加 (p. 99)</a> 」を参照してください。	2012 年 11 月 2 日
M3 インスタンス	2012-10-01	新しい Amazon Elastic Compute Cloud ( Amazon EC2 ) M3 エクストララージおよび M3 ダブルエクストララージインスタンスタイプに関する情報を追加しました。詳細については、「 <a href="#">インスタンスタイプ (p. 109)</a> 」を参照してください。	2012 年 10 月 31 日
Amazon EC2 スポットインスタンスリクエストステータス	2012-10-01	Amazon EC2 スポットリクエストのステータスを簡単に判断できるようにする、Amazon EC2 スポットインスタンスリクエストステータスに関する情報を追加しました。	2012 年 10 月 14 日
新しい Amazon Linux AMI リリース	2012-08-15	Amazon Linux AMI 2012.09 のリリースについての情報を追加しました。詳細については、「 <a href="#">Amazon Linux (p. 94)</a> 」を参照してください。	2012 年 10 月 11 日

機能	APIバージョン	説明	リリース日
Amazon EC2 リザーブドインスタンスマーケットプレイス	2012-08-15	リザーブドインスタンスマーケットプレイスでは、不要になった Amazon EC2 リザーブドインスタンスを持つ販売者と、追加の容量を求める購入者とが、互いに相手を見つけることができます。リザーブドインスタンスマーケットプレイスを通じて売買されたリザーブドインスタンスは、他のリザーブドインスタンス同様に動作します。異なるのは、残りの契約期間が短いことと、異なる価格で販売できることです。	2012 年 9 月 11 日
Amazon EBS 向けの Provisioned IOPS (input/output operations per second)	2012-07-20	Provisioned IOPS ポリユームは、整合性と応答時間の短さが重要な、データベースアプリケーションなど入出力を多用する作業負荷に対して、予測可能なハイパフォーマンスを提供します。詳細については、「 <a href="#">Amazon EBS ポリユームの種類 (p. 569)</a> 」を参照してください。	2012 年 7 月 31 日
Amazon EC2 向けのハイ I/O インスタンス	2012-06-15	ハイ I/O インスタンスは、SSD ベースのローカルインスタンスストレージを使用して遅延の少ないハイ I/O パフォーマンスを提供します。詳細については、「 <a href="#">HI1 インスタンス (p. 119)</a> 」を参照してください。	2012 年 7 月 18 日
Amazon EC2 インスタンスの IAM ロール	2012-06-01	Amazon EC2 向けの IAM ロールは次の機能を提供します。 <ul style="list-style-type: none"> <li>• Amazon EC2 インスタンスで実行中のアプリケーションの AWS アクセスキー。</li> <li>• Amazon EC2 インスタンスの AWS アクセスキーを自動更新します。</li> <li>• EC2 インスタンスで実行中の、AWS サービスにリクエストを送信するアプリケーションに対して、許可をより細かく設定できます。</li> </ul>	2012 年 6 月 11 日

機能	API バージョン	説明	リリース日
開始しやすく、中断の可能性に対応しやすくなるスポットインスタンス機能。		<p>Auto Scaling を使用して、スポットインスタンスを次のように管理できるようになりました。</p> <ul style="list-style-type: none"> <li>• Auto Scaling 起動設定を使用して EC2 スポットインスタンスに入札したり、スポットインスタンスの入札のスケジュールを設定したりできます。詳細については、「<a href="#">Auto Scaling によるスポットインスタンスの管理 (p. 166)</a>」を参照してください。</li> <li>• インスタンスの起動または終了の通知を受け取ることができます。詳細については、「<a href="#">Auto Scaling を使用してスポットインスタンスに関する通知を取得する (p. 179)</a>」を参照してください。</li> <li>• AWS CloudFormation テンプレートを使用して、AWS リソースのスタック内のスポットインスタンスを起動できます。詳細については、「<a href="#">CloudFormation テンプレートを使用してスポットインスタンスを起動する (p. 182)</a>」を参照してください。</li> </ul>	2012 年 6 月 7 日
Amazon EC2 のステータスチェックのための EC2 インスタンスのエクスポートとタイムスタンプ	2012-05-01	<p>当初 EC2 にインポートした Windows Server インスタンスのエクスポートのサポートを追加しました。</p> <p>ステータスチェックが失敗した日時を示す、インスタンスステータスとシステムステータスのタイムスタンプのサポートを追加しました。</p>	2012 年 5 月 25 日
EC2 インスタンスのエクスポート、および Amazon VPC に対するインスタンスとシステムのステータスチェックのタイムスタンプ	2012-05-01	<p>Citrix Xen、Microsoft Hyper-V、および VMware vSphere 向けの EC2 インスタンスのエクスポートのサポートを追加しました。</p> <p>インスタンスとシステムのステータスチェックのタイムスタンプのサポートを追加しました。</p>	2012 年 5 月 25 日
クラスターコンピュートエイトエクストララージインスタンス	2012-04-01	<p>Amazon Virtual Private Cloud ( Amazon VPC ) での cc2.8xlarge インスタンスのサポートを追加しました。</p>	2012 年 4 月 26 日
AWS Marketplace AMI	2012-04-01	<p>AWS Marketplace AMI のサポートを追加しました。</p>	2012 年 4 月 19 日
新しい Linux AMI リリース		<p>Amazon Linux AMI 2012.03 のリリースについての情報を追加しました。詳細については、<a href="#">Amazon Linux (p. 94)</a> を参照してください。</p>	2012 年 3 月 28 日
新しい AKI バージョン		<p>新しい AKI バージョン 1.03 のリリースと AWS GovCloud ( 米国 ) リージョン用の AKI のリリースについての情報を追加しました。詳細については、「<a href="#">独自の Linux カーネルの使用 (p. 102)</a>」を参照してください。</p>	2012 年 3 月 28 日

機能	API バージョン	説明	リリース日
メディアムインスタンス、すべての AMI での 64 ビットのサポート、および Java ベースの SSH クライアント		新しいインスタンスタイプと 64 ビット情報のサポートを追加しました。Linux/UNIX インスタンスに追加するための、Java ベースの SSH クライアントの使用手順を追加しました。	2012年3月7日
リザーブドインスタンス料金範囲		リザーブドインスタンス料金範囲に組み込まれた割引料金の利用方法に関する新しいセクションを追加しました。詳細については、「 <a href="#">リザーブドインスタンス料金範囲を理解する (p.226)</a> 」を参照してください。	2012年3月5日
Amazon Virtual Private Cloud での EC2 インスタンスのための Elastic Network Interfaces ( ENI )		VPC での EC2 インスタンスのための Elastic Network Interfaces ( ENI ) についての新しいセクションを追加しました。詳細については、「 <a href="#">Elastic Network Interface ( ENI ) ( p. 545 )</a> 」を参照してください。	2011年12月21日
新しい GRU リージョンと AKI		SA-East-1 リージョン用の新しい AKI リリースについての情報を追加しました。このリリースにより AKI バージョン 1.01 は廃止されます。AKI バージョン 1.02 は下位互換性があります。	2011年12月14日
新しい Amazon EC2 リザーブドインスタンスの提供タイプ	2011-11-01	インスタンスの使用目的に応じて、リザーブドインスタンス提供タイプを <i>重度使用</i> 、 <i>中度使用</i> 、または <i>軽度使用</i> から選択できるようになりました。 <a href="#">リザーブドインスタンス ( p. 218 )</a> を参照してください。	2011年12月1日
Amazon EC2 インスタンスのステータス		AWS で予定されている、インスタンスに影響を及ぼす可能性のあるイベントなど、インスタンスのステータスについて追加の詳細情報を表示できます。これら運用上の作業には、ソフトウェアアップデートやセキュリティパッチを適用するために必要なインスタンスの再起動や、ハードウェアに問題が生じた場合に必要となるインスタンスの廃棄などが含まれます。 <a href="#">インスタンスのステータスのモニタリング ( p. 386 )</a> を参照してください。	2011年11月16日
Amazon EC2 クラスターコンピューティングインスタンスタイプ		クラスターコンピューティングエクストララージ ( cc2.8xlarge ) のサポートを Amazon EC2 に追加しました。	2011年11月14日
新しい PDX リージョンと AKI		新しい US-West 2 リージョン用の新しい AKI のリリースについての情報を追加しました。	2011年11月8日

機能	API バージョン	説明	リリース日
Amazon VPC での Amazon EC2 スポット インスタンス		Amazon VPC での Amazon EC2 スポットインスタンスのサポートについての情報を追加しました。このアップデートにより、ユーザーは Amazon Virtual Private Cloud ( Amazon VPC ) 内でスポットインスタンスを起動できるようになります。Amazon VPC でスポットインスタンスを起動することにより、スポットインスタンスのユーザーは、Amazon VPC のすべてのコントロールと高度なセキュリティオプションを使用できるようになります。詳細については、「 <a href="#">Amazon Virtual Private Cloud でのスポットインスタンスの起動 (p. 183)</a> 」を参照してください。	2011年10月11日
新しい Linux AMI リリース		Amazon Linux AMI 2011.09 のリリースについての情報を追加しました。このアップデートにより、Amazon Linux AMI からベータのタグが削除され、リポジトリを特定のバージョンにロックする機能がサポートされ、インストールされているパッケージについてセキュリティアップデートを含むアップデートが利用可能になったときに通知が表示されます。詳細については、「 <a href="#">Amazon Linux (p. 94)</a> 」を参照してください。	2011年9月26日
CLI ツールのユーザーのための VM Import 処理の簡素化		CLI ユーザーのための VM Import 処理を簡素化し、 <code>ec2-import-instance</code> と <code>ec2-import-volume</code> の機能を拡張しました。これにより、インポートタスクの作成後に、イメージが Amazon EC2 にアップロードされます。さらに、 <code>ec2-resume-import</code> コマンドの導入により、アップロードが途中で止まった場合にその個所から再開できるようになりました。詳細については、「 <a href="#">ステップ 4: Amazon EC2 に VM をインポートする (p. 307)</a> 」を参照してください。	2011 年 9 月 15 日
VHD ファイル形式でのインポートのサポート		VM Import で、仮想マシンイメージファイルを VHD 形式でインポートできるようになりました。VHD ファイル形式は、Citrix Xen および Microsoft Hyper-V 仮想化プラットフォームと互換性があります。VM Import はこのリリースで、RAW、VHD、および VMDK ( VMware ESX 互換 ) イメージ形式をサポートしています。詳細については、「 <a href="#">ステップ 1: Amazon EC2 CLI をインストールする (p. 305)</a> 」を参照してください。	2011 年 8 月 24 日
Microsoft Windows Server 2003 R2 のサポート		VM Import が Windows Server 2003 ( R2 ) をサポートするようになりました。VM Import はこのリリースで、Amazon EC2 がサポートするすべてのバージョンの Microsoft Windows Server をサポートしています。	2011 年 8 月 24 日

機能	APIバージョン	説明	リリース日
VMware vCenter 用の Amazon EC2 VM Import Connector のアップデート		Amazon EC2 VM Import Connector for VMware vCenter 仮想アプライアンス ( Connector ) の1.1バージョンについての情報を追加しました。このアップデートには、インターネットアクセスのためのプロキシサポート、エラー処理の向上、タスクプログレスバーの精度向上、および数件のバグ修正が含まれます。詳細については、「 <a href="#">Amazon EC2 に VM をインポートする (p. 305)</a> 」を参照してください。	2011年6月27日
ユーザー提供のカーネルを実行するための Linux AMI の有効化		AKI のバージョン1.01から1.02への変更についての情報を追加しました。このバージョンでは、t1.microLinux インスタンスに関連する起動エラーに対処するため、PVGRUBが更新されています。詳細については、「 <a href="#">独自の Linux カーネルの使用 (p. 102)</a> 」を参照してください。	2011年6月20日
スポットインスタンスの Availability Zone の料金変更		スポットインスタンスの利用可能ゾーンの料金機能についての情報を追加しました。このリリースでは、スポットインスタンスリクエストおよびスポット価格履歴のクエリを行ったときに返される情報の一部として、新しい利用可能ゾーンの料金オプションが追加されました。これにより、特定の利用可能ゾーンでのスポットインスタンスの起動にかかる料金を調べやすくなりました。詳細については、「 <a href="#">スポットインスタンス (p. 134)</a> 」を参照してください。	2011年5月26日
AWS Identity and Access Management		AWS Identity and Access Management ( IAM ) についての情報を追加しました。IAM を使用すると、Amazon EC2 リソースで通常ユーザーが使用できる Amazon EC2 アクションをユーザーが指定できます。詳細については、「 <a href="#">Amazon EC2 のリソースに対するアクセスの制御 (p. 485)</a> 」を参照してください。	2011年4月26日
ユーザー提供のカーネルを実行するための Linux AMI の有効化		ユーザー提供のカーネルの実行用に PVGRUB Amazon Kernel Image ( AKI ) を使用するために Linux AMI を有効化する方法についての情報を追加しました。詳細については、「 <a href="#">独自の Linux カーネルの使用 (p. 102)</a> 」を参照してください。	2011年4月26日
専用インスタンス		専用インスタンスは、ホストのハードウェアレベルで物理的に隔離されているインスタンスであり、Amazon Virtual Private Cloud ( Amazon VPC ) 内で起動します。専用インスタンスを使うと、Amazon EC2 コンピュートインスタンスをハードウェアレベルで隔離しながら、オンデマンドの弾力性のあるプロビジョニングと、使用した分だけ支払うという料金システムを活用でき、Amazon VPC と AWS クラウドの利点を生かすことができます。詳細については、「 <a href="#">Amazon Virtual Private Cloud ユーザーガイド</a> 」の「 <a href="#">Using EC2 Dedicated Instances</a> 」を参照してください。	2011年3月27日

機能	APIバージョン	説明	リリース日
リザーブドインスタンスに関するAWSマネジメントコンソールの更新		AWS マネジメントコンソールが更新され、より簡単に、リザーブドインスタンスを表示したり、ハードウェア専有リザーブドインスタンスを含む追加のリザーブドインスタンスを購入したりできるようになりました。詳細については、「 <a href="#">リザーブドインスタンス (p. 218)</a> 」を参照してください。	2011年3月27日
Windows Server 2008 R2 のサポート		Amazon EC2 は、複数の定義済み Windows Server 2008 R2 AMI の提供を開始しました。これらの AMI は、あらゆるリージョンですぐに使用でき、t1.micro と HPC ファミリーを除くほとんどの 64 ビットインスタンスタイプで利用できます。AMI は複数の言語をサポートします。	2011年3月15日
新しい Amazon Linux 参照 AMI		CentOS 参照 AMI の代わりとなる、新しい Amazon Linux 参照 AMI についての情報を追加しました。Correcting Clock Drift for Cluster Instances on CentOS 5.4 AMI セクションなど、CentOS 参照 AMI に関する情報を削除しました。詳細については、「 <a href="#">GPU インスタンスの AMI (p. 125)</a> 」を参照してください。	2011年3月15日
メタデータ情報		2011-01-01 リリースでの変更を反映するため、メタデータについての情報を追加しました。詳細については、「 <a href="#">インスタンスメタデータとユーザーデータ (p. 289)</a> 」と「 <a href="#">インスタンスメタデータのカテゴリ (p. 297)</a> 」を参照してください。	2011年3月11日
Amazon EC2 VM Import Connector for VMware vCenter		Amazon EC2 VM Import Connector for VMware vCenter 仮想アプライアンス ( Connector ) についての情報を追加しました。このコネクタは、VMware vSphere Client と統合するための、VMware vCenter のプラグインです。コネクタの GUI を使用して、VMware 仮想マシンを Amazon EC2 にインポートすることができます。詳細については、「 <a href="#">Amazon EC2 に VM をインポートする (p. 305)</a> 」を参照してください。	2011年3月3日
ボリュームの強制デタッチ		AWS Management Console を使用して、Amazon EBS ボリュームをインスタンスから強制的にデタッチできるようになりました。詳細については、「 <a href="#">インスタンスからの Amazon EBS ボリュームのデタッチ (p. 591)</a> 」を参照してください。	2011年2月23日
インスタンス終了の防止		AWS Management Console を使用してインスタンスの終了を防止できるようになりました。詳細については、「 <a href="#">インスタンスの終了保護の有効化 (p. 354)</a> 」を参照してください。	2011年2月23日
CentOS 5.4 AMI のクラスタインスタンスのクロック同期ずれの修正		Amazon の CentOS 5.4 AMI 上で実行されているクラスタインスタンスのクロック同期ずれを修正する方法についての情報を追加しました。	2011年1月25日

機能	API パー ジョン	説明	リリース日
ドキュメントを再構成しました		次の更新を実装しました。 <ul style="list-style-type: none"> <li>すべての開発者ガイド情報をこのユーザーガイドに統合しました。</li> <li>AMI 作成についての情報を再構成し、更新しました。</li> <li>ユーザーガイドの導入セクションを更新しました。</li> </ul>	2011年1月 14日
VM Import		仮想マシンまたはボリュームを Amazon EC2 にインポートする VM Import についての情報を追加しました。詳細については、「 <a href="#">ステップ 1: Amazon EC2 CLI をインストールする (p. 305)</a> 」を参照してください。	2010年12 月15日
インスタンスの基本モニタリング		EC2 インスタンスの基本モニタリングについての情報を追加しました。	2010 年 12 月 12 日
クラスター GPU インスタンス		Amazon EC2 は、高性能コンピューティング ( HPC ) アプリケーション用にクラスター GPU インスタンス ( cg1.4xlarge ) を提供します。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「 <a href="#">Instance Type Details</a> 」を参照してください。	2010年11 月14日
フィルタとタグ		リソースの一覧表示、フィルタリング、およびタグ付けについての情報を追加しました。詳細については、 <a href="#">リソースのリスト表示とフィルタリング (p. 660)</a> と <a href="#">Amazon EC2 リソースにタグを付ける (p. 662)</a> を参照してください。	2010年9月 19日
インスタンス起動時の多重実行禁止		インスタンスを実行する際に多重実行を禁止する方法についての情報を追加しました。詳細については、「 <a href="#">べき等性の確保 (p. 728)</a> 」を参照してください。	2010 年 9 月 19 日
マイクロインスタンス		Amazon EC2 は、特定のタイプのアプリケーションに対して、t1.micro インスタンスタイプを提供します。詳細については、「 <a href="#">マイクロインスタンス (p. 110)</a> 」を参照してください。	2010年9月 8日
Amazon EC2 の AWS Identity and Access Management		Amazon EC2 が AWS Identity and Access Management ( IAM ) と統合されました。詳細については、「 <a href="#">Amazon EC2 のリソースに対するアクセスの制御 (p. 485)</a> 」を参照してください。	2010年9月 2日
クラスターインスタンス		Amazon EC2 は、高性能コンピューティング ( HPC ) アプリケーション用にクラスターコンピュートインスタンスを提供します。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「 <a href="#">Instance Type Details</a> 」を参照してください。	2010年7月 12日

機能	APIバージョン	説明	リリース日
Amazon VPC IP アドレス指定		Amazon VPC ユーザーは、VPC 内で起動されたインスタンスに割り当てる IP アドレスを指定できるようになりました。	2010年7月12日
Amazon CloudWatch による Amazon EBS ボリュームのモニタリング		Amazon CloudWatch による Amazon EBS ボリュームのモニタリングが自動的に行われるようになりました。詳細については、「 <a href="#">CloudWatch によるボリュームのモニタリング (p.581)</a> 」を参照してください。	2010年6月14日
ハイメモリエクストララージインスタンス		Amazon EC2 は、ハイメモリエクストララージ ( m2.xlarge ) インスタンスタイプをサポートするようになりました。各 Amazon EC2 インスタンスタイプのハードウェア仕様については、「 <a href="#">Instance Type Details</a> 」を参照してください。	2010年2月22日
Windows でのリザーブドインスタンス		Amazon EC2 は Windows でリザーブドインスタンスをサポートするようになりました。リザーブドインスタンスの詳細については、「 <a href="#">リザーブドインスタンス (p. 218)</a> 」を参照してください。	2010年2月22日