

---

# Amazon Simple Email Service Classic

デベロッパーガイド



## Amazon Simple Email Service Classic: デベロッパーガイド

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

Amazon の商標およびトレードドレスは、Amazon のものではない製品またはサービスと関連付けてはならず、また、お客様に混乱を招くような形や Amazon の信用を傷つけたり失わせたりする形で使用することはできません。Amazon が所有しない商標はすべてそれぞれの所有者に所属します。所有者は必ずしも Amazon と提携していたり、関連しているわけではありません。また、Amazon の後援を受けているとはかぎりません。

## Table of Contents

Amazon SES Classic とは .....	1
Amazon SES の機能 .....	1
Amazon SES を使用する理由 .....	1
Amazon SES とその他の AWS サービス .....	1
このガイドの内容 .....	2
クイックスタート .....	3
ステップ 1: AWS にサインアップ .....	3
ステップ 2: E メールアドレスの確認 .....	3
ステップ 3: 最初の E メールの送信 .....	3
ステップ 4: バウンスや苦情の処理方法の検討 .....	3
ステップ 5: Amazon SES サンドボックス外への移動 .....	3
次のステップ .....	4
E メールを送信する .....	5
Amazon SES を使用して E メールを送信する方法 .....	5
開始する方法 .....	6
概念 .....	7
Amazon SES と配信性能 .....	7
E メール送信プロセス .....	11
E メール形式と Amazon SES .....	14
認証情報のタイプ .....	17
開始方法 .....	19
Amazon SES コンソールの使用 .....	19
Simple Mail Transfer Protocol ( SMTP ) の使用 .....	19
AWS SDK の使用 .....	19
開始する前に .....	19
コンソールを使用して E メールを送信する .....	20
SMTP を使用して E メールを送信する .....	21
AWS SDK を使用して E メールを送信する .....	31
別のソリューションから Amazon SES への移行 .....	46
Eメールのセットアップ .....	47
AWS にサインアップする .....	48
ID の検証 .....	48
AWS アクセスキーの取得 .....	67
AWS SDK のダウンロード .....	67
カスタムの MAIL FROM ドメインの設定 .....	68
SPF レコードの設定 .....	73
SMTP 認証情報の取得 .....	73
サンドボックス外への移動 .....	74
オープンとクリックのカスタムドメインの設定 .....	80
VPC エンドポイントの設定 .....	84
Eメールの送信方法 .....	88
SMTP インターフェイスの使用 .....	88
API の使用 .....	113
パーソナライズされた E メールを送信する .....	122
Eメールの認証方法 .....	133
SPF での Eメールの認証 .....	134
DKIM での Eメールの認証 .....	135
DMARC への準拠 .....	147
送信クォータの管理 .....	149
送信クォータのモニタリング .....	150
送信クォータの引き上げ .....	151
送信クォータエラー .....	153
送信承認の使用 .....	154
の送信承認の概要 .....	155

送信承認ポリシー	157
送信承認ポリシーの例	161
ID 所有者のタスク	166
代理送信者のタスク	172
専用 IP アドレスの使用	178
容易なセットアップ	179
AWS が管理する評価	179
送信パターンの予測可能性	180
送信メールのボリューム	180
追加料金	180
送信者の評価に関する制御	180
送信者評価の分離	180
変わることがない既知の IP アドレス	181
専用 IP のリクエストと解放	181
専用 IP のウォームアップ	184
専用 IP プールの作成	185
自分の IP アドレスの使用	186
E メール送信のテスト	187
重要な考慮事項	188
メールボックスシミュレーターの使用	188
拒否イベントのテスト	189
アカウントレベルのサブプレッショリストの使用	190
アカウントレベルのサブプレッショリストに関する考慮事項	190
アカウントレベルのサブプレッショリストの有効化	191
設定セットに応じたアカウントレベルのサブプレッショリストの有効化	191
アカウントレベルのサブプレッショリストに個人の E メールアドレスを手動で追加する	192
アカウントレベルのサブプレッショリストに E メールアドレスを一括で追加	192
アカウントレベルのサブプレッショリストにあるアドレスの一覧表示	194
アカウントレベルのサブプレッショリストから E メールアドレスを削除する	195
アカウントレベルのサブプレッショリストから E メールアドレスを一括削除する	196
アカウントのインポートジョブのリストの表示	197
アカウントのインポートジョブに関する情報を取得する	198
アカウントレベルのサブプレッショリストの無効化	198
グローバルサブプレッショリストの使用	199
グローバルサブプレッショリストの考慮事項	199
グローバルサブプレッショリストからのアドレスの削除	200
リスト管理の使用	200
リスト管理の概要	201
リスト管理の設定	201
サブスクリプションの使用	206
購読管理の概要	206
購読取り消しフッターリンクの追加	207
購読取り消しヘッダーの考慮事項	207
サポートされるセキュリティプロトコル	207
E メール送信者から Amazon SES へ	207
Amazon SES から受信者へ	208
エンドツーエンドの暗号化	209
Eメールの受信	210
Eメール受信の概念	210
受信者ベースの制御	210
IP アドレスベースの制御	211
Eメール受信プロセス	212
Eメール受信の開始方法	212
ステップ 1: 開始する前に	212
ステップ 2: ドメインの検証	213
ステップ 3: 受信ルールの設定	214
ステップ 4: テスト Eメールの送信	216

ステップ 5: E メールを表示	217
ステップ 6: クリーンアップ	217
E メール受信のセットアップ	218
ユースケースの検討	218
ドメインの検証	220
MX レコードの公開	220
アクセス許可の付与	222
IP アドレスフィルタの作成	225
受信ルールセットの作成	226
受信ルールの作成	226
E メール受信の管理	239
受信ルールセットの管理	239
受信ルールの管理	242
IP アドレスフィルタの管理	244
エラーメトリクスの表示	245
通知の使用	246
設定セットの使用	257
設定セットの管理	257
設定セットの作成	257
設定セットの一覧表示	258
設定セットの詳細の表示	259
設定セットの削除	259
デフォルト設定セットの管理	260
イベント送信先の管理	262
イベント送信先の更新	262
イベント送信先の削除	263
イベント送信先の有効化または無効化	263
IP プールの管理	264
IP プールを設定セットに割り当てる	264
IP プールの割り当てを変更する	264
メール内で設定セットを指定する	264
評価メトリクスのエクスポート	265
設定セットの評価メトリクスのエクスポートの有効化	265
設定セットの評価メトリクスのエクスポートの無効化	266
送信アクティビティのモニタリング	267
コンソールを使用したモニタリング	269
API を使用したモニタリング	270
GetSendStatistics を使用した AWS CLI API オペレーションの呼び出し	270
プログラムを使用した GetSendStatistics オペレーションの呼び出し	270
通知を使用したモニタリング	273
重要な考慮事項	273
E メールでの通知	274
Amazon SNS の通知	276
イベント発行を使用した送信のモニタリング	296
イベント発行のしくみ	296
イベント発行を使用する方法	297
イベント発行の用語	297
イベント発行のセットアップ	298
イベントデータの使用	307
チュートリアル	348
送信者評価のモニタリング	377
評価ダッシュボードの使用	377
評価ダッシュボードのメッセージ	378
一般的なステータスメッセージ	379
バウンス率の通知	380
苦情率の通知	381
アンチスパム組織の通知	382

直接フィードバック通知	383
ドメインブロックリスト通知	384
内部レビュー通知	385
メールボックスプロバイダー通知	386
受信者フィードバック通知	387
関連アカウント通知	388
スパムトラップ通知	388
脆弱サイト通知	389
その他の通知	390
CloudWatch を使用したアラームの作成	390
専用 IP の SNS メトリクス	394
トラブルシューティングに関する質問	395
E メール送信の自動的な一時停止	395
アカウント全体の場合	395
設定セットの場合	400
アクセスの制御	406
Amazon SES にアクセスするための IAM ポリシーの作成	406
アクションの制限	406
E メールアドレスの制限	407
一般的な API の使用の制限	408
Amazon SES の IAM ポリシーの例	408
すべての Amazon SES アクションへのフルアクセスを許可	408
E メール送信アクションへのアクセスのみを許可	408
送信期間の制限	409
受取人アドレスの制限	409
"From" アドレスの制限	410
E メール送信者の表示名の制限	410
バウンスや苦情のフィードバックの制限と送信先	411
API コールのログ作成	412
CloudTrail 内の Amazon SES 情報	412
例: Amazon SES ログファイルのエントリ	414
API の使用	416
API リクエスト	416
GET リクエストの構造	416
POST リクエストの構造	417
リクエスト認証	418
認証エラー	419
GET と POST の例	420
GET リクエストの例	420
POST リクエストの例	420
API レスポンス	421
正常なレスポンスの構造	421
Amazon SES API エラーレスポンスの構造	421
API エラーコード	422
コードの例	425
AWS SDK を使用して E メールを送信する	425
SMTP インターフェイスを使用した Eメールの送信	439
AWS SDK を使用して raw Eメールを送信する	451
複数の E メールアドレスの検証	459
ID のレプリケート	460
リージョン	467
Amazon SES のリージョンとエンドポイント	467
サンドボックスのステータスと送信制限の引き上げ	468
E メールアドレスおよびドメインの検証	468
Easy DKIM	468
サブレーションリスト	468
フィードバック通知	468

SMTP 認証情報 .....	469
カスタムの MAIL FROM ドメイン .....	469
送信認可 .....	470
E メール受信 .....	470
クォータ .....	471
E メール送信クォータ .....	471
送信クォータ .....	471
メッセージのクォータ .....	471
送信者と受信者のクォータ .....	472
イベント発行に関連するクォータ .....	472
E メールテンプレートのクォータ .....	472
E メール受信に関連するクォータ .....	473
一般的なクォータ .....	473
Amazon SES API のクォータ .....	473
ベストプラクティス .....	475
成功のメトリクス .....	475
バウンス率 .....	475
苦情 .....	476
メッセージの品質 .....	477
ヒントとベストプラクティス .....	477
一般的な推奨事項 .....	477
ドメインおよび「From」アドレスに関する考慮事項 .....	478
Authentication .....	478
リストの構築とメンテナンス .....	478
Compliance .....	479
Amazon SES のトラブルシューティング .....	480
一般的な問題 .....	480
行った変更がすぐに表示されない .....	480
認証の問題 .....	481
E メール認証の問題 .....	481
ドメインの検証に関する問題 .....	481
ドメイン検証設定の確認 .....	482
DKIM に関する問題 .....	483
配信に関する問題 .....	484
受け取った E メールに関する問題 .....	485
通知に関する問題 .....	486
E メール送信エラー .....	486
スループットを上げる .....	488
SMTP に関する問題 .....	489
SMTP 応答コード .....	490
個人データを削除する .....	494
非表示の住所の削除 .....	494
送信された E メールに関するデータを削除する .....	495
保存されたイベントデータを削除する .....	495
ID に関するデータを削除する .....	496
送信者認証データを削除する .....	496
受信ルールに関連するデータを削除する .....	497
IP アドレスフィルターに関連するデータを削除する .....	497
E メールテンプレートのデータを削除する .....	498
カスタム検証 E メールテンプレートのデータを削除する .....	498
AWS アカウントを解約することですべてのデータを削除する .....	499
よくある質問 .....	500
送信レビュープロセス .....	500
アカウントのレビュー .....	500
送信一時停止 .....	503
バウンス .....	505
苦情 .....	507

---

スパムトラップ .....	512
手動調査 .....	514
DNSブラックホールリスト (DNSBL) .....	515
DNSBL よくある質問 Q1 .....	516
DNSBL よくある質問 Q2 .....	516
DNSBL よくある質問 Q3 .....	516
DNSBL よくある質問 Q4 .....	516
DNSBL よくある質問 Q5 .....	517
DNSBL よくある質問 Q6 .....	517
E メールメトリクス .....	518
全般 .....	518
オープンの追跡 .....	519
クリックの追跡 .....	520
リソース .....	522
付録 .....	523
ヘッダーフィールド .....	523
サポート対象外の添付ファイルのタイプ .....	525
ドキュメント履歴 .....	526



# Amazon SES Classic とは

Amazon SES Classic とは、オリジナルの Amazon SES コンソール、API、およびドキュメントを指します。これらは、廃止されるまで期間限定で利用できます。これは、SES Classic デベロッパーガイドと呼ばれる元の SES デベロッパーガイドです。この用語の変更は、新しい Amazon SES コンソールをサポートする新しい Amazon SES デベロッパーガイドと元のガイドをより明確に区別するためです。

新しい Amazon SES コンソールには、設定セットなどの、改善され性能が上がった機能が追加されています。これにより、Amazon SES Classic で行っていた操作すべてができる上に、オプションが増え、柔軟性が向上し、ワークフローが効率的になっています。これらの新機能の概要を確認するには、[新しい Amazon SES](#) を参照してください。[Amazon SES コンソール](#)と[新しい Amazon SES API \(V2 と呼ばれる\)](#)を使用して、これらの機能改善を検討し、活用することをお勧めします。詳細については、[新しい Amazon SES デベロッパーガイド](#)を参照してください。

## Amazon SES の機能

[Amazon SES](#) は、ユーザー自身の E メールアドレスとドメインを使用して E メールを送受信するための、簡単で費用効率の高い方法を提供する E メールプラットフォームです。

たとえば、特価販売などのマーケティング E メールや、注文確認などの取引 E メール、ニュースレターなどのその他のタイプの通信文の送信に使用できます。[Amazon SES](#) を使用してメールを受信するときは、E メール自動応答システム、E メール登録解除システム、受信 E メールからカスタマーサポートのチケットを生成するアプリケーションなどのソフトウェアソリューションを開発できます。

Amazon SES と新しい Amazon SES の両方で、使用した分のみ支払います。詳細については、「[Amazon SES の料金](#)」を参照してください。

## Amazon SES を使用する理由

大規模な E メールソリューションを構築することは、ビジネスにとってしばしば複雑で高コストな課題となります。E メールサーバーの管理、ネットワーク構成、IP アドレスの評価といったインフラストラクチャの課題に対処する必要があります。さらに、多くのサードパーティの E メールソリューションでは、契約、料金交渉、およびかなりの初期コストが必要となります。Amazon SES では、これらの課題を解消できます。お客様は、Amazon.com で蓄積された長年の経験と、自社の大規模な顧客ベースにサービスを提供するために構築してきた高度な E メールインフラストラクチャを利用できます。

## Amazon SES とその他の AWS サービス

Amazon SES では、他の AWS 製品がシームレスに統合されます。たとえば、以下のことが可能です。

- 任意のアプリケーションに E メール送信機能を追加します。[Amazon Elastic Compute Cloud \(Amazon EC2\)](#)でアプリケーションを実行する場合、Amazon SES を使用して毎月 6 万 2,000 件の Eメールの送信を追加料金なしでご利用いただけます。Amazon EC2 から Eメールを送信するには、[AWS SDK](#)を使用するか、[Amazon SES SMTP インターフェイス \(p. 88\)](#)を使用するか、または[Amazon SES API](#)を直接呼出します。
- [AWS Elastic Beanstalk](#)を使用して Eメール対応アプリケーション (Amazon SES を使用して顧客にニュースレターを送信するプログラムなど)を作成します。

- バウンスされた E メール、苦情が生成された E メール、受信者のメールサーバーに正常に配信された E メールに関する通知を受け取るように、[Amazon Simple Notification Service \(Amazon SNS\)](#) をセットアップします。Amazon SES を使用して E メールを受信するときに、Eメールのコンテンツを Amazon SNS トピックに公開することができます。
- AWS Management Console を使用し、Eメールの認証手段である Easy DKIM をセットアップします。任意の DNS プロバイダーで Easy DKIM を使用できますが、特に、[Route 53](#) でドメインを管理する場合はセットアップが簡単です。
- Eメール送信に対するユーザーアクセスを管理するには、[AWS Identity and Access Management \(IAM\)](#) を使用します。
- [Amazon Simple Storage Service \(Amazon S3\)](#) に受信する Eメールを保管します。
- [AWS Lambda](#) 関数をトリガーすることで、受信する Eメールに対してアクションを実行します。
- [AWS Key Management Service \(AWS KMS\)](#) を使用して、Amazon S3 バケットで受信する Eメールをオプションで暗号化します。
- [AWS CloudTrail](#) を使用して、コンソールまたは Amazon SES API で実行した Amazon SES API 呼び出しをログ付けします。
- [Amazon CloudWatch](#) または [Amazon Kinesis Data Firehose](#) に Eメール送信イベントを公開します。メール送信イベントを Kinesis Data Firehose に発行する場合、[Amazon Redshift](#)、[Amazon Elasticsearch Service](#)、または [Amazon S3](#) でそれらにアクセスできます。

## このガイドの内容

このガイドには以下のセクションがあります。

セクション	説明
<a href="#">Eメールを送信する (p. 5)</a>	Amazon SES を使用して Eメールを送信する方法について説明します。
<a href="#">Eメールの受信 (p. 210)</a>	Amazon SES を使用して Eメールを受信する方法について説明します。
<a href="#">アクセスの制御 (p. 406)</a>	Amazon SES と AWS Identity and Access Management (IAM) の使用方法を示し、どの Amazon SES リソースで、どの Amazon SES API アクションをユーザーが実行できるかを明確にします。
<a href="#">API コールのログ作成 (p. 412)</a>	AWS CloudTrail を使用してログを記録できる Amazon SES API のリストを示します。
<a href="#">認証情報のタイプ (p. 17)</a>	Amazon SES で使用できる認証情報の種類と、それらの認証情報を使用できる場合を説明します。
<a href="#">API の使用 (p. 416)</a>	Amazon SES API を操作する方法について説明します。
<a href="#">リージョン (p. 467)</a>	複数の AWS リージョンでの Amazon SES の使用についての情報が含まれています。
<a href="#">クォータ (p. 471)</a>	Amazon SES に適用されるクォータのリストを提供します。
<a href="#">リソース (p. 522)</a>	Amazon SES で作業するとき役に立つリソースを示します。
<a href="#">付録 (p. 523)</a>	ヘッダーフィールド、サポートされない添付の種類、スクリプトに関する補足情報を提供します。

# Amazon SES のクイックスタート

この手順では、AWS へのサインアップ、E メールアドレスの確認、最初のメールの送信、バウンスや苦情の処理方法の検討、Amazon Simple Email Service (Amazon SES) サンドボックスの外への移動の各ステップについて説明します。

以下の場合、この手順を使用します。

- Amazon SES を試用しています。
- プログラミングしないでテスト E メールを送信したい。
- できるだけ少ないステップでセットアップしたい。

## ステップ 1: AWS にサインアップ

Amazon SES を利用する前に、AWS にサインアップする必要があります。AWS にサインアップすると、アカウントが自動的にすべての AWS サービスにサインアップされます。

手順については、「[AWS にサインアップする \(p. 48\)](#)」を参照してください。

## ステップ 2: E メールアドレスの確認

Amazon SES を介してメールアドレスからメールを送信する場合は、その前に、E メールアドレスを確認してそのメールを所有していることを Amazon SES に示す必要があります。

手順については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。

## ステップ 3: 最初の E メールを送信

Amazon SES コンソールを使用して E メールを簡単に送信できます。アカウントはサンドボックスと呼ばれるテスト環境に新規ユーザーとして作成されるため、確認した E メールアドレスでの E メール送受信のみができます。

手順については、「[Amazon SES コンソールを使用して E メールを送信する \(p. 20\)](#)」を参照してください。

## ステップ 4: バウンスや苦情の処理方法の検討

次のステップに進む前に、バウンスや苦情の処理方法について検討する必要があります。少数の受信者に送信する場合、E メールで受け取るバウンスや苦情のフィードバックを調査し、それらの受信者をメールアドレスリストから削除するという簡単なプロセスになります。

## ステップ 5: Amazon SES サンドボックス外への移動

未確認の E メールアドレスに E メールを送信する、1 日あたりに送信できる Eメールの数を増やす、高速で E メールを送信するためには、アカウントをサンドボックスの外に移動する必要があります。このプロセスでは、サポートセンターで SES Sending Limits Increase ケースを開く必要があります。

サンドボックスの制限の詳細、また、サンドボックスの外への移動の申請方法の詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

## 次のステップ

- Amazon SES メールボックスシミュレーターへの E メールは送信クォータに加算されず、バウンス率や苦情率に影響しないため、自身宛てに数通のテストメールを送信した後、メールボックスシミュレーターを使用して詳細なテストを行います。メールボックスシミュレーターの詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。
- 送信した E メールの数、バウンス数または受け取った苦情の数など、送信アクティビティをモニタリングします。詳細については、「[Amazon SES 送信アクティビティのモニタリング \(p. 267\)](#)」を参照してください。
- アドレスを個別に検証せずにドメイン内の E メールアドレスから E メールを送信できるように、ドメイン全体を検証します。詳細については、「[Amazon SES でのドメインの検証 \(p. 61\)](#)」を参照してください。
- E メールを認証することにより、迷惑メールボックスではなく受取人の受信箱に E メールが配信される可能性を高めます。詳細については、「[Amazon SES での Eメールの認証 \(p. 133\)](#)」を参照してください。

# Amazon SES で E メールを送信する

E メールを送信すると、その E メールは何らかのタイプのアウトバウンド E メールサーバーを通過します。このような E メールサーバーは、インターネットサービスプロバイダー (ISP) や企業の IT 部門で用意されますが、ユーザーが自分でセットアップする場合があります。メールサーバーはメールコンテンツを受け取り、E メール標準に準拠するようにフォーマットしてから、インターネット経由で送信します。E メールは最終的に受取人 (受信者の代わりに E メールを受け取る ISP などのエンティティ) に到達する前に、他のサーバーを通過する場合があります。その後、受取人が E メールを受信者に送信します。次の図は、基本的な E メール送信プロセスを示しています。



Amazon SES を使用する場合、Amazon SES がアウトバウンド E メールサーバーになります。または、既存のメールサーバーを維持して、送信メールが Amazon SES 経由で送信されるように設定することもできます。この場合は、E メールクライアントの設定を変更する必要がありません。次の図は、Amazon SES の E メール送信プロセスでの役割を示しています。



送信者は、さまざまな方法で E メールコンテンツを生成できます。送信者は、メールクライアントアプリケーションでメールを作成するか、Eメールの自動生成プログラム (購入トランザクションのレスポンスとして注文確認を送信するアプリケーションなど) を使用します。

## Amazon SES を使用して E メールを送信する方法

Amazon SES を使用して E メールを送信するには、いくつかの方法があります。Amazon SES コンソール、Simple Mail Transfer Protocol (SMTP) インターフェイス、または Amazon SES API の呼び出しを使用できます。

- Amazon SES コンソール - システムをセットアップしていくつかのテスト E メールを送信するための最も迅速な方法です。E メールキャンペーン開始の準備が完了した後は、このコンソールは主に送信アクティビティを監視するために使用されます。たとえば、送信済みの Eメールの数や、受け取ったバウンスおよび苦情の数をすばやく表示できます。
- SMTP インターフェイス - SMTP インターフェイスを使用して Amazon SES にアクセスするには、2つの方法があります。最初の方法は、SMTP 対応ソフトウェアで Amazon SES 経由の Eメール送信を設定する方法です。コーディングは必要ありません。たとえば、既存の Eメールクライアントまたはソフトウェアプログラムで、現在のアウトバウンド Eメールサーバーの代わりに Amazon SES SMTP エンドポイントに接続するように設定します。

2 番目の方法は、SMTP と互換性のあるプログラミング言語 (Java など) で、その言語の組み込み SMTP 関数およびデータ型を使用して Amazon SES SMTP インターフェイスにアクセスする方法です。

- Amazon SES API — Amazon SES Query API は、HTTPS を介して直接呼び出すことも、[AWS Command Line Interface](#)、[AWS Tools for Windows PowerShell](#)、または [AWS SDK](#) を利用することもできます。AWS SDK は、Amazon SES API の低レベル機能を、詳細を自動的に処理する高レベルのデータ型および関数の呼び出しでラップします。AWS SDK は Amazon SES のオペレーションだけでなく、基本的な AWS 機能 (リクエスト認証、リクエスト再試行、エラー処理など) も提供します。

## 開始する方法

Amazon SES を初めて使用する場合は、まず次のセクションを読むことをお勧めします。

- [Amazon SES のクイックスタート \(p. 3\)](#) - できるだけ迅速にセットアップしてテスト E メールを送信する方法を示します。
- [Amazon SES による E メール送信のご利用開始にあたって \(p. 19\)](#) - Amazon SES コンソール、SMTP インターフェイス、および AWS SDK を使用して E メールを送信する方法を示します。C#、Java、および PHP の例を用意しています。
- [Amazon SES と配信性能 \(p. 7\)](#) - Amazon SES の使用時に精通しておくべき Eメールの配信性能の概念を説明します。
- [Amazon SES E メール送信プロセス \(p. 11\)](#) - Amazon SES を使用して E メールを送信するとどうなるかを示します。
- [E メール形式と Amazon SES \(p. 14\)](#) - Eメールのフォーマットを確認し、Amazon SES に提供する必要がある情報を識別します。

次の表に示すセクションを読むことで、Amazon SES を使用した E メール送信の詳細を学ぶことができます。

セクション	説明
<a href="#">Eメールのセットアップ (p. 47)</a>	AWS へのサインアップ、AWS アクセスキーの取得、AWS SDK のダウンロード、E メールアドレスやドメインの確認、Amazon SES サンドボックスの外への移動を行う方法を示します。
<a href="#">SMTP インターフェイスの使用 (p. 88)</a>	Amazon SES SMTP 認証情報を取得し、Amazon SES SMTP エンドポイントに接続する方法について説明します。また、Amazon SES 経由で E メールを送信するための E メールクライアントおよびソフトウェアパッケージの設定方法の例を示します。すべての送信 E メールが Amazon SES を通過するように、既存の E メールサーバーを設定する方法についても説明します。
<a href="#">API の使用 (p. 113)</a>	Amazon SES API を使用して、フォーマットされたメールおよび raw Eメールを送信する方法を示します。raw Eメールを送信するときに多目的インターネットメール拡張 (MIME) 標準を使用して、標準以外の文字を使用する方法および添付ファイルを送信する方法について説明します。
<a href="#">Eメールの認証方法 (p. 133)</a>	Amazon SES で DKIM を使用して、送信元のドメインを所有していることを ISP に示す方法について説明します。
<a href="#">送信クォータの管理 (p. 149)</a>	Amazon SES 送信クォータについて説明し、クォータの引き上げ手順を説明し、クォータを超えたときに受け取るエラーを文書化します。
<a href="#">送信承認の使用 (p. 154)</a>	お客様の代わりにお客様のアイデンティティから Eメールを送信することを他のユーザーに承認する方法を説明します。
<a href="#">専用 IP アドレスの使用 (p. 178)</a>	Amazon SES での送信に共有 IP アドレスを使用するか専用 IP アドレスを使用するかを判断します。専用 IP のリクエストと解放を行い、専用 IP のプールを作成する手順を提供します。

セクション	説明
<a href="#">E メール送信のテスト (p. 187)</a>	Amazon SES メールボックスシミュレーターを使用して、送信統計 (バウンスおよび苦情メトリクスなど) に影響を与えずに一般的な E メールシナリオをシミュレートする方法について説明します。テストできるシナリオは、正常な配信、バウンス、苦情、不在 (OOTO)、およびサブプレッションリストのアドレスです。

## Amazon SES E メール送信に関する概念

以下のセクションでは、Amazon SES での E メール送信について説明します。

このセクションのトピック:

- [Amazon SES と配信性能 \(p. 7\)](#)
- [Amazon SES E メール送信プロセス \(p. 11\)](#)
- [E メール形式と Amazon SES \(p. 14\)](#)
- [Amazon SES 認証情報の種類 \(p. 17\)](#)

### Amazon SES と配信性能

送信者が望むのは、受取人が E メールを読み、有益であると見なし、スパムのラベルを付けないことです。つまり、Eメールの配信性能 (メールが受信者の受信箱に到達する割合) を最大限に高める必要があります。このトピックでは、Amazon SES を使用する場合に理解を深めておく必要のある、Eメールの配信性能の概念について説明します。

メールの配信可能性を最大限に高めるには、Eメール配信の問題を理解して、問題を回避するステップを積極的に実施し、送信メールのステータス情報を継続的に入手することが必要です。さらに、配信が成功する確立を高めるために、必要に応じてメール送信プログラムを改善する必要があります。以下のセクションでは、これらのステップの基本となる概念について説明し、このプロセスの中で Amazon SES がどのように役立つかを示します。



## E メール配信問題の理解

ほとんどの場合、メッセージは対象の受取人に正常に配信されます。ただし、配信が失敗する場合や、送信されたメールの受け取りを受取人が拒否する場合があります。以下のセクションで説明するバウンス、苦情、およびサブプレッションリストは、これらの配信問題に関連しています。

### Bounce

受信者のレシーバー (E メールプロバイダーなど) がメッセージを受信者に配信できなかった場合、レシーバーはそのメッセージを Amazon SES にバウンスします。Amazon SES は、システムのセットアップ内容に応じて、バウンスされた E メールに関して、E メールまたは Amazon Simple Notification Service (Amazon SNS) を介して通知します。詳細については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。

以下に示すように、ハードバウンスとソフトバウンスがあります。

- **ハードバウンス** - 永続的な E メール配信の障害。たとえば、メールボックスは存在しません。Amazon SES は、DNS ルックアップの失敗以外では、ハードバウンスを再試行しません。ハードバウンスの原因となっている E メールアドレスに対して配信の試行を繰り返さないことを強くお勧めします。
- **ソフトバウンス** - 一時的な E メール配信の障害。たとえば、メールボックスがいっぱいである、接続が多すぎる (スロットリングとも呼ばれる)、または接続がタイムアウトになった場合です。Amazon SES はソフトバウンスを複数回再試行します。それでも E メールを配信できない場合、Amazon SES は再試行を停止します。



Amazon SES は、以後再試行されないハードバウンスおよびソフトバウンスについて通知します。ただし、Amazon SES コンソールまたは `GetSendStatistics` API を使用して取得されるバウンス率とバウンスメトリクスに反映されるのはハードバウンスのみです。

バウンスには同期または非同期があります。同期バウンスは、送信者と受取人の E メールサーバーがアクティブに通信している間に発生します。非同期バウンスは、最初に受取人が E メールメッセージの配信を承諾した後で、受取人への配信に失敗した場合に発生します。

## Complaint

ほとんどの E メールクライアントプログラムには、メッセージをスパムフォルダに移動して E メールプロバイダーに転送するためのボタン ([Mark as Spam (スパムとしてマーク)] など) が用意されています。また、ほとんどの E メールプロバイダーでは、ユーザーが不要な E メールメッセージを転送して E メールプロバイダーによる防止策をリクエストできる迷惑メール用アドレス (`abuse@example.net` など) を用意しています。どちらの場合も、受取人が苦情を申し立てます。E メールプロバイダーがスパム送信者であると判断し、Amazon SES により E メールプロバイダーにフィードバックループが設定されている場合、E メールプロバイダーは Amazon SES に苦情を送信します。Amazon SES はそのような苦情を受け取ると、スパム発信者と見なされた送信者のシステムのセットアップに応じて、E メールまたは Amazon SNS 通知でその送信者に苦情を転送します。詳細については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。苦情を申し立てている E メールアドレスに対して配信の試行を繰り返さないことをお勧めします。

## グローバルサブプレッションリスト

Amazon SES グローバルサブプレッションリストは、最近 Amazon SES カスタマーへのハードバウンスとなった受信者の E メールアドレスのリストです。サブプレッションリストに登録されているアドレスに Amazon SES 経由で E メールを送信しようとした場合、Amazon SES の呼び出しは成功しますが、Amazon SES は、この E メールを送信せず、代わりにハードバウンスとして扱います。ハードバウンスと同じく、ブラックリストのバウンスは送信量上限やバウンス率にカウントされます。E メールアドレスは、サブプレッションリストに最大 14 日まで残ります。送信しようとしている E メールアドレスが有効であることがわかっている場合は、サブプレッションリストの削除リクエストを送信できます。詳細については、「[Amazon SES グローバルサブプレッションリストの使用 \(p. 199\)](#)」を参照してください。

## 積極的な対応

インターネット経由の E メールに関する最大の問題の 1 つは、未承諾一括 E メール (スパム) です。E メールプロバイダーは、顧客にスパムが送信されないように広範な防止策を実行しています。また、Amazon SES では、E メールプロバイダーがお客様の E メールをスパムと見なす可能性を減らすための措置を講じています。Amazon SES は、検証、認証、送信クォータ、およびコンテンツフィルタリングを使用します。Amazon SES は、E メールプロバイダーとの信頼された評価も維持しており、高品質 Eメールの送信を求めます。Amazon SES では、これらの作業の一部を自動的に処理します (コンテンツのフィルタリングなど)。その他の場合、ツールが提供したり (認証など)、適切な指示で助言します (クォータの送信)。以下のセクションでは、それぞれの概念の詳細情報を提供します。

## Verification

残念ながらスパムの発信者は、E メールヘッダーを改ざんし、元のメールアドレスを偽装して、E メールが別の送信元から送られたように見せかけることができます。E メールプロバイダーと Amazon SES の信頼関係を維持するために、Amazon SES は送信者が本人であることを確認する必要があります。したがって、送信者は送信アイデンティティを保護するために、Amazon SES 経由で送信する Eメールの送信元となるすべての E メールアドレスを確認する必要があります。E メールアドレスを検証するには、Amazon SES コンソールまたは Amazon SES API を使用します。また、ドメイン全体を確認することもできます。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」および「[Amazon SES でのドメインの検証 \(p. 61\)](#)」を参照してください。

アカウントが Amazon SES サンドボックスにまだある場合は、Amazon SES のメールボックスシミュレーターから提供されているアドレスを除く、すべての受信者のアドレスも確認する必要があります。サン

ドボックスの外への移動については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。メールボックスシミュレーターの詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

## Authentication

認証は、本人であることを E メールプロバイダーに示す、もう 1 つの方法です。メールを認証するときに、自分がアカウントの所有者であり、E メールが転送中に変更されていないという証拠を提供します。場合によっては、E メールプロバイダーは、認証されない E メールを送信を拒否します。Amazon SES では、送信者ポリシーフレームワーク (SPF) とドメインキーアイデンティファイドメール (DKIM) の 2 つの認証方法がサポートされています。詳細については、「[Amazon SES での Eメールの認証 \(p. 133\)](#)」を参照してください。

## 送信クォータ

E メールプロバイダーでは、Eメールの数量または頻度が突然急増する現象を検出すると、その送信者がスパム発信者であることを疑い、Eメールをブロックする場合があります。したがって、すべての Amazon SES アカウントには、送信クォータのセットがあります。これらのクォータは、24 時間内に送信できる Eメールの数と、1 秒あたりに送信できる Eメールの数を制限します。これらの送信クォータは、Eメールプロバイダーとの信頼性の保護に役立ちます。

ほとんどの場合、新規ユーザーは、Amazon SES で毎日少量の Eメールを送信できます。送信したメールが Eメールプロバイダーに受け入れられる場合は、自動的にこのクォータが引き上げられます。送信クォータは時間の経過とともに次第に引き上げられ、より短い間隔でより多くの Eメールを送信できるようになります。また、[SES 送信制限の増加ケース](#)を作成して、追加のクォータ引き上げをリクエストします。

送信クォータの詳細およびクォータを引き上げる方法については、「[Amazon SES 送信クォータの管理 \(p. 149\)](#)」を参照してください。

## コンテンツのフィルタリング

多くの Eメールプロバイダーでは、コンテンツのフィルタリングを使用して、受信 Eメールがスパムであるかどうかを判断します。コンテンツフィルターは、疑わしいコンテンツを探し、Eメールがスパムのプロファイルに適合する場合はその Eメールをブロックします。Amazon SES では、コンテンツフィルターも使用します。アプリケーションから Amazon SES にリクエストを送ると、Amazon SES で自動的に EメールメッセージがアSEMBルされ、メッセージヘッダーおよび本文がスキャンされて、Eメールプロバイダーでスパムと見なされる可能性のあるコンテンツが含まれているかどうか判断されます。Amazon SES で使用されているコンテンツフィルターによってメッセージにスパムの疑いが生じた場合は、Amazon SES による評価が下がります。

また、Amazon SES はすべてのメッセージをスキャンして、ウイルスが含まれていないかを確認します。メッセージにウイルスが含まれている場合、Amazon SES はそのメッセージを受信者のメールサーバーに配信しようとしません。

## Reputation

Eメール送信に関して、評価は IP アドレス、Eメールアドレス、または送信ドメインがスパムの発信元ではないことを確認する測定指標であり、非常に重要です。送信者の Eメールが受信者の受信トレイに確実に配信されるように、Amazon SES は Eメールプロバイダーからの高い評価を維持しています。同様に、送信者も Amazon SES からの高信頼の評価を維持する必要があります。高品質なコンテンツを送信することにより、Amazon SES との信頼関係を構築します。常に高品質なコンテンツを送信することで信頼性の評価が徐々に高くなり、Amazon SES によって送信クォータが引き上げられます。バウンスおよび苦情が増えすぎると評価が下がり、Amazon SES でのアカウントの送信クォータが引き下げられたり、Amazon SES アカウントが停止されたりする場合があります。

高い評価を維持する方法の 1 つは、システムをテストするときに、自分で作成した Eメールアドレスに送信するのではなくメールボックスシミュレーターを使用することです。メールボックスシミュレーターに

送信された E メールは、バウンスや苦情のメトリクスには影響しません。メールボックスシミュレーターの詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

## 高品質 E メール

高品質 E メールは、受取人が価値を認めて受信を希望する E メールです。値は、受信者によって異なるものを意味し、その形式にはオファー、注文確認、領収書、ニュースレターなどがあります。E メールプロバイダーで品質が低いと見なされた E メールはブロックされるため、最終的に配信性能は送信する Eメールの品質に左右されます。

## 継続的な情報の入手

配信が失敗しても、受取人が E メールに対して苦情を申し立てても、Amazon SES によって E メールが受取人のメールサーバーに正常に配信されても、Amazon SES の使用統計を簡単に監視できる機能と通知を使用して問題を追跡することができます。

## Notifications

E メールがバウンスされると、E メールプロバイダーは Amazon SES に通知し、Amazon SES はユーザーに通知します。Amazon SES は、以後再試行しないハードバウンスおよびソフトバウンスについて通知します。多くの E メールプロバイダーも苦情を転送しています。Amazon SES では、主要なメールプロバイダーと苦情フィードバックループを設定しているため、必要ありません。Amazon SES では、バウンス、苦情、正常な配信について 2 つの方法で通知します。アカウントを設定して Amazon SNS を介して通知する方法と、E メールで通知する方法 (バウンスおよび苦情のみ) です。詳細については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。

## 使用統計

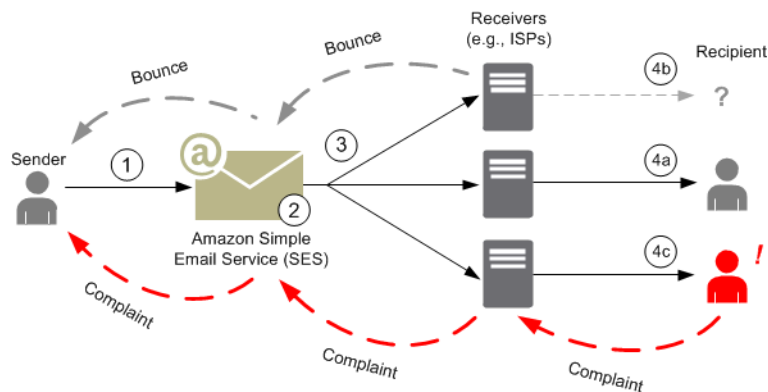
Amazon SES の使用統計を利用して、失敗した配信を調べ、根本原因を判断して解決することができます。使用統計を表示するには、Amazon SES コンソールを使用するか、Amazon SES API を呼び出します。配信、バウンス、苦情、およびウイルス感染により拒否された Eメールの数調べたり、送信クォータを超えていないかを確認したりできます。

## E メール送信プログラムの向上

多数のバウンスや苦情が発生する場合は、E メール送信戦略を再評価する必要があります。バウンス、苦情、および質の低い E メール送信の試行が多すぎると、迷惑メールの配信元と見なされ、AWS アカウントが停止されるおそれがあります。基本的に、Amazon SES を使用して高品質 E メールを送信し、受信を希望する受取人への E メールを送信する必要があります。

## Amazon SES E メール送信プロセス

このトピックでは、Amazon SES でメールを送信したときに行われる処理と、E メール送信後に想定されるさまざまな結果について説明します。次の図は、送信プロセスの概要を示しています。



1. クライアントアプリケーションは E メール送信者として動作し、1 人以上の受取人に E メールを送信するように Amazon SES にリクエストします。
2. リクエストが有効な場合、Amazon SES は E メールを受け入れます。
3. Amazon SES により、受取人の受信者宛てにインターネット経由でメッセージが送信されます。メッセージは Amazon SES に渡されると、通常は数ミリ秒内に発生する最初の配信の試行ですぐに送信されるのが一般的です。
4. この時点で、さまざまな可能性があります。以下に例を示します。
  - a. ISP が、受取人の受信箱にメッセージを正常に配信します。
  - b. 受信者の E メールアドレスが存在しないため、ISP が Amazon SES にバウンス通知を送信します。その後、Amazon SES は送信者に通知を転送します。
  - c. 受取人がメッセージを受信したものの、そのメッセージをスパムと見なして、ISP に苦情を登録します。その ISP に対して Amazon SES でフィードバックループがセットアップされている場合は、苦情が Amazon SES に送信され、そこから送信者に転送されます。

以下のセクションでは、送信者が E メールリクエストを Amazon SES に送信した後、または Amazon SES が E メールメッセージを受信者に送信した後に想定される結果について個々に説明します。

## 送信者が Amazon SES に E メールリクエストを送信した後

送信者が Amazon SES に E メール送信のリクエストを送ると、呼び出しが成功または失敗します。以下のセクションでは、それぞれのケースで発生する状況を説明します。

### リクエスト送信の成功

Amazon SES へのリクエストが成功すると、Amazon SES は成功のレスポンスを送信者に返します。このメッセージには、リクエストを一意に識別する文字列であるメッセージ ID が含まれます。メッセージ ID を使用して、送信された E メールを識別したり、送信中に発生した問題を追跡したりできます。Amazon SES では、リクエストパラメータに基づいて E メールメッセージをアSEMBルし、メッセージをスキャンして疑わしいコンテンツやウイルス感染を調べます。その後、SMTP (Simple Mail Transfer Protocol) を使用してインターネット経由でメッセージを送信します。通常、メッセージは即座に送信されます。一般的に最初の配信の試行は数ミリ秒以内に発生します。

#### Note

Amazon SES が送信者のリクエストを受け入れ、そのメッセージにウイルスが含まれていると判断した場合、Amazon SES はメッセージの処理を停止し、そのメッセージを受取人のメールサーバーに配信しようとしません。

### リクエスト送信の失敗

送信者が Amazon SES に送った E メール送信リクエストが失敗した場合、Amazon SES ではエラーを含むレスポンスを送信者に返して E メールを削除します。リクエストの失敗にはいくつかの理由が考えられます。リクエストが適切にフォーマットされていない場合や、E メールアドレスが送信者によって検証されていない場合などがあります。

リクエストが失敗したかどうかを判断する方法は、Amazon SES を呼び出す方法によって異なります。次の例は、エラーと例外がどのように返されるかを示しています。

- クエリ (HTTPS) API ( `SendEmail` または `SendRawEmail` ) で Amazon SES を呼び出している場合、そのアクションでエラーが返されます。詳細については、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。
- 例外を使用するプログラミング言語の AWS SDK を使用する場合は、Amazon SES の呼び出しによって `MessageRejectedException` がスローされます ( SDK によってはこの例外の名前が多少異なる場合があります )。

- SMTP インターフェイスを使用する場合、送信者は SMTP レスポンスコードを受け取りますが、エラーがどのように示されるかは送信者のクライアントによって異なります。一部のクライアントではエラーコードが表示されますが、表示されないクライアントもあります。

Amazon SES で E メールを送信するときに発生する可能性があるエラーについては、「[Amazon SES の E メール送信エラー \(p. 486\)](#)」を参照してください。

## Amazon SES から E メールを送信した後

送信者が Amazon SES に送ったリクエストが成功し、Amazon SES が E メールを送信すると、次の結果のいずれかが発生します。

- 配信が成功し、受信者が E メールを拒否しない場合 – E メールは ISP により受け入れられ、ISP が E メールを受信者に配信します。次の図は、正常な配信を示しています。



- ハードバウンス - メールは、永続的な状態が原因で ISP から拒否されるか、E メールアドレスが Amazon SES サプレッションリストに含まれているという理由で Amazon SES に拒否されます。Amazon SES のいずれかの顧客で最近ハードバウスの原因となった E メールアドレスは、Amazon SES のサプレッションリストに含まれます。ISP のハードバウンスは、受取人のアドレスが無効であるために発生する場合があります。ハードバウンスが発生すると ISP から Amazon SES に通知が送信され、さらに送信者に (送信者のセットアップに応じて) E メールまたは Amazon Simple Notification Service (Amazon SNS) で通知されます。Amazon SES は、同じ方法でサプレッションリストのバウンスを送信者に通知します。次の図は、ISP からのハードバウンスのパスを示しています。



- ソフトバウンス – ISP がリクエストを処理できないほどビジーであったり、受信者の E メールボックスがいっぱいであったり、一時的な状態のため、ISP が受信者に E メールを配信できません。ドメインが存在しない場合も、ソフトバウンスが発生することがあります。ISP は Amazon SES にソフトバウンス通知を返します。ドメインが存在しない場合、Amazon SES はドメインの E メールサーバーを見つけることができません。いずれの場合でも、Amazon SES は一定期間にわたって Eメールの配信を再試行します。その期間内に Eメールを配信できない場合、Amazon SES は Eメールまたは Amazon SNS を通じてバウンス通知を送信者に送信します。再試行中に Amazon SES が Eメールを受信者に配信できた場合、配信は成功です。次の図は、ソフトバウンスを示しています。このケースでは、Amazon SES は Eメールの送信を再試行し、ISP は最終的に Eメールを受取人に配信しています。



- 苦情 – メールは ISP に受け入れられて受信者に配信されますが、受信者がそのメールをスパムと見なし、E メールクライアントの [Mark as spam] などのボタンをクリックします。Amazon SES がその ISP に対してフィードバックループをセットアップしている場合は、苦情の通知が Amazon SES に送信され、そこから送信者に転送されます。ほとんどの ISP は苦情を申告した受取人の E メールアドレスを提供しないため、Amazon SES からの苦情通知で送信者に渡される受取人のリストは、Amazon SES が苦情を受け取った ISP や元のメッセージの受取人に基づいて推定された、苦情を申告した可能性のある受取人が記載されています。次の図は、苦情のパスを示しています。



- 自動応答 - E メールが ISP に受け入れられ、受信者に配信されます。ISP は、不在 (OOO) メッセージなどの自動応答を Amazon SES に送信します。Amazon SES は、自動応答通知を送信者に転送します。次の図は、自動応答を示しています。



Amazon SES に対応するプログラムで、自動レスポンスが生成されるメッセージの送信を再試行しないようにしてください。

#### Tip

Amazon SES メールボックスシミュレーターを使用して、正常な配信、バウンス、苦情、OOO をテストしたり、アドレスがサブプレッションリストに含まれている場合にどうなるかをテストしたりできます。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

## E メール形式と Amazon SES

クライアントが Amazon SES にリクエストを送ると、Amazon SES では、Internet Message Format 仕様 (RFC 5322) に準拠した E メールメッセージを構築します。以下に説明するように、E メールはヘッダー、本文、およびエンベロープで構成されます。

- ヘッダー - ルーティングの指示およびメッセージに関する情報が含まれています。例では、送信者のアドレス、受取人のアドレス、件名、および日付となっています。ヘッダーは、手紙の一番上に書かれる情報に似ていますが、ヘッダーにはその他の数多くのタイプの情報 (メッセージの形式など) を含めることができます。
- 本文 - メッセージ自体のテキストが含まれています。
- エンベロープ - SMTP セッション中に E メールクライアントとメールサーバーの間でやり取りされる実際のルーティング情報が含まれています。この E メールエンベロープ情報は、封筒に書かれる情報に似ています。E メールエンベロープのルーティング情報は、通常は E メールヘッダー内のルーティング情報と同じですが、異なる場合もあります。たとえば、ブラインドカーボンコピー (BCC) を送信した場合、エンベロープから取得される実際の受取人アドレスは、ヘッダーから取得されて受取人の E メールクライアントに表示される「To」アドレスと同じではありません。

Eメールのシンプルな例を次に示します。ヘッダーの後には空の行が続き、その後に Eメールの本文が続きます。エンベロープは Eメール自体には含まれず、SMTP セッション中にクライアントとメールサーバーの間で通信されるため、ここには示されていません。

```
Received: from abc.smtp-out.amazonses.com (123.45.67.89) by in.example.com (87.65.43.210);  
  Fri, 17 Dec 2010 14:26:22  
From: "Andrew" <andrew@example.com>;  
To: "Bob" <bob@example.com>  
Date: Fri, 17 Dec 2010 14:26:21 -0800  
Subject: Hello  
Message-ID: <61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com>  
Accept-Language: en-US  
Content-Language: en-US  
Content-Type: text/plain; charset="us-ascii"
```

```
Content-Transfer-Encoding: quoted-printable  
MIME-Version: 1.0
```

```
Hello, I hope you are having a good day.
```

```
-Andrew
```

以下のセクションでは、E メールヘッダーと本文について説明し、Amazon SES を使用するときを提供する必要のある情報を示します。

## E メールヘッダー

E メールメッセージごとに 1 個のヘッダーがあります。ヘッダーの各行にはフィールドが含まれ、その後にはコロン、その後にはフィールド本文が続きます。メールクライアントでメールを読むときに、通常、E メールクライアントには次のヘッダーフィールドの値が表示されます。

- To - メッセージの受信者の E メールアドレス。
- CC - メッセージのカーボンコピーの受信者の E メールアドレス。
- From - Eメールの送信元の E メールアドレス。
- Subject - メッセージトピックの概要。
- Date - Eメールが送信された日時。

これ以外にも、ルーティング情報を提供し、メッセージのコンテンツを説明するヘッダーフィールドが多数あります。E メールクライアントは、そのようなフィールドを、通常、ユーザーに表示しません。Amazon SES が受け入れるヘッダーフィールドの完全なリストについては、「[ヘッダーフィールド \(p. 523\)](#)」を参照してください。Amazon SES を使用する場合は、特に「From」、「Reply-To」、および「Return-Path」ヘッダーフィールドの違いを理解する必要があります。前述のように、「From」アドレスはメッセージ送信者の E メールアドレスですが、「Reply-To」および「Return-Path」は以下のとおりです。

- Reply-To - 返信が送信される E メールアドレス。デフォルトでは、返信は元の送信者の E メールアドレスに送信されます。
- Return-Path - メッセージのバウンスおよび苦情が送信される E メールアドレス。「Return-Path」は、「envelope from」、「envelope sender」、または「MAIL FROM」という名前になることもあります。

### Note

Amazon SES を使用する場合は、常に「Return-Path」パラメータを設定して、バウンスを常時把握し、バウンスの発生時に適切なアクションを実行できるようにすることをお勧めします。

バウンスされたメッセージと受取人を簡単に照合するために、可変エンベロープリターンパス (VERP) を使用できます。VERP を使用することで、メッセージがバウンスされた場合にどの受取人からバウンスされたかを自動的に把握できるように、受取人ごとに異なる「Return-Path」を設定できます。これにより、バウンスメッセージを開いて解析する必要がなくなります。

## E メール本文

E メール本文には、メッセージのテキストが含まれています。本文は、次の形式で送信できます。

- HTML - 受信者の E メールクライアントで HTML を解釈できる場合は、フォーマットされたテキストおよびハイパーリンクを本文に含めることができます。
- プレーンテキスト - 受信者の E メールクライアントがテキストベースの場合は、表示不可能な文字を本文に含めることはできません。
- HTML とプレーンテキスト - 1 つのメッセージで両方の形式を使用して同じコンテンツを送信すると、受信者の E メールクライアントでは機能に応じてどちらの形式で表示するかを決定します。

E メールメッセージを多数の受取人に送信する場合は、HTML とテキストの両方の形式で送信する方法が適しています。HTML 対応 E メールクライアントを使用している受取人は、メッセージに埋め込まれたハイパーリンクをクリックして表示できます。テキストベースの E メールクライアントを使用している受取人については、URL をコピーしてウェブブラウザで表示できるように、URL を記載しておく必要があります。

## Amazon SES に提供する必要がある E メール情報

Amazon SES で E メールを送信する場合、指定する必要がある E メール情報は、Amazon SES をどのように呼び出すかによって異なります。最小限の情報を提供した場合は、Amazon SES ですべてのフォーマットが自動的に処理されます。また、添付ファイルの送信などやや高度な作業を実行する場合は、raw メッセージを自分自身で提供できます。以下のセクションでは、Amazon SES API、Amazon SES SMTP インターフェイス、または Amazon SES コンソールを使用して E メールを送信するときに提供する必要のある情報について説明します。

### Amazon SES API

Amazon SES API を直接呼び出す場合は、SendEmail または SendRawEmail API を呼び出します。提供する必要のある情報の量は、呼び出す API によって異なります。

- SendEmail API では、送信元アドレス、宛先アドレス、メッセージの件名、メッセージ本文のみを提供するだけで済みます。必要に応じて、「Reply-To」アドレスを指定できます。この API を呼び出すと、Amazon SES では、E メールクライアントソフトウェアによる表示に最適化された、適切にフォーマット済みのマルチパート多目的インターネットメール拡張 (MIME) メールメッセージが自動的にアセンブルされます。詳細については、「[Amazon SES API を使用してフォーマット済み E メールを送信する \(p. 114\)](#)」を参照してください。
- SendRawEmail API には、ヘッダー、MIME パーツ、およびコンテンツの種類を指定し、独自の raw E メールメッセージをフォーマットして送信できる柔軟性があります。SendRawEmail は、通常は上級ユーザーに利用されています。メッセージの本文とすべてのヘッダーフィールドは、Internet Message Format 仕様 (RFC 5322) に準拠した形式で入力する必要があります。詳細については、「[Amazon SES API を使用して raw E メールを送信する \(p. 114\)](#)」を参照してください。

AWS SDK を使用して Amazon SES API を呼び出す場合は、対応する関数 (たとえば、Java の場合は SendEmail や SendRawEmail) に上記の情報を指定します。

Amazon SES API を使用した E メール送信の詳細については、「[Amazon SES API を使用して E メールを送信する \(p. 113\)](#)」を参照してください。

### Amazon SES SMTP インターフェイス

SMTP インターフェイスで Amazon SES にアクセスする場合、SMTP クライアントアプリケーションでメッセージがアセンブルされるため、提供する必要のある情報は使用するアプリケーションによって異なります。少なくとも、クライアントとサーバー間の SMTP 交換には、送信元アドレス、宛先アドレス、およびメッセージデータが必要です。SMTP インターフェイスを使用していて、フィードバック転送が有効になっている場合、バウンス、苦情、配信通知は "MAIL FROM" アドレスに送信されます。指定した "Reply-To" アドレスはいずれも使用されません。

Amazon SES SMTP インターフェイスを使用した E メール送信の詳細については、「[Amazon SES SMTP インターフェイスを使用した Eメールの送信 \(p. 88\)](#)」を参照してください。

### Amazon SES コンソール

Amazon SES コンソールを使用してメールを送信する場合、提供する必要のある情報の量は、フォーマットされたメールまたは raw Eメールのどちらを送信するかによって異なります。

- フォーマットされた E メールを送信するには、送信元アドレス、宛先アドレス、メッセージの件名、メッセージ本文を指定する必要があります。Amazon SES では、E メールクライアントソフトウェアに



よる表示に最適化された、適切にフォーマット済みのマルチパート MIME メールメッセージが自動的にアSEMBLされます。また、Reply-To および Return-Path フィールドを指定することもできます。

- raw E メールを送信する場合、送信元アドレス、宛先アドレス、およびメッセージコンテンツを指定します。このコンテンツには、Internet Message Format 仕様 (RFC 5322) に準拠した、メッセージ本文およびすべてのヘッダーフィールドを含める必要があります。

## Amazon SES 認証情報の種類

Amazon Simple Email Service (Amazon SES) と対話するには、セキュリティ認証情報を使用して、自分の身元と、Amazon SES と対話するためのアクセス許可があることを確認します。認証情報にはさまざまな種類があり、何をするかによって使用する認証情報は異なります。たとえば、Amazon SES API を使用して E メールを送信する場合は AWS アクセスキーを使用し、Amazon SES SMTP インターフェイスを使用して E メールを送信する場合は、SMTP 認証情報を使用します。

次の表に、目的に応じて Amazon SES で使用する可能性のある認証情報タイプを示します。

アクセス対象	使用する認証情報	認証情報の構成内容	認証情報の取得方法
Amazon SES API ( Amazon SES API には、直接アクセスするか、AWS SDK、AWS Command Line Interface、または AWS Tools for Windows PowerShell 経由で間接的にアクセスできます。)	AWS アクセスキー	アクセスキー ID とシークレットアクセスキー	AWS 全般のリファレンスの「 <a href="#">アクセスキー</a> 」を参照してください。  Note  セキュリティのベストプラクティスについては、AWS アカウントのアクセスキーではなく AWS Identity and Access Management (IAM) ユーザーのアクセスキーを使用します。AWS アカウントの認証情報によってすべての AWS リソースへのフルアクセス権を付与するため、この認証情報は安全な場所に保存しておき、AWS の日々の操作には IAM ユーザーの認証情報を使用してください。詳細については、AWS 全般のリファレンスの「 <a href="#">ルートアカウント認証情報と IAM ユーザー認証情報</a> 」を参照してください。
Amazon SES SMTP インターフェイス	SMTP 認証情報	ユーザー名とパスワード	「 <a href="#">Amazon SES SMTP 認証情報を取得 (p. 89)</a> 」を参照してください。  Note  Amazon SES SMTP 認証情報は、AWS アクセスキーと IAM ユー

アクセス対象	使用する認証情報	認証情報の構成内容	認証情報の取得方法
			<p>ザーアクセスキーとは異なります。Amazon SES SMTP 認証情報は、実際には IAM 認証情報の一種です。IAM ユーザーは、Amazon SES SMTP 認証情報を作成できますが、ルートアカウント所有者は IAM ユーザーのポリシーによってユーザーに IAM アクション「iam:ListUsers」、「iam:CreateUser」、「iam:CreateAccessKey」、および「iam:PutUserPolicy」へのアクセス許可が付与されることを確認する必要があります。</p>
Amazon SES コンソール	<p>IAM ユーザー名とパスワード</p> <p>または</p> <p>E メールアドレスとパスワード</p>	<p>IAM ユーザー名とパスワード</p> <p>または</p> <p>E メールアドレスとパスワード</p>	<p>AWS 全般のリファレンスの「IAM ユーザー名とパスワード」および「E メールアドレスとパスワード」を参照してください。</p> <p>Note</p> <p>セキュリティのベストプラクティスとして、E メールアドレスとパスワードではなく、IAM ユーザー名とパスワードを使用してください。E メールアドレスとパスワードの組み合わせは AWS アカウント用ですので、AWS の日々の操作に使用するのではなく、安全な場所に保存しておく必要があります。詳細については、AWS 全般のリファレンスの「ルートアカウント認証情報と IAM ユーザー認証情報」を参照してください。</p>

さまざまなタイプの AWS セキュリティ認証情報 (Amazon SES でのみ使用される SMTP 認証情報は除く) の詳細については、AWS 全般のリファレンスの「AWS セキュリティ認証情報」を参照してください。

# Amazon SES による E メール送信のご利用開始にあたって

この入門用チュートリアルでは、Amazon Simple Email Service (Amazon SES) の設定と Eメールの送信方法をステップごとに説明します。最初に「[Amazon SES を開始する前に \(p. 19\)](#)」の情報を確認します。次に、以下のいずれかの方法で Eメールを送信します。

## Amazon SES コンソールの使用

最小限の設定で Amazon SES からテスト Eメールの送信を開始する場合は、この方法を使用します。本番の Eメール送信キャンペーンを開始する準備が整ったら、別の送信方法を使用し、Amazon SES コンソールは主に送信業務をモニタリングします。

このチュートリアルを開始するには、「[Amazon SES コンソールを使用して Eメールを送信する \(p. 20\)](#)」を参照してください。

## Simple Mail Transfer Protocol ( SMTP ) の使用

プログラミングを利用して、または利用せずに Amazon SES SMTP インターフェイスから Eメールを送信する場合、この方法を以下のように使用します。

- SMTP 対応プログラミング言語を使用して、Amazon SES から Eメールを送信できるようにアプリケーションを有効にします。C#、Java、および PHP の例を用意しています。このチュートリアルを開始するには、「[プログラミングで Amazon SES の SMTP インターフェイスにアクセスして Eメールを送信する \(p. 22\)](#)」を参照してください。
- Amazon SES にメールを転送するようにメールサーバーをセットアップするか、Amazon SES を利用して Eメールを送信するように Eメールクライアントや SMTP 対応ソフトウェアパッケージをセットアップします。Postfix、Sendmail、Exim メールサーバーの例が用意されています。このチュートリアルを開始するには、「[既存の Eメールサーバーまたは SMTP 対応アプリケーションを設定して Amazon SES から Eメールを送信する \(p. 31\)](#)」を参照してください。

これらの SMTP 送信方法の基本的な情報については、「[SMTP を使用して Amazon SES から Eメールを送信する \(p. 21\)](#)」を参照してください。

## AWS SDK の使用

この方法では、基盤となる Amazon SES クエリインターフェイスの詳細を扱うライブラリを使用して、Amazon SES API を呼び出します。C#、Java、PHP、Ruby、および Python の例があります。このチュートリアルを開始するには、「[AWS SDK を使用して Amazon SES から Eメールを送信する \(p. 31\)](#)」を参照してください。

## Amazon SES を開始する前に

開始する前に、Amazon SES をセットアップする必要があります。Eメール送信に Amazon SES コンソール、SMTP インターフェイス、Amazon SES API のいずれを使用する場合も、以下の操作を行う必要があります。

- AWS にサインアップ - Amazon SES またはその他の AWS のサービスを使用する前に、AWS アカウントを作成する必要があります。詳細については、[AWS にサインアップする \(p. 48\)](#) を参照してください。
- Eメールアドレスまたはドメインの確認 - Amazon SES を使用して Eメールを送信する際には、常に「From」アドレスを見て、Eメールの所有者が自分であると示されていることを確認する必要があります。

- す。サンドボックスの場合、「To」アドレスも確認する必要があります。E メールアドレスまたはドメイン全体を検証できます。詳細については、[Amazon SES の ID の検証 \(p. 48\)](#) を参照してください。
- アカウントの詳細を入力 - アカウントの詳細を編集して、アカウントを最新の状態に保ちます。アカウントの詳細を入力することで、サンドボックスからの移動をリクエストすることもできます。サンドボックスで Eメールの送信先は、Amazon SES メールボックスシミュレーター、および検証済み Eメールアドレス、またはドメインに制限されます。詳細については、[Amazon SES サンドボックス外への移動 \(p. 74\)](#) を参照してください。

この一覧には、すべての E メール送信方法に必要なセットアップ作業が含まれています。E メール送信方法別のその他のセットアップ作業については、関連する開始方法セクションに記載されています。すべてのセットアップ作業の一覧を確認するには、「[Amazon SES での Eメールのセットアップ \(p. 47\)](#)」を参照してください。

## Amazon SES コンソールを使用して E メールを送信する

Amazon SES で E メールを送信する最も簡単な方法は、Amazon SES コンソールを使用することです。ただしコンソールでは手動で情報を入力する必要があるため、通常、テストメールの送信にのみ使用します。Amazon SES の使用を開始した後は、通常、Amazon SES の SMTP インターフェイスまたは API を使用して E メールを送信します。コンソールは、送信業務のモニタリングに役立ちます。

### Important

この入門用チュートリアルでは、受信されたかどうかを確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに計算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での Eメール送信のテスト \(p. 187\)](#)」を参照してください。

これらの手順を実行する前に、セットアップ手順を「[Amazon SES を開始する前に \(p. 19\)](#)」で必ず確認してください。

Amazon SES コンソールから E メールメッセージを送信するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>)を開きます。

### Note

AWS アカウントにサインインしていない場合、このリンクから自動的にサインインページにつながります。サインイン後、Amazon SES コンソールが表示されます。

2. Amazon SES コンソールの左側のナビゲーションペインで、[ID 管理] の [E メールアドレス] を選択し、「[Amazon SES Eメールアドレスの確認 \(p. 49\)](#)」で確認した E メールアドレスを表示します。
3. アイデンティティのリストで、確認した E メールアドレスのチェックボックスをオンにします。
4. [テストメール送信] を選択します。
5. [Send Test Email] で、[Email Format] を選択します。次の 2 つの選択肢があります。
  - フォーマット済み - これは最も簡単な選択肢です。[Body] テキストボックスにメッセージのみを入力する場合は、このオプションを選択します。E メールを送信するときに、Amazon SES はテキストを E メール形式に変換します。
  - Raw - HTML や添付ファイルなどが含まれるさらに複雑なメッセージを送信する場合には、このオプションを選択します。このような柔軟性により、メッセージは「[Amazon SES API を使用して raw Eメールを送信する \(p. 114\)](#)」の説明に従って各自でフォーマットし、次にフォーマット済み

メッセージの全体 (ヘッダーも含む) を [Body] テキストボックスに貼り付けます。HTML が含まれる次の例を使用して、Raw E メールフォーマットでテストメールを送信してみてください。このメッセージをコピーし、[Body] テキストボックスにすべてを貼り付けます。MIME-Version ヘッダーと Content-Type ヘッダーの間に空白行がないことを確認します。この 2 つの間に空白行がある場合、E メールが HTML ではなくプレーンテキストとしてフォーマットされます。

```
Subject: Amazon SES Raw Email Test
MIME-Version: 1.0
Content-Type: text/html

<!DOCTYPE html>
<html>
<body>
<h1>This text should be large, because it is formatted as a header in HTML.</h1>
<p>Here is a formatted link: <a href="https://docs.aws.amazon.com/ses/latest/DeveloperGuide/Welcome.html">Amazon Simple Email Service Developer Guide</a>.</p>
</body>
</html>
```

- [Send Test Email] で、残りのフィールドに入力します。Amazon SES サンドボックスで作業している場合は、[To (送信先)] フィールドのアドレスが検証済みの E メールアドレスになっていることを確認します。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- [テストメール送信] を選択します。
- Eメールの宛先の E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## SMTP を使用して Amazon SES から E メールを送信する

Amazon SES の SMTP インターフェイスで E メールを送信するには、SMTP 対応のプログラミング言語、E メールサーバー、またはアプリケーションを使用できます。開始する前に、「[Amazon SES を開始する前に \(p. 19\)](#)」の手順を確認します。また、以下の追加情報を取得する必要があります。

- Amazon SES SMTP エンドポイントに接続する Amazon SES SMTP のユーザーネームとパスワード。Amazon SES SMTP のユーザー名とパスワードを取得するには、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

### Important

SMTP 認証情報は、AWS の認証情報とは異なります。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。

- SMTP エンドポイントアドレス。Amazon SES における SMTP エンドポイントのリストについては、「[Amazon SES SMTP エンドポイントへの接続 \(p. 94\)](#)」を参照してください。
- Amazon SES SMTP インターフェイスポート番号。接続方法によって異なります。詳細については、「[Amazon SES SMTP エンドポイントへの接続 \(p. 94\)](#)」を参照してください。

SMTP 認証情報を取得した後、Amazon SES SMTP エンドポイントに接続して E メールを送信できます。この入門用チュートリアルでは、以下の方法による Amazon SES SMTP インターフェイスからの E メール送信について説明します。

- プログラミングで Amazon SES の SMTP インターフェイスにアクセスして E メールを送信する (p. 22)
- 既存の E メールサーバーまたは SMTP 対応アプリケーションを設定して Amazon SES から E メールを送信する (p. 31)

Amazon SES SMTP インターフェイスの詳細については、「[Amazon SES SMTP インターフェイスを使用した Eメールの送信 \(p. 88\)](#)」を参照してください。

## プログラミングで Amazon SES の SMTP インターフェイスにアクセスして E メールを送信する

SMTP 対応のプログラミング言語を使用して、Amazon SES の SMTP インターフェイスにアクセスできます。SMTP 認証情報と Amazon SES SMTP ホスト名およびポート番号を使用して、プログラミング言語の SMTP 生成機能によって E メールを送信します。

### Note

Amazon Elastic Compute Cloud (Amazon EC2) では、デフォルトでポート 25 経由での E メールトラフィックを制限しています。Amazon EC2 から SMTP エンドポイントを介して E メールを送信する際のタイムアウトを回避するには、これらの制限を解除するようリクエストします。詳細については、AWS ナレッジセンターの「[Amazon EC2 インスタンスまたは AWS Lambda 関数からポート 25 の制限を解除する方法](#)」を参照してください。  
このセクションのコード例で、この問題を回避するためにポート 587 を使用します。

このセクションのトピック:

- [C# で SMTP を使用して E メールを送信する \(p. 22\)](#)
- [Java で SMTP を使用して E メールを送信する \(p. 25\)](#)
- [PHP で SMTP を使用して E メールを送信する \(p. 28\)](#)

## C# で SMTP を使用して E メールを送信する

次の手順は、[Microsoft Visual Studio](#) を使用して、Amazon SES から E メールを送信する C# コンソールアプリケーションを作成する方法を示しています。このセクションの手順は Visual Studio 2017 に適用されますが、C# コンソールアプリケーションを作成するプロセスはすべての Microsoft Visual Studio エディションで似ています。

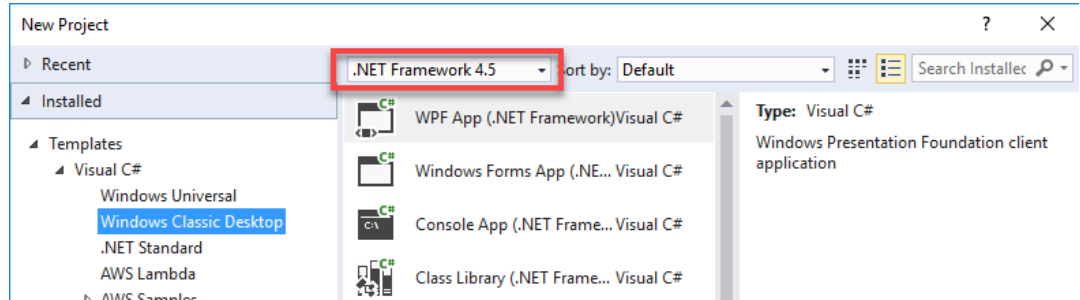
以下の手順を実行する前に、「[Amazon SES を開始する前に \(p. 19\)](#)」と「[SMTP を使用して Amazon SES から Eメールを送信する \(p. 21\)](#)」に記載されているセットアップ作業を完了します。

### Important

この入門用チュートリアルでは、受信されたかどうかを確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに計算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での Eメール送信のテスト \(p. 187\)](#)」を参照してください。

C# で Amazon SES の SMTP インターフェイスから E メールを送信するには

1. 次のステップを実行し、Visual Studio でコンソールプロジェクトを作成します。
  - a. Microsoft Visual Studio を開きます。
  - b. [ファイル] メニューで [New]、[Project] の順に選択します。
  - c. [New Project] ウィンドウの左ペインで、[Installed]、[Templates]、[Visual C#] の順に展開します。
  - d. [Visual C#] で、[Windows Classic Desktop] を選択します。
  - e. 次のイメージに示すように、ウィンドウ上部のメニューで [.NET Framework 4.5] を選択します。



### Note

必要に応じて、より新しいバージョンの .NET Framework を選択できます。

- f. [Console App (.NET Framework)] を選択します。
  - g. [Name (名前)] フィールドに、AmazonSESSample と入力します。
  - h. [OK] を選択します。
2. Visual Studio プロジェクトで、Program.cs のすべてのコンテンツを次のコードに置き換えます。

```
using System;
using System.Net;
using System.Net.Mail;

namespace AmazonSESSample
{
    class Program
    {
        static void Main(string[] args)
        {
            // Replace sender@example.com with your "From" address.
            // This address must be verified with Amazon SES.
            String FROM = "sender@example.com";
            String FROMNAME = "Sender Name";

            // Replace recipient@example.com with a "To" address. If your account
            // is still in the sandbox, this address must be verified.
            String TO = "recipient@amazon.com";

            // Replace smtp_username with your Amazon SES SMTP user name.
            String SMTP_USERNAME = "smtp_username";

            // Replace smtp_password with your Amazon SES SMTP password.
            String SMTP_PASSWORD = "smtp_password";

            // (Optional) the name of a configuration set to use for this message.
            // If you comment out this line, you also need to remove or comment out
            // the "X-SES-CONFIGURATION-SET" header below.
            String CONFIGSET = "ConfigSet";

            // If you're using Amazon SES in a region other than US West (Oregon),
            // replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
            // endpoint in the appropriate AWS Region.
            String HOST = "email-smtp.us-west-2.amazonaws.com";

            // The port you will connect to on the Amazon SES SMTP endpoint. We
            // are choosing port 587 because we will use STARTTLS to encrypt
            // the connection.
            int PORT = 587;

            // The subject line of the email
            String SUBJECT =
```

```
"Amazon SES test (SMTP interface accessed using C#)";

// The body of the email
String BODY =
    "<h1>Amazon SES Test</h1>" +
    "<p>This email was sent through the " +
    "<a href='https://aws.amazon.com/ses'>Amazon SES</a> SMTP interface " +
    "using the .NET System.Net.Mail library.</p>";

// Create and build a new MailMessage object
MailMessage message = new MailMessage();
message.IsBodyHtml = true;
message.From = new MailAddress(FROM, FROMNAME);
message.To.Add(new MailAddress(TO));
message.Subject = SUBJECT;
message.Body = BODY;
// Comment or delete the next line if you are not using a configuration set
message.Headers.Add("X-SES-CONFIGURATION-SET", CONFIGSET);

using (var client = new System.Net.Mail.SmtpClient(HOST, PORT))
{
    // Pass SMTP credentials
    client.Credentials =
        new NetworkCredential(SMTP_USERNAME, SMTP_PASSWORD);

    // Enable SSL encryption
    client.EnableSsl = true;

    // Try to send the message. Show status in console.
    try
    {
        Console.WriteLine("Attempting to send email...");
        client.Send(message);
        Console.WriteLine("Email sent!");
    }
    catch (Exception ex)
    {
        Console.WriteLine("The email was not sent.");
        Console.WriteLine("Error message: " + ex.Message);
    }
}
}
```

3. Program.cs で、以下の E メールアドレスを独自の値に置き換えます。

#### Important

E メールアドレスでは、大文字と小文字は区別されます。確認したアドレスと完全に一致することを確認してください。

- SENDER@EXAMPLE.COM - 「From」E メールアドレスに置き換えます。このアドレスを確認してから、プログラムを実行してください。詳細については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。
  - RECIPIENT@EXAMPLE.COM - 「To」E メールアドレスに置き換えます。アカウントがサンドボックスにまだある場合は、このアドレスを使用前に確認する必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。
4. Program.cs で、以下の SMTP 認証情報を「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」で取得した値に置き換えます。



### Important

SMTP 認証情報は、AWS の認証情報とは異なります。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。

- YOUR\_SMTP\_USERNAME - SMTP ユーザー名に置き換えます。SMTP ユーザー名の認証情報は 20 文字の文字と数字の並びであり、意味のある名前ではありません。
  - YOUR\_SMTP\_PASSWORD - 自分の SMTP パスワードに置き換えます。
5. (オプション) 米国西部 (オレゴン) 以外のリージョンで Amazon SES SMTP エンドポイントを使用する場合は、変数 HOST の値を実際に使用するエンドポイントに変更します。Amazon SES が使用可能な AWS リージョンの SMTP エンドポイント URL のリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。
  6. (オプション) この E メール送信時に設定セットを使用する場合は、変数 CONFIGSET の値を設定セットの名前に変更します。設定セットの詳細については、「[Amazon SES 設定セットの使用 \(p. 257\)](#)」を参照してください。
  7. Program.cs を保存します。
  8. プロジェクトを構築します。[Build]、[Build Solution] の順に選択します。
  9. プログラムを実行します。[Debug]、[Start Debugging] の順に選択します。
  10. 出力を確認します。E メールが正常に送信されると、コンソールに "Email sent!" が表示されます。送信に失敗すると、エラーメッセージが表示されます。
  11. 受信者のアドレスの E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## Java で SMTP を使用して E メールを送信する

この例では、[Eclipse IDE](#) と [JavaMail API](#) を使用して、SMTP インターフェイスで Amazon SES から E メールを送信します。

以下の手順を実行する前に、「[Amazon SES を開始する前に \(p. 19\)](#)」と「[SMTP を使用して Amazon SES から E メールを送信する \(p. 21\)](#)」に記載されているセットアップ作業を完了します。

### Important

この入門用チュートリアルでは、受信されたかどうかを確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに計算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

Java で Amazon SES SMTP インターフェイスを使用して E メールを送信するには

1. ウェブブラウザで、[JavaMail Github ページ](#)に移動します。[Downloads] で、[javax.mail.jar] を選択して最新バージョンの JavaMail をダウンロードします。

### Important

このチュートリアルには、JavaMail バージョン 1.5 以降が必要です。これらの手順は、JavaMail バージョン 1.6.1 を使用してテスト済みです。

2. 次のステップを実行し、Eclipse でプロジェクトを作成します。
  - a. Eclipse を起動します。
  - b. Eclipse で、[File] を選択し、[New]、[Java Project] の順に選択します。
  - c. [Create a Java Project] ダイアログボックスで、プロジェクト名を入力し、[Next] を選択します。
  - d. [Java Settings] ダイアログボックスの [Libraries] タブを選択します。
  - e. [Add External JARs] を選択します。

- f. JavaMail をダウンロードした先のフォルダを参照します。javax.mail.jar ファイルを選択し、[Open] を選択します。
- g. [Java Settings] ダイアログボックスの [Finish] を選択します。
3. Eclipse で、[Package Explorer] ウィンドウのプロジェクトを展開します。
4. プロジェクトの下の [src] ディレクトリを右クリックし、[New]、[Class] の順に選択します。
5. [New Java Class] ダイアログボックスの [Name] フィールドに AmazonSESSample と入力し、[Finish] を選択します。
6. AmazonSESSample.java の内容全体を次のコードに置き換えます。

```
import java.util.Properties;

import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class AmazonSESSample {

    // Replace sender@example.com with your "From" address.
    // This address must be verified.
    static final String FROM = "sender@example.com";
    static final String FROMNAME = "Sender Name";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    static final String TO = "recipient@example.com";

    // Replace smtp_username with your Amazon SES SMTP user name.
    static final String SMTP_USERNAME = "smtp_username";

    // Replace smtp_password with your Amazon SES SMTP password.
    static final String SMTP_PASSWORD = "smtp_password";

    // The name of the Configuration Set to use for this message.
    // If you comment out or remove this variable, you will also need to
    // comment out or remove the header below.
    static final String CONFIGSET = "ConfigSet";

    // Amazon SES SMTP host name. This example uses the US West (Oregon) region.
    // See https://docs.aws.amazon.com/ses/latest/DeveloperGuide/regions.html#region-
    endpoints
    // for more information.
    static final String HOST = "email-smtp.us-west-2.amazonaws.com";

    // The port you will connect to on the Amazon SES SMTP endpoint.
    static final int PORT = 587;

    static final String SUBJECT = "Amazon SES test (SMTP interface accessed using
    Java)";

    static final String BODY = String.join(
        System.getProperty("line.separator"),
        "<h1>Amazon SES SMTP Email Test</h1>",
        "<p>This email was sent with Amazon SES using the ",
        "<a href='https://github.com/javaee/javamail'>Javamail Package</a>",
        " for <a href='https://www.java.com'>Java</a>."
    );

    public static void main(String[] args) throws Exception {

        // Create a Properties object to contain connection configuration information.
```

```
Properties props = System.getProperties();
props.put("mail.transport.protocol", "smtp");
props.put("mail.smtp.port", PORT);
props.put("mail.smtp.starttls.enable", "true");
props.put("mail.smtp.auth", "true");

// Create a Session object to represent a mail session with the specified
properties.
Session session = Session.getDefaultInstance(props);

// Create a message with the specified information.
MimeMessage msg = new MimeMessage(session);
msg.setFrom(new InternetAddress(FROM, FROMNAME));
msg.setRecipient(Message.RecipientType.TO, new InternetAddress(TO));
msg.setSubject(SUBJECT);
msg.setContent(BODY, "text/html");

// Add a configuration set header. Comment or delete the
// next line if you are not using a configuration set
msg.setHeader("X-SES-CONFIGURATION-SET", CONFIGSET);

// Create a transport.
Transport transport = session.getTransport();

// Send the message.
try
{
    System.out.println("Sending...");

    // Connect to Amazon SES using the SMTP username and password you specified
above.
    transport.connect(HOST, SMTP_USERNAME, SMTP_PASSWORD);

    // Send the email.
    transport.sendMessage(msg, msg.getAllRecipients());
    System.out.println("Email sent!");
}
catch (Exception ex) {
    System.out.println("The email was not sent.");
    System.out.println("Error message: " + ex.getMessage());
}
finally
{
    // Close and terminate the connection.
    transport.close();
}
}
```

7. AmazonSESSample.java で、以下の E メールアドレスを独自の値に置き換えます。

#### Important

E メールアドレスでは、大文字と小文字は区別されます。確認したアドレスと完全に一致することを確認してください。

- SENDER@EXAMPLE.COM - 「From」E メールアドレスに置き換えます。このアドレスを確認してから、プログラムを実行してください。詳細については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。
  - RECIPIENT@EXAMPLE.COM - 「To」E メールアドレスに置き換えます。アカウントがサンドボックスにまだある場合は、このアドレスを使用前に確認する必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。
8. AmazonSESSample.java で、以下の SMTP 認証情報を「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」で取得した値に置き換えます。

## Important

SMTP 認証情報は、AWS の認証情報とは異なります。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。

- YOUR SMTP USERNAME - SMTP ユーザー名の認証情報に置き換えます。SMTP ユーザー名の認証情報は 20 文字の文字と数字の並びであり、意味のある名前ではありません。
  - YOUR SMTP PASSWORD - 自分の SMTP パスワードに置き換えます。
9. (オプション) 米国西部 (オレゴン) 以外の AWS リージョンで Amazon SES SMTP エンドポイントを使用する場合は、変数 HOST の値を実際に使用するエンドポイントに変更します。Amazon SES を使用できるリージョンのリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。
  10. (オプション) この E メール送信時に設定セットを使用する場合は、変数 CONFIGSET の値を設定セットの名前に変更します。設定セットの詳細については、「[Amazon SES 設定セットの使用 \(p. 257\)](#)」を参照してください。
  11. AmazonSESSample.java を保存します。
  12. プロジェクトを構築します。[Project]、[Build Project] の順に選択します。(このオプションが無効の場合、自動構築が有効になっている可能性があります。)
  13. プログラムを開始して E メールを送信します。[Run] を選択した後、もう一度 [Run] を選択します。
  14. 出力を確認します。E メールが正常に送信されると、コンソールに "Email sent!" が表示されます。送信に失敗すると、エラーメッセージが表示されます。
  15. 受信者のアドレスの E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## PHP で SMTP を使用して E メールを送信する

この例では PHPMailer クラスを使用し、SMTP インターフェイスで Amazon SES を介して E メールを送信します。

### Important

このチュートリアルでは、受信を確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに加算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

### Prerequisites

開始する前に、次のタスクを実行します。

- Amazon SES で E メールアドレスを検証する - Amazon SES で E メールを送信するには、送信者の E メールアドレスを所有していることを検証する必要があります。アカウントが Amazon SES サンドボックスにまだある場合は、受信者の E メールアドレスも検証する必要があります。E メールアドレスを検証する最も簡単な方法は、Amazon SES コンソールを使用することです。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- SMTP 認証情報を取得する - Amazon SES の SMTP インターフェイスにアクセスするには、Amazon SES の SMTP ユーザー名とパスワードが必要です。SMTP 認証情報と AWS 認証情報は同じではありません。Amazon SES コンソールの [SMTP の設定ページ](#)で、SMTP の認証情報を見つけることができます。SMTP 認証情報の詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。
- PHP をインストールする — PHP は、<http://php.net/downloads.php>。PHP をインストールした後、コマンドプロンプトから PHP を実行できるように環境変数に PHP のパスを追加します。

- Composer dependency manager をインストールする - Composer dependency manager では、PHPMailer クラスとその依存関係をダウンロードしてインストールできます。Composer をインストールするには、<https://getcomposer.org/download> のインストール手順を参照してください。
- PHPMailer クラスをインストール - Composer をインストールしたら、以下のコマンドを実行して PHPMailer をインストールします。

```
path/to/composer require phpmailer/phpmailer
```

上のコマンドで、`path/to/` を、Composer をインストールしたパスに置き換えます。

## Procedure

次の手順は、PHP で Amazon SES を介して E メールを送信する方法を示しています。

PHP で Amazon SES の SMTP インターフェイスを使用して E メールを送信するには

1. `amazon-ses-smtp-sample.php` という名前のファイルを作成します。テキストエディターでファイルを開き、次のコードを貼り付けます。

```
<?php

// Import PHPMailer classes into the global namespace
// These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

// If necessary, modify the path in the require statement below to refer to the
// location of your Composer autoload.php file.
require 'vendor/autoload.php';

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
$sender = 'sender@example.com';
$senderName = 'Sender Name';

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
$recipient = 'recipient@example.com';

// Replace smtp_username with your Amazon SES SMTP user name.
$usernameSmtplib = 'smtp_username';

// Replace smtp_password with your Amazon SES SMTP password.
$passwordSmtplib = 'smtp_password';

// Specify a configuration set. If you do not want to use a configuration
// set, comment or remove the next line.
$configSet = 'ConfigSet';

// If you're using Amazon SES in a region other than US West (Oregon),
// replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
// endpoint in the appropriate region.
$host = 'email-smtp.us-west-2.amazonaws.com';
$port = 587;

// The subject line of the email
$subject = 'Amazon SES test (SMTP interface accessed using PHP)';

// The plain-text body of the email
$bodyText = "Email Test\r\nThis email was sent through the
    Amazon SES SMTP interface using the PHPMailer class.";
```

```
// The HTML-formatted body of the email
$bodyHtml = '<h1>Email Test</h1>
  <p>This email was sent through the
  <a href="https://aws.amazon.com/ses">Amazon SES</a> SMTP
  interface using the <a href="https://github.com/PHPMailer/PHPMailer">
  PHPMailer</a> class.</p>';

$mail = new PHPMailer(true);

try {
  // Specify the SMTP settings.
  $mail->isSMTP();
  $mail->setFrom($sender, $senderName);
  $mail->Username = $usernameSmtp;
  $mail->Password = $passwordSmtp;
  $mail->Host = $host;
  $mail->Port = $port;
  $mail->SMTPAuth = true;
  $mail->SMTPSecure = 'tls';
  $mail->addCustomHeader('X-SES-CONFIGURATION-SET', $configurationSet);

  // Specify the message recipients.
  $mail->addAddress($recipient);
  // You can also add CC, BCC, and additional To recipients here.

  // Specify the content of the message.
  $mail->isHTML(true);
  $mail->Subject = $subject;
  $mail->Body = $bodyHtml;
  $mail->AltBody = $bodyText;
  $mail->Send();
  echo "Email sent!" , PHP_EOL;
} catch (phpmailerException $e) {
  echo "An error occurred. {$e->errorMessage()}", PHP_EOL; //Catch errors from
  PHPMailer.
} catch (Exception $e) {
  echo "Email not sent. {$mail->ErrorInfo}", PHP_EOL; //Catch errors from Amazon SES.
}

?>
```

2. `amazon-ses-smtp-sample.php` で、以下を独自の値に置き換えます。

- **sender@example.com** — Amazon SES で検証した E メールアドレスに置き換えます。詳細については、「[ID の検証 \(p. 48\)](#)」を参照してください。Amazon SES では、E メールアドレスの大文字と小文字が区別されます。検証したアドレスと完全に一致するアドレスを入力してください。
- **recipient@example.com** — 受信者のアドレスに置き換えます。アカウントがサンドボックスにまだある場合は、このアドレスを使用前に確認する必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。検証したアドレスと完全に一致するアドレスを入力してください。
- **smtp\_username** - SMTP ユーザー名の認証情報に置き換えます。これは、Amazon SES コンソールの [SMTP の設定](#) ページで取得できます。これは AWS アクセスキー ID とは異なります。SMTP ユーザー名の認証情報は 20 文字の文字と数字の並びであり、意味のある名前ではありません。
- **smtp\_password** - SMTP パスワードに置き換えます。このパスワードは、Amazon SES コンソールの [SMTP の設定](#) ページで取得できます。これは AWS シークレットアクセスキーとは異なります。
- (オプション) **ConfigSet** — この E メールを送信する際に設定セットを使用する場合、この値を設定セットの名前で置き換えます。設定セットの詳細については、「[Amazon SES 設定セットの使用 \(p. 257\)](#)」を参照してください。

- (オプション) `email-smtp.us-west-2.amazonaws.com` — 米国西部 (オレゴン) 以外のリージョンで Amazon SES SMTP エンドポイントを使用する場合は、この値を使用するリージョンの Amazon SES SMTP エンドポイントで置き換えます。Amazon SES が使用可能な AWS リージョンの SMTP エンドポイント URL のリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。
3. `amazon-ses-smtp-sample.php` を保存します。
  4. プログラムを実行するには、`amazon-ses-smtp-sample.php` と同じディレクトリでコマンドプロンプトを開き、`php amazon-ses-smtp-sample.php` と入力します。
  5. 出力を確認します。E メールが正常に送信されると、コンソールに "Email sent!" が表示されます。送信に失敗すると、エラーメッセージが表示されます。
  6. 受信者のアドレスの E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## 既存の E メールサーバーまたは SMTP 対応アプリケーションを設定して Amazon SES から E メールを送信する

プログラミングを行わずに、メールサーバー、E メールクライアント、E メール送信ソフトウェアパッケージを設定して、Amazon SES からメッセージを送信できます。

最初に、「[SMTP を使用して Amazon SES から E メールを送信する \(p. 21\)](#)」を確認します。次に、メールを Amazon SES に転送するメールサーバーの設定方法を紹介する、以下のいずれかのトピックを確認します。

- [Postfix の設定 \(p. 97\)](#)
- [Amazon SES と Sendmail の統合 \(p. 100\)](#)
- [Amazon SES と Exim の統合 \(p. 105\)](#)

## AWS SDK を使用して Amazon SES から E メールを送信する

Amazon SES API を使用して E メールを送信するには、Query インターフェイスを直接使用できます。または、AWS SDK を使用して HTTP のリクエストとレスポンスのアセンブルや解析など、低レベルの詳細を処理できます。

AWS SDK を使用して E メールを送信する前に、「[Amazon SES を開始する前に \(p. 19\)](#)」の手順を確認してください。このセクションのチュートリアルを完了するには、以下の操作も必要です。

- AWS SDK のダウンロード - AWS SDK をダウンロードしてインストールします。詳細については、「[AWS SDK のダウンロード \(p. 67\)](#)」を参照してください。
- AWS 認証情報を取得する — Amazon SES にプログラムでアクセスするには、AWS アクセスキーが必要です。詳細については、「[AWS アクセスキーの取得 \(p. 67\)](#)」を参照してください。
- 認証情報の共有ファイルの作成 - 「[認証情報の共有ファイルの作成 \(p. 31\)](#)」の手順に従って認証情報の共有ファイルを作成します。

上記の前提条件を満たしたら、「[AWS SDK を使用してプログラムで Amazon SES から E メールを送信します。 \(p. 32\)](#)」を参照してください。

### 認証情報の共有ファイルの作成

次の手順は、ホームディレクトリに認証情報の共有ファイルを作成する方法を示しています。SDK サンプルコードが正常に動作するには、このファイルを作成する必要があります。

1. テキストエディタで新規ファイルを作成します。ファイルに次のコードを貼り付けます。

```
[default]
aws_access_key_id = YOUR_AWS_ACCESS_KEY_ID
aws_secret_access_key = YOUR_AWS_SECRET_ACCESS_KEY
```

2. 先ほど作成したテキストファイルで、YOUR\_AWS\_ACCESS\_KEY を一意の AWS アクセスキー ID に置き換え、YOUR\_AWS\_SECRET\_ACCESS\_KEY を一意の AWS シークレットアクセスキーに置き換えます。
3. ファイルを保存します。次の表は、オペレーティングシステム別の正しい場所とファイル名を示しています。

使用しているクライアント	ファイルの保存名
Windows	C:\Users\<>yourUserName>\.aws\credentials
Linux、macOS、または Unix	~/.aws/credentials

#### Important

認証情報ファイルを保存するときは、ファイル拡張子を含めないでください。

## AWS SDK を使用してプログラムで Amazon SES から E メールを送信します。

AWS SDK を使用して Amazon SES から E メールを送信できます。AWS SDK は、複数のプログラミング言語で利用できます。詳細については、[Tools for Amazon Web Services](#) を参照してください。

#### Note

このセクションのチュートリアルを実施する前に、「[AWS SDK を使用して Amazon SES から E メールを送信する \(p. 31\)](#)」に示している前提条件を満たしてください (まだの場合)。

追加のコード例については、「[Amazon SES コード例 \(p. 425\)](#)」を参照してください。

このセクションのトピック:

- [AWS SDK for .NET を使用して E メールを送信する \(p. 32\)](#)
- [AWS SDK for Java を使用して E メールを送信する \(p. 35\)](#)
- [AWS SDK for PHP を使用して E メールを送信する \(p. 38\)](#)
- [AWS SDK for Ruby を使用して E メールを送信する \(p. 41\)](#)
- [AWS SDK for Python \(Boto\) を使用して E メールを送信する \(p. 43\)](#)

## AWS SDK for .NET を使用して E メールを送信する

以下の手順は、[Visual Studio](#) と AWS SDK for .NET を使用して、Amazon SES 経由で E メールを送信する方法を示しています。

このソリューションは次のコンポーネントを使用してテスト済みです。

- Microsoft Visual Studio コミュニティ 2017、バージョン 15.4.0。
- Microsoft .NET Framework バージョン 4.6.1。
- NuGet を使用してインストールされた AWSSDK.Core パッケージ (バージョン 3.3.19)。
- NuGet を使用してインストールされた AWSSDK.SimpleEmail パッケージ (バージョン 3.3.6.1)。



## Note

この入門用チュートリアルでは、受信されたかどうかを確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに加算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

## Prerequisites

開始する前に、次のタスクを実行します。

- Amazon SES で E メールアドレスを検証する - Amazon SES で E メールを送信するには、送信者の E メールアドレスを所有していることを検証する必要があります。アカウントが Amazon SES サンドボックスにまだある場合は、受信者の E メールアドレスも検証する必要があります。E メールアドレスを検証する最も簡単な方法は、Amazon SES コンソールを使用することです。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- AWS 認証情報を取得する — SDK を使用して Amazon SES にアクセスするには、AWS アクセスキー ID と AWS シークレットアクセスキーが必要です。認証情報を取得するには、AWS Management Console の「[セキュリティの認証情報](#)」のページを参照してください。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。
- Visual Studio のインストール - Visual Studio は <https://www.visualstudio.com/> から入手可能です。
- 共有認証情報ファイルの作成 — このセクションのサンプルコードが正常に機能するためには、共有認証情報ファイルを作成する必要があります。詳細については、「[」を参照してください](#) [認証情報の共有ファイルの作成 \(p. 31\)](#)

## Procedure

次の手順は、AWS SDK for .NET を使用して Amazon SES 経由で E メールを送信する方法を示しています。

AWS SDK for .NET を使用して E メールを送信するには

1. 以下のステップを実行して、新しいプロジェクトを作成します。
  - a. Visual Studio を起動します。
  - b. [ファイル] メニューで [New]、[Project] の順に選択します。
  - c. [New Project] ウィンドウの左側のパネルで、[Installed]、[Visual C#] の順に展開します。
  - d. 右側のパネルで、[Console App (.NET Framework)] を選択します。
  - e. [Name (名前)] に **AmazonSESSample** と入力し、[OK] を選択します。
2. 次のステップを実行して、NuGet を使用して Amazon SES パッケージをソリューションに含めます。
  - a. ソリューションエクスプローラーペインで、プロジェクトを右クリックして、コンテキストメニューの [NuGet パッケージの管理] を選択します。
  - b. [NuGet: AmazonSESSample] タブで、[参照] を選択します。
  - c. 検索ボックスに [**AWSSDK.SimpleEmail**] と入力します
  - d. [AWSSDK.SimpleEmail] パッケージを選択し、[インストール] を選択します。
  - e. [Preview Changes] ウィンドウで、[OK] を選択します。
3. [Program.cs] タブで、次のコードを貼り付けます。

```
using Amazon;  
using System;
```

```
using System.Collections.Generic;
using Amazon.SimpleEmail;
using Amazon.SimpleEmail.Model;

namespace AmazonSESSample
{
    class Program
    {
        // Replace sender@example.com with your "From" address.
        // This address must be verified with Amazon SES.
        static readonly string senderAddress = "sender@example.com";

        // Replace recipient@example.com with a "To" address. If your account
        // is still in the sandbox, this address must be verified.
        static readonly string receiverAddress = "recipient@example.com";

        // The configuration set to use for this email. If you do not want to use a
        // configuration set, comment out the following property and the
        // ConfigurationSetName = configSet argument below.
        static readonly string configSet = "ConfigSet";

        // The subject line for the email.
        static readonly string subject = "Amazon SES test (AWS SDK for .NET)";

        // The email body for recipients with non-HTML email clients.
        static readonly string textBody = "Amazon SES Test (.NET)\r\n"
            + "This email was sent through Amazon SES "
            + "using the AWS SDK for .NET.";

        // The HTML body of the email.
        static readonly string htmlBody = @"<html>
<head></head>
<body>
<h1>Amazon SES Test (AWS SDK for .NET)</h1>
<p>This email was sent with
<a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the
<a href='https://aws.amazon.com/sdk-for-net/'>
AWS SDK for .NET</a>.</p>
</body>
</html>";

        static void Main(string[] args)
        {
            // Replace USWest2 with the AWS Region you're using for Amazon SES.
            // Acceptable values are EUWest1, USEast1, and USWest2.
            using (var client = new
AmazonSimpleEmailServiceClient(RegionEndpoint.USWest2))
            {
                var sendRequest = new SendEmailRequest
                {
                    Source = senderAddress,
                    Destination = new Destination
                    {
                        ToAddresses =
                            new List<string> { receiverAddress }
                    },
                    Message = new Message
                    {
                        Subject = new Content(subject),
                        Body = new Body
                        {
                            Html = new Content
                            {
                                {
                                    Charset = "UTF-8",
                                    Data = htmlBody
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
        Text = new Content
        {
            Charset = "UTF-8",
            Data = textBody
        }
    },
    // If you are not using a configuration set, comment
    // or remove the following line
    ConfigurationSetName = configSet
};
try
{
    Console.WriteLine("Sending email using Amazon SES...");
    var response = client.SendEmail(sendRequest);
    Console.WriteLine("The email was sent successfully.");
}
catch (Exception ex)
{
    Console.WriteLine("The email was not sent.");
    Console.WriteLine("Error message: " + ex.Message);
}
}

Console.Write("Press any key to continue...");
Console.ReadKey();
}
}
```

4. コードエディタで、以下の作業を行います。

- `sender@example.com` を "差出人:" の E メールアドレスに置き換えます。このアドレスは確認する必要があります。詳細については、「[the section called "ID の検証" \(p. 48\)](#)」を参照してください。
- `recipient@example.com` を "宛先:" のアドレスに置き換えます。アカウントがサンドボックスにまだある場合は、このアドレスも確認する必要があります。
- `ConfigSet` を、この E メールを送信するとき使用する設定セットの名前に置き換えます。
- `USWest2` を、Amazon SES を使用して E メールを送信するために使用する AWS リージョンのエンドポイントの名前に置き換えます。Amazon SES を使用できるリージョンのリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。

終了したら、`Program.cs` を保存します。

5. 次の手順に従ってアプリケーションをビルドおよび実行します。

- a. [Build] メニューの [Build Solution] を選択します。
  - b. [Debug] メニューの [Start Debugging] を選択します。コンソールウィンドウが表示されます。
6. コンソールの出力を確認します。E メールが正常に送信されると、コンソールに "The email was sent successfully." と表示されます
  7. E メールが正常に送信されたら、受信者アドレスの E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## AWS SDK for Java を使用して E メールを送信する

以下の手順では、[Eclipse IDE for Java EE Developers](#)と [AWS Toolkit for Eclipse](#) を使用して AWS SDK プロジェクトを作成し、Java コードを変更して Amazon SES から E メールを送信する方法を示します。

## Important

この入門用チュートリアルでは、受信されたかどうかを確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに加算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

## Prerequisites

開始する前に、次のタスクを実行します。

- Amazon SES で E メールアドレスを検証する - Amazon SES で E メールを送信するには、送信者の E メールアドレスを所有していることを検証する必要があります。アカウントが Amazon SES サンドボックスにまだある場合は、受信者の E メールアドレスも検証する必要があります。E メールアドレスを検証する最も簡単な方法は、Amazon SES コンソールを使用することです。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- AWS 認証情報を取得する — SDK を使用して Amazon SES にアクセスするには、AWS アクセスキー ID と AWS シークレットアクセスキーが必要です。認証情報を取得するには、AWS Management Console の「[セキュリティの認証情報](#)」のページを参照してください。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。
- Eclipse のインストール - Eclipse は <https://www.eclipse.org/downloads> からダウンロードできます。このチュートリアルのコードは、バージョン 1.8 の Java Runtime Environment を実行する Eclipse Neon.3 (バージョン 4.6.3) でテスト済みです。
- AWS Toolkit for Eclipse のインストール - インストールした Eclipse に AWS Toolkit for Eclipse を追加する手順については、<https://aws.amazon.com/eclipse> を参照してください。このチュートリアルのコードはバージョン 2.3.1 の AWS Toolkit for Eclipse でテスト済みです。
- 共有認証情報ファイルの作成 — このセクションのサンプルコードが正常に機能するためには、共有認証情報ファイルを作成する必要があります。詳細については、「[」を参照してください](#) [認証情報の共有ファイルの作成 \(p. 31\)](#)

## Procedure

次の手順は、AWS SDK for Java を使用して Amazon SES 経由で E メールを送信する方法を示しています。

AWS SDK for Java を使用して E メールを送信するには

1. 以下のステップを実行し、Eclipse で AWS Java プロジェクトを作成します。
  - a. Eclipse を起動します。
  - b. [File] メニューで [New]、[Other] の順に選択します。[New] ウィンドウで、[AWS] フォルダを展開し、[AWS Java Project] を選択します。
  - c. [New AWS Java Project] ダイアログボックスで、以下の操作を行います。
    - i. [Project name] に、プロジェクト名を入力します。
    - ii. AWS SDK for Java のサンプルで、[Amazon Simple Email Service JavaMail のサンプル] を選択します。
    - iii. [Finish] を選択します。
2. Eclipse の [Package Explorer] ペインで、プロジェクトを展開します。
3. プロジェクトの src/main/java フォルダ、com.amazon.aws.samples フォルダの順に展開し、AmazonSESSample.java をダブルクリックします。
4. AmazonSESSample.java の内容全体を次のコードに置き換えます。

```
package com.amazonaws.samples;
```

```
import java.io.IOException;

import com.amazonaws.regions.Regions;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailService;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailServiceClientBuilder;
import com.amazonaws.services.simpleemail.model.Body;
import com.amazonaws.services.simpleemail.model.Content;
import com.amazonaws.services.simpleemail.model.Destination;
import com.amazonaws.services.simpleemail.model.Message;
import com.amazonaws.services.simpleemail.model.SendEmailRequest;

public class AmazonSESSample {

    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    static final String FROM = "sender@example.com";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    static final String TO = "recipient@example.com";

    // The configuration set to use for this email. If you do not want to use a
    // configuration set, comment the following variable and the
    // .withConfigurationSetName(CONFIGSET); argument below.
    static final String CONFIGSET = "ConfigSet";

    // The subject line for the email.
    static final String SUBJECT = "Amazon SES test (AWS SDK for Java)";

    // The HTML body for the email.
    static final String HTMLBODY = "<h1>Amazon SES test (AWS SDK for Java)</h1>"
        + "<p>This email was sent with <a href='https://aws.amazon.com/ses/'>"
        + "Amazon SES</a> using the <a href='https://aws.amazon.com/sdk-for-java/'>"
        + "AWS SDK for Java</a>";

    // The email body for recipients with non-HTML email clients.
    static final String TEXTBODY = "This email was sent through Amazon SES "
        + "using the AWS SDK for Java.";

    public static void main(String[] args) throws IOException {

        try {
            AmazonSimpleEmailService client =
                AmazonSimpleEmailServiceClientBuilder.standard()
                // Replace US_WEST_2 with the AWS Region you're using for
                // Amazon SES.
                .withRegion(Regions.US_WEST_2).build();
            SendEmailRequest request = new SendEmailRequest()
                .withDestination(
                    new Destination().withToAddresses(TO))
                .withMessage(new Message()
                    .withBody(new Body()
                        .withHtml(new Content()
                            .withCharset("UTF-8").withData(HTMLBODY))
                        .withText(new Content()
                            .withCharset("UTF-8").withData(TEXTBODY)))
                    .withSubject(new Content()
                        .withCharset("UTF-8").withData(SUBJECT)))
                .withSource(FROM)
                // Comment or remove the next line if you are not using a
                // configuration set
                .withConfigurationSetName(CONFIGSET);
            client.sendEmail(request);
            System.out.println("Email sent!");
        } catch (Exception ex) {
```

```
        System.out.println("The email was not sent. Error message: "
            + ex.getMessage());
    }
}
}
```

5. AmazonSESSample.java で、以下を独自の値に置き換えます。

#### Important

E メールアドレスでは、大文字と小文字は区別されます。確認したアドレスと完全に一致することを確認してください。

- SENDER@EXAMPLE.COM - 「From」E メールアドレスに置き換えます。このアドレスを確認してから、プログラムを実行してください。詳細については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。
  - RECIPIENT@EXAMPLE.COM - 「To」E メールアドレスに置き換えます。アカウントがサンドボックスにまだある場合は、このアドレスを使用前に確認する必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。
  - (オプション)us-west-2 — 米国西部 (オレゴン) 以外のリージョンで Amazon SES を使用する場合は、この値を使用するリージョンの名前で置き換えます。Amazon SES を使用できるリージョンのリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。
6. AmazonSESSample.java を保存します。
  7. プロジェクトを構築します。[Project]、[Build Project] の順に選択します。

#### Note

このオプションが無効の場合、自動構築が有効になっている可能性があります。その場合は、このステップをスキップします。

8. プログラムを開始して E メールを送信します。[Run] を選択した後、もう一度 [Run] を選択します。
9. Eclipse でコンソールペインの出力を確認します。E メールが正常に送信されると、コンソールに "Email sent!" が表示されます。送信に失敗すると、エラーメッセージが表示されます。
10. E メールが正常に送信されたら、受信者アドレスの E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## AWS SDK for PHP を使用して E メールを送信する

このトピックでは、[AWS SDK for PHP](#) を使用して Amazon SES 経由で E メールを送信する方法を示します。

#### Important

このチュートリアルでは、受信を確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに加算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

#### Prerequisites

開始する前に、次のタスクを実行します。

- Amazon SES で E メールアドレスを検証する - Amazon SES で E メールを送信するには、送信者の E メールアドレスを所有していることを検証する必要があります。アカウントが Amazon SES サンドボックスにまだある場合は、受信者の E メールアドレスも検証する必要があります。E メールアドレスを検証する最も簡単な方法は、Amazon SES コンソールを使用することです。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。

- AWS 認証情報を取得する — SDK を使用して Amazon SES にアクセスするには、AWS アクセスキー ID と AWS シークレットアクセスキーが必要です。認証情報を取得するには、AWS Management Console の「[セキュリティの認証情報](#)」のページを参照してください。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。
- PHP をインストールする — PHP は、<http://php.net/downloads.php> から入手できます。このチュートリアルでは、バージョン 5.5 以上の PHP が必要です。PHP をインストールした後、コマンドプロンプトから PHP を実行できるように環境変数に PHP のパスを追加します。このチュートリアルのコードは PHP 7.2.7 でテスト済みです。
- AWS SDK for PHP バージョン 3 をインストールする — ダウンロードとインストールの手順については、[AWS SDK for PHP のドキュメント](#)を参照してください。このチュートリアルのコードは SDK バージョン 3.64.13 でテスト済みです。
- 共有認証情報ファイルの作成 — このセクションのサンプルコードが正常に機能するためには、共有認証情報ファイルを作成する必要があります。詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください

## Procedure

次の手順は、AWS SDK for PHP を使用して Amazon SES 経由で E メールを送信する方法を示しています。

### AWS SDK for PHP を使用して Amazon SES から E メールを送信する

1. テキストエディタで `amazon-ses-sample.php` という名前のファイルを作成します。次のコードを貼り付けます。

```
<?php

// If necessary, modify the path in the require statement below to refer to the
// location of your Composer autoload.php file.
require 'vendor/autoload.php';

use Aws\Ses\SesClient;
use Aws\Exception\AwsException;

// Create an SesClient. Change the value of the region parameter if you're
// using an AWS Region other than US West (Oregon). Change the value of the
// profile parameter if you want to use a profile in your credentials file
// other than the default.
$SesClient = new SesClient([
    'profile' => 'default',
    'version' => '2010-12-01',
    'region' => 'us-west-2'
]);

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
$sender_email = 'sender@example.com';

// Replace these sample addresses with the addresses of your recipients. If
// your account is still in the sandbox, these addresses must be verified.
$recipient_emails = ['recipient1@example.com', 'recipient2@example.com'];

// Specify a configuration set. If you do not want to use a configuration
// set, comment the following variable, and the
// 'ConfigurationSetName' => $configuration_set argument below.
$configuration_set = 'ConfigSet';

$subject = 'Amazon SES test (AWS SDK for PHP)';
$plaintext_body = 'This email was sent with Amazon SES using the AWS SDK for PHP.' ;
$html_body = '<h1>AWS Amazon Simple Email Service Test Email</h1>'.
```

```
        '<p>This email was sent with <a href="https://aws.amazon.com/ses/">'.  
        'Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-  
php/">'.  
        'AWS SDK for PHP</a>.</p>';  
$char_set = 'UTF-8';  
  
try {  
    $result = $SesClient->sendEmail([  
        'Destination' => [  
            'ToAddresses' => $recipient_emails,  
        ],  
        'ReplyToAddresses' => [$sender_email],  
        'Source' => $sender_email,  
        'Message' => [  
            'Body' => [  
                'Html' => [  
                    'Charset' => $char_set,  
                    'Data' => $html_body,  
                ],  
                'Text' => [  
                    'Charset' => $char_set,  
                    'Data' => $plaintext_body,  
                ],  
            ],  
            'Subject' => [  
                'Charset' => $char_set,  
                'Data' => $subject,  
            ],  
        ],  
        // If you aren't using a configuration set, comment or delete the  
        // following line  
        'ConfigurationSetName' => $configuration_set,  
    ]);  
    $messageId = $result['MessageId'];  
    echo("Email sent! Message ID: $messageId." . "\n");  
} catch (AwsException $e) {  
    // output error message if fails  
    echo $e->getMessage();  
    echo("The email was not sent. Error message: ".$e->getAwsErrorMessage()."\n");  
    echo "\n";  
}
```

2. amazon-ses-sample.php で、以下を独自の値に置き換えます。

- **path\_to\_sdk\_inclusion** — プログラムに AWS SDK for PHP を含めるために必要なパスに置き換えます。詳細については、[AWS SDK for PHP ドキュメント](#)を参照してください。
- **sender@example.com** — Amazon SES で検証した E メールアドレスに置き換えます。詳細については、「[ID の検証 \(p. 48\)](#)」を参照してください。Amazon SES では、E メールアドレスの大文字と小文字が区別されます。検証したアドレスと完全に一致するアドレスを入力してください。
- **recipient1@example.com**、**recipient2@example.com** — 受取人のアドレスに置き換えます。アカウントがサンドボックスにまだある場合は、受取人のアドレスも確認済みである必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。検証したアドレスと完全に一致するアドレスを入力してください。
- (オプション)**ConfigSet** — この E メールを送信する際に設定セットを使用する場合、この値を設定セットの名前で置き換えます。設定セットの詳細については、「[Amazon SES 設定セットの使用 \(p. 257\)](#)」を参照してください。
- (オプション)**us-west-2** — 米国西部 (オレゴン) 以外のリージョンで Amazon SES を使用する場合は、この値を使用するリージョンの名前で置き換えます。Amazon SES を使用できるリージョンのリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。

3. amazon-ses-sample.php を保存します。



4. プログラムを実行するには、`amazon-ses-sample.php` と同じディレクトリでコマンドプロンプトを開き、次のコマンドを入力します。

```
$ php amazon-ses-sample.php
```

5. 出力を確認します。E メールが正常に送信されると、コンソールに "Email sent!" が表示されます。送信に失敗すると、エラーメッセージが表示されます。

#### Note

プログラムの実行時に "cURL error 60: SSL certificate problem" エラーが発生した場合は、[AWS SDK for PHP のドキュメント](#)に従って、最新の CA バンドルをダウンロードしてください。次に、`amazon-ses-sample.php` で、`SesClient::factory` 配列に以下の行を追加し、ダウンロードした CA バンドルのパスで `path_of_certs` を置き換えて、プログラムを再実行します。

```
'http' => [  
  'verify' => 'path_of_certs\ca-bundle.crt'  
]
```

6. 受信者のアドレスの E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## AWS SDK for Ruby を使用して E メールを送信する

このトピックでは、[AWS SDK for Ruby](#) を使用して Amazon SES 経由で E メールを送信する方法を示します。

#### Important

このチュートリアルでは、受信を確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに加算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

#### Prerequisites

開始する前に、次のタスクを実行します。

- Amazon SES で E メールアドレスを検証する - Amazon SES で E メールを送信するには、送信者の E メールアドレスを所有していることを検証する必要があります。アカウントが Amazon SES サンドボックスにまだある場合は、受信者の E メールアドレスも検証する必要があります。E メールアドレスを検証する最も簡単な方法は、Amazon SES コンソールを使用することです。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- AWS 認証情報を取得する — SDK を使用して Amazon SES にアクセスするには、AWS アクセスキー ID と AWS シークレットアクセスキーが必要です。認証情報を取得するには、AWS Management Console の「[セキュリティの認証情報](#)」のページを参照してください。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。
- Ruby のインストール - Ruby は <https://www.ruby-lang.org/en/downloads/> からダウンロードできます。このチュートリアルのコードは Ruby 1.9.3 でテスト済みです。Ruby をインストールした後、コマンドプロンプトから Ruby を実行できるように環境変数に Ruby のパスを追加します。
- AWS SDK for Ruby をインストール - ダウンロードとインストールの手順については、[AWS SDK for Ruby デベロッパーガイド](#)の「AWS SDK for Ruby」を参照してください。このチュートリアルのサンプルコードは AWS SDK for Ruby バージョン 2.9.36 でテスト済みです。
- 共有認証情報ファイルの作成 — このセクションのサンプルコードが正常に機能するためには、共有認証情報ファイルを作成する必要があります。詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください

## Procedure

次の手順は、AWS SDK for Ruby を使用して Amazon SES 経由で E メールを送信する方法を示しています。

### AWS SDK for Ruby を使用して Amazon SES から E メールを送信する

1. テキストエディタで `amazon-ses-sample.rb` という名前のファイルを作成します。ファイルに次のコードを貼り付けます。

```
require 'aws-sdk'

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = "sender@example.com"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
recipient = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable and the
# configuration_set_name: configsetname argument below.
configsetname = "ConfigSet"

# Replace us-west-2 with the AWS Region you're using for Amazon SES.
awsregion = "us-west-2"

# The subject line for the email.
subject = "Amazon SES test (AWS SDK for Ruby)"

# The HTML body of the email.
htmlbody =
  '<h1>Amazon SES test (AWS SDK for Ruby)</h1>'\
  '<p>This email was sent with <a href="https://aws.amazon.com/ses/">\
  Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-ruby/">\
  AWS SDK for Ruby</a>.'

# The email body for recipients with non-HTML email clients.
textbody = "This email was sent with Amazon SES using the AWS SDK for Ruby."

# Specify the text encoding scheme.
encoding = "UTF-8"

# Create a new SES resource and specify a region
ses = Aws::SES::Client.new(region: awsregion)

# Try to send the email.
begin

  # Provide the contents of the email.
  resp = ses.send_email({
    destination: {
      to_addresses: [
        recipient,
      ],
    },
  },
  message: {
    body: {
      html: {
        charset: encoding,
        data: htmlbody,
      },
      text: {
```

```
        charset: encoding,  
        data: textbody,  
      },  
    },  
    subject: {  
      charset: encoding,  
      data: subject,  
    },  
  },  
  source: sender,  
  # Comment or remove the following line if you are not using  
  # a configuration set  
  configuration_set_name: configsetname,  
  })  
  puts "Email sent!"  
  
# If something goes wrong, display an error message.  
rescue Aws::SES::Errors::ServiceError => error  
  puts "Email not sent. Error message: #{error}"  
  
end
```

- amazon-ses-sample.rb で、以下を独自の値に置き換えます。
  - sender@example.com** — Amazon SES で検証した E メールアドレスに置き換えます。詳細については、「[ID の検証 \(p. 48\)](#)」を参照してください。Amazon SES では、E メールアドレスの大文字と小文字が区別されます。検証したアドレスと完全に一致するアドレスを入力してください。
  - recipient@example.com** — 受信者のアドレスに置き換えます。アカウントがサンドボックスにまだある場合は、このアドレスを使用前に確認する必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。検証したアドレスと完全に一致するアドレスを入力してください。
  - (オプション)**us-west-2** — 米国西部 ( オレゴン ) 以外のリージョンで Amazon SES を使用する場合は、この値を使用するリージョンの名前で置き換えます。Amazon SES を使用できるリージョンのリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。
- amazon-ses-sample.rb を保存します。
- プログラムを実行するには、amazon-ses-sample.rb と同じディレクトリでコマンドプロンプトを開き、`ruby amazon-ses-sample.rb` と入力します
- 出力を確認します。E メールが正常に送信されると、コンソールに "Email sent!" が表示されます。送信に失敗すると、エラーメッセージが表示されます。
- 受信者のアドレスの E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## AWS SDK for Python (Boto) を使用して E メールを送信する

このトピックでは、[AWS SDK for Python \(Boto\)](#) を使用して Amazon SES 経由で E メールを送信する方法を示します。

### Important

このチュートリアルでは、受信を確認できるように自分宛に E メールを送信します。さらに詳しい実験や負荷テストには、Amazon SES メールボックスシミュレーターを使用してください。メールボックスシミュレーターに送信される E メールは、送信クォータに加算されず、バウンス率や苦情率の計算にも含まれません。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

### Prerequisites

開始する前に、次のタスクを実行します。

- Amazon SES で E メールアドレスを検証する - Amazon SES で E メールを送信するには、送信者の E メールアドレスを所有していることを検証する必要があります。アカウントが Amazon SES サンドボックスにまだある場合は、受信者の E メールアドレスも検証する必要があります。E メールアドレスを検証する最も簡単な方法は、Amazon SES コンソールを使用することです。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- AWS 認証情報を取得する — SDK を使用して Amazon SES にアクセスするには、AWS アクセスキー ID と AWS シークレットアクセスキーが必要です。認証情報を取得するには、AWS Management Console の「[セキュリティの認証情報](#)」のページを参照してください。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。
- Python のインストール - Python は <https://www.python.org/downloads/> からダウンロードできます。このチュートリアルは Python 2.7.6 および Python 3.6.1 でテスト済みです。Python をインストールした後、コマンドプロンプトから Python を実行できるように環境変数に Python のパスを追加します。
- AWS SDK for Python (Boto) のインストール — ダウンロードとインストールの手順については、[AWS SDK for Python \(Boto\) のドキュメント](#)を参照してください。このチュートリアルのサンプルコードは SDK for Python のバージョン 1.4.4 でテスト済みです。
- 共有認証情報ファイルの作成 — このセクションのサンプルコードが正常に機能するためには、共有認証情報ファイルを作成する必要があります。詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

## Procedure

次の手順は、SDK for Python を使用して Amazon SES 経由で E メールを送信する方法を示しています。

SDK for Python を使用して Amazon SES 経由で E メールを送信するには

1. テキストエディタで `amazon-ses-sample.py` という名前のファイルを作成します。ファイルに次のコードを貼り付けます。

```
import boto3
from botocore.exceptions import ClientError

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
SENDER = "Sender Name <sender@example.com>"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
RECIPIENT = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable, and the
# ConfigurationSetName=CONFIGURATION_SET argument below.
CONFIGURATION_SET = "ConfigSet"

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# The subject line for the email.
SUBJECT = "Amazon SES Test (SDK for Python)"

# The email body for recipients with non-HTML email clients.
BODY_TEXT = ("Amazon SES Test (Python)\r\n"
             "This email was sent with Amazon SES using the "
             "AWS SDK for Python (Boto).")

# The HTML body of the email.
BODY_HTML = ""<html>
<head></head>
```

```
<body>
  <h1>Amazon SES Test (SDK for Python)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the
    <a href='https://aws.amazon.com/sdk-for-python/'>
      AWS SDK for Python (Boto)</a>.</p>
</body>
</html>

"""

# The character encoding for the email.
CHARSET = "UTF-8"

# Create a new SES resource and specify a region.
client = boto3.client('ses', region_name=AWS_REGION)

# Try to send the email.
try:
    #Provide the contents of the email.
    response = client.send_email(
        Destination={
            'ToAddresses': [
                RECIPIENT,
            ],
        },
        Message={
            'Body': {
                'Html': {
                    'Charset': CHARSET,
                    'Data': BODY_HTML,
                },
                'Text': {
                    'Charset': CHARSET,
                    'Data': BODY_TEXT,
                },
            },
            'Subject': {
                'Charset': CHARSET,
                'Data': SUBJECT,
            },
        },
        Source=SENDER,
        # If you are not using a configuration set, comment or delete the
        # following line
        ConfigurationSetName=CONFIGURATION_SET,
    )
# Display an error if something goes wrong.
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print("Email sent! Message ID:"),
    print(response['MessageId'])
```

2. `amazon-ses-sample.py` で、以下を独自の値に置き換えます。

- **sender@example.com** — Amazon SES で検証した E メールアドレスに置き換えます。詳細については、「[ID の検証 \(p. 48\)](#)」を参照してください。Amazon SES では、E メールアドレスの大文字と小文字が区別されます。検証したアドレスと完全に一致するアドレスを入力してください。
- **recipient@example.com** — 受信者のアドレスに置き換えます。アカウントがサンドボックスにまだある場合は、このアドレスを使用前に確認する必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。検証したアドレスと完全に一致するアドレスを入力してください。
- (オプション)**us-west-2** — 米国西部 (オレゴン) 以外のリージョンで Amazon SES を使用する場合は、この値を使用するリージョンの名前で置き換えます。Amazon SES を使用できるリージョ

ンのリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。

3. `amazon-ses-sample.py` を保存します。
4. プログラムを実行するには、`amazon-ses-sample.py` と同じディレクトリでコマンドプロンプトを開き、`python amazon-ses-sample.py` と入力します
5. 出力を確認します。E メールが正常に送信されると、コンソールに "Email sent!" が表示されます。送信に失敗すると、エラーメッセージが表示されます。
6. 受信者のアドレスの E メールクライアントにサインインします。送信した E メールメッセージを確認します。

## 別の E メール送信ソリューションから Amazon SES への移行

このトピックでは、E メール送信ソリューションを、オンプレミスや Amazon EC2 インスタンスでホストされているソリューションから Amazon SES に移動する場合に必要な手順について概要を示します。

このセクションのトピック:

- [ドメインの検証](#) (p. 46)
- [本番稼働用アクセスのリクエスト](#) (p. 46)
- [ドメイン認証システムの設定](#) (p. 46)
- [SMTP 認証情報の生成](#) (p. 47)
- [SMTP エンドポイントへの接続](#) (p. 47)
- [次のステップ](#) (p. 47)

### ドメインの検証

Amazon SES を使用して E メールを送信する前に、Eメールの送信元の ID を検証する必要があります。Amazon SES の場合、ID は E メールアドレスまたはドメイン全体です。ドメインを検証すると、Amazon SES を使用してそのドメインの任意のアドレスから E メールを送信できます。ドメインの検証の詳細については、「[Amazon SES でのドメインの検証](#) (p. 61)」を参照してください。

### 本番稼働用アクセスのリクエスト

初めて Amazon SES の使用を開始した時点では、アカウントはサンドボックス環境内にあります。アカウントがサンドボックスにある間は、検証済みアドレスにのみ E メールを送信できます。さらに、1 日に送信できるメッセージの数と、1 秒あたりに送信できるメッセージの数に制限があります。本番稼働用アクセスのリクエストの詳細については、「[Amazon SES サンドボックス外への移動](#) (p. 74)」を参照してください。

### ドメイン認証システムの設定

DKIM や SPF などの認証システムを使用するようにドメインを設定できます。このステップは技術的にはオプションです。ただし、ドメインに DKIM または SPF (またはその両方) を設定することで、Eメールの配信性能を向上させ、顧客からの信頼度を高めることができます。SPF の設定の詳細については、「[Amazon SES における SPF を使った Eメールの認証](#) (p. 134)」を参照してください。DKIM の設定の詳細については、「[Amazon SES における DKIM を使った Eメールの認証](#) (p. 135)」を参照してください。

## SMTP 認証情報の生成

SMTP を使用するアプリケーションを使用して E メールを送信する場合は、SMTP 認証情報を生成する必要があります。SMTP 認証情報は、通常の AWS 認証情報とは異なります。また、これらの認証情報は AWS リージョンごとに一意です。SMTP 認証情報を生成する方法の詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

## SMTP エンドポイントへの接続

postfix や sendmail などのメッセージ転送エージェントを使用する場合は、アプリケーションが Amazon SES SMTP エンドポイントを参照するように設定を更新する必要があります。SMTP エンドポイントの詳細なリストについては、「[Amazon SES SMTP エンドポイントへの接続 \(p. 94\)](#)」を参照してください。前のステップで作成した SMTP 認証情報は、特定の AWS リージョンに関連付けられていることに注意してください。SMTP 認証情報を作成したリージョンの SMTP エンドポイントに接続する必要があります。

## 次のステップ

この時点で、Amazon SES を使用して Eメールの送信を開始する準備が整いました。ただし、オプションとして実行できるステップがいくつかあります。

- 設定セットを作成できます。これは、送信する Eメールに適用されるルールのセットです。たとえば、設定セットを使用して、Eメールを配信したときや、受取人がメッセージを開いたりメッセージ内のリンクをクリックしたりしたとき、Eメールがバウンスしたとき、受取人が Eメールをスパムとしてマークしたときに、通知を送信する先を指定できます。詳細については、「[Amazon SES 設定セットの使用 \(p. 257\)](#)」を参照してください。
- Amazon SES 経由で Eメールを送信するときは、アカウントのバウンスや苦情をモニタリングすることが重要です。Amazon SES には、アカウントのバウンスや苦情を追跡するために使用できる評価ダッシュボードが含まれています。詳細については、「[評価ダッシュボードを使用してバウンス率と苦情率を追跡する \(p. 377\)](#)」を参照してください。また、これらのレートが高くなりすぎた場合に警告する CloudWatch アラームを作成することもできます。CloudWatch アラームの作成の詳細については、「[CloudWatch を使用して評価モニタリングアラームを作成する \(p. 390\)](#)」を参照してください。
- 大量の Eメールを送信するユーザーや、自分の IP アドレスの評価を完全に制御したいユーザーは、追加の月額料金を支払って専用 IP アドレスをリースできます。詳細については、「[Amazon SES での専用 IP アドレスの使用 \(p. 178\)](#)」を参照してください。

# Amazon SES での Eメールのセットアップ

Amazon Simple Email Service (Amazon SES) で Eメールを設定するには、以下のタスクを実行する必要があります。

- Amazon SES またはその他の AWS サービスにアクセスする前に、AWS アカウントをセットアップする必要があります。詳細については、「[AWS にサインアップする \(p. 48\)](#)」を参照してください。
- Amazon SES を通じて Eメールを送信する前に、「From」アドレスを所有していることを確認する必要があります。アカウントが Amazon SES サンドボックスにまだある場合は、「To」アドレスも確認する必要があります。Eメールアドレスまたはドメイン全体を検証できます。詳細については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。

以下のタスクは、行う操作によってはオプションです。

- Amazon SES API を通じて Amazon SES にアクセスする場合は、クエリ (HTTPS) インターフェイスからアクセスするか、[AWS SDK](#)、[AWS Command Line Interface](#)、または [AWS Tools for Windows PowerShell](#) を通じて間接的にアクセスするかにかかわらず、AWS アクセスキーを取得する必要があります。詳細については、「[AWS アクセスキーの取得 \(p. 67\)](#)」を参照してください。

- クエリインターフェイスの低レベルの詳細を処理せずに Amazon SES API を呼び出したい場合は、AWS SDK を使用できます。詳細については、「[AWS SDK のダウンロード \(p. 67\)](#)」を参照してください。
- SMTP インターフェイスを通じて Amazon SES にアクセスする場合は、SMTP ユーザー名とパスワードを取得する必要があります。SMTP 認証情報は、AWS の認証情報とは異なります。詳細については、「[Amazon SES の SMTP 認証情報の取得 \(p. 73\)](#)」を参照してください。
- 初めて Amazon SES にサインアップすると、アカウントは Amazon SES サンドボックス環境にあります。サンドボックスでは、他の Amazon SES ユーザーと同じメール送信手段を使用して E メールを送信することができますが、最大で 1 秒間に 1 通の E メール、24 時間に 200 通までの E メール送信に制限され、また送信する E メールアドレスは確認済みのものに限られます。送信クォータを引き上げ、検証されていない E メールアドレスに E メールを送信するには、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。
- E メールが Sender Policy Framework (SPF) に基づいて DMARC (Domain-based Message Authentication, Reporting and Conformance) 認証に合格するには、「[カスタムの MAIL FROM ドメインの設定 \(p. 68\)](#)」の説明に従って、カスタムの MAIL FROM ドメインから送信できるようにアイデンティティを設定します。

## AWS にサインアップする

Amazon SES やその他の AWS のサービスを使用する前に、AWS アカウントを作成する必要があります。

AWS アカウントを作成するには

1. ウェブブラウザで、「<https://aws.amazon.com/ses>」にアクセスします。
2. [AWS アカウントの作成] を選択します。
3. 画面上の指示に従ってください。

## 次のステップ

AWS アカウントを作成すると、Amazon SES の設定を開始できます。

- Amazon SES で Eメールの送信を開始するには、まず [ID を検証 \(p. 48\)](#) する必要があります。ID とは、Eメールの送信元の Eメールアドレスまたはドメインです。
- Amazon SES を操作するには、アカウントの [IAM 認証情報を取得 \(p. 67\)](#) する必要があります。
- 初めて Amazon SES の使用を開始した時点では、アカウントは Amazon SES サンドボックス内にあります。サンドボックスでは、Amazon SES API および SMTP インターフェイスへのフルアクセスが可能です。ただし、次の制限が有効です。
  - Eメールを送信できる先は、[Amazon SES メールボックスシミュレーター \(p. 187\)](#)と[検証済み Eメール ID \(p. 48\)](#)のみです。
  - 最大で 24 時間あたり 200 メッセージを送信できます。
  - 最大で 1 秒あたり 1 メッセージを送信できます。

サンドボックスの外への移動については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

## Amazon SES の ID の検証

Amazon SES では、ID はユーザーが Eメールの送信に使用する Eメールアドレスまたはドメインです。Amazon SES を使用して Eメールを送信する前に、「From」、「Source」、「Sender」、または「Return-Path」のアドレスとして使用する各 ID を検証し、それらを所有していることを証明する必要があります。アカウントがまだ Amazon SES サンドボックス内にある場合は、[Amazon SES メールボック](#)



[スシミュレーター \(p. 187\)](#)から提供された E メールアドレスを除いて、すべての送信先の E メールアドレスも検証する必要があります。

Amazon SES コンソールまたは Amazon SES API を使用して、ID を検証することができます。

## Amazon SES E メールアドレスの確認

Amazon SES では、お客様の ID (メールを送信するドメインまたは E メールアドレス) を検証して、お客様がそれらを所有していることを確認し、不正使用を防止する必要があります。このセクションでは、E メールアドレス ID の検証について説明します。ドメイン ID の検証については、「[the section called “ドメインの検証” \(p. 61\)](#)」を参照してください。

Amazon SES で使用する E メールアドレスを確認するときは、次の要素を考慮してください。

- 「From」、「Source」、「Sender」、または「Return-Path」アドレスとして使用する各 ID を確認する必要があります。ただし、追加の検証ステップを実行せずに、既に検証済みの E メールアドレスにラベルを追加することができます (このリストの後半の情報を参照)。
- E メールアドレスでは、大文字と小文字が区別されます。たとえば、sender@EXAMPLE.com を検証した場合、sender@example.com も検証しないと、sender@example.com から E メールを送信できません。
- E メールアドレスとそのアドレスが属するドメインの両方を検証した場合は、E メールアドレスの設定によってドメインの設定が上書きされます。たとえば、ドメイン example.com でドメインキーアイデンティファイアードメール (DKIM) が有効になっているが、sender@example.com では有効になっていない場合、sender@example.com は DKIM で署名されません。
- Amazon SES では複数の AWS リージョンにエンドポイントがあり、E メールアドレスの検証ステータスはリージョンごとに個別に扱われます。複数のリージョンで同じ ID から E メールを送信する場合は、各リージョンの ID を検証する必要があります。複数のリージョンで Amazon SES を使用方法については、「[リージョン \(p. 467\)](#)」を参照してください。
- AWS リージョンごとに、10,000 個までの確認済み ID (ドメインと E メールアドレスの任意の組み合わせ) を含めることができます。
- 追加の検証手順を実行せずに、検証済みの E メールアドレスにラベルを追加することができます。E メールアドレスにラベルを追加するには、アカウント名と「at」記号 (@) の間にプラス記号 (+) を付け、続いてテキストラベルを付けます。たとえば、sender@example.com を既に検証している場合は、メールの「From」または「Return-Path」アドレスとして sender+myLabel@example.com を使用することができます。この機能を使用すると、可変エンベロープリターンパス (VERP) を実装できます。その後で、VERP を使用して、メーリングリストから配信不能な E メールアドレスを検出して削除することができます。
- 検証しようとしている E メールアドレスに送信されるメッセージをカスタマイズすることができます。詳細については、「[カスタム検証 E メールテンプレートの使用 \(p. 53\)](#)」を参照してください。

## E メールアドレスの検証

E メールアドレスを検証するには、Amazon SES コンソール、[VerifyEmailIdentity](#) Amazon SES API v1 オペレーション、または [CreateEmailIdentity](#) Amazon SES API v2 オペレーションを使用します。機能を強化するには、Amazon SES API v2 を使用することをお勧めします。

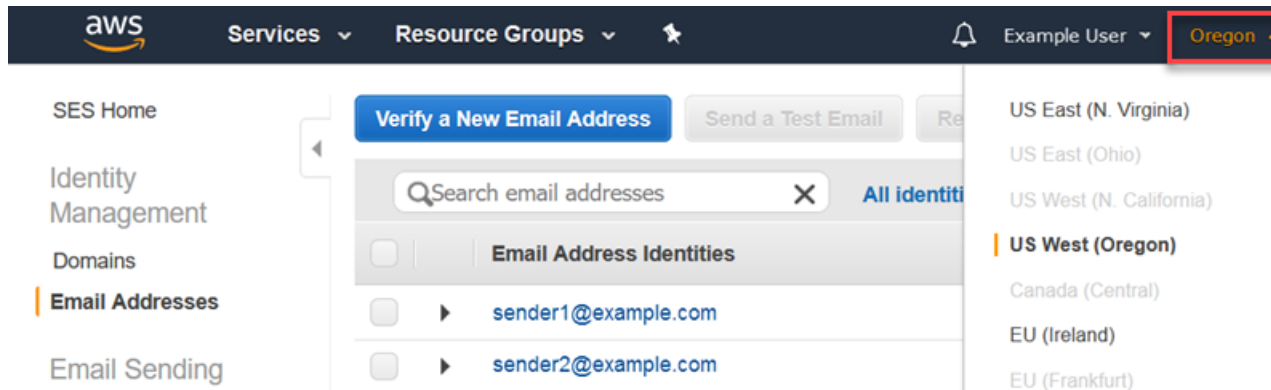
### Amazon SES コンソールを使用して E メールアドレスを検証する

Amazon SES コンソールを使用して E メールアドレスを検証するには、このセクションの手順を実行します。

Amazon SES コンソールを使用して E メールアドレスを検証するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。

2. コンソールでリージョンセクターを使用して、次のイメージに示すように、E メールアドレスを検証する AWS リージョンを選択します。



#### Note

複数のリージョンで使用する E メールアドレスを検証するには、このセクションの手順をリージョンごとに繰り返します。

3. ナビゲーションペインの [Identity Management] で [Email Addresses] を選択します。
4. [Verify a New Email Address] を選択します。
5. [Verify a New Email Address] ダイアログボックスの [Email Address] フィールドに E メールアドレスを入力し、[Verify This Email Address] を選択します。
6. 受信箱で、検証するメールアドレスをチェックします。「Amazon Web Services - Email Address Verification Request in region RegionName」という件名のメッセージが表示されます。ここで、RegionName は、ステップ 2 で選択した AWS リージョンの名前です。

メッセージ内のリンクをクリックします。

#### Note

検証メッセージ内のリンクは、メッセージが送信されてから 24 時間で期限切れになります。確認メールを受信してから 24 時間が経過した場合は、手順 1～5 を繰り返して、有効なリンクが設定された確認メールを受信してください。

7. Amazon SES コンソールの [Identity Management] で [Email Addresses] を選択します。E メールアドレスのリストで、検証する E メールアドレスを探します。E メールアドレスが検証された場合は、[Status] 列の値が [verified] になります。

### Amazon SES API を使用して E メールアドレスを検証するには

[VerifyEmailIdentity](#) Amazon SES API v1 オペレーション、または [CreateEmailIdentity](#) Amazon SES API v2 オペレーションを使用して、新しい E メール ID を作成します。個々の ID に対してこのオペレーションを呼び出す簡単な方法は、AWS CLI を使用することです。

E メール ID の検証中にデフォルトの設定セットを設定するには、「[Amazon SES デフォルト設定セットの管理 \(p. 260\)](#)」を参照してください。

#### Note

このセクションの手順を完了する前に、まず AWS CLI をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

### AWS CLI を使用して E メールアドレスを確認するには

1. コマンドラインで、[VerifyEmailIdentity](#) Amazon SES API v1 オペレーションを使用して 以下のコマンドを入力してください。

```
aws ses verify-email-identity --email-address sender@example.com
```

前述のコマンドの `sender@example.com` を、検証する E メールアドレスに置き換えます。

コマンドラインで、[CreateEmailIdentity](#) Amazon SES API v2 オペレーションを使用して以下のコマンドを入力してください。

```
aws sesv2 create-email-identity --email-identity sender@example.com
```

前述のコマンドの `sender@example.com` を、検証する E メールアドレスに置き換えます。

コマンドが正常に実行された場合、何の出力もなく終了します。

- 受信箱で、検証するメールアドレスをチェックします。「Amazon Web Services - Email Address Verification Request in region `RegionName`」という件名のメッセージが表示されます。ここで、`RegionName` は、E メールアドレスを検証しようとした AWS リージョンの名前です。

メッセージを開き、メッセージ内のリンクをクリックします。

#### Note

検証メッセージ内のリンクは、メッセージが送信されてから 24 時間で期限切れになります。確認メールを受信してから 24 時間が経過した場合は、手順 1〜5 を繰り返して、有効なリンクが設定された確認メールを受信してください。

- Amazon SES コンソールの [Identity Management] で [Email Addresses] を選択します。E メールアドレスのリストで、検証する E メールアドレスを探します。E メールアドレスが検証された場合は、[Status] 列の値が [verified] になります。

1 回のオペレーションで複数の E メール ID を検証するために使用できるスクリプトについては、「[the section called “複数の E メールアドレスの検証” \(p. 459\)](#)」を参照してください。

## E メールアドレス検証のトラブルシューティング

メールアドレスを確認しようとしても、AWS から確認メールが届かなかった場合は、次のトラブルシューティング手順を試してください。

- E メールクライアントの迷惑メールフォルダを確認します。
- E メールクライアントが、受信箱以外のフォルダに特定のメッセージを自動的に移動するルールを適用していないことを確認します。
- E メールクライアントで、アドレス帳または差出人のホワイトリストに `no-reply-aws@amazon.com` を追加します。システム管理者に、`no-reply-aws@amazon.com` からの受信メールをホワイトリストに加えるようリクエストすることもできます。
- 別の E メールサービスプロバイダ (個人の E メールアドレスなど) を使用する E メールアドレスで、検証するアドレスにメッセージを送信します。検証するアドレスにメッセージが受信されていることを確認します。このステップは、最近自分のドメインを設定した場合に特に重要です。場合によっては、新しいドメインが受信メールを正しく受信するように設定されていないことがあります。

または、個人の E メールアドレスなど、E メールを受信できることがわかっている E メールアドレスを確認してみてください。個人アドレスで確認メールを受け取った場合は、他のドメインに問題がある可能性があります。

これらのテストで、確認しようとしたアドレスで E メールが受信されないことが示された場合は、システム管理者または E メールサービスプロバイダに相談してください。

## Amazon SES での E メール ID の一覧表示

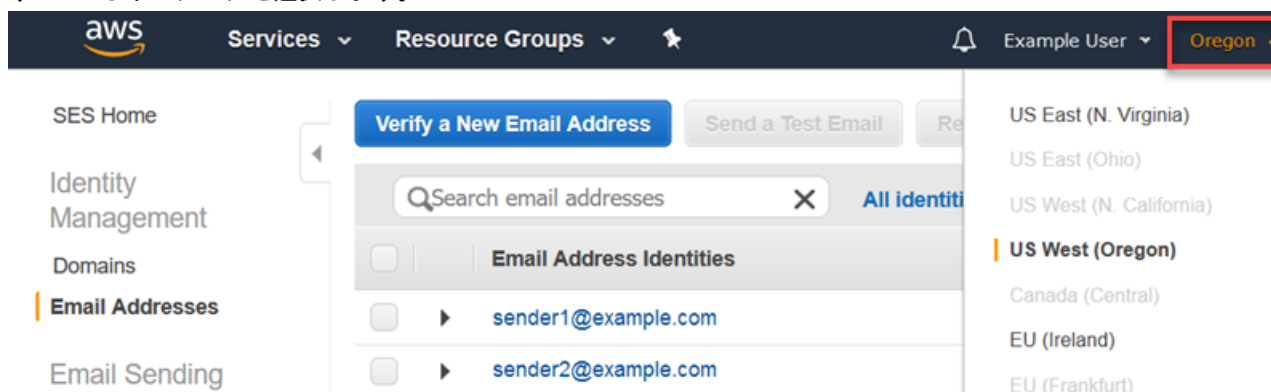
Amazon SES コンソールまたは [ListIdentities](#) API を使用して、E メール ID のリストを表示できます。

### Amazon SES での E メール ID の一覧表示

Amazon SES コンソールおよび API を使用して、検証済みまたは検証待ちの E メールアドレス、および検証プロセスに失敗した E メールアドレスのリストを表示できます。

検証済み E メールアドレスの一覧を表示するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. コンソールでリージョンセクターを使用して、次のイメージに示すように、E メール ID を一覧表示する AWS リージョンを選択します。



#### Note

これらの手順では、選択した AWS リージョンの E メールアドレスのリストのみが表示されます。

3. ナビゲーションペインの [Identity Management] で [Email Addresses] を選択します。

[Email Addresses] ページには、検証済み、検証待ち、検証プロセスに失敗した E メールアドレスのリストが表示されます。E メールアドレスをクリックすると、追加情報が表示されます。

### Amazon SES API を使用して E メール ID を一覧表示する

[ListIdentities](#) API オペレーションを使用して、そのステータスに関係なく、すべての E メール ID のリストを表示します。また、[GetIdentityVerificationAttributes](#) オペレーションを使用して、特定の ID の検証ステータスを確認することもできます。

AWS CLI を使用して ID のリストを表示するには、コマンドラインで `aws ses list-identities` コマンドを入力します。

`ListIdentities` オペレーションを実行すると、Amazon SES アカウントにあるすべての ID のリストをその検証ステータスに関係なく返します。1 つまたは複数の ID の検証ステータスを表示するには、`GetIdentityVerificationAttributes` オペレーションを使用します。AWS CLI を使用してアイデンティティの検証ステータスを確認するには、コマンドラインで次のコマンドを入力します。aws ses get-identity-verification-attributes --identities "**sender@example.com**"

前述のコマンドの `sender@example.com` を、検証ステータスを確認したい ID に置き換えます。このコマンドを使用して、単一の API 呼び出しで複数の ID の検証ステータスを確認することもできます。たとえば、ドメイン `example.com` と E メールアドレス `sender@example.co.uk` のステータスを検索するには、

コマンド `aws ses get-identity-verification-attributes --identities "example.com" "sender@example.co.uk"` を入力します。

## Amazon SES で E メール ID を削除する

検証済みの E メールアドレスを使用する必要がなくなった場合は、Amazon SES コンソールまたは `DeleteIdentity` API オペレーションを使用して削除できます。

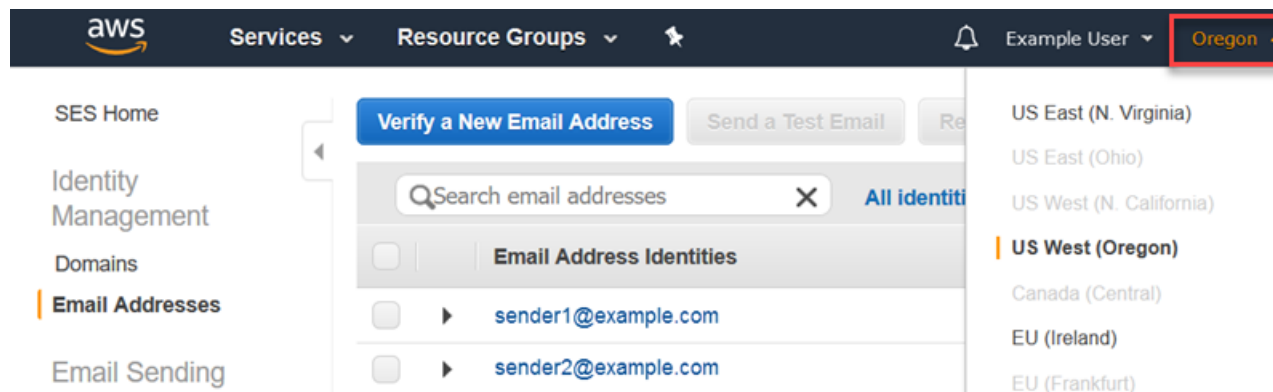
### Warning

この操作は元に戻すことができません。ただし、以前に削除された ID の検証プロセスを繰り返すことができます。

## Amazon SES で E メール ID を削除する

検証済み E メールアドレスを削除するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. コンソールでリージョンセレクターを使用して、次のイメージに示すように、E メール ID を削除する AWS リージョンを選択します。



### Note

これらの手順では、選択した AWS リージョンの E メールアドレスのみが削除されます。複数のリージョンで検証された E メールアドレスを削除するには、このセクションの手順をリージョンごとに繰り返します。

3. 削除する各 E メールアドレスを選択し、[Remove] を選択します。

## Amazon SES API を使用して E メール ID を削除する

`DeleteIdentity` API オペレーションを使用して E メールアドレスおよびドメインの ID を削除します。

AWS CLI を使用して E メールアドレスを削除するには、コマンドラインで次のコマンドを入力します。`aws ses delete-identity --identity "sender@example.com"`

前述のコマンドの `sender@example.com` を、削除する ID に置き換えます。

## カスタム検証 E メールテンプレートの使用

E メールアドレスを確認しようとする、Amazon SES はそのアドレスに次の図に示す例のような E メールを送信します。

Dear Amazon Web Services Customer,

We have received a request to authorize this email address for use with Amazon SES and Amazon Pinpoint in region US West (Oregon). If you requested this verification, please go to the following URL to confirm that you are authorized to use this email address:

<https://email-verification.us-west-2.amazonaws.com/?AWSAccessKeyId=AKIADQKF4EXAMPLE&Context=10987654321&Identity.IdentityName=recipient%40example.com&Identity.IdentityType=EmailAddress&Namespace=Bacon&Operation=ConfirmVerification&Signature=TjDufFhYYK1fSHCSBq4cjbodBQq%2FnyyZgziqZ%2BXsDYEXAMPLE&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2017-12-06T19%3A53%3A12.31Z>

Your request will not be processed unless you confirm the address using this URL. This link expires 24 hours after your original verification request.

If you did NOT request to verify this email address, do not click on the link. Please note that many times, the situation isn't a phishing attempt, but either a misunderstanding of how to use our service, or someone setting up email-sending capabilities on your behalf as part of a legitimate service, but without having fully communicated the procedure first. If you are still concerned, please forward this notification to [aws-email-domain-verification@amazon.com](mailto:aws-email-domain-verification@amazon.com) and let us know in the forward that you did not request the verification.

To learn more about sending email from Amazon Web Services, please refer to the Amazon SES Developer Guide at <http://docs.aws.amazon.com/SES/latest/DeveloperGuide/Welcome.html> and Amazon Pinpoint Developer Guide at <http://docs.aws.amazon.com/pinpoint/latest/userguide/welcome.html>.

Sincerely,  
The Amazon Web Services Team.

Amazon SES のお客様の中には、Amazon SES を通じて E メールを送信するアプリケーション (E メールマーケティングスイートや発券システムなど) を構築しています。これらのアプリケーションのエンドユーザーにとって、Eメールの検証プロセスはわかりにくい場合があります。確認 Eメールはアプリケーションのブランド化ではなく、Amazon SES ブランドを使用しています。エンドユーザーが直接 Amazon SES を使用するためにサインアップすることはありません。

Amazon SES のユースケースでは、Amazon SES で使用するために E メールアドレスを検証する必要がある場合、カスタマイズされた確認 Eメールを作成できます。これらのカスタマイズされた Eメールは、顧客の混乱を減らし、顧客が登録手続きを完了する率を上げるのに役立ちます。

#### Note

この機能を使用するには、Amazon SES アカウントがサンドボックス外にあることが必要です。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

このセクションのトピック:

- [カスタム検証 Eメールテンプレートの作成 \(p. 54\)](#)
- [カスタム検証 Eメールテンプレートの編集 \(p. 56\)](#)
- [カスタムテンプレートを使用した検証 Eメールの送信 \(p. 56\)](#)
- [カスタム検証 Eメールに関するよくある質問 \(p. 57\)](#)

## カスタム検証 Eメールテンプレートの作成

カスタム確認 Eメールを作成するには、CreateCustomVerificationEmailTemplate API オペレーションを使用します。このオペレーションでは、次の入力を使用します。

属性	説明
TemplateName	テンプレートの名前。指定する名前は一意である必要があります。
FromEmailAddress	確認 Eメールが送信された Eメールアドレス。指定したアドレスまたはドメインは、Amazon SES アカウントで使用するために検証する必要があります。  Note  FromEmailAddress 属性は表示名 (「差出人」名とも呼ばれます) をサポートしていません。
TemplateSubject	確認 Eメールの件名。

属性	説明
TemplateContent	Eメールの本文。Eメール本文には、一定の制限がある HTML を含めることができます。詳細については、「 <a href="#">カスタム検証 Eメールに関するよくある質問 (p. 57)</a> 」を参照してください。
SuccessRedirectionURL	ユーザーの Eメールアドレスが正常に確認された場合に送信される URL。
FailureRedirectionURL	ユーザーの Eメールアドレスが正常に確認されなかった場合に送信される URL。

AWS SDK または AWS CLI を使用して、CreateCustomVerificationEmailTemplate オペレーションでカスタム検証 Eメールテンプレートを作成できます。AWS SDK についての詳細は、「[Amazon Web Services のツール](#)」を参照してください。AWS CLI についての詳細は、「[AWS コマンドラインインターフェイス](#)」を参照してください。

次のセクションでは、AWS CLI を使用してカスタム確認 Eメールを作成する手順について説明します。これらの手順は、AWS CLI がインストールされ、設定されていることを前提としています。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

#### Note

このセクションの手順を完了するには、AWS CLI のバージョン 1.14.6 以降を使用する必要があります。最良の結果を得るには、AWS CLI の最新バージョンにアップグレードします。AWS CLI の更新の詳細については、ユーザーガイドの「[AWS Command Line Interface のインストール](#)」AWS Command Line Interface を参照してください。

1. テキストエディタで新規ファイルを作成します。エディタに、以下の内容を貼り付けます。

```
{
  "TemplateName": "SampleTemplate",
  "FromEmailAddress": "sender@example.com",
  "TemplateSubject": "Please confirm your email address",
  "TemplateContent": "<html>
    <head></head>
    <body style='font-family:sans-serif;'>
      <h1 style='text-align:center;'>Ready to start sending
        email with ProductName?</h1>
      <p>We here at Example Corp are happy to have you on
        board! There's just one last step to complete before
        you can start sending email. Just click the following
        link to verify your email address. Once we confirm that
        you're really you, we'll give you some additional
        information to help you get started with ProductName.</p>
    </body>
  </html>",
  "SuccessRedirectionURL": "https://www.example.com/verifysuccess",
  "FailureRedirectionURL": "https://www.example.com/verifyfailure"
}
```

#### Important

上記の例を読みやすくするために、TemplateContent 属性に改行が含まれています。前述の例をテキストファイルに貼り付ける場合は、続行する前に改行を削除してください。

TemplateName、FromEmailAddress、TemplateSubject、TemplateContent、SuccessRedirectionURL および FailureRedirectionURL の値を独自の値に置き換えます。

## Note

FromEmailAddress パラメータで指定する E メールアドレス は、検証される必要があるか、検証済みドメインのアドレスである必要があります。詳細については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。

終了したら、customverificationemail.json としてファイルを保存します。

2. コマンドラインで次のコマンドを入力して、カスタム検証 E メールテンプレートを作成します。

```
aws ses create-custom-verification-email-template --cli-input-json file://
customverificationemail.json
```

3. 必要に応じて、次のコマンドを入力して、テンプレートが作成されたことを確認できます。

```
aws ses list-custom-verification-email-templates
```

## カスタム検証 E メールテンプレートの編集

UpdateCustomVerificationEmailTemplate オペレーションを使用してカスタム確認 E メールテンプレートを編集できます。このオペレーションでは、CreateCustomVerificationEmailTemplate オペレーション (つまり TemplateName、FromEmailAddress、TemplateSubject、TemplateContent、SuccessRedirectionURL、および FailureRedirectionURL 属性) と同じ入力を受け入れます。ただし、UpdateCustomVerificationEmailTemplate オペレーションでは、これらの属性は必要ではありません。既存のカスタム確認 E メールテンプレートの名前と同じ TemplateName 値を渡すと、指定した属性がテンプレートに元々含まれていた属性を上書きします。

## カスタムテンプレートを使用した検証 Eメールの送信

少なくとも 1 つのカスタム確認 E メールテンプレートを作成したら、SendCustomVerificationEmail API オペレーションを呼び出して顧客に送信することができます。AWS SDK または SendCustomVerificationEmail のいずれかを使用して、AWS CLI オペレーションを呼び出すことができます。SendCustomVerificationEmail オペレーションでは、次の入力を使用します。

属性	説明
EmailAddress	確認されている E メールアドレス。
TemplateName	確認されている E メールアドレスに送信されるカスタム確認 E メールテンプレートの名前。
ConfigurationSetName	(オプション) 確認 E メールを送信するときに使用する設定セットの名前。

たとえば、顧客がアプリケーションのフォームを使用してサービスに登録するとします。顧客がフォームを完成させて送信すると、アプリケーションは SendCustomVerificationEmail オペレーションを呼び出し、顧客の E メールアドレスと使用するテンプレートの名前を渡します。

顧客は、作成したカスタマイズされた E メールテンプレートを使用する E メールを受信します。Amazon SES は、受信者へのユニークなリンクと簡単な免責事項を自動的に追加します。次の図は、[カスタム検証 E メールテンプレートの作成 \(p. 54\)](#) で作成されたテンプレートを使用する確認 Eメールのサンプルを示しています。



### Ready to start sending email with ProductName?

We here at Example Corp are happy to have you on board! There's just one last step to complete before you can start sending email. Just click the following link to verify your email address. Once we confirm that you're really you, we'll give you some additional information to help you get started with ProductName.

<https://email-verification.us-west-2.amazonaws.com/?AWSAccessKeyId=AKIADQKE4EXAMPLE&Context=10987654321&Identity.IdentityName=recipient%40example.com&Identity.IdentityType=EmailAddress&Namespace=Bacon&Operation=ConfirmVerification&Signature=TJDufFhYYK1fSHCSBq4cjbodBQq%2EnyyZgziqz%2BXsDYEXAMPLE&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2017-12-06T19%3A53%3A12.311Z>

If you did not request to verify this email address, please disregard this message. If you have any concerns, please forward this message to the following [email address](#) along with your questions or concerns.

## カスタム検証 E メールに関するよくある質問

このセクションでは、カスタム確認 E メールテンプレート機能に関するよくある質問に対する回答を示します。

### Q1. 作成できるカスタム確認 E メールテンプレートの数はいくつですか？

Amazon SES アカウントごとに最大 50 のカスタム確認 E メールテンプレートを作成できます。

### Q2. カスタム確認 E メールはどのように受信者に表示されますか？

カスタム確認 E メールには、テンプレートを作成したときに指定したコンテンツと、メールアドレスを確認するために受信者がクリックする必要があるリンクが含まれます。

### Q3. カスタム確認 E メールをプレビューすることはできますか？

カスタム確認 E メールをプレビューするには、SendCustomVerificationEmail オペレーションを使用して確認 E メールを所有しているアドレスに送信します。確認リンクをクリックしない場合は、Amazon SES は新しい ID を作成しません。確認リンクをクリックすると、DeleteIdentity オペレーションを使用して、新しく作成された ID を削除することもできます。

### Q4. カスタム確認 E メールテンプレートに画像を含めることはできますか？

Base64 エンコーディングを使用して、テンプレートの HTML に画像を埋め込むことができます。この方法で画像を埋め込むと、Amazon SES は画像を添付ファイルに自動的に変換します。次のコマンドのいずれかを発行して、画像をコマンドラインでエンコードすることができます。

Linux, macOS, or Unix

```
base64 -i imagefile.png | tr -d '\n' > output.txt
```

Windows

```
certutil -encodehex -f imagefile.png output.txt 0x40000001
```

*imagefile.png* を、エンコードするファイルの名前に置き換えます。上記の両方のコマンドで、Base64 でエンコードされた画像は *output.txt* に保存されます。

テンプレートの HTML に以下を含めることで、Base64 でエンコードされた画像を埋め込むことができます。

上記の例では、*png* をエンコードされた画像のファイルタイプ (jpg や gif など) に置き換え、*base64EncodedImage* を Base64 でエンコードされた画像 (つまり、前述のコマンドの *output.txt* の内容) に置き換えます。

### Q5. カスタム検証 E メールテンプレートに含めることのできるコンテンツに制限はありますか？

カスタム確認 E メールテンプレートのサイズは 10 MB を超えることはできません。また、HTML を含むカスタム検証 E メールテンプレートでは、以下の表に示しているタグと属性しか使用できません。

HTML タグ	許可された属性
abbr	class, id, style, title
acronym	class, id, style, title
address	class, id, style, title
area	class, id, style, title
b	class, id, style, title
bdo	class, id, style, title
big	class, id, style, title
blockquote	cite, class, id, style, title
body	class, id, style, title
br	class, id, style, title
button	class, id, style, title
caption	class, id, style, title
center	class, id, style, title
cite	class, id, style, title
code	class, id, style, title
col	class, id, span, style, title, width
colgroup	class, id, span, style, title, width
dd	class, id, style, title
del	class, id, style, title
dfn	class, id, style, title
dir	class, id, style, title
div	class, id, style, title
dl	class, id, style, title
dt	class, id, style, title
em	class, id, style, title
fieldset	class, id, style, title
font	class, id, style, title
form	class, id, style, title
h1	class, id, style, title
h2	class, id, style, title
h3	class, id, style, title

HTML タグ	許可された属性
h4	class, id, style, title
h5	class, id, style, title
h6	class, id, style, title
head	class, id, style, title
hr	class, id, style, title
html	class, id, style, title
i	class, id, style, title
img	align, alt, class, height, id, src, style, title, width
input	class, id, style, title
ins	class, id, style, title
kbd	class, id, style, title
label	class, id, style, title
legend	class, id, style, title
li	class, id, style, title
map	class, id, style, title
menu	class, id, style, title
ol	class, id, start, style, title, type
optgroup	class, id, style, title
option	class, id, style, title
p	class, id, style, title
pre	class, id, style, title
q	cite, class, id, style, title
s	class, id, style, title
samp	class, id, style, title
select	class, id, style, title
small	class, id, style, title
span	class, id, style, title
strike	class, id, style, title
strong	class, id, style, title
sub	class, id, style, title

HTML タグ	許可された属性
sup	class, id, style, title
table	class, id, style, summary, title, width
tbody	class, id, style, title
td	abbr, axis, class, colspan, id, rowspan, style, title, width
textarea	class, id, style, title
tfoot	class, id, style, title
th	abbr, axis, class, colspan, id, rowspan, scope, style, title, width
thead	class, id, style, title
tr	class, id, style, title
tt	class, id, style, title
u	class, id, style, title
ul	class, id, style, title, type
var	class, id, style, title

#### Note

カスタム検証 E メールテンプレートには、コメントタグを含めることはできません。

#### Q6. 確認済み E メールアドレスをアカウントにいくつ持つことができますか？

Amazon SES アカウントには、AWS リージョンごとに 10,000 個までの確認済み ID を含めることができます。Amazon SES で、ID には検証済みドメインと E メールアドレスの両方が含まれます。

#### Q7. Amazon SES コンソールを使用してカスタム確認 E メールテンプレートを作成できますか？

現在のところ、Amazon SES API を使用してカスタム確認 E メールを作成、編集、削除できます。

#### Q8. 顧客がカスタム確認 E メールを受け取ったときに発生するオープンイベントとクリックイベントを追跡できますか？

カスタム確認 E メールには、オープンまたはクリックの追跡を含めることはできません。

#### Q9. カスタム確認 E メールにカスタムヘッダーを含めることはできますか？

カスタム確認 E メールにカスタムヘッダーを含めることはできません。

#### Q10. カスタム確認 Eメールの下部に表示されるテキストを削除できますか？

次のテキストは、すべてのカスタム確認 Eメールの末尾に自動的に追加され、削除することはできません。

この Eメールアドレスの確認をリクエストしていない場合は、このメッセージを無視してください。不明な点がございましたら、ご質問やご意見とともに、このメールを次の Eメールアドレスに転送してください。

このテキストの E メールアドレスのリンクは [aws-email-domain-verification@amazon.com](mailto:aws-email-domain-verification@amazon.com) を指します。これは、Amazon SES チームが積極的にモニタリングしている受信トレイです。

#### Q11. カスタム確認 E メールは DKIM 署名されていますか？

確認 E メールを DKIM 署名するには、確認メールテンプレートを作成するときに `FromEmailAddress` 属性で指定した E メールアドレスが DKIM 署名を生成するように設定する必要があります。E メールアドレスの DKIM の設定の詳細については、[the section called “DKIM での Eメールの認証” \(p. 135\)](#) を参照してください。

#### Q12. カスタム検証 E メールテンプレートの API オペレーションが SDK または CLI がないのはなぜですか？

SDK または AWS CLI でカスタム検証 E メールテンプレートのオペレーションを使用できない場合は、古いバージョンの SDK または CLI を使用している可能性があります。カスタム検証 E メールテンプレートのオペレーションは以下の SDK および CLI で使用できます。

- AWS Command Line Interface のバージョン 1.14.6 以降
- AWS SDK for .NET のバージョン 3.3.205.0 以降
- AWS SDK for C++ のバージョン 1.3.20170531.19 以降
- AWS SDK for Go のバージョン 1.12.43 以降
- AWS SDK for Java のバージョン 1.11.245 以降
- AWS SDK for JavaScript のバージョン 2.166.0 以降
- AWS SDK for PHP のバージョン 3.45.2 以降
- AWS SDK for Python (Boto) のバージョン 1.5.1 以降
- `aws-sdk-ses` の AWS SDK for Ruby Gem のバージョン 1.5.0 以降

#### Q13. カスタム確認 E メールを送信するときに `ProductionAccessNotGranted` エラーが発生するのはなぜですか？

`ProductionAccessNotGranted` エラーは、アカウントが Amazon SES サンドボックスにまだ存在することを示しています。カスタム確認 E メールは、アカウントがサンドボックスから削除されている場合にのみ送信できます。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

## Amazon SES でのドメインの検証

Amazon SES では、E メールアドレスまたはドメインを検証して、それを所有していることを確認し、他のユーザーに使用されないようにする必要があります。ドメイン全体を検証すると、そのドメインのすべての E メールアドレスを検証することになるため、そのドメインの E メールアドレスを個別に検証する必要はありません。たとえば、ドメイン `example.com` を検証すると、`user1@example.com`、`user2@example.com`、または `example.com` の他の任意のユーザーから E メールを送信できます。

Amazon SES コンソールまたは Amazon SES API を使用して、検証済みドメインを管理できます。ドメインの検証に関連する API アクションの詳細説明については、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。このセクションでは、以下のトピックを取り上げて、Amazon SES コンソールを使用した操作について説明します。

- [Amazon SES でのドメインの検証 \(p. 62\)](#)
- [Amazon SES でのドメインの一覧表示 \(p. 64\)](#)
- [Amazon SES でドメイン ID を削除する \(p. 64\)](#)
- [Amazon SES ドメイン検証の失効 \(p. 65\)](#)
- [Amazon SES ドメイン検証 TXT レコード \(p. 65\)](#)

ドメインの検証については、以下のような点が重要です。

- Amazon SES は複数の AWS リージョンにエンドポイントを持っており、ドメインの検証は各 AWS リージョンに個別に適用されます。特定のドメインからの送信を行うリージョンごとに、ドメイン全体の検証手順を実行する必要があります。複数のリージョンにある同じドメインを検証する場合に、同じ名前の複数の TXT レコードを持つことを DNS プロバイダーが許可しないときは、「[ドメインの検証に関する一般的な問題 \(p. 481\)](#)」で回避策を参照してください。
- Amazon SES でドメインを検証する場合は、そのドメインの任意のサブドメインから送信することができます。サブドメインを個別に検証する必要はありません。たとえば、example.com を検証する場合、a.example.com や a.b.example.com を検証する必要はありません。[RFC 1034](#) で規定されているように、各 DNS ラベルには最大で 63 文字を指定することができ、ドメイン名全体の合計文字数は 255 を超えることはできません。
- ルートドメインを共有するドメイン、サブドメイン、および E メールアドレスを検証する場合、検証されるアイデンティティ設定 (フィードバック通知、Easy DKIM など) は検証する最も細かいレベルで適用されます。つまり、次のようになります。
  - 検証済み E メールアドレス設定は、検証済みドメイン設定をオーバーライドします。
  - 検証済みサブドメイン設定は、検証済みドメイン設定をオーバーライドします。より低いレベルのサブドメイン設定は、より高いレベルのサブドメイン設定をオーバーライドします。

たとえば、user@a.b.example.com、a.b.example.com、b.example.com、および example.com を検証するとします。これらは、以下のシナリオで使用される検証済みのアイデンティティ設定です。

- user@example.com (個別には検証されないアドレス) から送信される E メールは、example.com の設定を使用します。
- user@a.b.example.com (個別に検証されるアドレス) から送信される E メールは、user@a.b.example.com の設定を使用します。
- user@b.example.com (個別には検証されないアドレス) から送信される E メールは、b.example.com の設定を使用します。
- ドメイン名では大文字と小文字は区別されません。example.com を検証すると、EXAMPLE.com から送信できます。
- AWS リージョンごとに、10,000 個までの確認済み ID (ドメインと E メールアドレスの任意の組み合わせ) を含めることができます。

## Amazon SES でのドメインの検証

以下の手順では、Amazon SES コンソールを使用してドメインを検証する方法を示します。代わりに Amazon SES API v1 を使用する場合は、[Amazon Simple Email Service API リファレンス](#)を参照してください。

E メール ID の検証中にデフォルトの設定セットを設定するには、「[Amazon SES デフォルト設定セットの管理 \(p. 260\)](#)」を参照してください。

### Note

このセクションの手順を完了する代わりに、[Easy DKIM \(p. 135\)](#) を有効にすることもできます。ドメインの DNS 設定に DKIM が追加されたことを Amazon SES が検出すると、このセクションの手順を完了していなくても、そのドメインから E メールを送信できるようになります。

ドメインを検証するには

1. Amazon SES コンソールの[検証済みドメインリスト](#)に移動するか、次の手順に従って移動してください。
  - a. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
  - b. ナビゲーションペインで、[Identity Management] の [Domains] を選択します。

2. [Verify a New Domain] を選択します。
3. [Verify a New Domain] ダイアログボックスで、ドメイン名を入力します。

Tip

ドメインが `www.example.com` の場合は、ドメインとして「`example.com`」と入力します。  
「`www.`」部分は不要であり、これを含めるとドメイン検証プロセスが失敗します。

4. このドメインに DKIM 署名を設定する場合は、[Generate DKIM Settings (DKIM 設定の生成)] オプションを選択します。DKIM 署名の詳細については「[Amazon SES における DKIM を使った Eメールの認証 \(p. 135\)](#)」を参照してください。
5. [Verify This Domain] を選択します。
6. [Verify a New Domain] ダイアログボックスには、[Domain Verification Record Set] が表示されます。これには、[Name]、[Type]、および [Value] が含まれます。(ダイアログボックスを閉じた後でも、この情報はドメイン名を選択することで表示できます。)

ドメインの検証を完了するには、表示された Name および Value を指定した TXT レコードをドメインの DNS サーバーに追加します。Amazon SES の TXT レコードに関する情報、および DNS サーバーへの TXT レコードの追加方法に関する一般的なガイダンスについては、「[Amazon SES ドメイン検証 TXT レコード \(p. 65\)](#)」を参照してください。特に、次のことに注意してください。

- DNS プロバイダーがレコード名でのアンダースコアの使用を許可していない場合は、[名前] から `_amazonses` を削除できます。
- ドメインの DNS 設定内でこのレコードを識別しやすくするために、必要に応じて Value の前に `amazonses:` を付けることができます。
- DNS プロバイダーによっては、DNS レコード名にドメイン名が自動的に付加される場合があります。ドメイン名の重複を避けるために、DNS レコードのドメイン名の末尾にピリオドを追加できます。これにより、レコード名が完全修飾されて、DNS プロバイダーによる追加のドメイン名の付加が不要になります。

### Verify a New Domain

The domain `example.com` has been added to the list of Verified Senders with a Status of "pending verification". Further action is needed to complete verification of this domain. See details below.

In order to complete verification of `example.com`, you must create the following records in the DNS settings for the domain, with the following values:

Domain Verification Record Set

Name	Type	Value
<code>_amazonses.example.com</code>	TXT	<code>38BnuGvLZ13OZtQP+RRznLBdvhOqoz5tNqH4sKmB/pl=</code>

If your DNS provider does not allow underscores in record names, you can omit `_amazonses` from the record name. To help you easily identify this record within your domain's DNS settings, you can optionally prefix the record value with `amazonses:`. For more information about Amazon SES TXT records and general guidance on how to add a TXT record to a DNS server, see [Amazon SES Domain Verification TXT Records](#).

[Download Record Set as CSV >>](#)

Close

7. 検証対象のドメインが Route 53 の DNS サービスを使用しており、Route 53 用の同じアカウントで AWS Management Console にサインインしている場合は、Amazon SES で Amazon SES コンソール内からすぐに DNS サーバーを更新できます。

別の DNS プロバイダーを使用する場合、DNS レコードを更新する手順は、使用する DNS またはウェブホスティングプロバイダーによって異なります。次のテーブルでは、いくつかの主要なプロバ

イダーに関するドキュメントへのリンクを示しています。このリストにすべてが網羅されているわけではなく、このリストに含まれているからといって、他社の製品やサービスを支援または推奨するものではありません。お使いのプロバイダーがこの表にない場合でも、Amazon SES でそのドメインを使用できる可能性があります。

DNS/ホスティングプロバイダー	ドキュメントのリンク
GoDaddy	<a href="#">TXT レコードを追加する</a>
Dreamhost	<a href="#">カスタム DNS レコードを追加する方法</a>
Cloudflare	<a href="#">CloudFlare内でDNSレコードを管理する</a>
HostGator	<a href="#">HostGator/eNom で DNS レコードを管理する</a>
Namecheap	<a href="#">ドメインの TXT/SPF/DKIM/DMARC レコードを追加する方法</a>
Names.co.uk	<a href="#">ドメインの DNS 設定の変更</a>
Wix	<a href="#">Wix アカウントの TXT レコードの追加または更新</a>

検証が完了すると、Amazon SES コンソールでのドメインのステータスが「pending verification (検証中)」から「verified (検証済み)」に変わり、Amazon SES から通知 E メールが届きます。

- これで、Amazon SES を使用して検証済みドメインの任意のアドレスから E メールを送信できます。テストメールを送信するには、検証済みドメインの隣のボックスをオンにし、[Send a Test Email] を選択します。

DNS 設定が正しく更新されない場合は、Amazon SES からドメイン検証失敗の通知メールが届き、ドメインにより [Domains] タブに failed のステータスが表示されます。この場合は、「[ドメインの検証に関する一般的な問題 \(p. 481\)](#)」のトラブルシューティングに関するページにあるステップを完了します。TXT が正しく作成されたことを確認したら、failed ステータス通知の横にある [再試行リンク] を選択して、ドメイン検証プロセスを再開します。

## Amazon SES でのドメインの一覧表示

検証済みドメインを表示するには、以下の手順に従ってください。

検証済みドメインを表示するには

- Amazon SES コンソールの[検証済みドメインリスト](#)に移動するか、次の手順に従って移動してください。
  - AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
  - ナビゲーションペインで、[Identity Management] の [Domains] を選択します。
- 検証済みドメインのリストで、1 つ以上のドメインを展開して、詳細を表示することができます。

## Amazon SES でドメイン ID を削除する

検証済みドメインを削除するには、以下の手順に従ってください。



## 検証済みドメインを削除するには

1. Amazon SES コンソールの [検証済みドメインリスト](#) に移動するか、次の手順に従って移動してください。
  - a. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
  - b. ナビゲーションペインで、[Identity Management] の [Domains] を選択します。
2. 削除する各ドメインの横にあるボックスをオンにし、[Remove] を選択します。
3. 削除されたドメインからは E メールを送信できなくなります。

## Amazon SES ドメイン検証の失効

Amazon SES は定期的にドメインの検証ステータスを見直し、有効でなくなっている場合には検証を取り消します。ドメインの所有権の確認に必要な TXT レコード情報を Amazon SES が検出できない場合、Amazon SES 経由で Amazon SES ドメイン検証失効警告 (REVOCATION WARNING) の E メールが送信されます。

72 時間以内にドメインの DNS サーバーに TXT レコード情報を復元すると、Amazon SES 経由で Amazon SES ドメイン検証失効キャンセル (REVOCATION CANCELLATION) の E メールが送信されます。

### Note

以下の手順に従って、Amazon SES コンソールで必要な TXT レコードを確認できます。ナビゲーションペインで、[Identity Management] の [Domains] を選択します。ドメインのリストでドメインを選択 (展開するだけでなく) して、TXT レコード名と値を含むドメインの確認設定を表示します。

72 時間以内にドメインの DNS サーバーに TXT レコード情報を復元しないと、Amazon SES 経由で Amazon SES ドメイン検証失効 (REVOCATION) の E メールが送信されます。これにより、ドメインは [ドメイン] タブの 検証済みの送信者リストから削除され、ドメインからは送信できなくなります。

検証が失効したドメインを再検証するには、失効したドメインがまったく新しいドメインであるかのように、検証手順を最初から再開始する必要があります。

## Amazon SES ドメイン検証 TXT レコード

お客様のドメインは、DNS プロバイダーを介して管理する一連のドメインネームシステム (DNS) レコードに関連付けられます。TXT レコードは、ドメインに関する追加情報を提供する一種の DNS レコードです。各 TXT レコードは名前と値で構成されます。

Amazon SES コンソールまたは API を使用してドメインの検証を開始すると、Amazon SES によって TXT レコードに使用する名前と値が割り当てられます。たとえば、ドメインが example.com の場合、Amazon SES によって生成される TXT レコードの設定は以下のようになります。

名前	タイプ	値
_amazonses.example.com	TXT	pmBGN/7MjnfhTKUZ06Enqq1PeGUaOkw8IGhcfwefcHU=

指定した [名前] と [値] を使用して、TXT レコードをドメインの DNS サーバーに追加します。その TXT レコードがドメインの DNS 設定内にあることが検出されると、Amazon SES ドメインの検証は完了です。

### Note

一部の DNS プロバイダーは、DNS レコード名に自動的にドメイン名を追加します。ドメイン名の重複を避けるために、DNS レコードのドメイン名の末尾にピリオド (.) を追加するか、レコー

ド名からドメインを省略できます。詳細については、DNS プロバイダーのドキュメントを参照してください。

DNS プロバイダーが DNS レコード名でのアンダースコアの使用を許可していない場合は、[名前] から `_amazonses` を削除できます。この場合、前述の例では、TXT レコード名は `_amazonses.example.com` ではなく、`example.com` になります。レコードを識別しやすくメンテナンスしやすくするために、必要に応じて Value の前に `amazonses:` を付けることもできます。したがって、前の例で TXT レコードの値は `amazonses:pmBGN/7MjnfhTKUZ06Enqq1PeGUaOkw8lGhcfwefcHU=` になります。

#### Note

Amazon SES では以前、TXT レコード名にアンダースコアなしの `amazonses` を含めることができました。ドメインが検証済みで、TXT レコードにアンダースコアなしの `amazonses` が含まれている場合、そのドメインは引き続き検証済みとなります。お客様の側で必要になる操作はありません。ただし、新しいドメインの検証には、TXT レコード名の `amazonses` には先頭にアンダースコアをつけるか、または `_amazonses` を完全に TXT レコード名から削除する必要があります。

ドメイン検証の設定に関するトラブルシューティング情報と確認手順については、「[ドメインの検証に関する一般的な問題 \(p. 481\)](#)」を参照してください。

### ドメインの DNS サーバーに TXT レコードを追加する

ドメインの DNS サーバーに TXT レコードを追加する手順は DNS プロバイダーによって異なります。DNS プロバイダーは、Amazon Route 53 または GoDaddy などの別のドメイン名レジストラである可能性があります。このセクションでは、Route 53 に TXT レコードを追加する手順と、他の DNS プロバイダー向けの一般的な手順を示します。

#### Amazon Route 53 の手順

Amazon SES で使用する [新しいドメインの検証プロセス \(p. 62\)](#) を開始する場合、ドメイン検証用の TXT レコードを自動的に Route 53 設定に追加できます。ただし、TXT レコードを自動的に追加せずに、TXT レコードを手動で Route 53 設定に追加する場合は、このセクションの手順を実行できます。

TXT レコードを Route 53 マネージドドメインの DNS レコードに追加するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. [Identity Management (ID 管理)] の [Domains (ドメイン)] を選択します。
3. 検証するドメインを選択します。
4. [Verification (検証)] セクションを展開します。[TXT Value (TXT 値)] の横に表示されている値をコピーします。
5. Route 53 コンソール (<https://console.aws.amazon.com/route53/>) を開きます。
6. ナビゲーションペインで [Hosted zones] を選択します。
7. TXT レコードを追加する先のドメインを選択します。
8. [Create record (レコードを作成)] を選択します。
9. ルーティングポリシーで、[シンプルルーティング] を選択してから、[次へ] をクリックします。
10. [Define simple record (シンプルレコードを定義)] を選択します。
11. シンプルレコードを定義ペインで、次の操作を行います。
  - a. レコード名で、「`_amazonses`」と入力します。
  - b. 値/ルートへのトラフィック先で、IP アドレス、またはレコードタイプに応じた別の値を選択します。次に、テキスト領域に、Amazon SES コンソールからコピーした TXT レコードの値を貼り付けます。
  - c. レコードタイプで、[TXT] を選択します。
  - d. [TTL (Seconds) (TTL (秒))] に、[1800] と入力します。
  - e. [Define simple record (シンプルレコードを定義)] を選択します。

- レコードが反映されるまで約30分待ちます。そして、Amazon SES コンソールの [Domains (ドメイン)] ページで、ドメインの横にある [Status (ステータス)] 列の値を確認します。ステータスが [検証の保留中] である場合は、30分待ってから、[更新] (🔄) を選択します。ステータス列の値が [verified (検証済み)] になるまで、このプロセスを繰り返します。

### 他の DNS プロバイダー向けの一般的な手順

DNS 設定に TXT レコードを追加する手順は、プロバイダーによって異なります。詳しい手順については、DNS プロバイダーのドキュメントを参照してください。このセクションでは、ドメインの DNS 設定に TXT レコードを追加する場合の基本的なステップの概要を提供します。

ドメインの DNS サーバーに TXT レコードを追加するには (一般的な手順)

- DNS プロバイダーの管理コンソールに移動し、アカウントにサインインします。
- ドメインの DNS レコードを更新するためのページを見つけます。このページには、[DNS Records (DNS レコード)]、[DNS Zone File (DNS ゾーンファイル)]、または [Advanced DNS (アドバンスド DNS)] の例に示すような名前が表示されます。DNS サービスプロバイダーから提供されたドキュメントで詳細を参照してください。
- Amazon SES によって割り当てられた名前と値を指定した TXT レコードを追加します。

#### Important

DNS プロバイダーによっては、DNS レコードの末尾に GoDaddy のようなドメイン名が自動的に付加される場合があります。既にドメイン名が含まれているレコード (`_amazonses.example.com` など) を追加すると、ドメイン名が重複したレコード (`_amazonses.example.com.example.com` など) になる場合があります。ドメイン名の重複を避けるには、DNS レコードのドメイン名の末尾にピリオドを追加するか、単にレコード名からドメインを省略します。DNS サービスプロバイダーから提供されたドキュメントで詳細を参照してください。

- 変更を保存します。DNS レコードの更新が有効になるには、最大 48 時間かかることがあります、多くの場合それよりも大幅に早く有効になります。「[ドメイン検証設定の確認 \(p. 482\)](#)」の手順を使用して、TXT レコードが正しく公開されることを確認できます。

## AWS アクセスキーの取得

Amazon SES にサインアップした後、Amazon SES API 経由で Amazon SES にアクセスする場合は、直接 Query (HTTPS) インターフェイスから、または [AWS SDK](#)、[AWS Command Line Interface](#)、もしくは [AWS Tools for Windows PowerShell](#) から AWS アクセスキーを取得する必要があります。AWS アクセスキーは、アクセスキー ID とシークレットアクセスキーで構成されます。

Amazon SES で使用できるセキュリティキーのタイプの詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。AWS アクセスキーの取得についての詳細は、AWS 全般のリファレンスの「[AWS セキュリティ認証情報](#)」を参照してください

## AWS SDK のダウンロード

未加工 HTTP リクエストの組み立てなどの低レベルの詳細を処理せずに Amazon SES API を呼び出す場合は、AWS SDK を使用できます。AWS SDK は、Amazon SES や AWS の他のサービスの機能をカプセル化する関数とデータ型を提供します。AWS SDK をダウンロードするには、「[SDK](#)」にアクセスしてください。

このガイドの「[開始方法](#)」セクションでは、各種プログラミング言語で AWS SDK を使用して E メールを送信する方法の例を示しています。詳細については、「[AWS SDK を使用して Amazon SES から E メールを送信する \(p. 31\)](#)」を参照してください。

## カスタムの MAIL FROM ドメインの設定

E メールを送信する際、送信元を示す 2 つのアドレスを使用します。メッセージの受取人に表示される差出人アドレスと、メッセージの発信元を示す MAIL FROM アドレスです。MAIL FROM アドレスは、envelope sender、envelope from、bounce address、または Return Path アドレスと呼ばれることもあります。メールサーバーは MAIL FROM アドレスを使用して、バウンスメッセージやその他のエラー通知を返します。MAIL FROM アドレスは通常、受取人がメッセージのソースコードを表示する場合にのみ表示できます。

Amazon SES では、独自のドメインを指定する場合を除き、送信するメッセージの MAIL FROM ドメインがデフォルト値に設定されます。このセクションでは、カスタム MAIL FROM ドメインを設定する利点について説明し、設定手順を示します。

### カスタムの MAIL FROM ドメインを使用する理由

デフォルトでは、Amazon SES から送信するメッセージには、MAIL FROM ドメインとして `amazonses.com` のサブドメインが使用されます。デフォルトの MAIL FROM ドメインがメールを送信するアプリケーション (この場合 Amazon SES) と一致するため、これらのメッセージは SPF (Sender Policy Framework) 認証に合格します。

送信者が多い場合はこのレベルの認証で十分ですが、他の送信者は自分の所有するドメインに MAIL FROM ドメインを設定することを好みます。カスタム MAIL FROM ドメインを設定することにより、E メールは [ドメインベースのメッセージ認証、レポート、および適合性 \(DMARC\) \(p. 147\)](#) に準拠できます。DMARC により、送信者のドメインは、ドメインから送信された E メールが 1 つ以上の認証システムによって保護されていることを示すことができます。

DMARC 検証を実現するには、[Sender Policy Framework \(p. 134\) \(SPF\)](#) を使用する方法と、[DomainKeys Identified Mail \(p. 135\) \(DKIM\)](#) を使用する方法の 2 つがあります。SPF で DMARC に準拠する唯一の方法は、カスタム MAIL FROM ドメインを使用することです。SPF 検証では、差出人アドレスのドメインが MAIL FROM ドメインと一致する必要があるためです。独自の MAIL FROM ドメインを使用すると、SPF または DKIM、あるいはその両方を使用して、DMARC 認証に合格することができます。

### MAIL FROM ドメインの選択

MAIL FROM ドメインに使用するサブドメインは、次の要件を満たしている必要があります。

- MAIL FROM ドメインは、メールの送信元である検証済み ID (メールアドレスまたはドメイン) のサブドメインである必要があります。たとえば、`mail.example.com` は、ドメイン `example.com` の有効な MAIL FROM ドメインです。
- Eメールの送信元ドメインを MAIL FROM ドメインに設定することはできません。From アドレスに MAIL FROM ドメインを使用する必要がある場合は、[Eメールのフィードバック転送を無効 \(p. 275\)](#) にしてバウンスを Amazon SNS 通知から受け取るか、MAIL FROM ドメインがフィードバックの宛先でないことを確認します。Eメールのフィードバック転送の宛先を調べる方法については、「[Eメールのフィードバック転送先 \(p. 275\)](#)」を参照してください。
- MAIL FROM ドメインには、Eメールの受信に使用するドメインを指定することはできません。

### MAIL FROM ドメインの設定

カスタム MAIL FROM ドメインを設定するプロセスでは、ドメインの DNS 設定にレコードを追加する必要があります。ドメインがメールプロバイダーから送信されたバウンスと苦情の通知を受信できるようにするには、MX レコードを公開する必要があります。また、ドメインからメールを送信するアクセス許可が Amazon SES にあることを証明するためにも、SPF レコードを公開する必要があります。

ドメイン全体または個々のメールアドレスに対して、カスタム MAIL FROM ドメインを設定できます。次の手順は、Amazon SES コンソールを使用してカスタム MAIL FROM ドメインを設定する方法を示してい

ます。SetIdentityMailFromDomain API オペレーションを使用して、カスタム MAIL FROM ドメインを設定することもできます。

### 検証済みドメインの MAIL FROM ドメインを設定する

ドメイン全体に対して MAIL FROM ドメインを設定できます。この場合、そのドメインのアドレスから送信するメッセージではすべて、同じ MAIL FROM ドメインを使用します。

指定された MAIL FROM ドメインが確認済みドメインで使用されるように設定するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで、[Identity Management] の [Domains] を選択します。
3. ドメインのリストで、MAIL FROM ドメインの親ドメインが検証済みであることを確認します。ドメインを検証していない場合は、[Amazon SES でのドメインの検証 \(p. 61\)](#) の手順に従って、ドメインを確認します。それ以外の場合は、ドメインを選択し、次のステップに進みます。
4. [MAIL FROM ドメイン] で、[MAIL FROM ドメインの設定] を選択します。
5. [MAIL FROM ドメインの設定] ウィンドウで、以下の操作を行います。
  - a. [MAIL FROM ドメイン] に、MAIL FROM ドメインとして使用するサブドメインを入力します。
  - b. [Behavior if MX record not found] で、以下のいずれかのオプションを選択します。
    - Use region.amazonses.com as MAIL FROM – カスタム MAIL FROM ドメインの MX レコードが正しくセットアップされていない場合、Amazon SES は amazonses.com のサブドメインを使用します。サブドメインは、Amazon SES を使用する AWS リージョンによって異なります。
    - メッセージ拒否 – カスタム MAIL FROM ドメインの MX レコードが正しくセットアップされていない場合、Amazon SES はエラー MailFromDomainNotVerified を返します。このドメインから送信しようとした E メールは自動的に拒否されます。
  - c. [Set MAIL FROM Domain] を選択します。ドメインの DNS 設定に追加する必要がある MX および SPF レコードを含むウィンドウが表示されます。これらのレコードでは、次の表に示す形式を使用します。

名前	タイプ	値
<i>subdomain.domain.com</i>	MX	10 feedback-smtp.#### #.amazonses.com
<i>subdomain.domain.com</i>	TXT	「v=spf1 include:amazonses.com ~all」

前述のレコードで、*subdomain.domain.com* を MAIL FROM ドメインに、*region* を、MAIL FROM ドメインを検証する AWS リージョンの名前 (us-west-2、us-east-1、eu-west-1 など) に置き換えます。TXT レコードの値には引用符を含める必要がある点に注意してください。

これらの値を記録して、次のステップに進みます。

6. カスタム MAIL FROM ドメインの DNS サーバーに MX レコードを公開します。

#### Important

Amazon SES でカスタムの MAIL FROM ドメインを正しくセットアップするには、MAIL FROM ドメインの DNS サーバーに、MX レコードを 1 件のみ発行する必要があります。MAIL FROM ドメインに複数の MX レコードがあると、Amazon SES でのカスタム MAIL FROM のセットアップは失敗します。

MAIL FROM ドメインに対する DNS サービスの提供元が Route 53 であり、Route 53 で使用しているのと同じアカウントで AWS Management Console にサインインしている場合、[Route 53 を使用して

レコードを公開する]を選択します。DNS レコードは、ドメインの DNS 設定に自動的に適用されます。

別の DNS プロバイダーを使用する場合は、DNS レコードを MAIL FROM ドメインの DNS サーバーに手動で公開する必要があります。DNS レコードをドメインの DNS サーバーに追加する手順は、ウェブホスティングサービスまたは DNS プロバイダーによって異なります。

ドメインの DNS レコードを公開する手順は、使用している DNS プロバイダーによって異なります。このセクションには、いくつかの一般的な DNS プロバイダーに関するドキュメントへのリンクが含まれています。これは、プロバイダーの完全なリストではありません。以下にプロバイダーが示されていない場合でも、MAIL FROM ドメインを設定できる場合があります。このリストに含まれているからといって、他社の製品やサービスを支持または推奨するものではありません。

DNS/ホスティングプロバイダー名	ドキュメントのリンク
GoDaddy	<ul style="list-style-type: none"><li>• MX: <a href="#">MX レコードを追加する</a></li><li>• TXT: <a href="#">TXT レコードを追加する</a></li></ul>
DreamHost	<ul style="list-style-type: none"><li>• MX: <a href="#">MX レコードを変更するにはどうすればよいですか?</a></li><li>• TXT: <a href="#">カスタム DNS レコードを追加する方法</a></li></ul>
Cloudflare	<ul style="list-style-type: none"><li>• MX: <a href="#">メールまたは MX レコードを追加または編集するにはどうすればよいですか?</a></li><li>• TXT: <a href="#">Cloudflare内でDNSレコードを管理する</a></li></ul>
HostGator	<ul style="list-style-type: none"><li>• MX: <a href="#">MX レコードの変更 - Windows</a></li><li>• TXT: <a href="#">HostGator/eNom で DNS レコードを管理する</a></li></ul>
Namecheap	<ul style="list-style-type: none"><li>• MX: <a href="#">メールサービスに必要な MX レコードを設定するには、どうすればよいですか?</a></li><li>• TXT: <a href="#">ドメインの TXT/SPF/DKIM/DMARC レコードを追加する方法</a></li></ul>
Names.co.uk	<ul style="list-style-type: none"><li>• MX: <a href="#">ドメインの DNS 設定の変更</a></li><li>• TXT: <a href="#">ドメインの DNS 設定の変更</a></li></ul>
Wix	<ul style="list-style-type: none"><li>• MX: <a href="#">Wix アカウントの MX レコードの追加または更新</a></li><li>• TXT: <a href="#">Wix アカウントの TXT レコードの追加または更新</a></li></ul>

レコードが適切に配置されていることを Amazon SES が検出すると、カスタム MAIL FROM ドメインが正常に設定されたことを通知するメールが送信されます。DNS プロバイダーによっては、Amazon SES が MX レコードを検出するまでに最大 72 時間の遅延が発生する場合があります。

### 検証済みの E メールアドレスの MAIL FROM ドメインを設定する

また、特定のメールアドレスにカスタム MAIL FROM ドメインを設定することもできます。E メールアドレスにカスタム MAIL FROM ドメインを設定するには、E メールアドレスが関連付けられているドメインの DNS レコードを変更する必要があります。

#### Note

自分が所有していないドメインのアドレスにカスタム MAIL FROM ドメインを設定することはできません (たとえば、gmail.com ドメインのアドレスにカスタム MAIL FROM ドメインを作成することはできません。必要な DNS レコードをドメインに追加できないためです)。

指定された MAIL FROM ドメインが確認済み E メールアドレスで使用されるように設定するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインの [Identity Management] で [Email Addresses] を選択します。
3. E メールアドレスのリストで、カスタム MAIL FROM ドメインを設定する E メールアドレスが検証済みであることを確認します。E メールアドレスを検証していない場合は、[Amazon SES E メールアドレスの確認 \(p. 49\)](#) の手順に従って、E メールアドレスを確認します。それ以外の場合は、E メールアドレスを選択し、次のステップに進みます。
4. [MAIL FROM ドメイン] で、[MAIL FROM ドメインの設定] を選択します。
5. [MAIL FROM ドメインの設定] ウィンドウで、以下の操作を行います。
  - a. [MAIL FROM ドメイン] に、MAIL FROM ドメインとして使用するサブドメインを入力します。
  - b. [Behavior if MX record not found] で、以下のいずれかのオプションを選択します。
    - Use region.amazonses.com as MAIL FROM – カスタム MAIL FROM ドメインの MX レコードが正しくセットアップされていない場合、Amazon SES は `amazonses.com` のサブドメインを使用します。サブドメインは、Amazon SES を使用する AWS リージョンによって異なります。
    - メッセージ拒否 – カスタム MAIL FROM ドメインの MX レコードが正しくセットアップされていない場合、Amazon SES はエラー `MailFromDomainNotVerified` を返します。この E メールアドレスから送信しようとした E メールは自動的に拒否されます。
  - c. [Set MAIL FROM Domain] を選択します。E メールアドレスが属するドメインの DNS 設定に追加する必要がある MX レコードおよび SPF レコードを含むウィンドウが表示されます。これらのレコードでは、次の表に示す形式を使用します。

名前	タイプ	値
<code>subdomain.domain.com</code>	MX	10 feedback-smtp. <code>region</code> .amazonses.com
<code>subdomain.domain.com</code>	TXT	「v=spf1 include:amazonses.com ~all」

前述のレコードで、`subdomain.domain.com` を MAIL FROM ドメインに、`region` を、MAIL FROM ドメインを検証する AWS リージョンの名前 (`us-west-2`、`us-east-1`、`eu-west-1` など) に置き換えます。TXT レコードの値には引用符を含める必要がある点に注意してください。

これらの値を記録して、次のステップに進みます。

6. カスタム MAIL FROM ドメインの DNS サーバーに DNS レコードを公開します。

#### Important

Amazon SES でカスタムの MAIL FROM ドメインを正しくセットアップするには、MAIL FROM ドメインの DNS サーバーに、MX レコードを 1 件のみ発行する必要があります。MAIL FROM ドメインに複数の MX レコードがあると、Amazon SES でのカスタム MAIL FROM のセットアップは失敗します。

MAIL FROM ドメインに対する DNS サービスの提供元が Route 53 であり、Route 53 で使用しているのと同じアカウントで AWS Management Console にサインインしている場合、[Route 53 を使用してレコードを公開する] を選択します。DNS レコードは、ドメインの DNS 設定に自動的に適用されます。

別の DNS プロバイダーを使用する場合は、DNS レコードを MAIL FROM ドメインの DNS サーバーに手動で公開する必要があります。DNS レコードをドメインの DNS サーバーに追加する手順は、ウェブホスティングサービスまたは DNS プロバイダーによって異なります。

ドメインの DNS レコードを公開する手順は、使用している DNS プロバイダーによって異なります。このセクションには、いくつかの一般的な DNS プロバイダーに関するドキュメントへのリンクが含まれています。これは、プロバイダーの完全なリストではありません。以下にプロバイダーが示されていない場合でも、MAIL FROM ドメインを設定できる場合があります。このリストに含まれているからといって、他社の製品やサービスを支持または推奨するものではありません。

DNS/ホスティングプロバイダー名	ドキュメントのリンク
GoDaddy	<ul style="list-style-type: none"> <li>MX: <a href="#">MX レコードを追加する</a></li> <li>TXT: <a href="#">TXT レコードを追加する</a></li> </ul>
DreamHost	<ul style="list-style-type: none"> <li>MX: <a href="#">MX レコードを変更するにはどうすればよいですか？</a></li> <li>TXT: <a href="#">カスタム DNS レコードを追加する方法</a></li> </ul>
Cloudflare	<ul style="list-style-type: none"> <li>MX: <a href="#">メールまたは MX レコードを追加または編集するにはどうすればよいですか？</a></li> <li>TXT: <a href="#">Cloudflare内でDNSレコードを管理する</a></li> </ul>
HostGator	<ul style="list-style-type: none"> <li>MX: <a href="#">MX レコードの変更 - Windows</a></li> <li>TXT: <a href="#">HostGator/eNom で DNS レコードを管理する</a></li> </ul>
Namecheap	<ul style="list-style-type: none"> <li>MX: <a href="#">メールサービスに必要な MX レコードを設定するには、どうすればよいですか？</a></li> <li>TXT: <a href="#">ドメインの TXT/SPF/DKIM/DMARC レコードを追加する方法</a></li> </ul>
Names.co.uk	<ul style="list-style-type: none"> <li>MX: <a href="#">ドメインの DNS 設定の変更</a></li> <li>TXT: <a href="#">ドメインの DNS 設定の変更</a></li> </ul>
Wix	<ul style="list-style-type: none"> <li>MX: <a href="#">Wix アカウントの MX レコードの追加または更新</a></li> <li>TXT: <a href="#">Wix アカウントの TXT レコードの追加または更新</a></li> </ul>

レコードが適切に配置されていることを Amazon SES が検出すると、カスタム MAIL FROM ドメインが正常に設定されたことを通知するメールが送信されます。DNS プロバイダーによっては、Amazon SES が MX レコードを検出するまでに最大 72 時間の遅延が発生する場合があります。

## Amazon SES での MAIL FROM ドメインのセットアップ状態

カスタムの MAIL FROM ドメインを使用するようにアイデンティティを設定すると、DNS 設定内の必要な MX レコードを Amazon SES で検出するまでは、セットアップの状態が [pending] になります。Amazon SES で MX レコードを検出したかどうかで状態は変わります。次の表は、各状態に対応する E メール送信動作と Amazon SES のアクションの一覧です。状態が変わるたびに、AWS アカウントに関連付けられている E メールアドレスに、Amazon SES から通知が送信されます。

状態	E メール送信動作	Amazon SES のアクション
Pending	カスタムの MAIL FROM フォールバック設定を使用	Amazon SES は、必要な MX レコードの検出を 72 時間試行します。検出できない場合、



状態	E メール送信動作	Amazon SES のアクション
		状態は "Failed" に変化します。
成功	カスタムの MAIL FROM ドメインを使用	Amazon SES では、必要な MX レコードがあることを継続的に確認します。
TemporaryFailure	カスタムの MAIL FROM フォールバック設定を使用	Amazon SES は、必要な MX レコードの検出を 72 時間試行します。検出できない場合、状態は "Failed" に変化します。検出できた場合、状態は "Success" に変化します。
失敗	カスタムの MAIL FROM フォールバック設定を使用	Amazon SES は、必要な MX レコードを検出するための試行を停止しました。カスタムの MAIL FROM ドメインを使用するには、 <a href="#">MAIL FROM ドメインの設定 (p. 68)</a> のセットアッププロセスをやり直す必要があります。

## Amazon SES の SPF レコードの設定

SPF レコードは ISP に対し、ドメインの E メールを送信することを Amazon SES が許可されていることを示します。Amazon SES を使用する場合、SPF レコードを発行すべきかどうかは、E メールが受信メールサーバーによる SPF チェックにのみ合格すればよいのか、SPF に基づく DMARC (Domain-based Message Authentication, Reporting and Conformance) 認証に合格するために必要な追加要件も満たす必要があるのかによって決まります。詳細については、「[Amazon SES における SPF を使った Eメールの認証 \(p. 134\)](#)」を参照してください。

## Amazon SES の SMTP 認証情報の取得

Amazon SES SMTP インターフェイスを使用するには、まず、SMTP ユーザー名とパスワードを作成する必要があります。SMTP 認証情報を取得するには、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

### Important

SMTP ユーザー名とパスワードは、AWS アクセスキー ID およびシークレットアクセスキーと同じではありません。Amazon SES SMTP エンドポイントに対して自身の承認を行う際に、AWS

認証情報を使用しないでください。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。

## Amazon SES サンドボックス外への移動

不正使用や悪用を防止し、送信者としての評価を保つため、新しい Amazon SES アカウントには一定の制限が適用されます。

すべての新しいアカウントが Amazon SES サンドボックスに配置されます。アカウントはサンドボックスにありますが、Amazon SES のすべての機能を使用することができます。ただし、アカウントがサンドボックスにある場合、アカウントに次の制限が適用されます。

- Eメールの送信先は、検証済み Eメールアドレスおよびドメイン、または [Amazon SES メールボックスシミュレーター \(p. 187\)](#) に制限されます。
- Eメールは、検証済み Eメールアドレスまたはドメインからのみ送信できます。

### Note

この制限は、アカウントがサンドボックスにない場合でも適用されます。

- 最大で 24 時間あたり 200 メッセージを送信できます。
- 最大で 1 秒あたり 1 メッセージを送信できます。

アカウントがサンドボックスの外にあるときは、受信者のアドレスまたはドメインが検証されたかどうかにかかわらず、任意の受信者に Eメールを送信できます。ただし、「From」、「Source」、「Sender」または「Return-Path」アドレスとして使用するすべての ID は引き続き検証する必要があります。

アカウントがサンドボックスから削除されることをリクエストするには、このセクションの手順を完了します。

### Note

Amazon SES を使用して Amazon EC2 インスタンスから Eメールを送信する場合、Amazon EC2 インスタンスのポート 25 からスロットルを削除するようにリクエストする必要がある場合もあります。詳細については、AWS ナレッジセンターの「[EC2 インスタンスからポート 25 のスロットルを解除する方法を教えてください。](#)」を参照してください。

AWS Management Console を使用して、アカウントが Amazon SES サンドボックスから削除されるようにリクエストするには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインの、[Email Sending (Eメール送信)] で、[Sending Statistics (送信ステータス)] と選択します。
3. アカウントの詳細については、次の図に示されているとおり、[アカウントの詳細を編集する] を選択します。

## Your account details

---



**Your Amazon SES account has "sandbox" access in region** the Amazon SES mailbox simulator and verified email addresses in your account details. [Learn more](#).

**Can't find your existing account settings?** Your account may be in a different region. See the [navigation bar](#).

<b>Production Access</b>	<b>Sandbox</b>
<b>Mail type</b>	none
<b>Website URL</b>	none
<b>Use case description</b>	none
<b>Additional contact addresses</b>	none
<b>Preferred contact language</b>	none

[Edit your account details](#)

4. アカウントの詳細モデルで、以下のアカウントの詳細を入力します。

## Edit your account details

Provide your account details, including a description of how you plan to email, to update your account. Choose Yes if you wish to enable produc

**Enable Production Access**

Yes

No

**Mail type**

Transactional

**Website URL**

**Use case description**

**Additional contact addresses**

**Preferred contact language**

English

I agree to the [AWS S](#)

- 本番稼働用アクセスの有効化で、[はい]または[いいえ]を選択します。 サンドボックスから移動するには、[はい]を選択します。
- [Mail Type (メールの種類)] で、送信する予定の E メールタイプを選択します。複数の値が当てはまる場合は、送信する Eメールの大部分に当てはまるオプションを選択します。

- [Website URL (ウェブサイトの URL)] に、ウェブサイトの URL を入力します。この情報は、お客様が送信を予定しているコンテンツのタイプを AWS が正しく理解するために役立ちます。
- ユースケースの説明で、Amazon SES を使用して E メールを送信する方法を説明します。AWS がお客様のリクエストを処理できるように、以下の質問にも答えます。
  - メーリングリストをどのように構築または取得しますか？
  - バウンスや苦情はどのように処理しますか？
  - 受取人はお客様からの Eメールの受信をどのようにオプトアウトできますか？
  - このリクエストで指定した送信レートまたは送信クォータは、どのように選択しましたか？
- 5. 追加の連絡先住所で、アカウントに関するコミュニケーションの受信場所をお知らせください。最大 4 つの E メールアドレスをカンマで区切って指定できます。
- 6. [優先する問い合わせ言語] で、本件に関して連絡を受けるのに、[英語] か [日本語] を選択します。
- 7. 完了したら、[提出してレビュー] を選択します。

#### Note

アカウントの詳細のレビューを送信すると、レビューが完了するまで詳細を編集できなくなります。

AWS Management Console を使用して本番稼働用アクセスリクエストを送信する代わりに、AWS CLI を使用してリクエストを送信できます。AWS CLI を使用してリクエストを送信すると、多数の ID に対して本番稼働用アクセスをリクエストする場合や、Amazon SES のセットアッププロセスを自動化する場合に便利です。

AWS CLI を使用して、アカウントが Amazon SES サンドボックスから削除されるようにリクエストするには

#### Note

このセクションの手順を完了する前に、まず AWS CLI をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 put-account-details \  
--production-access-enabled \  
--mail-type TRANSACTIONAL \  
--website-url https://example.com \  
--use-case-description "Use case description" \  
--additional-contact-email-addresses info@example.com \  
--contact-language EN
```

上記のコマンドで、次の操作を行います。

- **#####**を Amazon SES 経由で送信する予定の Eメールのタイプに置き換えます。TRANSACTIONAL または PROMOTIONAL のどちらかを指定できます。複数の値が当てはまる場合は、送信する Eメールの大部分に当てはまるオプションを指定します。
- **https://example.com** を、ウェブサイトの URL に置き換えます。この情報は、お客様が送信を予定しているコンテンツのタイプを AWS が正しく理解するために役立ちます。
- **#####**を、Amazon SES を使用して Eメールを送信する方法の説明に置き換えます。AWS がお客様のリクエストを処理できるように、以下の質問にも答えます。
  - メーリングリストをどのように構築または取得しますか？
  - バウンスや苦情はどのように処理しますか？
  - 受取人はお客様からの Eメールの受信をどのようにオプトアウトできますか？
  - このリクエストで指定した送信レートまたは送信クォータは、どのように選択しましたか？

- `info@example.com` を、アカウントに関するお知らせの受信先となるメールアドレスに置き換えます。最大 4 つの E メールアドレスをカンマで区切って指定できます。
- `EN` 任意の言語を使用します。英語の場合は `EN` を、日本語の場合は `JP` を指定します。

#### Note

アカウントの詳細のレビューを送信すると、レビューが完了するまで詳細を編集できなくなります。

AWS Support チームは、お客様のリクエストに対して、24 時間以内に一次回答を行います。

迷惑なコンテンツや悪意のあるコンテンツを送信するためにシステムが悪用されないように、各リクエストを慎重に検討する必要があります。可能であれば、24 時間以内にリクエストを承認します。ただし、お客様から追加情報を取得する必要がある場合は、お客様のリクエストの解決に時間がかかる場合があります。

お客様のユースケースが AWS の方針と一致しない場合は、リクエストを承認できない場合があります。

## アカウントのサンドボックスステータスを確認する

お使いのアカウントがまだサンドボックスに入っているかどうかを確認するには、Amazon SES コンソールを使用します。

アカウントがサンドボックスにあるかどうかを判断するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで、[Email Sending (E メール送信)]、[Sending Statistics (送信ステータス)] の順に選択します。
3. アカウントの詳細で、アカウントのステータスが表示されます。

アカウントがまだ審査中の場合は、アカウントがまだ審査中であることを示すバナーが表示されます。お客様の本番稼働用アクセスステータスは引き続きサンドボックスだと分かります。

## Your account details



Your account details are currently under review. [See case 7266](#)

<b>Production Access</b>	<b>Sandbox</b>
<b>Mail type</b>	Transactional
<b>Website URL</b>	https://example.com
<b>Use case description</b>	A description of how you plan to use A
<b>Additional contact addresses</b>	info@example.com
<b>Preferred contact language</b>	English

バナーには、サンドボックスからの移動リクエストが拒否された場合や失敗した場合の、アカウントのステータスも反映されます。

サンドボックスからの移動リクエストが許可されている場合、本番稼働用アクセスステータスは[有効]になっています。このステータスは、現在のリージョンのサンドボックスにはないことを意味します。

## Your account details

<b>Production Access</b>	Enabled
<b>Mail type</b>	Transactional
<b>Website URL</b>	https://example.com
<b>Use case description</b>	A description of how you plan to use A
<b>Additional contact addresses</b>	info@example.com
<b>Preferred contact language</b>	English

[Edit your account details](#)

確認していないアドレスに E メールを送信して、自分のアカウントがサンドボックスに入っているかどうかを確認することもできます。アカウントがサンドボックスにある場合は、宛先アドレスが確認されていないことを示すエラーメッセージが表示されます。

## カスタムドメインを設定してオープンとクリックの追跡を処理する

[イベント発行 \(p. 296\)](#)を使用して、オープンおよびクリックイベントをキャプチャすると、Amazon SES は送信する E メールに小さな変更を加えます。オープンイベントをキャプチャするために、Amazon SES は 1 ピクセル × 1 ピクセルの透明イメージを各 Eメールの最後に追加します。このイメージには、Eメールごとに一意のファイル名があり、Amazon SES が運営するサーバー上でホストされています。リンククリックイベントをキャプチャするために、Amazon SES は Eメールのリンクを Amazon SES が運営するサーバーへのリンクで置き換えます。これにより、受信者はただちに目的の宛先にリダイレクトされます。一部の Amazon SES の顧客は、Amazon SES が所有および運営するドメインではなく、自分のドメインを使用して、受信者のためのより一貫性のあるエクスペリエンスを作成することができます。

複数のカスタムドメインを設定して、オープンとクリックの追跡イベントを処理できます。これらのカスタムドメインは設定セットと関連付けられています。設定セットを使用して Eメールを送信すると、その設定セットがカスタムドメインを使用するように設定されている場合、その Eメール内のオープンとクリックのリンクは、その設定セットで指定されたカスタムドメインを自動的に使用します。

このセクションには、Amazon SES によって運営されているオープンとクリックの追跡サーバーにユーザーを自動的にリダイレクトするために、所有するサーバー上にサブドメインを設定する手順が含まれています。これらのドメインの設定には 2 つのステップがあります。まず、サブドメイン自体を設定し、次にカスタムドメインを使用するよう設定セットを設定します。このトピックには、これらのステップの両方を完了する手順が含まれています。



## パート 1: オープンとクリックのリンクのリダイレクトを処理するためのドメインの設定

リダイレクトドメインを設定する固有の手順は、ウェブホスティングプロバイダ (および HTTPS サーバを使用する場合は Content Delivery Network) によって異なります。次のセクションの手順では、特定のステップではなく一般的なガイダンスを示します。

### オプション 1: HTTP ドメインの設定

HTTP ドメインを使用してオープンとリンクのリンクを処理する予定の場合 (HTTPS ドメインではなく)、サブドメインを設定するプロセスには、数ステップのみが含まれます。

#### Note

HTTP プロトコルを使用するカスタムドメインを設定し、HTTPS プロトコルを使用するリンクを含む E メールを送信する場合、Eメールのリンクをクリックすると警告メッセージが表示されることがあります。HTTPS プロトコルを使用するリンクが含まれる E メールを送信する場合は、HTTPS ドメインを使用して、オープンとクリックの追跡イベントを処理する必要があります。

HTTPS のサブドメインを使用する予定がある場合は、「[オプション 2: HTTPS ドメインの設定 \(p. 82\)](#)」の手順に従います。

オープンとクリックのリンクの処理のために、HTTP のサブドメインを設定するには

1. まだそうしていない場合は、オープンとクリックの追跡リンクに使用するサブドメインを作成します。これらのリンクの処理専用のサブドメインを作成することをお勧めします。
2. Amazon SES で使用するサブドメインを検証します。詳細については、「[Amazon SES でのドメインの検証 \(p. 61\)](#)」を参照してください。
3. サブドメインの DNS レコードを変更します。DNS レコードで、リクエストを Amazon SES 追跡ドメインにリダイレクトする新しい CNAME レコードを追加します。リダイレクトするアドレスは、Amazon SES を使用する AWS リージョンによって異なります。次の表には、Amazon SES が利用可能な AWS リージョンの追跡ドメインの一覧が含まれます。

AWS リージョン	AWS 追跡ドメイン
米国東部 (オハイオ)	r.us-east-2.awstrack.me
米国東部 (バージニア北部)	r.us-east-1.awstrack.me
米国西部 (オレゴン)	r.us-west-2.awstrack.me
アジアパシフィック (ムンバイ)	r.ap-south-1.awstrack.me
アジアパシフィック (ソウル)	r.ap-northeast-2.awstrack.me
アジアパシフィック (シンガポール)	r.ap-southeast-1.awstrack.me
アジアパシフィック (シドニー)	r.ap-southeast-2.awstrack.me
アジアパシフィック (東京)	r.ap-northeast-1.awstrack.me
カナダ (中部)	r.ca-central-1.awstrack.me
欧州 (フランクフルト)	r.eu-central-1.awstrack.me
欧州 (アイルランド)	r.eu-west-1.awstrack.me

AWS リージョン	AWS 追跡ドメイン
欧州 (ロンドン)	r.eu-west-2.awstrack.me
南米 (サンパウロ)	r.sa-east-1.awstrack.me
AWS GovCloud (US)	r.us-gov-west-1.awstrack.me

#### Note

ウェブホスティングプロバイダによっては、サブドメインの DNS レコードの変更が有効になるまでに数分かかることがあります。ウェブホスティングプロバイダや IT 組織はこれらの遅延に関する追加情報を提供できます。

## オプション 2: HTTPS ドメインの設定

HTTPS ドメインを使用して、リンクのクリックを追跡することもできます。リンクを追跡するために HTTPS ドメインを設定するには、[HTTP ドメインの設定 \(p. 81\)](#)に必要なステップ以外に、いくつかの追加ステップを実行する必要があります。

#### Note

リンクのクリックを追跡するには、HTTPS ドメインのみ使用できます。Amazon SES は HTTP ドメイン経由のオープンの追跡のみをサポートしています。

クリックを処理するための HTTPS サブドメインを設定するには

1. クリックの追跡リンクに使用するサブドメインを作成します。これらのリンクの処理専用のサブドメインを作成することをお勧めします。
2. Amazon SES で使用するサブドメインを検証します。詳細については、「[Amazon SES でのドメインの検証 \(p. 61\)](#)」を参照してください。
3. [Amazon CloudFront](#) などのコンテンツ配信ネットワーク (CDN) を使用して新しいアカウントを作成します。
4. リクエストを Amazon SES 追跡ドメインに転送するように CDN を設定します。リダイレクトするアドレスは、Amazon SES を使用する AWS リージョンによって異なります。次の表には、Amazon SES が利用可能な AWS リージョンの追跡ドメインの一覧が含まれます。

AWS リージョン	AWS 追跡ドメイン
米国東部 (オハイオ)	r.us-east-2.awstrack.me
米国東部 (バージニア北部)	r.us-east-1.awstrack.me
米国西部 (オレゴン)	r.us-west-2.awstrack.me
アジアパシフィック (ムンバイ)	r.ap-south-1.awstrack.me
アジアパシフィック (ソウル)	r.ap-northeast-2.awstrack.me
アジアパシフィック (シンガポール)	r.ap-southeast-1.awstrack.me
アジアパシフィック (シドニー)	r.ap-southeast-2.awstrack.me
アジアパシフィック (東京)	r.ap-northeast-1.awstrack.me
カナダ (中部)	r.ca-central-1.awstrack.me

AWS リージョン	AWS 追跡ドメイン
欧州 (フランクフルト)	r.eu-central-1.awstrack.me
欧州 (アイルランド)	r.eu-west-1.awstrack.me
欧州 (ロンドン)	r.eu-west-2.awstrack.me
南米 (サンパウロ)	r.sa-east-1.awstrack.me
AWS GovCloud (US)	r.us-gov-west-1.awstrack.me

- Amazon CloudFront を CDN として使用している場合、次の手順を実行してください。
  - [CloudFront Distributions] ページで、CDN に対応するディストリビューションを選択します。
  - [Behaviors] タブでデフォルトの動作を選択してから、[Edit] を選択します。
  - [Cache Based on Selected Request Headers] で、[All] を選択します。
  - [Query String Forwarding and Caching] で、[Forward all, cache based on all] を選択します。
  - ディストリビューションに代替ドメイン名を追加します。使用するサブドメインは Amazon SES で確認する必要があります。詳細については、Amazon CloudFront デベロッパーガイドの「[代替ドメイン名とHTTPSを使用する](#)」を参照してください。

CloudFront 以外の CDN を使用する場合は、同様のステップを実行する必要があります。詳細については、CDN のドキュメントを参照してください。

- Route 53 を使用して、ドメインの DNS 設定を管理し、CDN として CloudFront を管理する場合、Route 53 で CloudFront ディストリビューションを参照するエイリアスレコードを作成します (d1111111abcdef8.cloudfront.net など)。詳細については、Amazon Route 53 デベロッパーガイドの「[Amazon Route 53 コンソールを使用したレコードの作成](#)」を参照してください。

それ以外の場合、サブドメインの DNS 設定で、CDN のアドレスを参照する CNAME レコードを追加します。

- 信頼できる認証機関から SSL 証明書を取得します。証明書は、ステップ 1 で作成したサブドメインと、ステップ 3 ~ 5 で設定した CDN の両方をカバーする必要があります。証明書を CDN にアップロードします。

## パート 2: オープンとクリックのカスタム追跡ドメインを参照するための設定セットのセットアップ

オープンとクリックの追跡リダイレクトを処理するようにドメインを設定した後、カスタムドメインを参照するように設定セットでイベント発行先を設定する必要があります。このステップは Amazon SES コンソール、または `CreateConfigurationSetTrackingOptions` API オペレーションを使用して完了できます。このセクションには、Amazon SES コンソールを使用してこれらのタスクを完了する手順が含まれます。API の使用に関する詳細については、[Amazon Simple Email Service API リファレンス](#)の「[CreateConfigurationSetTrackingOptions](#)」を参照してください。

カスタム追跡ドメインを参照する新しい設定セットイベント発行先を作成するには

- AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
- 画面左側のナビゲーションバーで、[Configuration Sets] を選択します。
- [Create Configuration Set] を選択します。
- [Configuration Set Name] に、設定セットの名前を入力し、[Create Configuration Set] を選択します。
- 設定セットのリストで、前のステップで作成した設定セットの横にあるボックスを選択します。[Actions] メニューで、[Edit] を選択します。

6. [Event Destinations] の [Add Destination] で、イベント発行先タイプを選択します。このメニューのオプションの詳細については、「[ステップ 2: イベント送信先を追加する \(p. 299\)](#)」を参照してください。
7. [Event types] で、追跡するイベントのタイプに応じて、[Click]、[Open]、またはその両方を選択します。
8. [Domain] で、[Use your own subdomain] を選択します。
9. [Select a verified domain] で、オープンとクリックのイベント追跡に使用するドメインを選択します。メニューの左側のテキストフィールドには、親ドメインのサブドメインを指定することもできます。
10. 通常どおり、残りのオプションを設定します。イベント発行先のセットアップの詳細については、「[ステップ 2: イベント送信先を追加する \(p. 299\)](#)」を参照してください。
11. [Save] を選択します。

## Amazon SES を用いた VPC エンドポイントの設定

Amazon SES のお客様の多くは、会社のポリシーによって、内部システムからパブリックインターネットに接続することを制限されています。そのため、これらのお客様は、パブリック Amazon SES エンドポイントを使用することができません。

これらの制限内で作業するために、Amazon Virtual Private Cloud (Amazon VPC) を使用することができます。Amazon VPC を使うと、AWS クラウドの分離されたエリアに存在する仮想ネットワークに AWS リソースをデプロイできます。Amazon VPC の詳細については、[Amazon VPC ユーザーガイド](#)を参照してください。

Amazon VPC で Amazon SES を使用するには、まず組織の VPC に Amazon EC2 インスタンスを作成する必要があります。続いてそのインスタンスに接続し、Amazon SES を介した E メール送信にそのインスタンスを使用します。このセクションでは、Amazon EC2 インスタンスを設定し、Amazon SES 用の Amazon VPC エンドポイントを作成する手順について説明します。

### Note

Amazon Elastic Compute Cloud (Amazon EC2) では、デフォルトでポート 25 経由での E メールトラフィックを制限しています。Amazon EC2 から SMTP エンドポイントを介して E メールを送信する際のタイムアウトを回避するには、これらの制限を解除するようリクエストします。詳細については、AWS ナレッジセンターの「[Amazon EC2 インスタンスまたは AWS Lambda 関数からポート 25 の制限を解除する方法](#)」を参照してください。別のポート (587 や 2587 など) を使用して、この問題を回避することもできます。

## Prerequisites

このセクションの手順を実行する前に、以下の手順を完了してください。

- 仮想プライベートクラウドを作成する。手順については、「[Amazon VPC の IPv4 の使用開始](#)」を参照してください。
- VPC 内の Amazon EC2 インスタンスの起動。詳細については、「[EC2 インスタンスをデフォルト VPC 内に起動する](#)」を参照してください。

## Amazon VPC での Amazon SES のセットアップ

Amazon SES で使用する VPC エンドポイントをセットアップするプロセスは、いくつかのステップに分かれています。まず、VPC エンドポイントで使用する Amazon EC2 インスタンスのプライベート IP アドレスを特定する必要があります。次に、セキュリティグループを作成してインスタンスが SMTP ポートと通信できるようにします。その後、Amazon SES の VPC エンドポイントを作成します。最後に、VPC エンドポイントへの接続をテストして、適切に設定されていることを確認します。

## ステップ 1: Amazon EC2 インスタンスのプライベート IP アドレスを特定する

Amazon SES VPC エンドポイントを使用する Amazon EC2 インスタンスをセットアップするには、まずインスタンスのプライベート IP を見つける必要があります。この IP アドレスは、後のステップで使用します。

Amazon EC2 インスタンスのプライベート IP を見つけるには

1. Amazon EC2 コンソール (<https://console.aws.amazon.com/ec2/>) を開きます。
2. ナビゲーションペインで、[Instances] の下にある [Instances] を選択します。
3. Amazon EC2 インスタンスのリストで、VPC エンドポイントへの接続に使用するインスタンスを選択します。
4. 画面の下部にある詳細ペインの [説明] タブで、[Private IP] の横にある IP アドレスをコピーします。

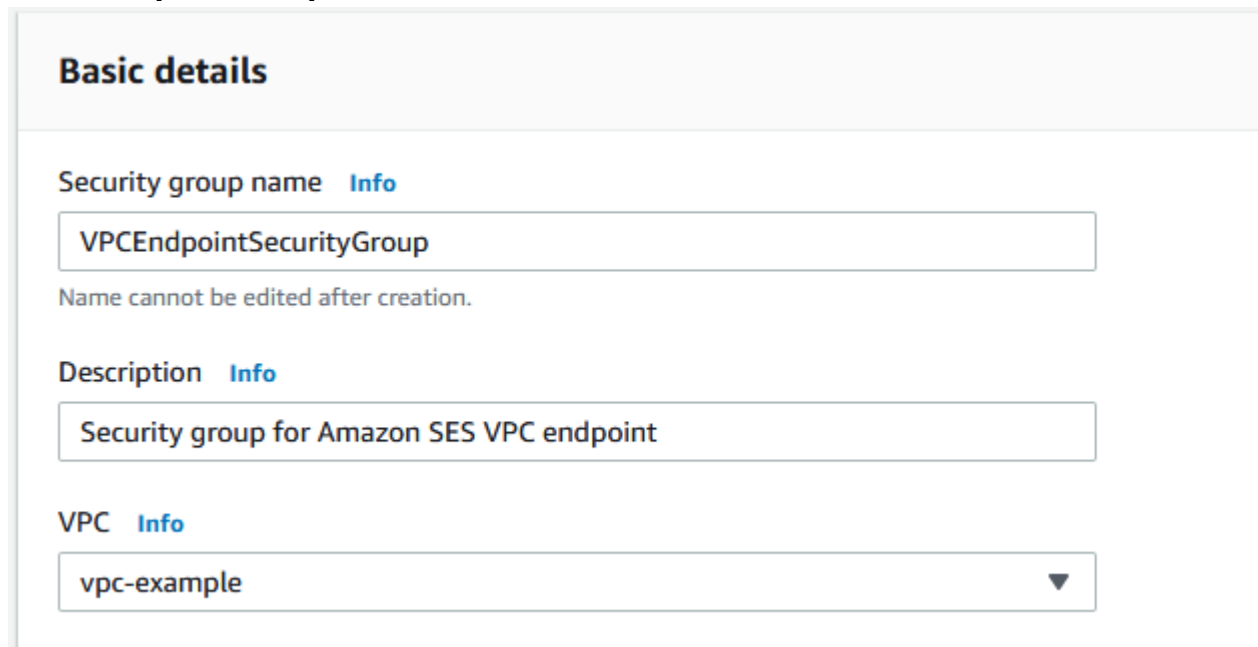
## ステップ 2: セキュリティグループの作成

Amazon EC2 では、セキュリティグループを使用して、VPC との間のインバウンドおよびアウトバウンドの通信を制御できます。このステップでは、Amazon EC2 インスタンスが SMTP エンドポイントと通信できるように、セキュリティグループを作成します。

セキュリティグループを作成するには

1. Amazon EC2 コンソールのナビゲーションペインで、[ネットワークとセキュリティ] の下にある [セキュリティグループ] を選択します。
2. [セキュリティグループの作成] を選択します。
3. [基本的な詳細] で、次の操作を行います。
  - [セキュリティグループ名] に、セキュリティグループの一意的な名前を入力します。
  - [説明] に、セキュリティグループの目的を説明するテキストを入力します。
  - [VPC] で、Amazon SES で使用する VPC を選択します。

完了すると、[基本的な詳細] セクションは次のイメージの例のようになります。



**Basic details**

Security group name [Info](#)

VPCEndpointSecurityGroup

Name cannot be edited after creation.

Description [Info](#)

Security group for Amazon SES VPC endpoint

VPC [Info](#)

vpc-example ▼

4. [インバウンドルール] で、[ルールの追加] を選択します。
5. [インバウンドルール 1] で、次の操作を行います。
  - [タイプ] で [カスタム TCP] を選択します。
  - [ポート範囲] に、E メール送信に使用するポート番号を入力します。25、465、587、2465、2587 のいずれかのポート番号が使用できます。
  - [Source タイプ] で、[Custom] を選択します。
  - [送信元] に、Amazon EC2 インスタンスのプライベート IP (先ほど見つけたアドレス) を入力します。
6. (オプション) その他のポートにインバウンドルールを追加する場合は、もう一度 [ルールの追加] を選択します。その後、上記のステップを繰り返してポートを追加します。前のステップでリストしたポート番号の一部に対してルールを作成することも、全部に対してルールを作成することもできます。
7. 完了したら、[Create security group] を選択します。

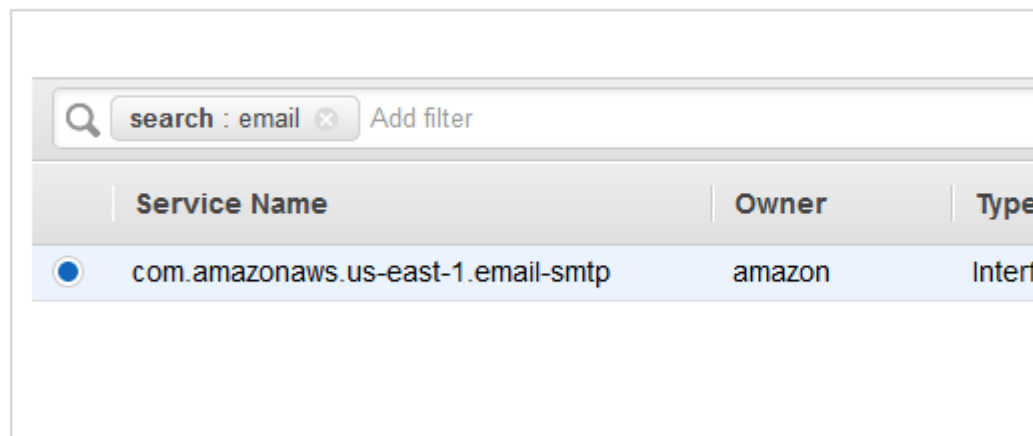
### ステップ 3: VPC エンドポイントの作成

Amazon VPC では、VPC エンドポイントを使用して、サポートされている AWS サービスに VPC を接続できます。この場合、Amazon EC2 セキュリティグループが Amazon SES に接続できるように、Amazon VPC を設定します。

#### VPC エンドポイントを作成する

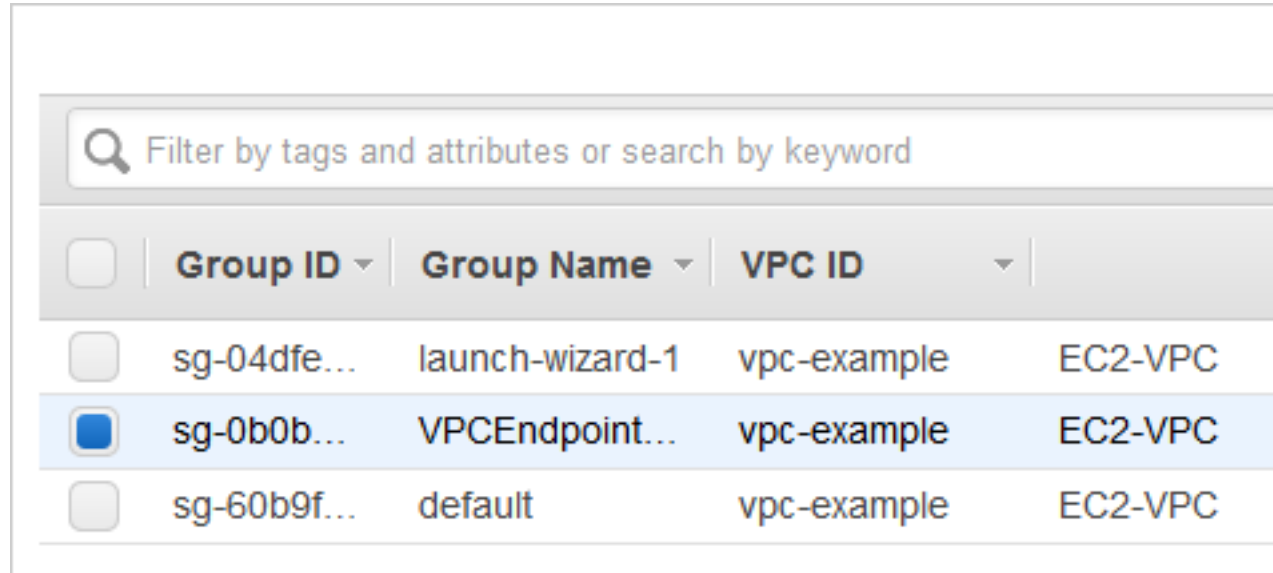
1. Amazon VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. [仮想プライベートクラウド] で、[エンドポイント] を選択します。
3. [] を選択します。
4. [エンドポイントの作成] ページの [サービスカテゴリ] で [AWS サービス] を選択します。
5. 次の図に示すように、[サービス名] の検索ボックスを使用して「email」を検索します。

**Service Name** `com.amazonaws.us-east-1.email-smtp` ⓘ

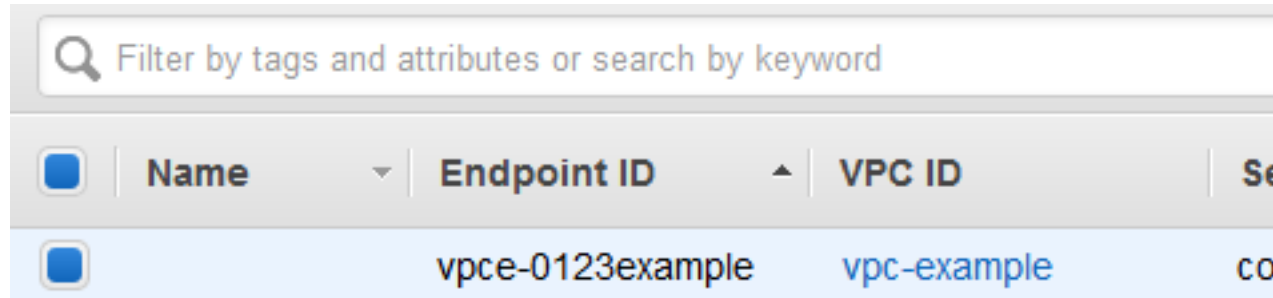


現在の AWS リージョンの email-smtp サービスを選択します。

6. [VPC] で、使用する仮想プライベートクラウドを選択します。
7. 次の図に示すように、[セキュリティグループ] で先ほど作成したセキュリティグループを選択します。



8. [Create endpoint (エンドポイントの作成)] を選択します。Amazon VPC がエンドポイントを作成している間、5 分ほど待ちます。エンドポイントの準備ができたら、次の図に示すように、[ステータス] 列の値が「available」に変わります。



#### ステップ 4: VPC エンドポイントへの接続をテストする

VPC エンドポイントの設定プロセスが完了したら、接続をテストして、VPC エンドポイントが正しく設定されていることを確認します。接続のテストには、ほとんどのオペレーティングシステムに用意されているコマンドラインツールを使用できます。

VPC エンドポイントへの接続をテストするには

1. Amazon EC2 インスタンスに接続します。

Linux インスタンスに接続する方法の詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「[Linux インスタンスへの接続](#)」を参照してください。

Windows インスタンスへの接続の詳細については、Windows インスタンス用の Amazon EC2 ユーザーガイドの「[開始方法](#)」を参照してください。

2. [コマンドラインを使用し、Amazon SES SMTP インターフェイスを介して E メールを送信する \(p. 109\)](#) の手順を実行して、テストメールを送信します。

#### Note

Amazon SES 経由で E メールを送信する前に、メールアドレスまたはドメインを検証する必要があります。検証の方法については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。

## Amazon SES での Eメールの送信方法

Amazon Simple Email Service (Amazon SES) では、Amazon SES コンソール、Amazon SES Simple Mail Transfer Protocol (SMTP) インターフェイス、または Amazon SES API を使用して E メールを送信することができます。通常、テスト E メールを送信して送信アクティビティを管理する場合は、コンソールを使用します。一括 E メールを送信する場合には、SMTP インターフェイスまたは API を使用します。Amazon SES の料金に関する詳細については、「[Amazon SES の料金](#)」を参照してください。

- SMTP 対応のソフトウェアパッケージ、アプリケーション、またはプログラミング言語を使用し、Amazon SES を介して E メールを送信する場合や、Amazon SES を既存のメールサーバーに統合する場合には、Amazon SES SMTP インターフェイスを使用します。詳細については、「[Amazon SES SMTP インターフェイスを使用した Eメールの送信 \(p. 88\)](#)」を参照してください。
- raw HTTP リクエストを使用して、Amazon SES を呼び出す場合には、Amazon SES API を使用します。詳細については、「[Amazon SES API を使用して Eメールを送信する \(p. 113\)](#)」を参照してください。

Eメールを送信する前に、「[Amazon SES での Eメールのセットアップ \(p. 47\)](#)」を参照してください。

#### Important

複数の受取人 ( 「To」 、 「CC」 、 「BCC」 の各アドレスで受取人が指定されている ) にメールを送信する場合、Amazon SES の呼び出しに失敗すると、Eメール全体が拒否されて、どの受信者も目的のメールを受信できません。そのため、1回につき1人の受信者にEメールを送信することをお勧めします。

## Amazon SES SMTP インターフェイスを使用した Eメールの送信

Amazon SES を介して本稼働 E メールを送信する場合には、Simple Mail Transfer Protocol ( SMTP ) インターフェイスまたは Amazon SES API を使用できます。Amazon SES API の詳細については、「[Amazon SES API を使用して Eメールを送信する \(p. 113\)](#)」を参照してください。このセクションでは、SMTP インターフェイスについて説明します。

Amazon SES は、インターネットで最も一般的な E メールプロトコルである SMTP を使用してメールを送信します。SMTP 対応の各種プログラミング言語やソフトウェアを使用して Amazon SES SMTP インターフェイスに接続することで、Amazon SES を介して E メールを送信できます。このセクションでは、Amazon SES を介して E メールを送信するために、Amazon SES の SMTP 認証情報を取得する方法、SMTP インターフェイスを使用して E メールを送信する方法、および、各種のソフトウェアおよびメールサーバーの設定方法を説明します。

#### Note

SMTP インターフェイスを介した Amazon SES の使用に伴う一般的な問題の解決策については、「[Amazon SES SMTP の問題 \(p. 489\)](#)」を参照してください。

Amazon SES SMTP インターフェイスを使用して E メールを送信するには、次の情報が必要です。



- AWS アカウント。詳細については、「[AWS にサインアップする \(p. 48\)](#)」を参照してください。
- SMTP エンドポイントアドレス。SMTP エンドポイントの Amazon SES リストについては、[Amazon SES SMTP エンドポイントへの接続 \(p. 94\)](#) を参照してください。
- SMTP インターフェイスのポート番号。ポート番号は接続方法によって変わります。詳細については、「[Amazon SES SMTP エンドポイントへの接続 \(p. 94\)](#)」を参照してください。
- SMTP ユーザー名とパスワード。SMTP 認証情報は、各 AWS リージョンに固有です。複数の AWS リージョンで SMTP インターフェイスを使用して E メールを送信する予定がある場合は、リージョンごとにユーザー名とパスワードが必要になります。

#### Important

SMTP ユーザー名とパスワードは、Amazon SES コンソールにサインインするときに使用する AWS アクセスキーや認証情報と同じではありません。SMTP ユーザー名とパスワードを生成する方法については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

- Transport Layer Security (TLS) を使用して通信できるクライアントソフトウェア。詳細については、「[Amazon SES SMTP エンドポイントへの接続 \(p. 94\)](#)」を参照してください。
- Amazon SES で検証済みの E メールアドレス。詳細については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。
- 大量の E メールを送信する場合は、送信クォータを引き上げが必要です。詳細については、「[Amazon SES 送信クォータの管理 \(p. 149\)](#)」を参照してください。

次に、以下を実行することにより、E メールを送信できます：

- SMTP 対応ソフトウェアを、Amazon SES SMTP インターフェイスを介して E メールを送信するように設定するには、「[ソフトウェアパッケージを使用し、Amazon SES を介して E メールを送信する \(p. 94\)](#)」を参照してください。
- Amazon SES を介して E メールを送信するようにアプリケーションをプログラムするには、「[アプリケーションから Amazon SES を介して E メールを送信する \(p. 96\)](#)」を参照してください。
- Amazon SES を介してすべての送信 E メールを送信するために、既存の E メールサーバーを設定する方法については、「[Amazon SES を既存の E メールサーバーと統合する \(p. 96\)](#)」を参照してください。
- コマンドラインを使用して、Amazon SES SMTP インターフェイスを操作する (テストを行うときに有効) 方法については、「[コマンドラインを使用して Amazon SES SMTP インターフェイスへの接続をテストする \(p. 107\)](#)」を参照してください。

SMTP 応答コードのリストについては、「[Amazon SES から返される SMTP 応答コード \(p. 490\)](#)」を参照してください。

## 提供する E メール情報

SMTP インターフェイスで Amazon SES にアクセスする場合、SMTP クライアントアプリケーションでメッセージがアセンブルされるため、提供する必要がある情報は使用するアプリケーションによって異なります。少なくとも、クライアントとサーバー間の SMTP 交換には、送信元アドレス、宛先アドレス、およびメッセージデータが必要です。

SMTP インターフェイスを使用していて、フィードバック転送が有効になっている場合、バウンス、苦情、配信通知は "MAIL FROM" アドレスに送信されます。指定した "Reply-To" アドレスはいずれも使用されません。

## Amazon SES SMTP 認証情報を取得

Amazon SES SMTP インターフェイスにアクセスするには、Amazon SES SMTP のユーザー名とパスワードが必要です。

Amazon SES SMTP インターフェイスを介して E メールを送信するために使用する認証情報は、各 AWS リージョンに固有のもので、Amazon SES SMTP インターフェイスを使用して複数のリージョンに E メールを送信する場合は、使用しようとしている各リージョンで [SMTP 認証情報のセットを生成 \(p. 89\)](#)する必要があります。

SMTP パスワードは、AWS シークレットアクセスキーとは異なります。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。

## Amazon SES コンソールを使用して Amazon SES SMTP 認証情報を取得

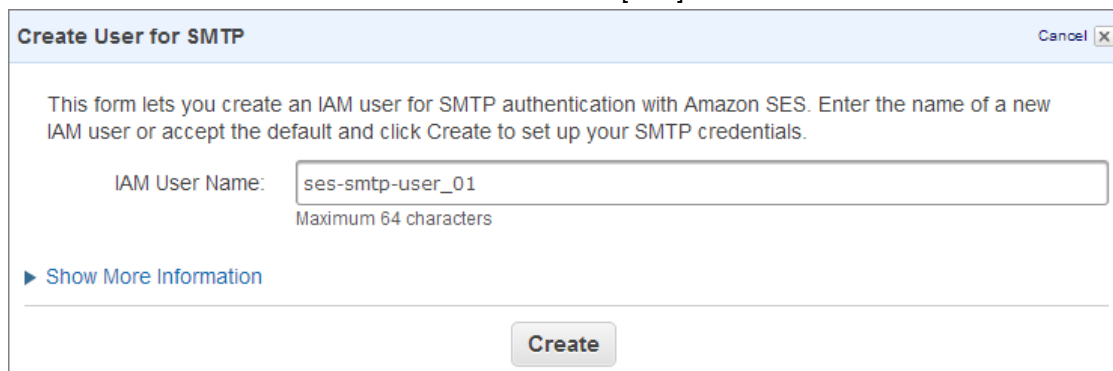
Amazon SES コンソールを使用して SMTP 認証情報を生成する場合、Amazon SES コンソールは、Amazon SES を呼び出すための適切なポリシーを持った IAM ユーザーを作成した上で、そのユーザーに関連付けられている SMTP 認証情報を提示します。

### Requirement

IAM ユーザーは Amazon SES SMTP 認証情報を作成できます。ただし、Amazon SES SMTP 認証情報の作成には IAM を使用するため、IAM ユーザーのポリシーによって IAM 自体を使用するアクセス許可をユーザーに付与する必要があります。IAM ポリシーで許可する必要がある IAM アクションは、iam:ListUsers、iam:CreateUser、iam:CreateAccessKey、および iam:PutUserPolicy です。Amazon SES SMTP 認証情報をコンソールで作成する場合、これらのアクセス許可が IAM ユーザーにないと、アカウントに「iam:ListUsers を実行する権限がない」というエラーメッセージが表示されます。

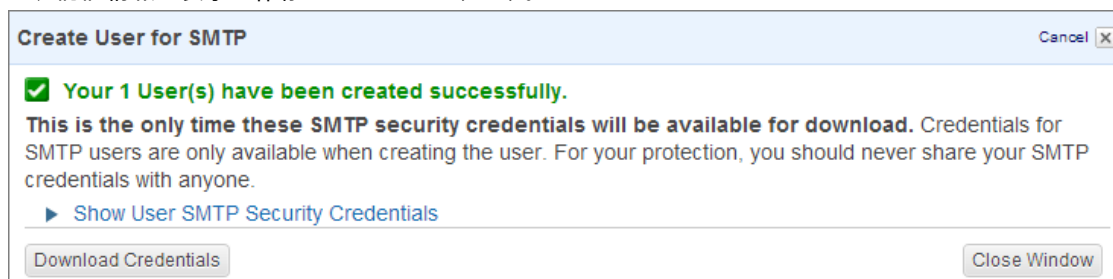
SMTP 認証情報を作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで [SMTP Settings] を選択します。
3. コンテンツペインで、[Create My SMTP Credentials] を選択します。
4. [SMTP のユーザーを作成] で、SMTP ユーザーの名前を入力します。または、このフィールドに提供されているデフォルト値を使用できます。完了したら、[作成] を選択します。



The screenshot shows a dialog box titled "Create User for SMTP" with a "Cancel" button in the top right. The main text reads: "This form lets you create an IAM user for SMTP authentication with Amazon SES. Enter the name of a new IAM user or accept the default and click Create to set up your SMTP credentials." Below this is a text input field for "IAM User Name" containing "ses-smtp-user\_01" and a note "Maximum 64 characters". There is a "Show More Information" link and a "Create" button at the bottom.

5. [Show User SMTP Credentials] を選択します。SMTP 認証情報が画面に表示されます。これらの認証情報をダウンロードまたはコピーして、安全な場所に保存します。このダイアログボックスを閉じると、認証情報の表示や保存はできなくなります。



The screenshot shows the same dialog box after successful creation. A green checkmark icon is followed by the text: "Your 1 User(s) have been created successfully." Below this is a warning: "This is the only time these SMTP security credentials will be available for download. Credentials for SMTP users are only available when creating the user. For your protection, you should never share your SMTP credentials with anyone." There is a "Show User SMTP Security Credentials" link and a "Download Credentials" button at the bottom left, and a "Close Window" button at the bottom right.

6. [Close Window] を選択します。

この手順で作成した既存の SMTP 認証情報を一覧表示するには、IAM コンソール (<https://console.aws.amazon.com/iam/>) に移動します。ナビゲーションペインの [アクセス管理] で、[ユーザー] を選択します。検索バーを使用して、SMTP 認証情報を割り当てたすべてのユーザーを検索します。

IAM コンソールを使用して、既存の SMTP ユーザーを削除することもできます。ユーザーの削除についての詳細は、IAM 使用開始ガイドの <https://docs.aws.amazon.com/IAM/latest/UserGuide/Managing IAM Users> を参照してください。

SMTP 認証情報をローテーションする場合は、上の手順を実行して、新しい SMTP 認証情報のセットを生成します。次に、新しい認証情報をテストして、期待どおりに機能することを確認します。最後に、IAM コンソールで古い SMTP 認証情報に関連付けられた IAM ユーザーを削除します。IAM でのユーザーの削除についての詳細は、IAM ユーザーガイドの「[ユーザーの管理](#)」を参照してください。

## 既存の Amazon AWS 認証情報を変換して Amazon SES SMTP 認証情報を取得する

IAM インターフェイスを使用して IAM ユーザーを設定した場合、ユーザーの Amazon SES SMTP 認証情報は AWS 認証情報から派生できます。

### Important

一時的な AWS 認証情報を使用して、SMTP 認証情報を派生させないでください。Amazon SES SMTP インターフェイスは、一時的なセキュリティ認証情報から生成された SMTP 認証情報をサポートしていません。

IAM ユーザーが Amazon SES SMTP インターフェイスを使用して E メールを送信できるようにするには、次の 2 つのステップを実行する必要があります。

- SMTP 認証情報を AWS 認証情報から派生させるには、このセクションで示すアルゴリズムを使用します。AWS 認証情報から開始するため、SMTP ユーザー名は AWS アクセスキー ID と同じです。したがって、生成する必要があるのは SMTP パスワードのみです。
- IAM ユーザーに次のポリシーを適用します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ses:SendRawEmail",
      "Resource": "*"
    }
  ]
}
```

IAM での Amazon SES 使用の詳細については、「[Amazon SES へのアクセスのコントロール \(p. 406\)](#)」を参照してください。

### Note

Amazon SES SMTP 認証情報はどの IAM ユーザーに対しても生成できますが、SMTP 認証情報用に別個の IAM ユーザーを作成することをお勧めします。目的別にユーザーを作成することが推奨される理由については、「[IAM のベストプラクティス](#)」を参照してください。

次の擬似コードでは、AWS シークレットアクセスキーを Amazon SES SMTP パスワードに変換するアルゴリズムを示します。

```
// Modify this variable to include your AWS secret access key
key = "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY";
```

```
// Modify this variable to refer to the AWS Region that you want to use to send email.
region = "us-west-2";

// The values of the following variables should always stay the same.
date = "11111111";
service = "ses";
terminal = "aws4_request";
message = "SendRawEmail";
version = 0x04;

kDate = HmacSha256(date, "AWS4" + key);
kRegion = HmacSha256(region, kDate);
kService = HmacSha256(service, kRegion);
kTerminal = HmacSha256(terminal, kService);
kMessage = HmacSha256(message, kTerminal);
signatureAndVersion = Concatenate(version, kMessage);
smtpPassword = Base64(signatureAndVersion);
```

一部のプログラミング言語に含まれているライブラリを使用して、IAM シークレットアクセスキーを SMTP パスワードに変換できます。このセクションのコード例では、Python を使用して AWS シークレットアクセスキーを Amazon SES SMTP パスワードに変換します。

#### Note

次の例では、Python 3.6で導入された f 文字列を使用しています。古いバージョンでは使用できません。

現在、Python SDK (Boto3) は公式に 2.7 と 3.6 (またはそれ以降) をサポートしています。ただし、2.7 のサポートは廃止され、2021 年 7 月 15 日に削除されるため、少なくとも 3.6 にアップグレードする必要があります。

#### Python

```
#!/usr/bin/env python3

import hmac
import hashlib
import base64
import argparse

SMTP_REGIONS = [
    'us-east-2',      # US East (Ohio)
    'us-east-1',     # US East (N. Virginia)
    'us-west-2',     # US West (Oregon)
    'ap-south-1',    # Asia Pacific (Mumbai)
    'ap-northeast-2', # Asia Pacific (Seoul)
    'ap-southeast-1', # Asia Pacific (Singapore)
    'ap-southeast-2', # Asia Pacific (Sydney)
    'ap-northeast-1', # Asia Pacific (Tokyo)
    'ca-central-1',  # Canada (Central)
    'eu-central-1',  # Europe (Frankfurt)
    'eu-west-1',     # Europe (Ireland)
    'eu-west-2',     # Europe (London)
    'sa-east-1',     # South America (Sao Paulo)
    'us-gov-west-1', # AWS GovCloud (US)
]

# These values are required to calculate the signature. Do not change them.
DATE = "11111111"
SERVICE = "ses"
MESSAGE = "SendRawEmail"
TERMINAL = "aws4_request"
VERSION = 0x04
```

```
def sign(key, msg):
    return hmac.new(key, msg.encode('utf-8'), hashlib.sha256).digest()

def calculate_key(secret_access_key, region):
    if region not in SMTP_REGIONS:
        raise ValueError(f"The {region} Region doesn't have an SMTP endpoint.")

    signature = sign(("AWS4" + secret_access_key).encode('utf-8'), DATE)
    signature = sign(signature, region)
    signature = sign(signature, SERVICE)
    signature = sign(signature, TERMINAL)
    signature = sign(signature, MESSAGE)
    signature_and_version = bytes([VERSION]) + signature
    smtp_password = base64.b64encode(signature_and_version)
    return smtp_password.decode('utf-8')

def main():
    parser = argparse.ArgumentParser(
        description='Convert a Secret Access Key for an IAM user to an SMTP password.')
    parser.add_argument(
        'secret', help='The Secret Access Key to convert.')
    parser.add_argument(
        'region',
        help='The AWS Region where the SMTP password will be used.',
        choices=SMTP_REGIONS)
    args = parser.parse_args()
    print(calculate_key(args.secret, args.region))

if __name__ == '__main__':
    main()
```

このスクリプトを使用して SMTP パスワードを取得するには、上記のコードを `smtp_credentials_generate.py` として保存します。次に、コマンドラインで以下のコマンドを実行します。

```
python path/to/smtp_credentials_generate.py wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY us-east-1
```

上記のコマンドで、次の操作を行います。

- `path/to/` を保存した場所へのパスに置き換えます `smtp_credentials_generate.py`。
- `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` を SMTP パスワードに変換するシークレットアクセスキーに置き換えます。
- `us-east-1` を SMTP 認証情報を使用する AWS リージョンに置き換えます。

このスクリプトが正常に実行されると、SMTP パスワードだけが出力されます。

このスクリプトを使用するには、まず前のコードを `smtp_credentials_generate.py` として保存します。次に、コマンドラインで以下のコマンドを実行します。

```
python path/to/smtp_credentials_generate.py wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY us-
east-1
```

上記のコマンドで、次の操作を行います。

- `path/to/` を保存した場所へのパスに置き換えます `smtp_credentials_generate.py`。

- `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY` を SMTP パスワードに変換するシークレットアクセスキーに置き換えます。
- `us-east-1` を SMTP 認証情報を使用する AWS リージョンに置き換えます。

このスクリプトが正常に実行されると、SMTP パスワードだけが出力されます。

## Amazon SES SMTP エンドポイントへの接続

Amazon SES SMTP インターフェイスを使用して E メールを送信するには、SMTP エンドポイントに接続します。Amazon SES SMTP エンドポイントの完全なリストについては、AWS 全般のリファレンスの「[Amazon Simple Email サービスエンドポイントとクォータ](#)」を参照してください。

Amazon SES SMTP エンドポイントでは、すべての接続が Transport Layer Security ( TLS ) を使用して暗号化されている必要があります。(TLS は、以前のプロトコルの名前である「SSL」と呼ばれることが多いことに注意してください。) Amazon SES は、TLS で暗号化された接続を確立するために、STARTTLS および TLS ラッパーという 2 つのメカニズムをサポートしています。ソフトウェアが STARTTLS および TLS ラッパーをサポートしているかどうかは、ソフトウェアのドキュメントを参照してください。

### Note

Amazon Elastic Compute Cloud (Amazon EC2) では、デフォルトでポート 25 経由での E メールトラフィックを制限しています。Amazon EC2 から SMTP エンドポイントを介して E メールを送信する際のタイムアウトを回避するには、これらの制限を解除するようリクエストします。詳細については、AWS ナレッジセンターの「[Amazon EC2 インスタンスまたは AWS Lambda 関数からポート 25 の制限を解除する方法](#)」を参照してください。または、別のポートを使用するか、[Amazon VPC エンドポイント \(p. 84\)](#)を使用することもできます。

## STARTTLS

STARTTLS とは、暗号化されていない接続を暗号化された接続にアップグレードする方法です。STARTTLS には、様々なプロトコルに対応したバージョンがあります。SMTP バージョンは、「[RFC 3207](#)」に定義されています。

STARTTLS 接続を設定する場合、SMTP クライアントは、ポート 25、587、または 2587 で Amazon SES SMTP エンドポイントに接続し、EHLO コマンドを発行します。次に、サーバーから STARTTLS SMTP 拡張機能をサポートしているという通知が来るのを待ちます。通知を受けたクライアントは、STARTTLS コマンドを発行し、TLS ネゴシエーションを開始します。ネゴシエーションが完了すると、クライアントが暗号化された新しい接続で EHLO コマンドを発行し、SMTP セッションが正常に進行します。

## TLS ラッパー

TLS ラッパー ( SMTPS またはハンドシェイクプロトコルとも呼ばれる ) は、最初に暗号化されていない接続を確立するのではなく、最初から暗号化された接続を開始する方法です。TLS ラッパーを使用する場合、Amazon SES SMTP エンドポイントは TLS ネゴシエーションを実行しません。TLS を使用してエンドポイントに接続し、通信全体で TLS の使用を継続するのはクライアントの役割です。TLS ラッパーは古いプロトコルですが、数多くのクライアントが今もサポートしています。

TLS ラッパー接続を設定する場合、SMTP クライアントは、Amazon SES SMTP エンドポイントにポート 465 または 2465 で接続します。サーバーが自身の証明書を提示すると、クライアントが EHLO コマンドを発行し、SMTP セッションが正常に進行します。

## ソフトウェアパッケージを使用し、Amazon SES を介して E メールを送信する

SMTP を介した E メール送信をサポートしている市販/オープンソースのソフトウェアパッケージは多数あります。次に例を示します。

- ブログプラットフォーム
- RSS アグリゲータ
- リスト管理ソフトウェア
- ワークフローシステム

上記の SMTP 対応ソフトウェアは、Amazon SES SMTP インターフェイスを介して E メールを送信するように設定できます。個々のソフトウェアパッケージの SMTP 設定手順については、各ソフトウェアのドキュメントを参照してください。

次の手順は、一般的な問題追跡ソリューションである JIRA で Amazon SES 送信を設定する方法を示しています。この設定により、ソフトウェアの問題のステータスが変化したときに JIRA からユーザーに E メールで通知できるようになります。

Amazon SES を介して E メールを送信するために、JIRA を設定するには

1. ウェブブラウザを使用し、管理者認証情報で JIRA にログインします。
2. ブラウザのウィンドウで、[Administration] を選択します。
3. System メニューで、[Mail] を選択します。
4. [Mail administration] ページで、[Mail Servers] を選択します。
5. [Configure new SMTP mail server] を選択します。
6. [Add SMTP Mail Server] フォームで、次のフィールドに入力します。
  - a. Name - このサーバーの名前 (わかりやすいもの)。
  - b. From address - Eメールの送信元アドレス。この E メールアドレスから送信する前に、Amazon SES でこの E メールアドレスを検証する必要があります。検証の詳細については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。
  - c. Email prefix - 送信前に、件名行の前に JIRA が付加する文字列。
  - d. Protocol - [SMTP] を選択します。

#### Note

この設定を使用して Amazon SES に接続できない場合は、SECURE\_SMTP を試してください。

- e. ホスト名 - Amazon SES SMTP エンドポイントのリストについては、「[Amazon SES SMTP エンドポイントへの接続 \(p. 94\)](#)」を参照してください。たとえば、米国西部 (オレゴン) リージョンで Amazon SES エンドポイントを使用する場合、ホスト名は email-smtp.us-west-2.amazonaws.com です。
- f. SMTP Port - 25、587、2587 のいずれか (STARTTLS を使用して接続する場合)、または 465、2465 のいずれか (TLS ラッパーを使用して接続する場合)。
- g. TLS - このチェックボックスをオンにします。
- h. Username - SMTP ユーザー名。
- i. Password - SMTP パスワード。

TLS ラッパーの設定を次に示します。

The screenshot shows the 'Update SMTP Mail Server' page in the JIRA administration interface. The page title is 'Mail' and the sub-page is 'Update SMTP Mail Server'. The main content area contains the following fields and options:

- Name \***: Amazon SES (The name of this server within JIRA.)
- Description**: (Empty text field)
- From address \***: bob@example.com (The default address this server will use to send emails from.)
- Email prefix \***: JIRA (This prefix will be prepended to all outgoing email subjects.)
- Server Details**: Enter either the host name of your SMTP server or the JNDI location of a javax.mail.Session object to use.
- SMTP Host**:
  - Protocol**: SMTP (Dropdown menu)
  - Host Name \***: .us-east-1.amazonaws.com (The SMTP host name of your mail server.)
  - SMTP Port**: 465 (Optional - SMTP port number to use. Leave blank for default (defaults: SMTP - 25, SMTPS - 465).)
  - Timeout**: 10000 (Timeout in milliseconds - 0 or negative values indicate infinite timeout. Leave blank for default (10000 mSecs).)
  - TLS**:  (Optional - the mail server requires the use of TLS security.)

7. [Test Connection] を選択します。Amazon SES を介して JIRA が送信したテスト E メールが正しく到着すれば、設定は完了です。

## アプリケーションから Amazon SES を介して E メールを送信する

多くのプログラミング言語は、SMTP を使用した Eメールの送信をサポートしています。この機能は、プログラミング言語自体に組み込まれている場合もありますが、アドオン、プラグイン、またはライブラリとして利用できる場合もあります。この機能を利用することで、ユーザーが書き込むアプリケーションプログラム内から Amazon SES を介して Eメールを送信できます。

C#、および Java の例については、「はじめに」セクションの「[プログラミングで Amazon SES の SMTP インターフェイスにアクセスして Eメールを送信する \(p. 22\)](#)」を参照してください。

## Amazon SES を既存の Eメールサーバーと統合する

現在、Eメールサーバーをお客様が管理している場合には、Amazon SES SMTP エンドポイントを使用して、外部へのメールをすべて Amazon SES に送信できます。既存の Eメールクライアントや Eメールアプリケーションに変更を加える必要はありません。Amazon SES への転送は、これらのクライアントやアプリケーションに対して透過的です。

メール転送エージェント (MTA) の中には、SMTP リレーを介した Eメールの送信をサポートしているものもあります。このセクションでは、一般的ないくつかの MTA について、Amazon SES SMTP インターフェイスを使用して Eメールを送信するための設定方法について、一般的なガイダンスを提供します。

Amazon SES SMTP エンドポイントでは、すべての接続が Transport Layer Security (TLS) を使用して暗号化されている必要があります。

### トピック

- [Amazon SES と Postfix の統合 \(p. 97\)](#)
- [Amazon SES と Sendmail の統合 \(p. 100\)](#)



- [Amazon SES を Microsoft Windows Server IIS SMTP に統合する \(p. 104\)](#)
- [Amazon SES と Exim の統合 \(p. 105\)](#)

## Amazon SES と Postfix の統合

Postfix は、広く使用されている Sendmail Message Transfer Agent (MTA) に代わる手段です。Postfix の詳細については、「<http://www.postfix.org>」を参照してください。このトピックの手順では、Linux、macOS、または Unix で動作します。

### Note

Postfix はサードパーティー製アプリケーションであり、Amazon Web Services によって開発またはサポートされていません。このセクションの手順は情報提供のみを目的としており、予告なく変更される場合があります。

### Prerequisites

このセクションの手順を完了する前に、以下のタスクを実行する必要があります。

- システムに Sendmail がインストール済みである場合、これをアンインストールします。このステップを完了する手順は使用するオペレーティングシステムによって異なります。
- Postfix をインストールします。このステップを完了する手順は使用するオペレーティングシステムによって異なります。
- SASL 認証パッケージをインストールします。このステップを完了する手順は使用するオペレーティングシステムによって異なります。たとえば、RedHat ベースのシステムを使用している場合は、`cyrus-sasl-plain` パッケージをインストールする必要があります。Debian または Ubuntu ベースのシステムを使用している場合は、`libsasl2-modules` パッケージをインストールする必要があります。
- E メール送信に使用する E メールアドレスまたはドメインを確認します。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- アカウントがまだサンドボックスにある場合、検証済み E メールアドレスにのみ E メールを送信できます。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

### Postfix の設定

Postfix を使用して Amazon SES 経由で E メールを送信するようにメールサーバーを設定するには、次の手順を実行します。

Postfix を設定するには

1. コマンドラインから、以下のコマンドを入力します。

```
sudo postconf -e "relayhost = [email-smtp.us-west-2.amazonaws.com]:587" \  
"smtp_sasl_auth_enable = yes" \  
"smtp_sasl_security_options = noanonymous" \  
"smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd" \  
"smtp_use_tls = yes" \  
"smtp_tls_security_level = encrypt" \  
"smtp_tls_note_starttls_offer = yes"
```

### Note

米国西部 (オレゴン) 以外の AWS リージョンで Amazon SES を使用する場合は、`email-smtp.us-west-2.amazonaws.com` を、前述のコマンドで、該当するリージョンの SMTP エンドポイントに置き換えます。詳細については、「[リージョン \(p. 467\)](#)」を参照してください。

2. テキストエディタで、`/etc/postfix/master.cf` ファイルを開きます。次のエントリを検索します。

```
-o smtp_fallback_relay=
```

このエントリが見つかった場合は、行の先頭に `#` (ハッシュ) 文字を配置してコメントアウトします。保存してファイルを閉じます。

それ以外で、このエントリが存在しない場合、次の手順に進みます。

3. テキストエディタで、`/etc/postfix/sasl_passwd` ファイルを開きます。このファイルが存在していない場合は、作成します。
4. `/etc/postfix/sasl_passwd` に次の行を追加します。

```
[email-smtp.us-west-2.amazonaws.com]:587 SMTPUSERNAME:SMTPPASSWORD
```

#### Note

**SMTPUSERNAME** と **SMTPPASSWORD** を、それぞれ SMTP ユーザー名とパスワードに置き換えます。SMTP ユーザー名とパスワードは、AWS アクセスキー ID およびシークレットアクセスキーと同じではありません。認証情報の詳細については、「[the section called “SMTP 認証情報の取得” \(p. 89\)](#)」を参照してください。

米国西部 (オレゴン) 以外の AWS リージョンで Amazon SES を使用する場合は、上の例の **email-smtp.us-west-2.amazonaws.com** を、該当するリージョンの SMTP エンドポイントに置き換えます。詳細については、「[リージョン \(p. 467\)](#)」を参照してください。

`sasl_passwd` を保存して閉じます。

5. コマンドプロンプトで次のコマンドを入力し、SMTP 認証情報を含む `hashmap` データベースファイルを作成します。

```
sudo postmap hash:/etc/postfix/sasl_passwd
```

6. (オプション) 前の手順で作成した `/etc/postfix/sasl_passwd` ファイルおよび `/etc/postfix/sasl_passwd.db` ファイルは暗号化されていません。これらのファイルには SMTP 認証情報が含まれているため、ファイルへのアクセスを制限するために、ファイルの所有権とアクセス許可を変更することをお勧めします。これらのファイルへのアクセスを制限するには。
  - a. コマンドプロンプトで次のコマンドを入力して、ファイルの所有権を変更します。

```
sudo chown root:root /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
```

- b. コマンドプロンプトで次のコマンドを入力してファイルのアクセス許可を変更して、ルートユーザーのみがこれらのファイルで読み込みや書き込みを実行できるようにします。

```
sudo chmod 0600 /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
```

7. Postfix が CA 証明書の場所を認識できるようにします (Amazon SES サーバー証明書を検証するために必要です)。このステップで使用するコマンドは、オペレーティングシステムによって異なります。

- Amazon Linux、Red Hat Enterprise Linux、あるいは関連するディストリビューションを使用している場合には、次のコマンドを入力します。

```
sudo postconf -e 'smtp_tls_CAfile = /etc/ssl/certs/ca-bundle.crt'
```

- Ubuntu あるいは関連するディストリビューションを使用している場合には、次のコマンドを入力します。

```
sudo postconf -e 'smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt'
```

- macOS を使用する場合、システムのキーチェーンで証明書を生成できます。証明書を生成するには、コマンドラインで次のコマンドを入力します。

```
sudo security find-certificate -a -p /System/Library/Keychains/  
SystemRootCertificates.keychain > /etc/ssl/certs/ca-bundle.crt
```

証明書を生成したら、次のコマンドを入力します。

```
sudo postconf -e 'smtp_tls_CAfile = /etc/ssl/certs/ca-bundle.crt'
```

8. 次のコマンドを入力して Postfix サーバーを起動します (サーバーがすでに起動している場合は、設定を再ロードしてください)。

```
sudo postfix start; sudo postfix reload
```

9. コマンドラインで以下のように入力し、各行の最後で Enter キーを押して、テスト E メールを送信します。 `sender@example.com` を送信元 E メールアドレスに置き換えてください。送信元アドレスの Amazon SES での使用を確認する必要があります。 `recipient@example.com` を送信先アドレスに置き換えます。アカウントがサンドボックスにまだある場合は、受信者アドレスも確認する必要があります。最後に、メッセージの最後の行には 1 つのピリオドが (.) 含まれていることが必要です (他に何も含まない)。

```
sendmail -f sender@example.com recipient@example.com  
From: Sender Name <sender@example.com>  
Subject: Amazon SES Test  
This message was sent using Amazon SES.  
.
```

10. 受信者のアドレスに関連付けられているメールボックスを確認します。E メールが届かない場合は、迷惑メールフォルダを確認します。それでも E メールが見つからない場合には、E メールを送信するために使用したシステムのメールログで詳細を確認します (通常は、`/var/log/maillog` にあります)。

## 高度な使用例

この例は、[設定セット \(p. 257\)](#) を使用する E メール、および MIME マルチパートエンコードを使用してプレーンテキストと HTML バージョンのメッセージの両方を添付ファイルと共に送る E メールを送信する方法を示します。また、それにはクリックイベントの分類に利用できる [リンクタグ \(p. 521\)](#) が含まれています。Eメールのコンテンツは外部ファイルで指定されるため、Postfix セッションで手動でコマンドを入力する必要はありません。

Postfix を使用してマルチパートの MIME E メールを送信するには

1. テキストエディタで、`mime-email.txt` という名前の新規ファイルを作成します。
2. テキストファイルで次のコンテンツを貼り付け、赤の値を自分のアカウントの適切な値に置き換えます。

```
X-SES-CONFIGURATION-SET: ConfigSet  
From: Sender Name <sender@example.com>  
Subject: Amazon SES Test  
MIME-Version: 1.0  
Content-Type: multipart/mixed; boundary="YWVhZDF1Y2QzMzMGQ2N2U0YTZmODU"  
  
--YWVhZDF1Y2QzMzMGQ2N2U0YTZmODU
```

```
Content-Type: multipart/alternative; boundary="3NjM0N2QwMTE4MWQ0ZTg2NTYxZQ"

--3NjM0N2QwMTE4MWQ0ZTg2NTYxZQ
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

Amazon SES Test

This message was sent from Amazon SES using the SMTP interface.

For more information, see:
http://docs.aws.amazon.com/ses/latest/DeveloperGuide/send-email-smtp.html

--3NjM0N2QwMTE4MWQ0ZTg2NTYxZQ
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

<html>
  <head>
</head>
  <body>
    <h1>Amazon SES Test</h1>
    <p>This message was sent from Amazon SES using the SMTP interface.</p>
    <p>For more information, see
    <a ses:tags="samplekey0:samplevalue0;samplekey1:samplevalue1;"
href="http://docs.aws.amazon.com/ses/latest/DeveloperGuide/send-email-smtp.html">
Using the Amazon SES SMTP Interface to Send Email</a> in the <em>Amazon SES
Developer Guide</em>.</p>
  </body>
</html>
--3NjM0N2QwMTE4MWQ0ZTg2NTYxZQ--
--YVvhZDFly2QzMGQ2N2U0YTZmODU
Content-Type: application/octet-stream
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="customers.txt"

SUQsRmlyc3ROYW11LExhc3ROYW11LENvdW50cnkKMzQ4LEpvaG4sU3RpbGVzLENh
bmFkYQo5MjM4OSxKaWUStG11LENoaW5hCjczNCxTaGlybGV5LFJvZHJpZ3VleixV
bm10ZWQgU3RhdGVzCjI4OTMsQW5heWESX11bmdhcixJbRmRyQ==
--YVvhZDFly2QzMGQ2N2U0YTZmODU--
```

保存してファイルを閉じます。

3. コマンドラインから、以下のコマンドを入力します。*sender@example.com* をユーザーの E メールアドレスに、*recipient@example.com* を受信者の E メールアドレスに置き換えます。

```
sendmail -f sender@example.com recipient@example.com < mime-email.txt
```

コマンドが正常に実行された場合、何の出力もなく終了します。

4. Eメールの受信箱を確認します。メッセージが配信されなかった場合は、システムのメールログを確認します。

## Amazon SES と Sendmail の統合

Sendmail は 1980 年代の初めにリリースされ、それ以降、継続的に改善されてきました。これは、大規模なユーザーコミュニティを持つ柔軟で設定可能な Message Transfer Agent (MTA) です。Sendmail は 2013 年に Proofpoint により買収されましたが、Proofpoint は Sendmail のオープンソースバージョンの提供を続けています。[Sendmail のオープンソースバージョン](#)は、Proofpoint のウェブサイトからダウンロードするか、ほとんどの Linux ディストリビューションのパッケージマネージャを経由してダウンロードできます。

このセクションの手順は、Amazon SES を介して E メールを送信するよう Sendmail を設定する方法を示しています。この手順は Ubuntu 18.04.2 LTS を実行しているサーバーでテストされています。

#### Note

Sendmail はサードパーティーのアプリケーションであり、Amazon Web Services によって開発またはサポートされていません。このセクションの手順は情報提供のみを目的としており、予告なく変更される場合があります。

### Prerequisites

このセクションの手順を完了する前に、以下の手順を完了する必要があります。

- Sendmail パッケージをサーバーにインストールします。

#### Note

使用しているオペレーティングシステムのディストリビューションに応じて、`sendmail-cf`、`m4`、および `cyrus-sasl-plain` のパッケージのインストールが必要になる場合があります。

- 差出人アドレスとして使用する ID を確認します。詳細については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。

アカウントが Amazon SES サンドボックスにある場合は、Eメールの送信先のアドレスも検証する必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

Amazon SES を使用して Amazon EC2 インスタンスから Eメールを送信する場合は、次の手順も実行する必要があります。

- Amazon SES を使用して Amazon EC2 インスタンスから Eメールを送信する場合は、Elastic IP アドレスを Amazon EC2 インスタンスに割り当て、Eメールプロバイダーが Eメールを受信できるようにする必要があります。詳細については、Linux インスタンス向け Amazon EC2 ユーザーガイドの「[Amazon EC2 elastic IP アドレス](#)」を参照してください。
- Amazon Elastic Compute Cloud (Amazon EC2) では、デフォルトでポート 25 経由での Eメールトラフィックを制限しています。Amazon EC2 から SMTP エンドポイントを介して Eメールを送信する際のタイムアウトを回避するには、これらの制限を解除するようリクエストします。詳細については、AWS ナレッジセンターの「[Amazon EC2 インスタンスまたは AWS Lambda 関数 からポート 25 の制限を解除する方法](#)」を参照してください。

または、このセクションの手順を変更して、ポート 25 ではなく、ポート 587 を使用することもできます。

### Sendmail を設定する

このセクションの手順に従って、Amazon SES を使用して Eメールを送信するよう Sendmail を設定します。

#### Important

このセクションの手順は、米国西部 (オレゴン) AWS リージョンで Amazon SES を使用することを前提としています。別のリージョンを使用する場合は、この手順の `email-smtp.us-west-2.amazonaws.com` のすべてのインスタンスをご希望リージョンの SMTP エンドポイントに置き換えてください。Amazon SES が使用可能な AWS リージョンの SMTP エンドポイント URL のリストについては、[AWS 全般のリファレンス](#)の「Amazon Simple Email Service (Amazon SES)」を参照してください。

## Sendmail を設定するには

1. ファイルエディタで、ファイル `/etc/mail/authinfo` を開きます。このファイルが存在しない場合は、作成します。

以下の行を `/etc/mail/authinfo` に追加します。

```
AuthInfo:email-smtp.us-west-2.amazonaws.com "U:root" "I:smtpUsername" "P:smtpPassword"
"M:PLAIN"
```

上の例に、以下の変更を加えます。

- `email-smtp.us-west-2.amazonaws.com` を、使用する Amazon SES SMTP エンドポイントに置き換えます。
- `smtpUsername` を Amazon SES SMTP ユーザー名に置き換えます。
- `smtpPassword` を Amazon SES SMTP パスワードに置き換えます。

### Note

SMTP ユーザー名およびパスワードは、AWS アクセスキー ID およびシークレットアクセスキーとは異なります。SMTP ユーザー名とパスワードを取得する方法の詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

終了したら、`authinfo` を保存します。

2. コマンドラインで、以下のコマンドを入力して `/etc/mail/authinfo.db` ファイルを生成します。

```
sudo sh -c 'makemap hash /etc/mail/authinfo.db < /etc/mail/authinfo'
```

3. コマンドラインで、以下のコマンドを入力して Amazon SES SMTP エンドポイントに中継するためのサポートを追加します。

```
sudo sh -c 'echo "Connect:email-smtp.us-west-2.amazonaws.com RELAY" >> /etc/mail/access'
```

上のコマンドで、`email-smtp.us-west-2.amazonaws.com` を、使用する Amazon SES SMTP エンドポイントのアドレスに置き換えます。

4. コマンドラインで、以下のコマンドを入力して `/etc/mail/access.db` を再生成します。

```
sudo sh -c 'makemap hash /etc/mail/access.db < /etc/mail/access'
```

5. コマンドラインで、以下のコマンドを入力して `sendmail.cf` および `sendmail.mc` ファイルのバックアップを作成します。

```
sudo sh -c 'cp /etc/mail/sendmail.cf /etc/mail/sendmail_cf.backup && cp /etc/mail/sendmail.mc /etc/mail/sendmail_mc.backup'
```

6. 以下の行を、`/etc/mail/sendmail.mc` ファイルのすべての `MAILER()` 定義の前に追加します。

```
define(`SMART_HOST', `email-smtp.us-west-2.amazonaws.com')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 25')dnl
define(`confAUTH_MECHANISMS', `LOGIN PLAIN')dnl
FEATURE(`authinfo', `hash -o /etc/mail/authinfo.db')dnl
MASQUERADE_AS(`example.com')dnl
FEATURE(masquerade_envelope)dnl
FEATURE(masquerade_entire_domain)dnl
```

上のテキストで、次の操作を行います。

- `email-smtp.us-west-2.amazonaws.com`を、使用する Amazon SES SMTP エンドポイントに置き換えます。
- `example.com` を、Eメールの送信に使用するドメインに置き換えます。

終了したら、ファイルを保存します。

#### Note

Amazon EC2 は、デフォルトでポート 25 経由の通信を制限します。Amazon EC2 インスタンスの Sendmail を使用する場合は、[Eメール送信制限解除申請](#)に入力する必要があります。

7. コマンドラインで、以下のコマンドを入力して `sendmail.cf` を書き込み可能にします。

```
sudo chmod 666 /etc/mail/sendmail.cf
```

8. コマンドラインで、以下のコマンドを入力して `sendmail.cf` を再生成します。

```
sudo sh -c 'm4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf'
```

#### Note

「コマンドが見つかりません」や「そのようなファイルまたはディレクトリはありません」などといったエラーが発生した場合は、`m4` パッケージと `sendmail-cf` パッケージがシステムにインストールされていることを確認します。

9. コマンドラインで、以下のコマンドを入力して `sendmail.cf` の権限を読み取り専用にリセットします。

```
sudo chmod 644 /etc/mail/sendmail.cf
```

10. コマンドラインで、以下のコマンドを入力して Sendmail を再起動します。

```
sudo /etc/init.d/sendmail restart
```

11. 以下の手順を実行して、テスト Eメールを送信します。

- a. コマンドラインで、以下のコマンドを入力します。

```
/usr/sbin/sendmail -vf sender@example.com recipient@example.com
```

`sender@example.com` を送信元 Eメールアドレスに置き換えてください。`recipient@example.com` を送信先アドレスに置き換えます。終了したら、Enter キーを押します。

- b. 以下のメッセージの内容を入力します。各行の終わりで Enter キーを押します。

```
From: sender@example.com  
To: recipient@example.com  
Subject: Amazon SES test email  
  
This is a test message sent from Amazon SES using Sendmail.
```

Eメールの内容を入力し終わったら、Ctrl+D キーを押して送信します。

12. 受取人の Eメールクライアントで Eメールをチェックします。Eメールが見つからない場合は、迷惑メールフォルダを確認します。それでも Eメールが見つからない場合は、メールサーバー上の Sendmail ログを確認します。通常、ログは `/var/log/mail.log` または `/var/log/maillog` にあります。

## Amazon SES を Microsoft Windows Server IIS SMTP に統合する

Amazon SES を介して E メールを送信するように Microsoft Windows Server の IIS SMTP サーバーを設定します。以下の手順は、Amazon EC2 インスタンスで、Microsoft Windows Server 2012 を使用している場合のもので、Microsoft Windows Server 2008 と Microsoft Windows Server 2008 R2 のいずれを使用している場合も同じ設定を使用できます。

### Note

Windows Server はサードパーティーのアプリケーションであり、Amazon Web Services によって開発またはサポートされていません。このセクションの手順は情報提供のみを目的としており、予告なく変更される場合があります。

Amazon SES と Microsoft Windows Server IIS SMTP を統合するには

- まず、以下の手順を使用して Microsoft Windows Server 2012 をセットアップします。
  - [Amazon EC2 マネジメントコンソール](#)から、Microsoft Windows Server 2012 ベースの新しい Amazon EC2 インスタンスを起動します。
  - 「[Amazon EC2 Windows インスタンスの使用開始](#)」の手順に従って、リモートデスクトップによりそのインスタンスに接続してログインします。
  - Server Managerのダッシュボードを起動します。
  - [Web Server] ロールをインストールします。[IIS 6 Management Compatibility tools] ( [Web Server] チェックボックスの下にあるオプション ) をオンにすることを忘れないでください。
  - [SMTP Server] 機能をインストールします。
- 次に、以下の手順を使用して IIS SMTP サービスを設定します。
  - Server Manager のダッシュボードに戻ります。
  - [Tools] メニューの [Internet Information Services (IIS) 6.0 Manager] を選択します。
  - [SMTP Virtual Server #1] を右クリックし、[Properties] を選択します。
  - [Access] タブの [Relay Restrictions] で、[Relay] を選択します。
  - [Relay Restrictions] ダイアログボックスで、[Add] を選択します。
  - [Single Computer] で、IP アドレスとして「127.0.0.1」と入力します。これで、IIS SMTP サービスを介して E メールを Amazon SES に中継する権限がこのサーバーに割り当てられました。

この手順では、E メールがこのサーバーで生成されることを前提としています。E メールを生成するアプリケーションが別のサーバーで実行されている場合は、IIS SMTP を介して E メールを中継する権限をそのサーバーに割り当て必要があります。

### Note

SMTP リレーをプライベートサブネットに拡張するには、[Relay Restriction] で [Single Computer] 127.0.0.1 と [Group of Computers] 172.1.1.0 ~ 255.255.255.0 (ネットマスクセクション) を使用します。[Connection] で、[Single Computer] 127.0.0.1 と [Group of Computers] 172.1.1.0 ~ 255.255.255.0 (ネットマスクセクション) を使用します。

- 最後に、以下の手順を使用して、Amazon SES を介して E メールを送信するようにサーバーを設定します。
  - SMTP Virtual Server #1 Properties ダイアログボックスに戻り、[Delivery] タブを選択します。
  - [Delivery] タブで、[Outbound Security] を選択します。
  - [Basic Authentication] を選択し、Amazon SES SMTP のユーザー名とパスワードを入力します。これらの認証情報は、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」の手順に従って Amazon SES コンソールから取得できます。



## Important

SMTP ユーザー名とパスワードは、AWS アクセスキー ID およびシークレットアクセスキーと同じではありません。SMTP エンドポイントに対して認証を行う際に、AWS 認証情報を使用しないでください。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。

- d. [TLS encryption] が選択されていることを確認します。
- e. [Delivery] タブに戻ります。
- f. [Outbound Connections] を選択します。
- g. [Outbound Connections] ダイアログボックスで、ポートが 25 または 587 であることを確認します。

## Note

メールサーバーが Amazon EC2 でホストされていて、ポート 25 を使用して E メールを送信する場合は、まずアカウントのポート 25 の制限を削除する必要があります。詳細については、AWS ナレッジセンターの「[Amazon EC2 インスタンスまたは AWS Lambda 関数 からポート 25 の制限を解除する方法](#)」を参照してください。

- h. [Advanced] を選択します。
- i. スマートホスト名で、使用する Amazon SES エンドポイントを入力します (たとえば、email-smtp.us-west-2.amazonaws.com)。Amazon SES が使用可能な AWS リージョンのエンドポイント URL のリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。
- j. Server Manager のダッシュボードに戻ります。
- k. Server Manager のダッシュボードで、[SMTP Virtual Server #1] を右クリックし、新しい設定が選択されるようにサービスを再起動します。
- l. このサーバーを介して E メールを送信します。メッセージヘッダーを調べると、そのメッセージが Amazon SES を介して配信されたことを確認できます。

## Amazon SES と Exim の統合

Exim は、非常に柔軟で構成可能なメール転送エージェント (MTA) です。Exim の詳細については、「[Exim のウェブサイト](#)」にアクセスしてください。

## Note

Exim はサードパーティーのアプリケーションであり、Amazon Web Services によって開発またはサポートされていません。このセクションの手順は情報提供のみを目的としており、予告なく変更される場合があります。

Amazon SES を介して E メールを送信するように Exim を設定するには

1. テキストエディタで、`/etc/exim.conf.local` ファイルを開きます。ファイルが存在しない場合は、`/etc/exim4/exim4.conf.template` からテンプレートをコピーします。
2. `/etc/exim.conf.local` で、以下の変更を加えます。
  - a. `routers` セクションの `begin routers` 行の後に以下を追加します。

```
send_via_ses:
driver = manualroute
domains = ! +local_domains
transport = ses_smtp
route_list = * email-smtp.us-west-2.amazonaws.com;
```

上のコマンドで、[email-smtp.us-west-2.amazonaws.com](#) を、メッセージ送信に使用する SMTP エンドポイントに置き換えます。詳細については、「[リージョンと Amazon SES \(p. 467\)](#)」を参照してください。

- b. transports セクションの begin transports 行の後に以下を追加します。

```
ses_smtp:  
driver = smtp  
port = 587  
hosts_require_auth = *  
hosts_require_tls = *
```

- c. authenticators セクションの begin authenticators 行の後に以下を追加します。

```
ses_login:  
driver = plaintext  
public_name = LOGIN  
client_send = : USERNAME : PASSWORD
```

前述のコードの **USERNAME** を SMTP ユーザー名に、**PASSWORD** を SMTP パスワードに置き換えます。

#### Important

SMTP 認証情報は、AWS アクセスキー ID やシークレットアクセスキーと同じではありません。SMTP 認証情報を取得する方法については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

3. /etc/exim.conf.local を保存します。
4. 設定の更新が完了したら、次のコマンドを入力して Exim を再起動します。

```
sudo /etc/init.d/exim4 restart
```

#### Note

このコマンドは、使用するオペレーティングシステムによって異なる場合があります。

5. コマンドラインで以下の手順を実行して、テストメッセージを送信します。
  - a. 次のコマンドを入力します。

```
exim -v recipient@example.com
```

前述のコマンドの **recipient@example.com** をメッセージの送信先アドレスに置き換えます。

- b. 次のように入力し、各行の最後で Enter を押します。

```
From: sender@example.com  
Subject: Test message  
This is a test.  
.
```

前述のコマンドの **sender@example.com** をメッセージの送信元アドレスに置き換えます。

最後のピリオド(.)の後に Enter を押すと、Exim は SMTP サーバーとの接続を開始します。メッセージの送信後も接続が開いたままの場合は、Ctrl+D を押して閉じます。

## Tip

メッセージが配信されない場合は、システムのメールログにエラーが記録されていないか確認します。Exim のメールログは通常、`/var/log/exim4/mainlog` にあります。

## コマンドラインを使用して Amazon SES SMTP インターフェイスへの接続をテストする

Amazon SES SMTP インターフェイスは、オペレーティングシステムのコマンドラインから操作できます。このセクションで説明する方法は、Amazon SES SMTP エンドポイントへの接続のテスト、SMTP 認証情報の検証、および接続問題のトラブルシューティングでの使用を目的としています。以下の手順では、代表的なオペレーティングシステムに搭載されているツールやライブラリを使用します。

SMTP 接続問題のトラブルシューティングの詳細については、「[Amazon SES SMTP の問題 \(p. 489\)](#)」を参照してください。

### Prerequisites

Amazon SES SMTP インターフェイスに接続するときは、SMTP 認証情報のセットを指定する必要があります。SMTP 認証情報は、標準の AWS 認証情報とは異なります。2 つのタイプの認証情報は、一方をもう一方の代わりに使用することはできません。SMTP 認証情報を取得する方法の詳細については、「[the section called "SMTP 認証情報の取得" \(p. 89\)](#)」を参照してください。

### Amazon SES SMTP インターフェイスへの接続のテスト

コマンドラインを使用すると、認証情報の入力やメッセージの送信を行うことなく、Amazon SES SMTP インターフェイスへの接続をテストできます。この手順は、基本的な接続問題のトラブルシューティングに役立ちます。

このセクションでは、OpenSSL (ほとんどの Linux、macOS、Unix デイストリビューションに搭載され、Windows でも使用可能) と PowerShell の `Test-NetConnection` コマンドレット (ほとんどの最新バージョンの Windows に搭載) の両方を使用して接続をテストする手順を示します。

Linux, macOS, or Unix

OpenSSL を使用して Amazon SES SMTP インターフェイスに接続するには、ポート 587 経由で明示的な SSL を使用する方法と、ポート 465 経由で暗黙的な SSL を使用する方法の 2 つがあります。

明示的な SSL を使用して SMTP インターフェイスに接続するには

- コマンドラインで、次のコマンドを入力して Amazon SES SMTP サーバーに接続します。

```
openssl s_client -crlf -quiet -starttls smtp -connect email-smtp.us-west-2.amazonaws.com:587
```

上のコマンドで、*email-smtp.us-west-2.amazonaws.com* を、使用する AWS リージョンの Amazon SES SMTP エンドポイントの URL に置き換えます。詳細については、「[リージョン \(p. 467\)](#)」を参照してください。

正常に接続されると次のような出力が表示されます。

```
depth=2 C = US, O = Amazon, CN = Amazon Root CA 1
verify return:1
depth=1 C = US, O = Amazon, OU = Server CA 1B, CN = Amazon
verify return:1
depth=0 CN = email-smtp.us-west-2.amazonaws.com
verify return:1
```

```
250 Ok
```

約 10 秒間何もしないと、接続は自動的に終了します。

または、暗黙的な SSL を使用して、ポート 465 経由で SMTP インターフェイスに接続することもできます。

暗黙的な SSL を使用して SMTP インターフェイスに接続するには

- コマンドラインで、次のコマンドを入力して Amazon SES SMTP サーバーに接続します。

```
openssl s_client -crlf -quiet -connect email-smtp.us-west-2.amazonaws.com:465
```

上のコマンドで、*email-smtp.us-west-2.amazonaws.com* を、使用する AWS リージョンの Amazon SES SMTP エンドポイントの URL に置き換えます。詳細については、「[リージョン \(p. 467\)](#)」を参照してください。

正常に接続されると次のような出力が表示されます。

```
depth=2 C = US, O = Amazon, CN = Amazon Root CA 1
verify return:1
depth=1 C = US, O = Amazon, OU = Server CA 1B, CN = Amazon
verify return:1
depth=0 CN = email-smtp.us-west-2.amazonaws.com
verify return:1
220 email-smtp.amazonaws.com ESMTP SimpleEmailService-d-VCSHDP1YZ
A1b2C3d4E5f6G7h8I9j0
```

約 10 秒間何もしないと、接続は自動的に終了します。

## PowerShell

PowerShell の [Test-NetConnection](#) コマンドレットを使用して Amazon SES SMTP サーバーに接続します。

### Note

[Test-NetConnection](#) コマンドレットでは、コンピュータが Amazon SES SMTP エンドポイントに接続できるかどうかを判断できます。ただし、コンピュータが SMTP エンドポイントに暗黙的な SSL 接続や明示的な SSL 接続を確立できるかどうかはテストしません。SSL 接続をテストするには、OpenSSL for Windows をインストールするか、「[コマンドラインを使用し、Amazon SES SMTP インターフェイスを介して E メールを送信する \(p. 109\)](#)」の手順を実行して、テスト E メールを送信できます。

**Test-NetConnection** コマンドレットを使用して SMTP インターフェイスに接続するには

- Amazon SES で、次のコマンドを入力して Amazon SES SMTP サーバーに接続します。

```
Test-NetConnection -Port 587 -ComputerName email-smtp.us-west-2.amazonaws.com
```

上記のコマンドで、*email-smtp.us-west-2.amazonaws.com* を AWS リージョンの Amazon SES SMTP エンドポイントの URL に置き換え、587 をポート番号に置き換えます。Amazon SES のリージョンのエンドポイントの詳細については、の「[リージョン \(p. 467\)](#)」を参照してください。

接続に成功すると、次の例のような出力が表示されます。

```
ComputerName      : email-smtp.us-west-2.amazonaws.com
RemoteAddress     : 198.51.100.126
RemotePort        : 587
InterfaceAlias    : Ethernet
SourceAddress     : 203.0.113.46
TcpTestSucceeded : True
```

## コマンドラインを使用し、Amazon SES SMTP インターフェイスを介して E メールを送信する

コマンドラインを使用して、Amazon SES SMTP インターフェイスを介してメッセージを送信することもできます。この手順は、SMTP 認証情報をテストする場合や、Amazon SES を使用して送信するメッセージを特定の受信者が受信できるかどうかをテストする場合に役立ちます。

Linux, macOS, or Unix

E メール送信者が SMTP サーバーに接続すると、クライアントはスタンダードなリクエストのセットを発行し、サーバーは各リクエストにスタンダードのレスポンスで応答します。この一連のリクエストとレスポンスは SMTP 対話と呼ばれます。OpenSSL を使用して Amazon SES SMTP サーバーに接続すると、サーバーは SMTP 対話が発生することを予想します。

OpenSSL を使用して SMTP インターフェイスに接続する場合は、base64 エンコードを使用して SMTP 認証情報をエンコードする必要があります。このセクションでは、base64 を使用して認証情報をエンコードする手順を示します。

SMTP インターフェイスを使用してコマンドラインから E メールを送信するには

1. コマンドラインで、次のコマンドを入力して SMTP ユーザー名をエンコードします。この際、***SMTPUsername*** をお使いの SMTP ユーザー名に置き換えます。

```
echo -n "SMTPUsername" | openssl enc -base64
```

このコマンド出力を書き留めます。

2. コマンドラインで、次のコマンドを入力して SMTP パスワードをエンコードします。この際、***SMTPPassword*** をお使いの SMTP パスワードに置き換えます。

```
echo -n "SMTPPassword" | openssl enc -base64
```

このコマンド出力を書き留めます。

3. テキストエディタで新規ファイルを作成します。ファイルに次のコードを貼り付けます。

```
EHLO example.com
AUTH LOGIN
Base64EncodedSMTPUserName
Base64EncodedSMTPPassword
MAIL FROM: sender@example.com
RCPT TO: recipient@example.com
DATA
X-SES-CONFIGURATION-SET: ConfigSet
From: Sender Name <sender@example.com>
To: recipient@example.com
Subject: Amazon SES SMTP Test

This message was sent using the Amazon SES SMTP interface.
.
```

```
QUIT
```

4. 前のステップで作成したファイルを次のように変更します。

- `example.com` を送信ドメインに置き換えます。
- `Base64EncodedSMTPUserName` をお使いの base64 エンコード SMTP ユーザー名に置き換えます。
- `Base64EncodedSMTPPassword` をお使いの base64 エンコード SMTP パスワードに置き換えます。
- `sender@example.com` を送信元 E メールアドレスに置き換えます。この ID は検証する必要があります。
- `recipient@example.com` を送信先アドレスに置き換えます。Amazon SES アカウントがサンドボックスにまだある場合は、このアドレスも検証済みである必要があります。
- `ConfigSet` を、この E メールを送信する際に使用する [設定セット \(p. 257\)](#) の名前に置き換えます。

#### Note

設定セットを使用しない場合は、`X-SES-CONFIGURATION-SET` で始まる行全体を省略できます。

終了したら、`input.txt` としてファイルを保存します。

5. コマンドラインで、次のいずれかのオプションを選択します。

- 明示的な SSL を使用してポート 587 経由で送信するには 次のコマンドを入力します。

```
openssl s_client -crlf -quiet -starttls smtp -connect email-smtp.us-west-2.amazonaws.com:587 < input.txt
```

- 暗黙的な SSL を使用してポート 465 経由で送信するには 次のコマンドを入力します。

```
openssl s_client -crlf -quiet -connect email-smtp.us-west-2.amazonaws.com:465 < input.txt
```

#### Note

`email-smtp.us-west-2.amazonaws.com` AWS リージョンの Amazon SES SMTP エンドポイントの URL に置き換えます。詳細については、「[リージョン \(p. 467\)](#)」を参照してください。

メッセージが Amazon SES によって受け入れられた場合は、次の例のような出力が表示されます。

```
250 Ok 01010160d7de98d8-21e57d9a-JZho-416c-bbe1-8ebaAexample-000000
```

250 Ok に続く数字の文字列とテキストは、Eメールのメッセージ ID です。

#### Note

約 10 秒間何もしないと、接続は自動的に終了します。

## PowerShell

`Net.Mail.SmtpClient` クラスを使用すると、明示的な SSL を通じてポート 587 経由で Eメールを送信できます。

## Note

Net.Mail.SmtpClient クラスは正式に廃止されており、Microsoft はサードパーティー製のライブラリを使用することを推奨しています。このコードはテスト専用であり、本番環境用のワークロードには使用しないでください。

明示的な SSL を使用して PowerShell 経由で E メールを送信するには

1. テキストエディタで新規ファイルを作成します。ファイルに次のコードを貼り付けます。

```
function SendEmail($Server, $Port, $Sender, $Recipient, $Subject, $Body) {
    $Credentials = [Net.NetworkCredential](Get-Credential)

    $SMTPClient = New-Object Net.Mail.SmtpClient($Server, $Port)
    $SMTPClient.EnableSsl = $true
    $SMTPClient.Credentials = New-Object
    System.Net.NetworkCredential($Credentials.Username, $Credentials.Password);

    try {
        Write-Output "Sending message..."
        $SMTPClient.Send($Sender, $Recipient, $Subject, $Body)
        Write-Output "Message successfully sent to $($Recipient)"
    } catch [System.Exception] {
        Write-Output "An error occurred:"
        Write-Error $_
    }
}

function SendTestEmail(){
    $Server = "email-smtp.us-west-2.amazonaws.com"
    $Port = 587

    $Subject = "Test email sent from Amazon SES"
    $Body = "This message was sent from Amazon SES using PowerShell (explicit SSL,
    port 587)."
```

終了したら、SendEmail.ps1 としてファイルを保存します。

2. 前のステップで作成したファイルを次のように変更します。
  - `sender@example.com` を、メッセージの送信元の E メールアドレスに置き換えます。
  - `recipient@example.com` をメッセージの送信先アドレスに置き換えます。
  - `email-smtp.us-west-2.amazonaws.com` with the URL of the Amazon SES SMTP endpoint for your AWS Region. を AWS リージョンの Amazon SES SMTP エンドポイントの URL に置き換えます。詳細については、「[リージョンと Amazon SES \(p. 467\)](#)」を参照してください。
3. PowerShell を使用して、次のコマンドを入力します。

```
.\path\to\SendEmail.ps1
```

上のコマンドで、`path\to\SendEmail.ps1` をステップ 1 で作成したファイルへのパスに置き換えます。

4. プロンプトが表示されたら、SMTP ユーザー名とパスワードを入力します。

または、[System.Web.Mail.SmtpMail](#) クラスを使用して、ポート 465 経由で暗黙的な SSL を通じて E メールを送信することもできます。

#### Note

`System.Web.Mail.SmtpMail` クラスは正式に廃止されており、Microsoft はサードパーティー製のライブラリを使用することを推奨しています。このコードはテスト専用であり、本番環境用のワークロードには使用しないでください。

暗黙的な SSL を使用して PowerShell 経由で E メールを送信するには

1. テキストエディタで新規ファイルを作成します。ファイルに次のコードを貼り付けます。

```
[System.Reflection.Assembly]::LoadWithPartialName("System.Web") > $null

function SendEmail($Server, $Port, $Sender, $Recipient, $Subject, $Body) {
    $Credentials = [Net.NetworkCredential](Get-Credential)

    $mail = New-Object System.Web.Mail.MailMessage
    $mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/smtpserver",
    $Server)
    $mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/
smtpserverport", $Port)
    $mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/smtpusessl",
    $true)
    $mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/sendusername",
    $Credentials.UserName)
    $mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/sendpassword",
    $Credentials.Password)
    $mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/
smtpconnectiontimeout", $timeout / 1000)
    $mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/sendusing", 2)
    $mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/
smtpauthenticate", 1)

    $mail.From = $Sender
    $mail.To = $Recipient
    $mail.Subject = $Subject
    $mail.Body = $Body

    try {
        Write-Output "Sending message..."
        [System.Web.Mail.SmtpMail]::Send($mail)
        Write-Output "Message successfully sent to $($Recipient)"
    } catch [System.Exception] {
        Write-Output "An error occurred:"
        Write-Error $_
    }
}

function SendTestEmail(){
    $Server = "email-smtp.us-west-2.amazonaws.com"
    $Port = 465

    $Subject = "Test email sent from Amazon SES"
    $Body = "This message was sent from Amazon SES using PowerShell (implicit SSL,
port 465)."
```

```
    $Sender = "sender@example.com"
    $Recipient = "recipient@example.com"

    SendEmail $Server $Port $Sender $Recipient $Subject $Body
}
```



```
SendTestEmail
```

終了したら、SendEmail.ps1 としてファイルを保存します。

2. 前のステップで作成したファイルを次のように変更します。
  - `sender@example.com` を、メッセージの送信元の E メールアドレスに置き換えます。
  - `recipient@example.com` をメッセージの送信先アドレスに置き換えます。
  - `email-smtp.us-west-2.amazonaws.com` with the URL of the Amazon SES SMTP endpoint for your AWS Region.を AWS リージョンの Amazon SES SMTP エンドポイントの URL に置き換えます。詳細については、「[リージョンと Amazon SES \(p. 467\)](#)」を参照してください。
3. PowerShell を使用して、次のコマンドを入力します。

```
.\path\to\SendEmail.ps1
```

上のコマンドで、`path\to\SendEmail.ps1` をステップ 1 で作成したファイルへのパスに置き換えます。

4. プロンプトが表示されたら、SMTP ユーザー名とパスワードを入力します。

## Amazon SES API を使用して E メールを送信する

Amazon SES を介して本稼働 E メールを送信する場合には、Simple Mail Transfer Protocol (SMTP) インターフェイスまたは Amazon SES API を使用できます。SMTP インターフェイスの詳細については、「[Amazon SES SMTP インターフェイスを使用した Eメールの送信 \(p. 88\)](#)」を参照してください。このセクションでは、API を使用して E メールを送信する方法について説明します。

また、Amazon SES API を使用して E メールを送信することもできます。API を使用して E メールを送信する場合は、メッセージの内容を指定し、Amazon SES は MIME E メールを組み立てます。また、メッセージの内容を完全に制御できるように、E メールを自分で構成することもできます。API の詳細については、[Amazon Simple Email Service API リファレンス](#) を参照してください。Amazon SES が使用可能な AWS リージョンのエンドポイント URL のリストについては、[AWS 全般のリファレンス](#) の「Amazon Simple Email Service エンドポイントとクォータ」を参照してください。

以下の方法で API を呼び出すことができます。

- 直接 HTTPS リクエストを作成する - これは、リクエストの認証と署名を手動で処理し、リクエストを手動で作成する必要があるため、最も高度な方法です。リクエストの実行方法の詳細については、「[Amazon SES API の使用 \(p. 416\)](#)」を参照してください。
- AWS SDK を使用する - AWS SDK を使用して、Amazon SES を含む AWS の複数のサービスにアクセスできます。SDK を使用すると、認証、リクエスト署名、再試行ロジック、エラー処理などの低レベルの機能が自動的に実行されるため、充実したアプリケーションの構築に専念できます。
- コマンドラインインターフェイスの使用 - [AWS Command Line Interface](#) は、Amazon SES 用のコマンドラインツールです。さらに、PowerShell 環境でのスクリプト作成用に [AWS Tools for Windows PowerShell](#) も用意しています。

Amazon SES API に直接アクセスする場合であれ、AWS SDK、AWS Command Line Interface、または AWS Tools for Windows PowerShell を介して間接的にアクセスする場合であれ、Amazon SES API では、E メールメッセージの構成に対する制御の程度に応じて、2 つの方法で E メールを送信できます。

- フォーマット済み - Amazon SES は、正しくフォーマットされた E メールメッセージを構成して送信します。ユーザーは、送信元と宛先のアドレス、件名、メッセージ本文を入力するだけです。Amazon SES は残りのすべてを処理します。詳細については、「[Amazon SES API を使用してフォーマット済み Eメールを送信する \(p. 114\)](#)」を参照してください。

- Raw - E メールメッセージを手動で構成して送信します。自分で E メールヘッダーおよび MIME の種類を指定します。Eメールのフォーマット作業を自分で行った経験があれば、raw インターフェイスにより、メッセージの構成をより詳細に制御できます。詳細については、「[Amazon SES API を使用して raw E メールを送信する \(p. 114\)](#)」を参照してください。

このセクションのトピック:

- [Amazon SES API を使用してフォーマット済み E メールを送信する \(p. 114\)](#)
- [Amazon SES API を使用して raw E メールを送信する \(p. 114\)](#)

## Amazon SES API を使用してフォーマット済み E メールを送信する

フォーマット済み E メールを送信するには、AWS Management Console を使用する方法、アプリケーションから Amazon SES API を直接呼び出す方法、AWS SDK、AWS Command Line Interface、または AWS Tools for Windows PowerShell から間接的に呼び出す方法があります。

Amazon SES API は、SendEmail アクションを提供し、フォーマット済み E メールを構成して送信できるようにします。SendEmail には、送信元アドレス、宛先アドレス、メッセージの件名、およびメッセージの本文 (テキスト、HTML、またはその両方) が必要です。SendEmail の詳細については、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。

### Note

E メールアドレスは、7 ビット ASCII 文字列になっている必要があります。送信先または送信元の E メールアドレス内で、ドメインの部分に unicode 文字が含まれる場合は、Punycode を使用してドメインをエンコードする必要があります。詳細については、[RFC 3492](#)を参照してください。

AWS SDK for Java または AWS SDK for .NET を使用したフォーマット済みメッセージを構成する方法の例は、それぞれ「[AWS SDK for Java を使用して E メールを送信する \(p. 35\)](#)」または「[AWS SDK for .NET を使用して E メールを送信する \(p. 32\)](#)」を参照してください。

SendEmail に複数の呼び出しを作成する場合の E メール送信速度向上方法については、「[Amazon SES のスループットを上げる \(p. 488\)](#)」を参照してください。

## Amazon SES API を使用して raw E メールを送信する

高度にカスタマイズされたメッセージを受信者に送信するために、Amazon SES SendRawEmail オペレーションを使用できます。

このセクションには、Amazon SES API を使用して raw E メールを作成および送信する手順が含まれています。

### E メールヘッダーフィールドについて

Simple Mail Transfer Protocol (SMTP) は、E メールエンベロープとそのパラメータのいくつかを定義することにより、E メールメッセージの送信方法を定義しますが、メッセージの内容については何も定義しません。一方、Internet Message Format ( [RFC 5322](#) ) は、メッセージの構成方法を定義します。

Internet Message Format の仕様に従って、すべてのメッセージ E メールはヘッダーと本文から構成されます。ヘッダーはメッセージのメタデータで構成され、本文にメッセージそのものが含まれます。E メールヘッダーと本文の詳細については、「[E メール形式と Amazon SES \(p. 14\)](#)」を参照してください。

### MIME の使用

SMTP プロトコルはもともと 7 ビット ASCII 文字のみを含む E メールメッセージを送信するように設計されていました。この仕様により、ASCII 以外のテキストエンコード (Unicode など)、バイナリコンテン

ツ、または添付ファイルでは SMTP が不十分になります。多目的インターネットメール拡張 (MIME) 標準は、SMTP を使用して、他の多くの種類のコンテンツを送信できるようにするために開発されたものです。

MIME 標準には、メッセージ本文を複数のパートに分割し、パートごとに、どのような操作を行うかを指定する機能があります。たとえば、E メールメッセージ本文の、あるパートはプレーンテキスト、別のパートは HTML という場合があります。さらに、MIME では、E メールメッセージに 1 つ以上の添付ファイルを含めることができます。メッセージの受取人は、E メールクライアント内から添付ファイルを見たり、保存したりできます。

メッセージヘッダーとコンテンツとは空白行で分離されます。Eメールの各パートは、boundary で分離されます。boundary は、各パートの開始と終了を示す文字列です。

次の例のマルチパートメッセージには、テキストと HTML パートが含まれています。また、添付ファイルが含まれます。

```
From: "Sender Name" <sender@example.com>
To: recipient@example.com
Subject: Customer service contact info
Content-Type: multipart/mixed;
    boundary="a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a"

--a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a
Content-Type: multipart/alternative;
    boundary="sub_a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a"

--sub_a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: quoted-printable

Please see the attached file for a list of customers to contact.

--sub_a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a
Content-Type: text/html; charset=iso-8859-1
Content-Transfer-Encoding: quoted-printable

<html>
<head></head>
<body>
<h1>Hello!</h1>
<p>Please see the attached file for a list of customers to contact.</p>
</body>
</html>

--sub_a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a--

--a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a
Content-Type: text/plain; name="customers.txt"
Content-Description: customers.txt
Content-Disposition: attachment;filename="customers.txt";
    creation-date="Sat, 05 Aug 2017 19:35:36 GMT";
Content-Transfer-Encoding: base64

SUQsRmlzc3ROYW11LEExhc3ROYW11LENvdW50cnkKMzQ4LEpvaG4sU3RpbGVzLENhbmFkYQo5MjM4
OSxKaWUsTG11LENoaW5hCjczNCxTaGlybGV5LFJvZHZHJpZ3VleixVbml0ZWQgU3RhdGVzCjI4OTMs
QW5heWESsX1lbmdhcixJbmRyYQ==

--a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a--
```

メッセージのコンテンツタイプが multipart/mixed であることから、メッセージに多数のパートがあり (この例では、本文および添付ファイル)、受信するクライアントは各パートを別々に扱う必要があることがわかります。本文セクション内に入れ子になっているのは、multipart/alternative コンテンツタイプを使用する 2 番目のパートです。このコンテンツタイプは、各パートに同じコンテンツの代替パー

ジョンが含まれる (この場合はテキストバージョンおよび HTML バージョン) ことを示します。受取人の E メールクライアントで HTML コンテンツを表示できる場合は、メッセージ本文の HTML バージョンが表示されます。受取人の E メールクライアントで HTML コンテンツを表示できない場合は、メッセージ本文のプレーンテキストバージョンが表示されます。メッセージの両方のバージョンには添付ファイルも含まれます (この場合、一部の顧客名を含むショートテキストファイル)。

この例のように MIME パートを他のパートに入れ子にすると、入れ子になったパートは、親のパートの boundary パラメータとは異なる boundary パラメータを使用する必要があります。これらの境界は固有の文字列でなければなりません。MIME パーツ間の境界を定義するには、2 つのハイフンを (--) タイプし、その後に境界文字列が続きます。MIME パーツの最後に、境界文字列の先頭および末尾の両方に 2 つのハイフンを置きます。

## MIME エンコード

古いシステムとの互換性を維持するために、Amazon SES は [RFC 2821](#) で定義されている SMTP の 7 ビット ASCII 制限を優先します。非 ASCII 文字を含むコンテンツを送信する場合は、これらの文字を 7 ビット ASCII 文字を使用する形式にエンコードする必要があります。

## E メールアドレス

メッセージエンベロープで使用された E メールアドレスをエンコードするには、Punycode エンコードを使用します。

たとえば、张伟@example.com に E メールを送信するには、アドレスのローカル部分 (@ 記号の前部分) で Punycode エンコーディングを使用します。結果として得られるエンコードされたアドレスは、xn--cpqy30b@example.com です。

### Note

このルールは、メッセージヘッダーではなく、メッセージエンベロープで指定する E メールアドレスにのみ適用されます。SendRawEmail API を使用する場合、Source および Destinations パラメータで指定したアドレスは、それぞれエンベロープの送信者と受取人を定義します。

Punycode エンコードの詳細については、[RFC 3492](#) を参照してください。

## E メールヘッダー

メッセージヘッダーをエンコードするには、MIME encoded-word 構文を使用します。MIME encoded word 構文では、次の形式が使用されます。

```
=?charset?encoding?encoded-text?=
```

*encoding* の値は *Q* または *B* となります。エンコードの値が *Q* の場合、値 *encoded-text* には Q エンコードを使用する必要があります。エンコードの値が *B* の場合、*encoded-text* の値には base64 エンコードを使用する必要があります。

たとえば、「Як ти поживаєш?» を使用する場合 Eメールの件名に次のエンコードのいずれかを使用することができます。

- Q エンコード

```
=?utf-8?Q?=D0=AF=D0=BA_ =D1=82=D0=B8_ =D0=BF=D0=BE=D0=B6=D0=B8=D0=B2=D0=B0=D1=94=D1=88=3F?=?
```

- Base64 エンコード

```
=?utf-8?B?0K/QuiDRgtC4INC/0L7QttC40LLQsNGU0Yg/?=?
```

Q エンコードの詳細については、[RFC 2047](#) を参照してください。base64 エンコードの詳細については、[RFC 2045](#) を参照してください。

## メッセージ本文

メッセージの本文をエンコードするには、quoted-printable エンコードまたは base64 エンコードを使用できます。次に、Content-Transfer-Encoding ヘッダーを使用して、使用するエンコードスキームを指定します。

たとえば、メッセージの本文に次のテキストが含まれているとします。

१९७२ मे रे टॉमलंसिन ने पहला ई-मेल सेंदश भेजा | रे टॉमलंसिन ने ही सर्वप्रथम @ चनिह का चयन किया और इन्ही को ईमेल का आविष्कारक माना जाता है

base64 エンコードを使用してこのテキストをエンコードする場合、最初に以下のヘッダーを指定します。

```
Content-Transfer-Encoding: base64
```

次に、E メールの本文セクションに、base64 でエンコードされたテキストを含めます。

```
4KWn4KWv4KWt4KWoIOckruClhyDgpLdGpYcg4KSf4KWJ4KSu4KSy4KS/4KSC4KS44KS0IOckqOC1  
hyDgpKrgpLngpLlGpL4g4KSILeCkruClh+CksidGpLjgpILgpKbgpYfgpLYg4KSt4KWH4KSc4KS+  
IHwg4KSv4KWHIOckn+ClieCkruCksuCkv+CkguCkuOCkqCDgpKjgpYcg4KS54KWAIOckuOCksOC1  
jeCkteCkquCljeCksOCkpeCkriBAIOckmuCkv+CkqOC1jeCkuSDgpJXgpL4g4KSa4KSv4KS0IOck  
leCkv+Ckr+CkviDgpJTgpLAg4KSH4KS04KWN4KS54KWAIOckleClidyDgpIjgpK7gpYfgpLIg4KSv  
4KS+IOckhuCkteCkv+Ckt+CljeCkleCkvuCksOCk1SDgpK7gpL7gpKjgpL4g4KSc4KS+4KSk4KS+  
IOckueCliaO=
```

### Note

場合によっては、Amazon SES を使用して送信するメッセージに 8 ビットの Content-Transfer-Encoding を使用できます。ただし、Amazon SES がメッセージを変更する必要がある場合 (たとえば、[オープンとクリックの追跡 \(p. 518\)](#)を使用した場合)、8 ビットでエンコードされたコンテンツは、受取人の受信トレイに届いたときに正しく表示されないことがあります。このため、7 ビットの ASCII 以外のコンテンツは常にエンコードする必要があります。

## 添付ファイル

E メールにファイルを添付するには、base64 エンコードを使用して添付ファイルをエンコードする必要があります。添付ファイルは通常、次のヘッダーを含む専用の MIME メッセージ部分に配置されています。

- Content-Type: 添付ファイルの種類。一般的な MIME Content-Type 宣言の例を以下に示します。
  - プレーンテキストファイル: Content-Type: text/plain; name="sample.txt"
  - Microsoft Word ドキュメント: Content-Type: application/msword; name="document.docx"
  - JPG イメージ: Content-Type: image/jpeg; name="photo.jpeg"
- Content-Disposition: 受取人の E メールクライアントがコンテンツをどのように処理するかを指定します。添付ファイルの場合、この値は Content-Disposition: attachment です。
- Content-Transfer-Encoding: 添付ファイルのエンコードに使用されるスキーム。添付ファイルでは、ほとんどの場合この値は base64 です。

Amazon SES は最も一般的なファイルタイプに対応しています。Amazon SES が対応していないファイルの種類の一覧については、「[サポート対象外の添付ファイルのタイプ \(p. 525\)](#)」を参照してください。

## Amazon SES API を使用して raw E メールを送信する

Amazon SES API は SendRawEmail アクションを提供します。これにより、指定したフォーマットの E メールメッセージを構成して送信できるようになります。SendRawEmail の詳細については、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。

## Note

SendRawEmail に複数の呼び出しを作成する場合の E メール送信速度向上方法については、「[Amazon SES のスループットを上げる \(p. 488\)](#)」を参照してください。

メッセージ本文には、正しくフォーマットされ、適切なヘッダーフィールドとメッセージ本文がエンコードされた raw E メールメッセージを含める必要があります。アプリケーション内で raw メッセージを手動で構成することはできますが、既存のメールライブラリを使用して構成するほうが、はるかに簡単です。

## Java

次のコード例は、[JavaMail](#) ライブラリと [AWS SDK for Java](#) を使用して raw E メールを作成および送信する方法を示しています。

```
package com.amazonaws.samples;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.nio.ByteBuffer;
import java.util.Properties;

// JavaMail libraries. Download the JavaMail API
// from https://javaee.github.io/javamail/
import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;

// AWS SDK libraries. Download the AWS SDK for Java
// from https://aws.amazon.com/sdk-for-java
import com.amazonaws.regions.Regions;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailService;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailServiceClientBuilder;
import com.amazonaws.services.simpleemail.model.RawMessage;
import com.amazonaws.services.simpleemail.model.SendRawEmailRequest;

public class AmazonSESSample {

    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    private static String SENDER = "Sender Name <sender@example.com>";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    private static String RECIPIENT = "recipient@example.com";

    // Specify a configuration set. If you do not want to use a configuration
    // set, comment the following variable, and the
    // ConfigurationSetName=CONFIGURATION_SET argument below.
    private static String CONFIGURATION_SET = "ConfigSet";

    // The subject line for the email.
    private static String SUBJECT = "Customer service contact info";

    // The full path to the file that will be attached to the email.
    // If you're using Windows, escape backslashes as shown in this variable.
```

```
private static String ATTACHMENT = "C:\\Users\\sender\\customers-to-contact.xlsx";

// The email body for recipients with non-HTML email clients.
private static String BODY_TEXT = "Hello,\r\n"
    + "Please see the attached file for a list "
    + "of customers to contact.";

// The HTML body of the email.
private static String BODY_HTML = "<html>"
    + "<head></head>"
    + "<body>"
    + "<h1>Hello!</h1>"
    + "<p>Please see the attached file for a "
    + "list of customers to contact.</p>"
    + "</body>"
    + "</html>";

public static void main(String[] args) throws AddressException, MessagingException,
IOException {

    Session session = Session.getDefaultInstance(new Properties());

    // Create a new MimeMessage object.
    MimeMessage message = new MimeMessage(session);

    // Add subject, from and to lines.
    message.setSubject(SUBJECT, "UTF-8");
    message.setFrom(new InternetAddress(SENDER));
    message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(RECIPIENT));

    // Create a multipart/alternative child container.
    MimeMultipart msg_body = new MimeMultipart("alternative");

    // Create a wrapper for the HTML and text parts.
    MimeBodyPart wrap = new MimeBodyPart();

    // Define the text part.
    MimeBodyPart textPart = new MimeBodyPart();
    textPart.setContent(BODY_TEXT, "text/plain; charset=UTF-8");

    // Define the HTML part.
    MimeBodyPart htmlPart = new MimeBodyPart();
    htmlPart.setContent(BODY_HTML, "text/html; charset=UTF-8");

    // Add the text and HTML parts to the child container.
    msg_body.addBodyPart(textPart);
    msg_body.addBodyPart(htmlPart);

    // Add the child container to the wrapper object.
    wrap.setContent(msg_body);

    // Create a multipart/mixed parent container.
    MimeMultipart msg = new MimeMultipart("mixed");

    // Add the parent container to the message.
    message.setContent(msg);

    // Add the multipart/alternative part to the message.
    msg.addBodyPart(wrap);

    // Define the attachment
    MimeBodyPart att = new MimeBodyPart();
    DataSource fds = new FileDataSource(ATTACHMENT);
    att.setDataHandler(new DataHandler(fds));
    att.setFileName(fds.getName());
```

```
// Add the attachment to the message.
msg.addBodyPart(att);

// Try to send the email.
try {
    System.out.println("Attempting to send an email through Amazon SES "
        +"using the AWS SDK for Java...");

    // Instantiate an Amazon SES client, which will make the service
    // call with the supplied AWS credentials.
    AmazonSimpleEmailService client =
        AmazonSimpleEmailServiceClientBuilder.standard()
        // Replace US_WEST_2 with the AWS Region you're using for
        // Amazon SES.
        .withRegion(Regions.US_WEST_2).build();

    // Print the raw email content on the console
    PrintStream out = System.out;
    message.writeTo(out);

    // Send the email.
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    message.writeTo(outputStream);
    RawMessage rawMessage =
        new RawMessage(ByteBuffer.wrap(outputStream.toByteArray()));

    SendRawEmailRequest rawEmailRequest =
        new SendRawEmailRequest(rawMessage)
        .withConfigurationSetName(CONFIGURATION_SET);

    client.sendRawEmail(rawEmailRequest);
    System.out.println("Email sent!");
} catch (Exception ex) {
    // Display an error if something goes wrong.
    System.out.println("Email Failed");
    System.err.println("Error message: " + ex.getMessage());
    ex.printStackTrace();
}
}
```

## Python

次のコード例は、[Python email.mime](#) パッケージと [AWS SDK for Python \(Boto\)](#) を使用して raw E メールを作成および送信する方法を示しています。

```
import os
import boto3
from botocore.exceptions import ClientError
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.application import MIMEApplication

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
SENDER = "Sender Name <sender@example.com>"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
RECIPIENT = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable, and the
# ConfigurationSetName=CONFIGURATION_SET argument below.
```



```
CONFIGURATION_SET = "ConfigSet"

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# The subject line for the email.
SUBJECT = "Customer service contact info"

# The full path to the file that will be attached to the email.
ATTACHMENT = "path/to/customers-to-contact.xlsx"

# The email body for recipients with non-HTML email clients.
BODY_TEXT = "Hello,\r\nPlease see the attached file for a list of customers to
contact."

# The HTML body of the email.
BODY_HTML = """\
<html>
<head></head>
<body>
<h1>Hello!</h1>
<p>Please see the attached file for a list of customers to contact.</p>
</body>
</html>
""

# The character encoding for the email.
CHARSET = "utf-8"

# Create a new SES resource and specify a region.
client = boto3.client('ses',region_name=AWS_REGION)

# Create a multipart/mixed parent container.
msg = MIMEMultipart('mixed')
# Add subject, from and to lines.
msg['Subject'] = SUBJECT
msg['From'] = SENDER
msg['To'] = RECIPIENT

# Create a multipart/alternative child container.
msg_body = MIMEMultipart('alternative')

# Encode the text and HTML content and set the character encoding. This step is
# necessary if you're sending a message with characters outside the ASCII range.
textpart = MIMEText(BODY_TEXT.encode(CHARSET), 'plain', CHARSET)
htmlpart = MIMEText(BODY_HTML.encode(CHARSET), 'html', CHARSET)

# Add the text and HTML parts to the child container.
msg_body.attach(textpart)
msg_body.attach(htmlpart)

# Define the attachment part and encode it using MIMEApplication.
att = MIMEApplication(open(ATTACHMENT, 'rb').read())

# Add a header to tell the email client to treat this part as an attachment,
# and to give the attachment a name.
att.add_header('Content-
Disposition', 'attachment', filename=os.path.basename(ATTACHMENT))

# Attach the multipart/alternative child container to the multipart/mixed
# parent container.
msg.attach(msg_body)

# Add the attachment to the parent container.
msg.attach(att)
#print(msg)
```

```
try:
    #Provide the contents of the email.
    response = client.send_raw_email(
        Source=SENDER,
        Destinations=[
            RECIPIENT
        ],
        RawMessage={
            'Data':msg.as_string(),
        },
        ConfigurationSetName=CONFIGURATION_SET
    )
# Display an error if something goes wrong.
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print("Email sent! Message ID:"),
    print(response['MessageId'])
```

## Amazon SES API を使用してパーソナライズされた E メールを送信する

Eメールのテンプレートを作成するには、[CreateTemplate](#) API オペレーションを使用できます。これらのテンプレートには、件名、Eメール本文のテキストパートと HTML パートが含まれています。件名および本文セクションには、受信者ごとにパーソナライズされたユニークな値を含めることもできます。

これらの機能を使用する際に、いくつかの制限と考慮事項があります。

- Amazon SES アカウントごとに最大 10,000 の E メールテンプレートを作成できます。
- 各テンプレートは、テキストと HTML パートの両方を含めて、最大 500 KB のサイズまで可能です。
- 各テンプレートに置換変数を回数の制限なく含めることができます。
- `SendBulkTemplatedEmail` オペレーションの呼び出しごとに、最大 50 個の送信先に E メールを送信できます。送信先には受取人のリスト、CC および BCC の受取人が含まれます。1 回の API の呼び出しで連絡できる送信先がアカウントの最大送信レートによって制限される場合があります。詳細については、「[Amazon SES 送信クォータの管理 \(p. 149\)](#)」を参照してください。

このセクションでは、Eメールテンプレートの作成手順と、パーソナライズされた Eメールの送信手順について説明します。

### Note

これらの手順では、AWS CLI がすでにインストールされ、設定されていることを前提としています。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

## パート 1: レンダリング失敗イベント通知を設定する

無効なパーソナライズコンテンツを含む Eメールを送信する場合、Amazon SES はメッセージを受け入れることもありますが、配信することはできません。このため、パーソナライズされた Eメールを送信する場合、Amazon SNS を通じてレンダリング失敗イベント通知を送信するように Amazon SES を設定する必要があります。レンダリング失敗イベント通知を受信した場合、無効なコンテンツが含まれていたメッセージを確認し、問題を修正して、もう一度メッセージを送信できます。

このセクションの手順はオプションですが、強くお勧めします。

レンダリング失敗イベント通知を設定するには

1. Amazon SNS トピックを作成する。手続きについては、Amazon Simple Notification Service デベロッパーガイドの「[トピックの作成](#)」を参照してください。
2. Amazon SNS トピックを購読します。たとえば、E メールでレンダリング失敗通知を受信する場合は、そのトピックを E メールエンドポイント (E メールアドレス) で購読する必要があります。

手順については、Amazon Simple Notification Service デベロッパーガイドの「[トピックを購読する](#)」を参照してください。

3. 「[the section called “Amazon SNS 送信先のセットアップ” \(p. 304\)](#)」の手順を完了して、設定セットをセットアップし、レンダリング失敗イベントを Amazon SNS トピックに発行します。

## パート 2: Eメールのテンプレートを作成する

このセクションでは、CreateTemplate API オペレーションを使用して、パーソナライゼーション属性を持つ新しい E メールテンプレートを作成します。

この手順は、AWS CLI がすでにインストールされ、設定されていることを前提としています。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

テンプレートを作成するには

1. テキストエディタで新規ファイルを作成します。ファイルに次のコードを貼り付けます。

```
{
  "Template": {
    "TemplateName": "MyTemplate",
    "SubjectPart": "Greetings, {{name}}!",
    "HtmlPart": "<h1>Hello {{name}},</h1><p>Your favorite animal is
    {{favoriteanimal}}.</p>",
    "TextPart": "Dear {{name}},\r\nYour favorite animal is {{favoriteanimal}}."
  }
}
```

このコードには次のプロパティが含まれています。

- **TemplateName** – テンプレートの名前。E メールを送信する場合に、この名前を参照します。
- **SubjectPart** – Eメールの件名。このプロパティには、置換タグが含まれる場合があります。これらのタグは以下のフォーマットを使用します: `{{tagname}}`。Eメールを送信するとき、各宛先の `tagname` に対する値を指定できます。

前述の例には、`{{name}}` および `{{favoriteanimal}}` の 2 つのタグが含まれます。

- **HtmlPart** – Eメールの HTML 本文。このプロパティには、置換タグが含まれる場合があります。
  - **TextPart** – Eメールのテキスト本文。Eメールクライアントに HTML Eメールが表示されない受信者には、このバージョンが表示されます。このプロパティには、置換タグが含まれる場合があります。
2. 前の例をニーズに合わせてカスタマイズし、`mytemplate.json` ファイルとして保存します。
  3. コマンドラインで次のコマンドを入力して、CreateTemplate API オペレーションを使用して新しいテンプレートを作成します。

```
aws ses create-template --cli-input-json file://mytemplate.json
```

## パート 3: パーソナライズされた E メールを送信する

作成した E メールテンプレートは、Eメールの送信に使用できます。テンプレートを使用して E メールを送信するのに使用できる 2 つの API オペレーションがあり、それは `SendTemplatedEmail` および `SendBulkTemplatedEmail` です。`SendTemplatedEmail` オペレーションは、一つへの送信先 (同じメールを受信する「To」、「CC」、「BCC」の集合) にカスタマイズされたメールを送信するのに役立ちます。`SendBulkTemplatedEmail` オペレーションは、単一の Amazon SES API 呼び出しで複数の宛先それぞれにユニークな E メールを送信するのに役立ちます。このセクションでは、AWS CLI で両方のオペレーションを使用して E メールを送信する方法の例を提供します。

### 単一の宛先にテンプレートに基づく E メールを送信する

単一の宛先に E メールを送信するには、`SendTemplatedEmail` オペレーションを使用できます。`Destination` オブジェクトにあるすべての受信者に、同じ E メールが届きます。

単一の宛先にテンプレートに基づく E メールを送信するには

1. テキストエディタで新規ファイルを作成します。ファイルに次のコードを貼り付けます。

```
{
  "Source": "Mary Major <mary.major@example.com>",
  "Template": "MyTemplate",
  "ConfigurationSetName": "ConfigSet",
  "Destination": {
    "ToAddresses": [ "alejandro.rosalez@example.com"
  ]
},
  "TemplateData": "{ \"name\": \"Alejandro\", \"favoriteanimal\": \"alligator\" }"
}
```

このコードには次のプロパティが含まれています。

- Source – 送信者の E メールアドレス。
- Template – E メールに対して適用するテンプレートの名前。
- ConfigurationSetName – E メールを送信するときに使用する設定セットの名前。

#### Note

レンダリング失敗イベントを Amazon SNS に発行するように設定された設定セットを使用することをお勧めします。詳細については、「[the section called “パート 1: 通知のセットアップ” \(p. 122\)](#)」を参照してください。

- Destination - 受信者のアドレス。複数の「To」、「CC」、「BCC」アドレスを含めることができます。`SendTemplatedEmail` オペレーションを使用した場合、すべての受信者に同じ E メールが届きます。
  - TemplateData – キーと値のペアを含むエスケープされた JSON 文字列。キーはテンプレート内の変数 (`{{name}}` など) に対応します。値は E メール内の変数を置き換える内容を表します。
2. ニーズに合わせて上記のコードの値を変更し、ファイルを `myemail.json` という名前で保存します。
  3. コマンドラインで、次のコマンドを入力して E メールを送信します。

```
aws ses send-templated-email --cli-input-json file://myemail.json
```

## 複数の送信先にテンプレートに基づく E メールを送信する

SendBulkTemplatedEmail オペレーションを使用して、単一の API 呼び出しで、複数の送信先にメールを送信できます。Amazon SES は、各 Destination オブジェクトで、1名または複数の受信者に一意の E メールを送信します。

複数の送信先にテンプレートに基づく E メールを送信するには

1. テキストエディタで新規ファイルを作成します。ファイルに次のコードを貼り付けます。

```
{
  "Source": "Mary Major <mary.major@example.com>",
  "Template": "MyTemplate",
  "ConfigurationSetName": "ConfigSet",
  "Destinations": [
    {
      "Destination": {
        "ToAddresses": [
          "anaya.iyengar@example.com"
        ]
      },
      "ReplacementTemplateData": "{ \"name\": \"Anaya\", \"favoriteanimal\": \"angelfish\" }"
    },
    {
      "Destination": {
        "ToAddresses": [
          "liu.jie@example.com"
        ]
      },
      "ReplacementTemplateData": "{ \"name\": \"Liu\", \"favoriteanimal\": \"lion\" }"
    },
    {
      "Destination": {
        "ToAddresses": [
          "shirley.rodriguez@example.com"
        ]
      },
      "ReplacementTemplateData": "{ \"name\": \"Shirley\", \"favoriteanimal\": \"shark\" }"
    },
    {
      "Destination": {
        "ToAddresses": [
          "richard.roe@example.com"
        ]
      },
      "ReplacementTemplateData": "{}"
    }
  ],
  "DefaultTemplateData": "{ \"name\": \"friend\", \"favoriteanimal\": \"unknown\" }"
}
```

このコードには次のプロパティが含まれています。

- Source – 送信者の E メールアドレス。
- Template – E メールに対して適用するテンプレートの名前。
- ConfigurationSetName – E メールを送信するときに使用する設定セットの名前。

## Note

レンダリング失敗イベントを Amazon SNS に発行するように設定された設定セットを使用することをお勧めします。詳細については、「[the section called “パート 1: 通知のセットアップ” \(p. 122\)](#)」を参照してください。

- Destinations – 1 つ以上の宛先が含まれている配列。
  - Destination - 受信者のアドレス。複数の「To」、「CC」、「BCC」アドレスを含めることができます。SendBulkTemplatedEmail オペレーションを使用した場合、同一の Destination オブジェクト内のすべての受信者に同じ E メールが届きます。
  - ReplacementTemplateData – キーと値のペアを含む JSON オブジェクト。キーはテンプレート内の変数 ({{name}} など) に対応します。値は E メール内の変数を置き換える内容を表します。
  - DefaultTemplateData – キーと値のペアを含む JSON オブジェクト。キーはテンプレート内の変数 ({{name}} など) に対応します。値は E メール内の変数を置き換える内容を表します。このオブジェクトにはフォールバックデータが含まれています。Destination オブジェクトの ReplacementTemplateData プロパティに空の JSON オブジェクトが含まれる場合、DefaultTemplateData プロパティの値が使用されます。
- 2. ニーズに合わせて上記のコードの値を変更し、ファイルを mybulkemail.json という名前で保存します。
- 3. コマンドラインで、次のコマンドを入力してバルク E メールを送信します。

```
aws ses send-bulk-templated-email --cli-input-json file://mybulkemail.json
```

## Eメールの高度なパーソナライズ

Amazon SES のテンプレート機能は、Handlebars テンプレートシステムを基盤としています。Handlebars を使用して、入れ子の属性、配列の反復処理、基本的な条件ステートメント、インラインパーシャルの作成などの高度な機能を含むテンプレートを作成することができます。このセクションでは、これらの機能の例を示します。

このセクションに記載されている機能以外にも、Handlebars にはさまざまな機能を提供します。詳細については、[handlebarsjs.com](http://handlebarsjs.com) の「[Built-In Helpers](#)」を参照してください。

### トピック

- [入れ子の属性の解析 \(p. 126\)](#)
- [リストに対する反復処理 \(p. 127\)](#)
- [基本的な条件ステートメントの使用 \(p. 128\)](#)
- [インラインパーシャルの作成 \(p. 130\)](#)

### 入れ子の属性の解析

Handlebars には入れ子パスのサポートが含まれています。入れ子パスを使うことで、複雑な顧客データを簡単に整理し、E メールテンプレート内でそのデータを参照できるようになります。

たとえば、受信者のデータをいくつかの一般的なカテゴリで整理し、各カテゴリに詳細な情報を含めることができます。以下のコード例は、このような構造を 1 人の受信者で実現しています。

```
{
  "meta": {
    "userId": "51806220607"
  },
  "contact": {
    "firstName": "Anaya",
    "lastName": "Iyengar",
  }
}
```

```
    "city": "Bengaluru",
    "country": "India",
    "postalCode": "560052"
  },
  "subscription": [
    {
      "interest": "Sports"
    },
    {
      "interest": "Travel"
    },
    {
      "interest": "Cooking"
    }
  ]
}
```

E メールテンプレートで入れ子の属性を参照するには、親属性の名前、ピリオド (.)、値を含める属性の名前の順に指定します。たとえば前述の例で示したデータ構造を使用し、それぞれの受信者のファーストネームを E メールテンプレートに含める場合は、次のテキストを E メールテンプレートに含めます。Hello `{{contact.firstName}}`!

Handlebars は、複数の階層にわたって入れ子になったパスを解析できます。つまり、テンプレートのデータ構造を柔軟に設定することができます。

## リストに対する反復処理

each ヘルパー関数は、配列内の項目を反復処理します。以下のコード例の E メールテンプレートでは、each ヘルパー関数を使用して、それぞれの受信者の関心を項目別に記載したリストを作成しています。

```
{
  "Template": {
    "TemplateName": "Preferences",
    "SubjectPart": "Subscription Preferences for {{contact.firstName}}
    {{contact.lastName}}",
    "HtmlPart": "<h1>Your Preferences</h1>
    <p>You have indicated that you are interested in receiving
    information about the following subjects:</p>
    <ul>
      {{#each subscription}}
      <li>{{interest}}</li>
      {{/each}}
    </ul>
    <p>You can change these settings at any time by visiting
    the <a href=https://www.example.com/preferences/i.aspx?
    id={{meta.userId}}>
    Preference Center</a>.</p>",
    "TextPart": "Your Preferences\n\nYou have indicated that you are interested in
    receiving information about the following subjects:\n
    {{#each subscription}}
    - {{interest}}\n
    {{/each}}
    \nYou can change these settings at any time by
    visiting the Preference Center at
    https://www.example.com/preferences/i.aspx?id={{meta.userId}}"
  }
}
```

### Important

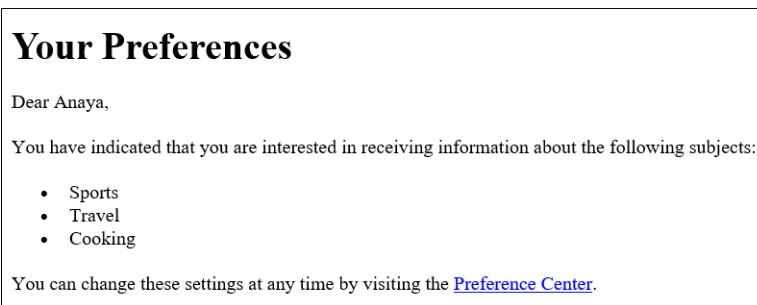
上記のコード例の `HtmlPart` および `TextPart` 属性には、読みやすくするための改行が含まれています。テンプレートの JSON ファイルでは、これらの値に改行を含めることはできません。こ

の例を独自の JSON ファイルにコピーして貼り付ける場合は、HtmlPart および TextPart セクションの改行と不要なスペースを削除してから次に進んでください。

テンプレートを作成したら、SendTemplatedEmail または SendBulkTemplatedEmail オペレーションを使用し、このテンプレートを使って受信者に E メールを送信します。Interests オブジェクトに値が少なくとも 1 つあれば、それぞれの受信者は、項目別の関心リストを含む E メールを受信します。以下の例は、上記のテンプレートを使用した複数の受信者への E メール送信に使用できる JSON ファイルを示しています。

```
{
  "Source": "Sender Name <sender@example.com>",
  "Template": "Preferences",
  "Destinations": [
    {
      "Destination": {
        "ToAddresses": [
          "anaya.iyengar@example.com"
        ]
      },
      "ReplacementTemplateData": "{\"meta\":{\"userId\":\"51806220607\"},\"contact\":{\"firstName\":\"Anaya\",\"lastName\":\"Iyengar\"},\"subscription\":[{\"interest\":\"Sports\"},{\"interest\":\"Travel\"},{\"interest\":\"Cooking\"}]}"
    },
    {
      "Destination": {
        "ToAddresses": [
          "shirley.rodriguez@example.com"
        ]
      },
      "ReplacementTemplateData": "{\"meta\":{\"userId\":\"1981624758263\"},\"contact\":{\"firstName\":\"Shirley\",\"lastName\":\"Rodriguez\"},\"subscription\":[{\"interest\":\"Technology\"},{\"interest\":\"Politics\"}]}"
    }
  ],
  "DefaultTemplateData": "{\"meta\":{\"userId\":\"\"},\"contact\":{\"firstName\":\"Friend\",\"lastName\":\"\"},\"subscription\":[ ]}"
}
```

上記の例でリストされた受信者に SendBulkTemplatedEmail オペレーションを使用して E メールを送信すると、受信者には以下の画像のようなメッセージが届きます。



## 基本的な条件ステートメントの使用

このセクションの内容は、前のセクションで説明した例に基づいています。前のセクションの例では、each ヘルパーを使用して関心のリストを反復処理しています。ただし、関心が指定されていない受信者には、空白のリストが記載された E メールが届くことになります。{{if}} ヘルパーを使用すると、テンプレート内のデータに特定の属性が存在する場合に、E メールに異なるフォーマットを適用することができます。以下のコードでは、{{if}} ヘルパーを使用して、Subscription 配列に何らかの値が含まれている場合に、前のセクションの箇条書きリストを表示しています。配列が空の場合、別のテキストブロックが表示されます。



```
{
  "Template": {
    "TemplateName": "Preferences2",
    "SubjectPart": "Subscription Preferences for {{contact.firstName}}
    {{contact.lastName}}",
    "HtmlPart": "<h1>Your Preferences</h1>
    <p>Dear {{contact.firstName}},</p>
    {{#if subscription}}
    <p>You have indicated that you are interested in receiving
    information about the following subjects:</p>
    <ul>
    {{#each subscription}}
    <li>{{interest}}</li>
    {{/each}}
    </ul>
    <p>You can change these settings at any time by visiting
    the <a href=https://www.example.com/preferences/i.aspx?
    id={{meta.userId}}>
      Preference Center</a>.</p>
    {{else}}
    <p>Please update your subscription preferences by visiting
    the <a href=https://www.example.com/preferences/i.aspx?
    id={{meta.userId}}>
      Preference Center</a>.
    {{/if}}",
    "TextPart": "Your Preferences\n\nDear {{contact.firstName}},\n\n
    {{#if subscription}}
    You have indicated that you are interested in receiving
    information about the following subjects:\n
    {{#each subscription}}
    - {{interest}}\n
    {{/each}}
    \nYou can change these settings at any time by visiting the
    Preference Center at https://www.example.com/preferences/i.aspx?
    id={{meta.userId}}.
    {{else}}
    Please update your subscription preferences by visiting the
    Preference Center at https://www.example.com/preferences/i.aspx?
    id={{meta.userId}}.
    {{/if}}"
  }
}
```

### Important

上記のコード例の `HtmlPart` および `TextPart` 属性には、読みやすくするための改行が含まれています。テンプレートの JSON ファイルでは、これらの値に改行を含めることはできません。この例を独自の JSON ファイルにコピーして貼り付ける場合は、`HtmlPart` および `TextPart` セクションの改行と不要なスペースを削除してから次に進んでください。

以下の例は、上記のテンプレートを使用した複数の受信者への E メール送信に使用できる JSON ファイルを示しています。

```
{
  "Source": "Sender Name <sender@example.com>",
  "Template": "Preferences2",
  "Destinations": [
    {
      "Destination": {
        "ToAddresses": [
          "anaya.iyengar@example.com"
        ]
      }
    }
  ],
}
```

```
"ReplacementTemplateData": {"meta": {"userId": "51806220607"}, "contact": {"firstName": "Anaya", "lastName": "Iyengar"}, "subscription": [{"interest": "Sports"}, {"interest": "Cooking"}]}, {"Destination": {"ToAddresses": ["shirley.rodriguez@example.com"]}, "ReplacementTemplateData": {"meta": {"userId": "1981624758263"}, "contact": {"firstName": "Shirley", "lastName": "Rodriguez"}}, {"DefaultTemplateData": {"meta": {"userId": ""}, "contact": {"firstName": "Friend", "lastName": ""}, "subscription": []}]}
```

この例の場合、テンプレートのデータに関心リストが含まれている受信者は、前のセクションで示した例と同じ E メールを受け取ります。一方、テンプレートデータに関心が含まれていない受信者には、以下の画像のような E メールが届きます。

## Your Preferences

Dear Shirley,

Please update your subscription preferences by visiting the [Preference Center](#).

## インラインパーシャルの作成

インラインパーシャルを使用して、文字列の繰り返しを含むテンプレートを簡素化することができます。たとえば、テンプレートの先頭に以下のコードを追加することで、受信者のファーストネームと (存在する場合は) ラストネームを含むインラインパーシャルを作成できます。

```
{{#* inline \"fullName\"}}{{firstName}}{{#if lastName}} {{lastName}}{{/if}}{{/inline}}\n
```

### Note

テンプレートの内容と \n ブロックを分離するためには、改行文字 ({{inline}}) が必要です。最終的な出力には改行は表示されません。

fullName パーシャルを作成したら、以下の例のように、大なり記号 (>)、スペース、パーシャルの名前を指定することで ({{> fullName}})、このパーシャルをテンプレートの任意の場所に含めることができます。インラインパーシャルは、メールの特定部分をまたがって適用されません。たとえば E メール of HTML バージョンとテキストバージョンに同じインラインパーシャルを使用する場合、HtmlPart および TextPart セクションの両方に、このインラインパーシャルを定義する必要があります。

インラインパーシャルを配列を反復処理する際に使用することもできます。以下のコードを使用して、fullName インラインパーシャルを使用するテンプレートを作成できます。この例では、受信者の名前とその他の名前の配列の両方に、インラインパーシャルが適用されています。

```
{
  "Template": {
    "TemplateName": "Preferences3",
    "SubjectPart": "{{firstName}}'s Subscription Preferences",
    "HtmlPart": "{{#* inline \"fullName\"}}
      {{firstName}}{{#if lastName}} {{lastName}}{{/if}}
    {{/inline~}}\n
    <h1>Hello {{> fullName}}!</h1>
    <p>You have listed the following people as your friends:</p>
    <ul>
```

```
        {{#each friends}}
        <li>{{> fullName}}</li>
        {{/each}}</ul>",
    "TextPart": "{{#* inline \"fullName\"}}
        {{firstName}}{{#if lastName}} {{lastName}}{{/if}}
        {{/inline-}}\n
        Hello {{> fullName}}! You have listed the following people
        as your friends:\n
        {{#each friends}}
        - {{> fullName}}\n
        {{/each}}"
    }
}
```

### Important

上記のコード例の `HtmlPart` および `TextPart` 属性には、読みやすくするための改行が含まれています。テンプレートの JSON ファイルでは、これらの値に改行を含めることはできません。この例を独自の JSON ファイルにコピーして貼り付ける場合は、これらのセクションの改行と不要なスペースを削除してください。

## E メールテンプレートの管理

[E メールテンプレートの作成 \(p. 122\)](#)に加えて、Amazon SES API を使用して、既存のテンプレートを更新または削除したり、既存のテンプレートを一覧表示したり、テンプレートの内容を表示したりすることができます。

このセクションでは、AWS CLI を使用して Amazon SES テンプレートに関連するタスクを実行する手順について説明します。

### Note

これらの手順では、AWS CLI がすでにインストールされ、設定されていることを前提としています。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

## E メールテンプレートの一覧表示

Amazon SES API の `ListTemplates` オペレーションを使用して、既存の E メールテンプレートのリストを表示します。

E メールテンプレートの一覧を表示するには

- コマンドラインで以下のコマンドを入力します。

```
aws ses list-templates
```

現在のリージョンの Amazon SES アカウントに既存の E メールテンプレートがある場合、このコマンドは次の例のようなレスポンスを返します。

```
{
  "TemplatesMetadata": [
    {
      "Name": "SpecialOffers",
      "CreatedTimestamp": "2020-08-05T16:04:12.640Z"
    },
    {
      "Name": "NewsAndUpdates",
      "CreatedTimestamp": "2019-10-03T20:03:34.574Z"
    }
  ]
}
```

```
}
```

テンプレートをまだ作成していない場合、このコマンドはメンバーのいない `TemplatesMetadata` オブジェクトを返します。

## 特定の E メール テンプレートの内容の表示

Amazon SES API の `GetTemplate` オペレーションを使用して、特定の E メールテンプレートの内容を表示できます。

E メールテンプレートの内容を表示するには

- コマンドラインで以下のコマンドを入力します。

```
aws ses get-template --template-name MyTemplate
```

上記のコマンドで、*MyTemplate* を、表示したいテンプレートの名前に置き換えます。

指定したテンプレート名が Amazon SES アカウントに存在するテンプレートと一致する場合、このコマンドは次の例のようなレスポンスを返します。

```
{
  "Template": {
    "TemplateName": "TestMessage",
    "SubjectPart": "Amazon SES Test Message",
    "TextPart": "Hello! This is the text part of the message.",
    "HtmlPart": "<html>\n<body>\n<h2>Hello!</h2>\n<p>This is the HTML part of the
message.</p></body>\n</html>"
  }
}
```

指定したテンプレート名が Amazon SES アカウントに存在するテンプレートと一致しない場合、コマンドは `TemplateDoesNotExist` エラーを返します。

## E メールテンプレートの削除

Amazon SES API の `DeleteTemplate` オペレーションを使用して、特定の E メールテンプレートを削除できます。

E メールテンプレートを削除する

- コマンドラインで以下のコマンドを入力します。

```
aws ses delete-template --template-name MyTemplate
```

前述のコマンドで、*MyTemplate* を削除するテンプレート名で置き換えます。

このコマンドは出力を提供しません。`GetTemplate` (p. 132) オペレーションを使用して、テンプレートが削除されたことを確認できます。

## E メールテンプレートの更新

Amazon SES API の `UpdateTemplate` オペレーションを使用して、既存の E メールテンプレートを更新します。たとえば、このオペレーションは、E メールテンプレートの件名を変更する場合や、メッセージの本文を変更する必要がある場合に役立ちます。

## Eメールテンプレートを更新する

1. `GetTemplate` コマンドを使用し、コマンドラインに次のコマンドを入力して、既存のテンプレートを取得できます。

```
aws ses get-template --template-name MyTemplate
```

上記のコマンドで、*MyTemplate*を、テンプレートの更新名に置き換えます。

指定したテンプレート名が Amazon SES アカウントに存在するテンプレートと一致する場合、このコマンドは次の例のようなレスポンスを返します。

```
{
  "Template": {
    "TemplateName": "TestMessage",
    "SubjectPart": "Amazon SES Test Message",
    "TextPart": "Hello! This is the text part of the message.",
    "HtmlPart": "<html>\n<body>\n<h2>Hello!</h2>\n<p>This is the HTML part of the message.</p></body>\n</html>"
  }
}
```

2. テキストエディタで新規ファイルを作成します。前のコマンドの出力をファイルに貼り付けます。
3. 必要に応じてテンプレートを変更します。省略した行は、テンプレートから削除されます。たとえば、テンプレートの `SubjectPart` を変更したい場合のみ、`TextPart` および `HtmlPart` のプロパティを含める必要があります。

終了したら、`update_template.json` としてファイルを保存します。

4. コマンドラインで以下のコマンドを入力します。

```
aws ses update-template --cli-input-json file:///path/to/update_template.json
```

前述の上記のコマンドで、*path/to/update\_template.json* を、前のステップで作成した `update_template.json` ファイルのパスに置き換えます。

テンプレートが正常に更新された場合、このコマンドは出力を提供しません。[GetTemplate \(p. 132\)](#) オペレーションを使用して、テンプレートが更新されたことを確認できます。

指定したテンプレートが存在しない場合、このコマンドは `TemplateDoesNotExist` エラーを返します。テンプレートに `TextPart` または `HtmlPart` のプロパティ (またはその両方) が含まれていない場合、このコマンドは `InvalidParameterValue` エラーを返します。

## Amazon SES での Eメールの認証

Amazon Simple Email Service (Amazon SES) は、Simple Mail Transfer Protocol (SMTP) を使用して Eメールを送信します。SMTP 自体は認証を提供しません。実際の送信元を隠蔽したスパムの発信者から、他人を装って Eメールメッセージが送信される可能性があります。スパムの発信者は、Eメールヘッダーを改ざんし、送信元 IP アドレスを偽装することにより、そのメールメッセージが本物であると受取人に思い込ませることが可能です。

メールトラフィックを転送するほとんどの ISP は、Eメールの正当性を評価するための対策を講じています。Eメールが認証されているかどうかの確認は、ISP が講じているそうした対策の 1 つです。認証において送信者は、自分がその送信元アカウントの所有者であることを証明する必要があります。場合によっては、ISP は認証されない Eメールの送信を拒否します。配信性能を最大限確保するために、Eメールを認証することをお勧めします。

以降のセクションでは、ISP で使用される 2 つの認証メカニズム、Sender Policy Framework (SPF) とドメインキーアイデンティファイドメール (DKIM)、について説明するとともに、Amazon SES でそれらの標準を使用する方法について説明します。

- SPF では、E メールメッセージをその送信元のシステムにまでさかのぼって追跡することができます。「[Amazon SES における SPF を使った Eメールの認証 \(p. 134\)](#)」を参照してください。
- DKIM は、E メールメッセージに署名することで、自分のメッセージが本物であることと送信中に改ざんされていないことを ISP に証明するための標準規格です。「[Amazon SES における DKIM を使った Eメールの認証 \(p. 135\)](#)」を参照してください。
- SPF および DKIM に基づく DMARC (Domain-based Message Authentication, Reporting and Conformance) に準拠する方法については、「[Amazon SES での DMARC への準拠 \(p. 147\)](#)」を参照してください。

## Amazon SES における SPF を使った Eメールの認証

Sender Policy Framework (SPF) は、Eメールのなりすましを防ぐために設計された Eメールの検証標準です。ドメイン所有者は SPF を使用して、自分のドメインからメールを送信できるサーバーをメールプロバイダーに通知します。SPF は [RFC 7208](#) で定義されています。

SPF を設定するには、TXT レコードをドメインの DNS 設定に公開します。このレコードには、ドメインから Eメールを送信することを許可するサーバーの一覧が含まれます。Eメールプロバイダーは、ドメインからメッセージを受信すると、ドメインの DNS レコードをチェックして、承認されたサーバーから Eメールが送信されたことを確認します。

Amazon SES 経由で Eメールを送信する場合、送信するメッセージは、デフォルトで SPF チェックに合格します。Amazon SES は、AmazonSES.com のサブドメインであるメッセージごとに MAIL FROM ドメインを指定し、メッセージの送信メールサーバーはこのドメインと一致します。

オプションで、独自の SPF レコードを公開できます。SPF レコードを公開することにより、Eメールはドメインベースのメッセージ認証、レポート、および適合性 (DMARC) に準拠できます。詳細については、「[DMARC への準拠 \(p. 147\)](#)」を参照してください。

### SPF レコードの追加

SPF レコードを公開するには、ドメインの DNS 設定に新しい TXT レコードを追加する必要があります。DNS レコードを更新する手順は、使用する DNS プロバイダーやウェブホスティングプロバイダーによって異なります。

次のテーブルには、いくつかの主要なプロバイダーに関するドキュメントへのリンクが含まれます。このリストにはすべての内容が網羅されているわけではなく、このリストに含まれているからといって、他社の製品やサービスを支援または推奨するものではありません。お使いのプロバイダーがこの表にない場合でも、SPF レコードを公開できる場合があります。

DNS/ホスティングプロバイダー	ドキュメントのリンク
Amazon Route 53	<a href="#">Amazon Route 53 コンソールを使用して、レコードを作成</a>
GoDaddy	<a href="#">SPF レコードを追加</a>
Dreamhost	<a href="#">SPF レコードを追加する方法</a>
Cloudflare	<a href="#">CloudFlare内でDNSレコードを管理する</a>
HostGator	<a href="#">SPF レコード</a>

DNS/ホスティングプロバイダー	ドキュメントのリンク
Namecheap	<a href="#">ドメインの TXT/SPF/DKIM/DMARC レコードを追加する方法</a>
Names.co.uk	<a href="#">ドメインの DNS 設定の変更</a>
Wix	<a href="#">Wix アカウントで SPF レコードを追加または更新する</a>

ドメインに既存の SPF レコードがない場合は、次の値を使用して TXT レコードを発行します。レコード名は、DNS サービスによって空白または @ になります。

```
"v=spf1 include:amazonses.com ~all"
```

SPF レコードには、複数の include ステートメントを含めることができます。ドメインにすでに SPF レコードがある場合は、以下の形式を使用して、Amazon SES の include ステートメントを追加できません。

```
"v=spf1 include:example.com include:amazonses.com ~all"
```

## Amazon SES における DKIM を使った Eメールの認証

ドメインキーアイデンティファイドメール (DKIM) は、送信者が暗号化キーで E メールメッセージを署名できる規格です。E メールプロバイダーはこの署名を使用して、メッセージが転送中に第三者によって改ざんされていないことを検証します。

DKIM を使って送信される E メールメッセージには、DKIM-Signature というヘッダーフィールドがあり、メッセージに対し、暗号化を用いて署名したデータが格納されます。メッセージを受信したプロバイダーは、送信者の DNS レコードで発行されるパブリックキーを使用して、署名をデコードします。E メールプロバイダーはこの情報を使用して、メッセージが本物であるかどうかを判断します。

DKIM 署名はオプションです。DKIM 署名を使って E メールに署名することで、DKIM を準拠する E メールプロバイダーによる配信性能を強化することができます。Amazon SES には、DKIM 署名を使用してメッセージに署名するための 3 つのオプションがあります。

- Amazon SES によって自動的にすべての送信メッセージに DKIM 署名を追加する送信元の ID をセットアップするには、「[Amazon SES の Easy DKIM \(p. 135\)](#)」を参照してください。
- DKIM 認証に独自のパブリック/プライベートのキーペアを使用するには、「[独自の DKIM 認証トークンを Amazon SES で提供する \(p. 143\)](#)」を参照してください。
- SendRawEmail API を使って送信する E メールに独自の DKIM 署名を追加する場合は、「[Amazon SES 内で手動での DKIM 署名 \(p. 146\)](#)」を参照してください。

## Amazon SES の Easy DKIM

ID で Easy DKIM をセットアップすると、Amazon SES はその ID から送信するすべての E メールに 1024 ビットの DKIM キーを自動的に追加します。Easy DKIM を設定するには、Amazon SES コンソールあるいは API を使用できます。

### Note

Easy DKIM をセットアップするには、ドメインの DNS 設定を変更する必要があります。Route 53 を DNS プロバイダーとして使用している場合、Amazon SES は適切なレコードを

自動的に作成します。別の DNS プロバイダーを使用する場合は、使用するプロバイダーのドキュメントを参照して、ドメインの DNS 設定を変更する詳細を確認します。

Easy DKIM を正常に設定すると、「[Amazon SES でのドメインの検証 \(p. 62\)](#)」の手順を完了していない場合でも、DKIM 対応ドメインからの Eメールの送信を開始できます。

## Easy DKIM に関する考慮事項

Eメールの認証に Easy DKIM を使用する場合、次のルールが適用されます。

- 「送信元」アドレスで使用するドメインでのみ Easy DKIM を設定する必要があります。「Return-Path」あるいは「Reply-to」アドレスで使用するドメインに Easy DKIM をセットアップする必要はありません。
- Amazon SES は、複数の AWS リージョンで利用できます。複数の AWS リージョンを使用して Eメールを送信する場合、リージョンごとに Easy DKIM のセットアップを完了して、すべての Eメールに DKIM 署名があることを確認する必要があります。
- ドメインの検証時、特定のサブドメインに Easy DKIM がセットアップされていない限り、Easy DKIM 設定はこのドメインのすべてのサブドメインにも適用されます。
- 親ドメイン、サブドメインおよび Eメールアドレスに対して Easy DKIM をセットアップした場合、Amazon SES は以下の方法で Easy DKIM 設定を適用します。
  - サブドメインの DKIM 設定は、親ドメインの設定を上書きします。
  - Eメールアドレスの DKIM 設定の設定は、サブドメインの設定 (適用される場合) と親ドメインの設定を上書きします。

このトピックは、以下のセクションで構成されます。

- [ドメインにおける Easy DKIM のセットアップ \(p. 136\)](#)
- [Eメールアドレスで Easy DKIM をセットアップする \(p. 138\)](#)
- [Easy DKIM の管理 \(p. 139\)](#)

## ドメインにおける Easy DKIM のセットアップ

このセクションの手順では、ドメインに Easy DKIM をセットアップする方法を示します。ドメインに Easy DKIM をセットアップすると、[ドメインを検証する手順 \(p. 61\)](#)を完了していない場合でも、そのドメインから Eメールの送信を開始できます。

ドメインに Easy DKIM をセットアップするには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで、[Identity Management] の [Domains] を選択します。
3. ドメインのリストで、Easy DKIM をセットアップするドメインを選択します。

### Note

ドメインの検証プロセスをまだ開始していない場合は、「[Amazon SES でのドメインの検証 \(p. 62\)](#)」の手順を参照してください。

4. [DKIM で] で、[DKIM 設定の生成] を選択します。
5. このセクションに表示される 3 つの CNAME レコードをコピーします。または、[レコードセットを CSV としてダウンロード] を選択してコンピューターにこのレコードを保存することもできます。

以下のイメージは、[DKIM] セクションの例を示しています。



## ▼ DKIM

**DKIM settings for your domain have been generated.** The information below depends on who provides your DNS service; if your DNS service is provided by Amazon Route 53, you can use the records below. [Learn more](#)

**DKIM:** waiting on sender verification...

**DKIM Verification Status:** pending verification

To enable DKIM signing for your domain, the records below must be entered into your DNS and allow DKIM signing at that time. Note that verification of these settings

Name
hirjd4exampled5477y22yd23ettobiho._domainkey.example.com
v3rnz522czcl46quexamplek3efo5o6x._domainkey.example.com
y4examplexbhyhnsjcmvtvzotfvqjmdqoj._domainkey.example.com

[Download Record Set as CSV »](#)

- ドメインの DNS 設定に CNAME レコードを追加します。ドメインの DNS レコードを更新するには
  - DNS プロバイダーとして Route 53 を使用する場合は、Amazon SES を使用して E メールを送信するときに使用するアカウントと同じアカウントで Route 53 を使用することは、Route 53 の使用を選択して、ドメインの DNS 設定を自動的に更新します。それ以外の場合は、[Amazon Route 53 デベロッパーガイド](#)のレコードの編集に示す手順を完了します。
  - 別の DNS プロバイダーを使用している場合は、それぞれのプロバイダーでは、DNS レコードを更新するために異なる手順を設けています。次のテーブルでは、いくつかの主要なプロバイダーに関するドキュメントへのリンクを示しています。このリストにすべてが網羅されているわけではなく、このリストに含まれているからといって、他社の製品やサービスを支援または推奨するものではありません。お使いのプロバイダーがこの表にない場合でも、Amazon SES でそのドメインを使用できる可能性があります。

DNS/ホスティングプロバイダー	ドキュメントのリンク
GoDaddy	<a href="#">CNAME レコードを追加します</a>
Dreamhost	<a href="#">カスタム DNS レコードを追加する方法</a>
Cloudflare	<a href="#">CNAME レコードを追加する方法</a>

DNS/ホスティングプロバイダー	ドキュメントのリンク
HostGator	<a href="#">HostGator/eNom で DNS レコードを管理する</a>
Namecheap	<a href="#">ドメインの TXT/SPF/DKIM/DMARC レコードを追加する方法</a>
Names.co.uk	<a href="#">ドメインの DNS 設定の変更</a>
Wix	<a href="#">Adding or Updating CNAME Records in Your Wix Account</a>

#### Note

いくつかの DNS プロバイダーでは、レコード名に下線 ( \_ ) を含めることが許可されていません。ただし、DKIM レコード名には下線が必要となります。DNS プロバイダーがレコード名に下線を含めることを許可しない場合、プロバイダーのカスタマーサポートにお問い合わせください。

- DNS プロバイダーが不明な場合 - システム管理者にお問い合わせください。

Amazon SES は通常、72 時間以内に DNS 設定への変更を検出します。

## E メールアドレスで Easy DKIM をセットアップする

このセクションの手順では、Amazon SES で検証済みの特定の E メールアドレスに Easy DKIM をセットアップする方法を示しています。既に所有しているドメインに属する E メールアドレスに対してのみ Easy DKIM を設定できます。これは、E メールアドレスに Easy DKIM をセットアップするためにはドメインの DNS 設定を変更する必要があるためです。

#### Important

所有していないドメインの E メールアドレスに対して Easy DKIM をセットアップすることはできません。たとえば、gmail.com や hotmail.com アドレスに Easy DKIM をセットアップすることはできません。

E メールアドレスが属するドメインで既に Easy DKIM をセットアップしている場合には、E メールアドレスにも同じく Easy DKIM をセットアップする必要はありません。ドメインで Easy DKIM をセットアップする場合、Amazon SES はこのドメインのすべてのアドレスからのすべての E メールを自動的に認証します。特定の E メールアドレスの Easy DKIM 設定は、アドレスが属するドメインの設定を自動的に上書きします。

E メールアドレスに Easy DKIM をセットアップするには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインの [Identity Management] で [Email Addresses] を選択します。
3. E メールアドレスのリストで、Easy DKIM をセットアップするアドレスを選択します。
4. [DKIM で] で、[DKIM 設定の生成] を選択します。
5. このセクションに表示される 3 つの CNAME レコードをコピーします。または、[レコードセットを CSV としてダウンロード] を選択してコンピューターにこのレコードを保存することもできます。
6. ドメインの DNS 設定に CNAME レコードを追加します。ドメインの DNS レコードを更新するには

- DNS プロバイダーとして Route 53 を使用する場合 - Amazon Route 53 デベロッパーガイドの [レコードの編集](#) に示す手順を完了します。

- 別の DNS プロバイダーを使用している場合 – それぞれのプロバイダーでは、DNS レコードを更新するために異なる手順を設けています。DNS サービスプロバイダーから提供されたドキュメントで詳細を参照してください。

#### Note

いくつかの DNS プロバイダーでは、レコード名に下線 ( \_ ) を含めることが許可されていません。ただし、DKIM レコード名には下線が必要となります。DNS プロバイダーがレコード名に下線を含めることを許可しない場合、プロバイダーのカスタマーサポートにお問い合わせください。

- DNS プロバイダーが不明な場合 – システム管理者にお問い合わせください。

Amazon SES は通常、72 時間以内に DNS 設定への変更を検出します。

## Easy DKIM の管理

ID で Easy DKIM を管理する方法は 2 つあります。ウェブベースの Amazon SES コンソールを使用する方法と Amazon SES API を使用する方法です。これらのいずれかの方法を使用して ID の DKIM レコードを取得するか、または、ID に対して Easy DKIM を有効または無効にすることができます。

### ID で Easy DKIM レコードを取得する

Amazon SES コンソールを使用して、ドメインあるいは E メールアドレスに Easy DKIM レコードをいつでも取得できます。

コンソールを使用して、ID に Easy DKIM レコードを取得するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインの [ID 管理] で、Easy DKIM レコードを取得する ID のタイプを選択します。
3. ID のリストで、Easy DKIM レコードを取得する ID を選択します。
4. [DKIM] セクションで、3 つの CNAME レコードをコピーします。以下のイメージは、[DKIM] セクションの例を示しています。

## ▼ DKIM

**DKIM settings for your domain have been generated.** The information below depends on who provides your DNS service; if your DNS service is provided by a third party, you may need to create DNS records. [Learn more](#)

**DKIM:** waiting on sender verification...

**DKIM Verification Status:** pending verification

To enable DKIM signing for your domain, the records below must be entered into your DNS and allow DKIM signing at that time. Note that verification of these settings may take some time.

Name
hirjd4exampled5477y22yd23ettobiho._domainkey.example.com
v3rnz522czcl46quexamplek3efo5o6x._domainkey.example.com
y4examplexbhyhnsjcmvtvzotfvqjmdqoj._domainkey.example.com

[Download Record Set as CSV »](#)

Amazon SES API を使用して、ID に CNAME レコードを取得することもできます。API とやり取りする一般的な方法は、AWS CLI を使用することです。

AWS CLI を使用して、ID に Easy DKIM レコードを取得するには

1. コマンドラインから、以下のコマンドを入力します。

```
aws ses get-identity-dkim-attributes --identities "example.com"
```

前述の例で、*example.com* を Easy DKIM レコードを取得する ID で置き換えます。E メールアドレスまたはドメインを指定できます。

2. このコマンドの出力には、次の例に示すように、DkimTokens セクションが含まれています。

```
{
  "DkimAttributes": {
    "example.com": {
      "DkimEnabled": true,
      "DkimVerificationStatus": "Success",
      "DkimTokens": [
        "hirjd4exampled5477y22yd23ettobi",
        "v3rnz522czcl46quexamplek3efo5o6x",

```

```
        "y4examplexbhynsjcmtvzotfvqjmdqoj"  
      ]  
    }  
  }  
}
```

トークンを使用してドメインの DNS 設定に追加する CNAME レコードを作成します。CNAME レコードを作成するには、次のテンプレートを使用します。

```
token1._domainkey.example.com CNAME token1.dkim.amazonses.com  
token2._domainkey.example.com CNAME token2.dkim.amazonses.com  
token3._domainkey.example.com CNAME token3.dkim.amazonses.com
```

**token1** の各インスタンスを、`aws ses get-identity-dkim-attributes` コマンドを実行したときに受け取ったリストの最初のトークンに置き換え、**token2** のインスタンスすべてを、リスト内の 2 番目のトークンに置き換え、**token3** のインスタンスすべてを、リストの 3 番目のトークンに置き換えます。

たとえば、前述の例に示されるトークンにこのテンプレートを適用すると、次のレコードが生成されます。

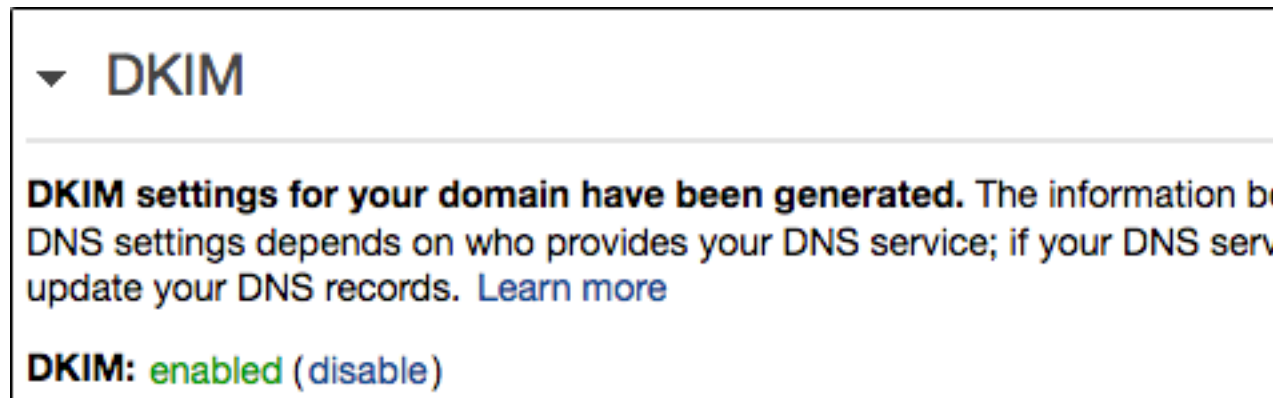
```
hirjd4exampled5477y22yd23ettobi._domainkey.example.com CNAME  
hirjd4exampled5477y22yd23ettobi.dkim.amazonses.com  
v3rnz522czcl46quexamplek3efo5o6x._domainkey.example.com CNAME  
v3rnz522czcl46quexamplek3efo5o6x.dkim.amazonses.com  
y4examplexbhynsjcmtvzotfvqjmdqoj._domainkey.example.com CNAME  
y4examplexbhynsjcmtvzotfvqjmdqoj.dkim.amazonses.com
```

## ID の Easy DKIM を無効にする

Amazon SES コンソールを使用して、ID の DKIM 認証を素早く無効にできます。

ID で Easy DKIM を無効にするには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインの [ID 管理] で、Easy DKIM を無効にする ID のタイプを選択します。
3. ID のリストで、Easy DKIM レコードを無効にする ID を選択します。
4. [DKIM] セクションの [DKIM: enabled (DKIM: 有効)] の横で、次の図に示すように [無効化] を選択します。



また、Amazon SES API を使用して、ID の Easy DKIM を無効にすることもできます。API とやり取りする一般的な方法は、AWS CLI を使用することです。

AWS CLI を使用して、ID の Easy DKIM を無効にするには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses set-identity-dkim-enabled --identity example.com --no-dkim-enabled
```

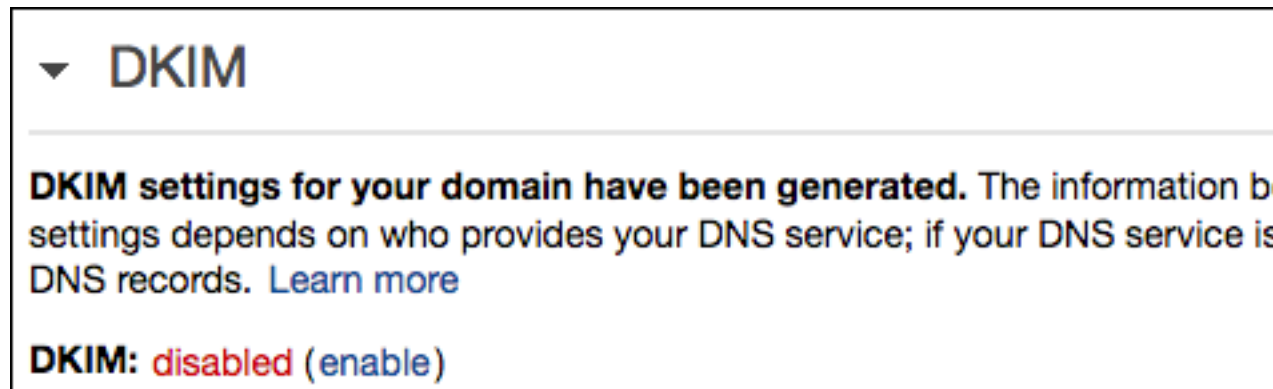
前述の例で、*example.com* を Easy DKIM を無効にする ID で置き換えます。E メールアドレスまたはドメインを指定できます。

### ID の Easy DKIM を有効にする

前に ID で Easy DKIM を無効化した場合、Amazon SES コンソールを使用して再度有効化できます。

ID で Easy DKIM を有効にするには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインの [ID 管理] で、Easy DKIM を有効にする ID のタイプを選択します。
3. ID のリストで、Easy DKIM レコードを有効にする ID を選択します。
4. [DKIM] セクションの [DKIM: disabled (DKIM: 無効)] の横で、次の図に示すように [有効化] を選択します。



また、Amazon SES API を使用して、ID の Easy DKIM を有効にすることもできます。API とやり取りする一般的な方法は、AWS CLI を使用することです。

AWS CLI を使用して、ID の Easy DKIM を有効にするには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses set-identity-dkim-enabled --identity example.com --dkim-enabled
```

前述の例で、*example.com* を Easy DKIM を有効にする ID で置き換えます。E メールアドレスまたはドメインを指定できます。

## 独自の DKIM 認証トークンを Amazon SES で提供する

[Easy DKIM \(p. 135\)](#) を使用する代わりに、独自のパブリック/プライベートのキーペアを使用して DKIM 認証を設定することもできます。このプロセスは、「Bring Your Own DKIM (BYODKIM)」として知られています。

BYODKIM では、単一の DNS レコードを使用してドメインの DKIM 認証を設定できます。一方 Easy DKIM では、3 つの個別の DNS レコードを発行する必要があります。さらに、BYODKIM を使用すると、ドメインの DKIM キーを必要な回数だけローテーションできます。

このセクションのトピック:

- [ステップ 1: キーペアを作成する \(p. 143\)](#)
- [ステップ 2: ドメインの DNS 設定にパブリックキーを追加する \(p. 143\)](#)
- [ステップ 3: BYODKIM を使用するようにドメインを設定する \(p. 144\)](#)

### ステップ 1: キーペアを作成する

独自の DKIM 機能を使用するには、まずキーペアを作成する必要があります。

生成するプライベートキーは、1024 ビットの RSA エンコードを使用する必要があります。プライベートキーは PKCS #1 形式である必要があります。

このセクションでは、ほとんどの Linux、macOS、または Unix オペレーティングシステムに組み込まれている `openssl` コマンドを使用してキーペアを作成する方法について説明します。

#### Note

Windows コンピュータを使用している場合は、サードパーティーアプリケーションを使用して RSA キーペアを生成できます。サードパーティーアプリケーションを使用する場合は、1,024 ビットの RSA キーペアを PKCS#1 形式で生成できる必要があります。

Linux、macOS、または Unix コマンドラインからキーペアを作成するには

1. コマンドラインで、以下のコマンドを入力してプライベートキーを生成します。

```
openssl genrsa -f4 -out private.key 1024
```

2. コマンドラインで、以下のコマンドを入力してパブリックキーを生成します。

```
openssl rsa -in private.key -outform PEM -pubout -out public.key
```

### ステップ 2: ドメインの DNS 設定にパブリックキーを追加する

キーペアを作成したので、パブリックキーを TXT レコードとしてドメインの DNS 設定に追加する必要があります。

ドメインの DNS 設定にパブリックキーを追加するには

1. DNS またはホスティングプロバイダーの管理コンソールにサインインします。
2. ドメインの DNS 設定に新しいテキストレコードを追加します。レコードは、次の形式を使用する必要があります。

名前	タイプ	値
<code>selector._domainkey.example</code>	<code>TXT</code>	<code>p=<i>yourPublicKey</i></code>

上の例に、以下の変更を加えます。

- `selector` は、キーを識別する一意の名前に置き換えます。
- `example.com` をドメインに置き換えます。
- `yourPublicKey` は、以前に作成したパブリックキーに置き換えます。

#### Note

生成されたパブリックキーの最初と最後の行 (それぞれ -----BEGIN PUBLIC KEY----- と -----END PUBLIC KEY-----) を削除する必要があります。さらに、生成されたパブリックキーの改行を削除する必要があります。結果の値は、スペースや改行を含まない文字列になります。

プロバイダーによって、DNS レコードを更新する手順は異なります。次のテーブルでは、いくつかの主要なプロバイダーに関するドキュメントへのリンクを示しています。このリストにはすべての内容が網羅されているわけではなく、このリストに含まれているからといって、他社の製品やサービスを支援または推奨するものではありません。

DNS/ホスティングプロバイダー	ドキュメントのリンク
Amazon Route 53	詳細については、『 <a href="#">Amazon Route 53 デベロッパーガイド</a> 』の「 <a href="#">レコードの編集</a> 」を参照してください。
GoDaddy	<a href="#">TXT レコードを追加する</a>
Dreamhost	<a href="#">カスタム DNS レコードを追加する方法</a>
Cloudflare	<a href="#">CloudFlare内でDNSレコードを管理する</a>
HostGator	<a href="#">HostGator/eNom で DNS レコードを管理する</a>
Namecheap	<a href="#">ドメインの TXT/SPF/DKIM/DMARC レコードを追加する方法</a>
Names.co.uk	<a href="#">ドメインの DNS 設定の変更</a>
Wix	<a href="#">Wix アカウントの TXT レコードの追加または更新</a>

## ステップ 3: BYODKIM を使用するようにドメインを設定する

BYODKIM は、新しいドメイン (現在 Amazon SES を経由した Eメールの送信に使用していないドメイン) と既存のドメイン (Amazon SES で使用するように設定済みのドメイン) の両方に対して設定できます。新しいドメインを設定するには、Amazon SES API で `CreateEmailIdentity` オペレーションを実行します。既存のドメインを設定するには、`PutEmailIdentityDkimSigningAttributes` オペレーションを実行します。

このセクションでは、AWS CLI を使用して新規ドメインと既存のドメインを設定する手順について説明します。このセクションの手順を完了する前に、まず AWS CLI をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

### オプション 1: BYODKIM を使用する新しいドメイン ID を作成する

このセクションでは、BYODKIM を使用する新しいドメイン ID を作成する手順について説明します。新しいドメイン ID とは、Amazon SES を使用して Eメールを送信するように設定していないドメインです。



BYODKIM を使用するように既存のドメインを設定する場合は、代わりに「[オプション 2: 既存のドメイン ID を設定する \(p. 145\)](#)」の手順を実行します。

ID を作成するには

1. テキストエディタで、次のコードを貼り付けます。

```
{
  "EmailIdentity": "example.com",
  "DkimSigningAttributes": {
    "DomainSigningPrivateKey": "privateKey",
    "DomainSigningSelector": "selector"
  }
}
```

上の例に、以下の変更を加えます。

- `example.com` を、作成するドメインに置き換えます。
- `privateKey` をプライベートキーに置き換えます。
- `selector` は、ドメインの DNS 設定で TXT レコードを作成したときに指定した一意のセレクトに置き換えます。

終了したら、`create-identity.json` としてファイルを保存します。

2. コマンドラインで以下のコマンドを入力します。

```
aws sesv2 create-email-identity --cli-input-json file://path/to/create-identity.json
```

上記のコマンドで、`path/to/create-identity.json` を、前のステップで作成したファイルの完全なパスに置き換えます。

## オプション 2: 既存のドメイン ID を設定する

このセクションでは、BYODKIM を使用するために既存のドメイン ID を更新する手順について説明します。既存のドメイン ID は、Amazon SES を使用して Eメールを送信するようにすでに設定されているドメインです。

ドメイン ID を更新するには

1. テキストエディタで、次のコードを貼り付けます。

```
{
  "SigningAttributes": {
    "DomainSigningPrivateKey": "privateKey",
    "DomainSigningSelector": "selector"
  },
  "SigningAttributesOrigin": "EXTERNAL"
}
```

上の例に、以下の変更を加えます。

- `privateKey` をプライベートキーに置き換えます。
- `selector` は、ドメインの DNS 設定で TXT レコードを作成したときに指定した一意のセレクトに置き換えます。

終了したら、`update-identity.json` としてファイルを保存します。

2. コマンドラインで以下のコマンドを入力します。

```
aws sesv2 put-email-identity-dkim-signing-attributes --email-identity example.com --cli-input-json file://path/to/update-identity.json
```

上のコマンドに、以下の変更を加えます。

- *path/to/update-identity.json* を、前のステップで作成したファイルへの完全なパスに置き換えます。
- *example.com* を、更新するドメインに置き換えます。

### BYODKIM を使用するドメインの DKIM ステータスを確認する

BYODKIM を使用するようにドメインを設定したら、GetEmailIdentity オペレーションを実行して、DKIM が正しく設定されていることを確認できます。

ドメインの DKIM ステータスを確認するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 get-email-identity --email-identity example.com
```

上記のコマンドで、*example.com* をドメインに置き換えます。

このコマンドは、次の例のようなセクションを含む JSON オブジェクトを返します。

```
{
  ...
  "DkimAttributes": {
    "SigningAttributesOrigin": "EXTERNAL",
    "SigningEnabled": true,
    "Status": "SUCCESS",
    "Tokens": [ ]
  },
  ...
}
```

BYODKIM は、以下のすべてに該当する場合、ドメインに対して適切に設定されています。

- SigningAttributesOrigin プロパティの値は EXTERNAL です。
- SigningEnabled の値は true です。
- Status の値は SUCCESS です。

## Amazon SES 内で手動での DKIM 署名

Easy DKIM を使用する代わりに、手動で DKIM 署名をメッセージに追加し、Amazon SES を使用してこのメッセージを送信することもできます。手動でメッセージを署名することを選択する場合、まず DKIM 署名を作成する必要があります。メッセージと DKIM 署名を作成したら、[SendRawEmail API](#) を使用してメッセージを送信できます。

Eメールを手動で署名する場合、次の要素を考慮してください。

- Amazon SES を使用して送信するすべてのメッセージには、amazonses.com のドメインの署名を参照する DKIM ヘッダーが含まれています(つまり、d=amazonses.com という文字列が含まれています)。メッセージを手動で署名する場合、このメッセージには 2 つの DKIM ヘッダー (ドメインのヘッダーと amazonses.com のヘッダー) が含まれている必要があります。

- Amazon SES は、メッセージに手で追加した DKIM 署名を検証しません。メッセージの DKIM 署名にエラーがある場合、このメッセージは E メールプロバイダーによって拒否される可能性があります。
- メッセージに署名する場合は、少なくとも 1024 ビットのビット長を使用する必要があります。
- 次のフィールドには署名しないでください。メッセージ ID、日付、Return-Path、Bounces-To。

#### Note

Amazon SES SMTP インタフェースでの Eメールの送信に Eメールクライアントを使用する場合、クライアントはメッセージで自動的に DKIM 署名を実行することがあります。一部のクライアントは上述のフィールドのいずれかに署名することがあります。Eメールクライアントのドキュメントを参照して、デフォルトでどのフィールドが署名されるかを確認してください。

## Amazon SES での DMARC への準拠

DMARC (Domain-based Message Authentication, Reporting and Conformance) は、SPF (Sender Policy Framework) およびドメインキーアイデンティファイドメール (DKIM) を使用してメールスプーフィングを検出するためのメール認証プロトコルです。DMARC に準拠するため、メッセージは SPF または DKIM のいずれか、または両方で認証される必要があります。

このトピックには、送信する Eメールが SPF および DKIM の両方に準拠するよう Amazon SES を設定するのに役立つ情報が含まれます。これらの認証システムのいずれかに従うことで、Eメールを DMARC に準拠させることができます。DMARC の仕様については、<http://www.dmarc.org> を参照してください。

### ドメインの DMARC ポリシーのセットアップ

DMARC をセットアップするには、ドメインの DNS 設定を変更する必要があります。ドメインの DNS 設定に、ドメインの DMARC 設定を指定する TXT レコードが含まれている必要があります。DNS 設定に TXT レコードを追加する手順は、DNS またはホスティングプロバイダーによって異なります。Route 53 を使用する場合は、[Amazon Route 53 デベロッパーガイド](#)の「レコードで作業」を参照してください。別のプロバイダーを使用する場合は、DNS 設定についてはプロバイダーのドキュメントを参照してください。

作成する TXT レコードの名前は、`_dmarc.example.com` の必要があります。ここで、`example.com` はお客様のドメインです。TXT レコードの値には、ドメインに適用される DMARC ポリシーが含まれています。以下に、DMARC ポリシーを含む TXT レコードの例を示します。

名前	タイプ	値
<code>_dmarc.example.com</code>	TXT	<code>"v=DMARC1;p=quarantine;pct=25;rua=ma</code>

単純な言語で、このポリシーは Eメールプロバイダーに以下のことを指示します。

- `example.com` の「From」ドメインを持ち、SPF または DKIM 認証に合格しないすべての Eメールを探します。
- 認証に失敗した Eメールの 25% をスパムフォルダに送信して隔離します。(また、`p=none` を使用して何も実行しない、あるいは `p=reject` を使用してメッセージを完全に拒否することもできます。)
- 認証が失敗したすべての Eメールに関するレポートをダイジェスト (つまり、イベントごとに個別のレポートを送信するのではなく、特定の期間のデータを集計したレポート) で送信します。ポリシーはプロバイダーごとに異なりますが、通常、Eメールプロバイダーは 1 日 1 回レポートを送信します。

ドメインの DMARC 設定の詳細については、DMARC ウェブサイトの「[概要](#)」を参照してください。

DMARC システムの完全な仕様については、IETF ウェブサイトの「[RFC 7489](#)」を参照してください。このドキュメントのセクション 6.3 には、ドメインの DMARC ポリシーを設定するために使用できるタグの完全なリストが含まれています。

## SPF による DMARC への準拠

E メールが SPF に基づいて DMARC に準拠するためには、次の両方の条件を満たすことが求められています。

- E メールが SPF チェックに合格する必要があります。
- メールヘッダーに含まれる From アドレスのドメインは、送信側メールサーバーが受信側メールサーバーに指定する MAIL FROM ドメインと合致する必要があります。ドメインの DMARC ポリシーが SPF で strict アラインメントを指定している場合、From ドメインと MAIL FROM ドメインが完全に一致する必要があります。ドメインの DMARC ポリシーが SPF で relaxed アラインメントを指定している場合、From ヘッダーに指定されたドメインのサブドメインを MAIL FROM ドメインにすることができます。

これらの要件に準拠するためには、次のステップを実行します。

- [the section called “カスタムの MAIL FROM ドメインの設定” \(p. 68\)](#) の手順を実行して、カスタム MAIL FROM ドメインを設定します。
- 送信元ドメインが SPF に relaxed ポリシーを使用していることを確認します。ドメインのポリシーアラインメントを変更していない場合は、デフォルトで relaxed ポリシーが使用されます。

### Note

コマンドラインで以下のコマンドを入力して、[example.com](#) をドメインで置き換えることで、SPF での DMARC アラインメントを選択できます。

```
nslookup -type=TXT _dmarc.example.com
```

このコマンドの出力の [Non-authoritative answer] から、v=DMARC1 で始まるレコードを探します。このレコードに文字列 aspf=r が含まれるか、または aspf 文字列がまったく存在しない場合、ドメインは SPF に relaxed アラインメントを使用します。レコードに文字列 aspf=s が含まれる場合、ドメインは SPF に strict アラインメントを使用します。システム管理者は、ドメインの DNS 設定の DMARC TXT レコードからこのタグを削除する必要があります。または、ウェブベースの DMARC ルックアップツール (dmarcian ウェブサイトの [DMARC Inspector](#) や Proofpoint ウェブサイトの [DMARC Check](#) ツールなど) を使用して、お客様のドメインのポリシーが SPF に準拠しているかどうかを判断できます。

## DKIM による DMARC への準拠

E メールが DKIM に基づいて DMARC に準拠するためには、次の両方の条件を満たすことが求められています。

- 有効な DKIM 署名がメッセージに含まれている必要があります。
- E メールヘッダーの From アドレスが DKIM 署名の d= ドメインに合致すること。ドメインの DMARC ポリシーで DKIM に strict アラインメントが指定されている場合は、これらのドメインが完全に一致する必要があります。ドメインの DMARC ポリシーが DKIM に relaxed アラインメントを指定している場合、From ドメインのサブドメインを d= ドメインにすることができます。

これらの要件に準拠するためには、次のステップを実行します。

- [the section called “Easy DKIM” \(p. 135\)](#) の手順を実行して Easy DKIM を設定します。Easy DKIM を使用すると、Amazon SES は自動的に E メールに署名します。

#### Note

Easy DKIM を使用せずに、[メッセージに手動で署名する \(p. 146\)](#) こともできます。ただし、Amazon SES は手動で構築した DKIM 署名を検証しないため、この選択は慎重に行ってください。そのため、Easy DKIM を使用することを強くお勧めします。

- 送信元ドメインが DKIM に relaxed ポリシーを使用していることを確認します。ドメインのポリシーアラインメントを変更していない場合は、デフォルトで relaxed ポリシーが使用されます。

#### Note

コマンドラインで以下のコマンドを入力して、`example.com` をドメインで置き換えることで、DKIM での DMARC アラインメントを選択できます。

```
nslookup -type=TXT _dmarc.example.com
```

このコマンドの出力の [Non-authoritative answer] から、`v=DMARC1` で始まるレコードを探します。このレコードに文字列 `adkim=r` が含まれるか、または `adkim` 文字列がまったく存在しない場合、ドメインは DKIM に relaxed アラインメントを使用します。レコードに文字列 `adkim=s` が含まれる場合、ドメインは DKIM に strict アラインメントを使用します。システム管理者は、ドメインの DNS 設定の DMARC TXT レコードからこのタグを削除する必要があります。

または、ウェブベースの DMARC ルックアップツール (dmarcian ウェブサイトの [DMARC Inspector](#) や Proofpoint ウェブサイトの [DMARC Check](#) ツールなど) を使用して、お客様のドメインのポリシーが DKIM に準拠しているかどうかを判断できます。

## Amazon SES 送信クォータの管理

Amazon SES アカウントには、送信できる E メールメッセージの数と送信レートを規制するための送信クォータのセットがあります。送信クォータは、Amazon SES と E メールプロバイダーとの間の信頼関係を保つという意味で、Amazon SES を利用するすべてのお客様に利益をもたらします。E メール送信のボリュームやレートが不意に急増すると、E メールプロバイダーによって E メールがブロックされますが、送信クォータがあることで、送信アクティビティを徐々に増やすことができ、その可能性を低く抑えることができます。

以下のクォータは、Amazon SES 経由の E メール送信に適用されます。

- 最大日次送信数 - 24 時間に送信できるメールの最大数 このクォータは、ローリング期間で計算されます。メールを送信しようとするたびに、Amazon SES は過去 24 時間以内に送信したメールの数を決定します。過去 24 時間以内に送信した Eメールの総数が 1 日の最大数より少ない限り、送信リクエストは受け入れられ、Eメールが送信されます。

メッセージの送信がアカウントの 1 日の最大数を超える場合、Amazon SES への呼び出しは拒否されません。

- 最大送信レート - Amazon SES が 1 秒あたりにアカウントから受け付ける Eメールの最大数。このクォータを瞬間的に超えることはできますが、制限を超えた状態が長時間続くことは許可されません。

#### Note

ご使用のアカウントで、Amazon SES のメッセージ受信レートが、この最大送信レートよりも少なくなる場合があります。

Amazon SES 送信クォータは、AWS リージョンごとに独立しています。複数の AWS リージョンでの Amazon SES の使用については、「[リージョンと Amazon SES \(p. 467\)](#)」を参照してください。

アカウントが Amazon SES サンドボックス内にある場合、24 時間あたり 200 メッセージのみ送信でき、最大送信レートは毎秒 1 メッセージです。サンドボックスからアカウントを削除するリクエストを送信す

るとき、同時にクォータの引き上げをリクエストすることもできます。アカウントをサンドボックスから削除する方法については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

アカウントがサンドボックスから削除されたら、AWS Support Center で新しいケースを作成することで、いつでもさらにクォータの引き上げをリクエストできます。詳細については、「[Amazon SES 送信クォータの引き上げ \(p. 151\)](#)」を参照してください。

#### Note

メッセージ数ではなく、受取人数に基づいた送信クォータがあります。たとえば、受取人数が 10 人である E メールは、クォータに対しては 10 通とカウントされます。ただし、SendEmail API オペレーションの 1 回の呼び出しで複数の受信者にメールを送信することはお勧めしません。呼び出しが失敗すると、メール全体が拒否されるためです。SendEmail は、受取人ごとに 1 回呼び出すようお勧めします。

- 送信クォータの引き上げについては、「[Amazon SES 送信クォータの引き上げ \(p. 151\)](#)」を参照してください。
- 送信クォータに達したときにアプリケーションが受け取るエラーについては、「[Amazon SES アカウントの送信クォータに関するエラー \(p. 153\)](#)」を参照してください。
- Amazon SES コンソールまたは Amazon SES API を使用して送信クォータをモニタリングするには、「[Amazon SES 送信クォータのモニタリング \(p. 150\)](#)」を参照してください。

## Amazon SES 送信クォータのモニタリング

送信クォータをモニタリングするには、Amazon SES コンソールまたは Amazon SES API を使用します。これを行うには、クエリ (HTTPS) インターフェイスを直接呼び出すか、[AWS SDK](#)、[AWS Command Line Interface](#)、または [AWS Tools for Windows PowerShell](#) を使用して間接的に呼び出します。

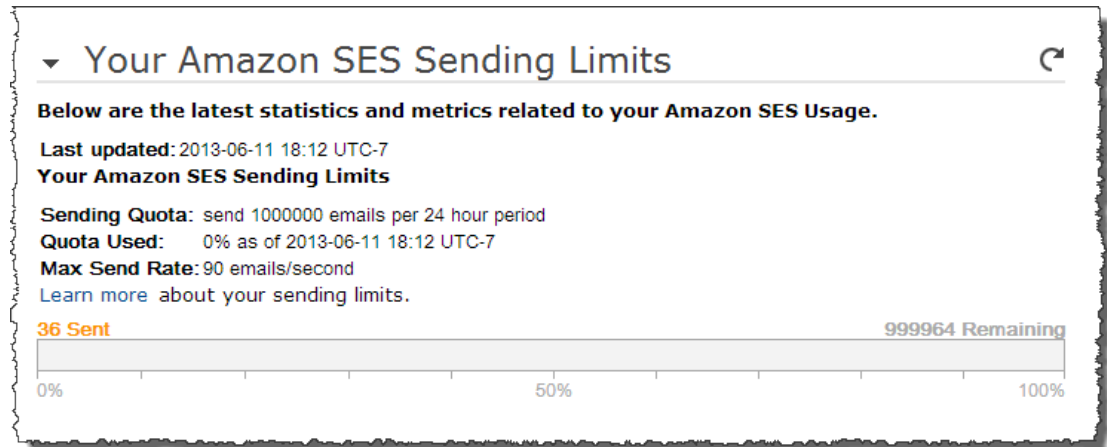
#### Important

送信クォータに達しないように、頻繁に送信統計を確認することをお勧めします。送信クォータに近付いた場合は、クォータを引き上げる方法について「[Amazon SES 送信クォータの引き上げ \(p. 151\)](#)」を参照してください。送信クォータに達する前に、クォータの引き上げを検討してください。

## Amazon SES コンソールを使用した送信クォータのモニタリング

次の手順では、Amazon SES コンソールを使用した送信クォータの表示方法を示します。

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで、[Sending Statistics] を選択します。送信クォータが [Amazon SES の送信制限] に表示されます。



3. 表示を更新するには、[Refresh] を選択します。

## Amazon SES API を使用した送信クォータのモニタリング

Amazon SES API には送信クォータを返す `GetSendQuota` アクションがあります。GetSendQuota アクションを呼び出すと、以下の情報が返されます。

- 過去 24 時間に送信した E メールの数
- 現在の 24 時間の送信クォータ
- 最大送信レート

### Note

「GetSendQuota」の説明については、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。

## Amazon SES 送信クォータの引き上げ

アカウントがサンドボックスの外にあり、質の高い E メールを送信しようとする場合、本稼働アカウント用に送信クォータが自動的に引き上げられる可能性があります。多くの場合、これらのクォータを実際に引き上げる必要が生じる前に、クォータは自動的に引き上げられます。

レートが自動的に引き上げられる対象になるには、次のすべての条件が満たされている必要があります。

- 受取人が受信を希望する高品質のコンテンツを送信する – 受取人の希望や期待に沿ったコンテンツを送信します。E メールを希望しないお客様へのメール送信を停止します。
- 実際の本稼働コンテンツを送信する フェイクの E メールアドレスにテストメッセージを送信する – バウンス率や苦情率に悪い影響が出る可能性があります。また、内部の受取人へのみメッセージを送信すると、お客様が受信を希望するコンテンツを送信しているかどうかの判断が難しくなります。ただし、内部以外の受信者に本稼働メッセージを送信する場合は、E メール送信プラクティスを正確に評価できません。
- 現在のクォータに近いボリュームを送信する - 自動クォータ引き上げの対象となるには、1 日あたりの E メール量がアカウントの 1 日あたりの最大メール量を超えることはないが定期的にそれに近づいている必要があります。
- バウンス率と苦情率が低い – 受信するバウンスと苦情の件数は最小限に抑えます。バウンスと苦情の件数の多さが、送信クォータに悪影響を及ぼすことがあります。

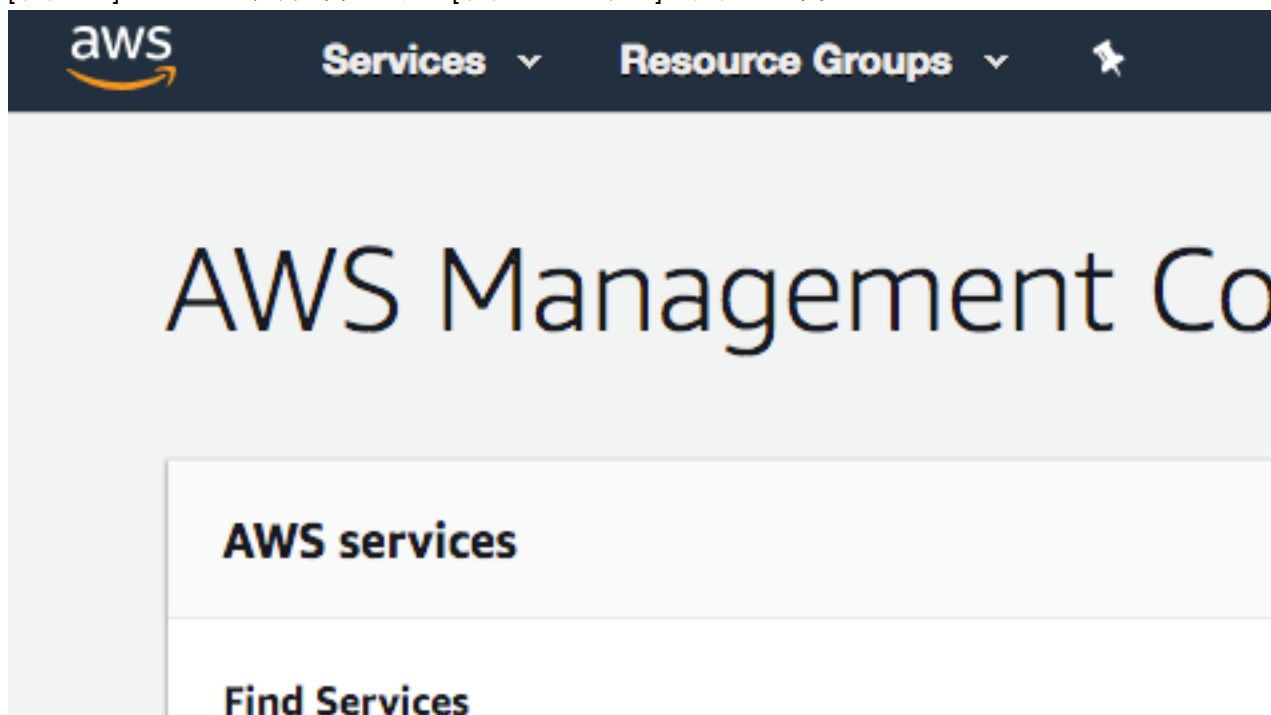
現在の送信クォータがニーズに見合っていないが自動的に引き上げられていない場合、引き上げをリクエストできます。送信クォータ引き上げのリクエストの詳細については、「[Amazon SES の送信クォータを引き上げるためのケースを開く \(p. 152\)](#)」を参照してください。

## Amazon SES の送信クォータを引き上げるためのケースを開く

Amazon SES の送信クォータの引き上げを申請するには、以下のステップを実行してサポートセンターでケースを開きます。

送信クォータの引き上げをリクエストするには

1. AWS Management Console (<https://console.aws.amazon.com/>) にサインインします。
2. [サポート] メニューで、次の図のように [サポートセンター] を選択します。



3. My support cases (サポートケース) タブで、[Create case (ケースの作成)] を選択します。
4. [Create case (ケースの作成)] で [Service limit increase (サービスの上限緩和)] を選択します。
5. [Case classification (ケースの分類)] で、以下のセクションに入力します。
  - [Limit type (制限のタイプ)] で SES Service Limits (SES サービスの制限) を選択します。
  - [Mail Type (メールの種類)] で、送信する予定の E メールタイプを選択します。複数の値が当てはまる場合は、送信する Eメールの大部分に当てはまるオプションを選択します。
  - [Website URL (ウェブサイトの URL)] に、ウェブサイトの URL を入力します。この情報は、お客様が送信を予定しているコンテンツのタイプを AWS が正しく理解するために役立ちます。
  - [My email sending complies with the AWS Service Terms and AUP] で、ユースケースに当てはまるオプションを選択します。
  - [I only send to recipients who have specifically requested my mail (私は明確にリクエストされた受信者にもみメールを送信します)] で、ユースケースに当てはまるオプションを選択します。
  - [I have a process to handle bounces and complaints (バウンスや苦情を処理するプロセスがあります)] で、ユースケースに当てはまるオプションを選択します。
6. [Requests (リクエスト)] で、以下のセクションに入力します。
  - [Region (リージョン)] で、リクエストが適用される AWS リージョンを選択します。



- [Limit (制限)] で、リクエストするクォータ引き上げのタイプを選択します。以下のオプションから選択できます。
  - 希望する最大送信クォータ – 選択したリージョンでアカウントが 24 時間あたりに送信できる E メール数の引き上げをリクエストする場合は、このオプションを選択します。  
  
希望する最大送信レート – 選択したリージョンでアカウントが 1 秒あたりに送信できる E メール数の引き上げをリクエストする場合は、このオプションを選択します。
- [New limit value (新しい制限値)] に、引き上げを希望するクォータを入力します。メール量は必要分のみをリクエストしてください。ただし、リクエストした量が保証されるわけではありません。

#### Note

送信クォータの引き上げと送信レートの引き上げの両方をリクエストする場合、または別の AWS SDK for PHP リージョンでの送信クォータの引き上げをリクエストする場合は、[Add another request (別のリクエストを追加)] を選択します。その後、このステップを繰り返します。

7. [Case Description (ケースの説明)] の [Use case description (ユースケースの説明)] で、Amazon SES を使用して E メールを送信する方法を説明します。AWS がお客様のリクエストを処理できるように、以下の質問にも答えます。
  - メーリングリストをどのように構築または取得しますか？
  - バウンスや苦情はどのように処理しますか？
  - 受取人はお客様からの Eメールの受信をどのようにオプトアウトできますか？
  - このリクエストで指定した新しい送信レートまたは送信クォータは、どのように選択しましたか？

AWS がケースを評価する際に検討する必要がある追加情報がある場合は、その情報もこのセクションで提供します。

8. [Contact options (問い合わせ先のオプション)] の [Preferred contact language (優先される問い合わせ言語)] で、このケースに関する AWS からの問い合わせは [English (英語)] がいいか [Japanese (日本語)] がいいかを選択します。
9. 完了したら、[Submit] を選択します。

AWS サポートチームは、お客様のリクエストに対して、24 時間以内に一次回答を行います。

迷惑なコンテンツや悪意のあるコンテンツを送信するためにシステムが悪用されないように、各リクエストを慎重に検討する必要があります。可能であれば、24 時間以内にリクエストを承認します。ただし、お客様から追加情報を取得する必要がある場合は、お客様のリクエストの解決に時間がかかる場合があります。

お客様のユースケースが AWS の方針と一致しない場合は、リクエストを承認できない場合があります。

## Amazon SES アカウントの送信クォータに関するエラー

1 日あたりの送信クォータ (24 時間以内に送信できる Eメールの最大数) や最大送信レート (1 秒間あたりに送信できるメッセージの最大数) に達した後で Eメールを送信しようとする、Amazon SES はメッセージを削除し、再配信を試行しません。また、Amazon SES は問題を説明するエラーメッセージを提供します。このエラーメッセージを Amazon SES で生成する方法は、Eメールの送信を試行した方法によって異なります。このトピックでは、Amazon SES API と SMTP インターフェイスを介して受信したメッセージに関する情報を示します。

最大送信レートに達したときに使用できる手法については、AWS メッセージングとターゲティングプログラムの「[Throttling – Maximum sending rate exceeded \(スロットリング – 最大送信レートの超過\)](#)」エラーの[対処法](#)を参照してください。

## Amazon SES API で送信制限に達した場合

Amazon SES API (または AWS SDK) を使用して E メールを送信しようとしたときに、アカウントの送信制限をすでに超えている場合は、API によって `ThrottlingException` エラーが生成されます。このエラーメッセージには、以下のいずれかのメッセージが含まれます。

- Daily message quota exceeded
- Maximum sending rate exceeded

スロットリングエラーが発生した場合は、最大 10 分間待機した後で送信リクエストを再試行するようにアプリケーションをプログラムする必要があります。

## SMTP で送信制限に達した場合

Amazon SES SMTP を使用して E メールを送信しようとしたときに、アカウントの送信制限をすでに超えている場合、SMTP クライアントは以下のいずれかのエラーを表示する場合があります。

- 454 Throttling failure: Maximum sending rate exceeded
- 454 Throttling failure: Daily message quota exceeded

これらのエラーの処理方法は、SMTP クライアントごとに異なります。

# Amazon SES での送信承認の使用

Amazon SES を設定することで、別のユーザーが、そのユーザーの Amazon SES アカウントを使用して、自分が所有するアドレスまたはドメイン (アイデンティティ) から E メールを送信することを許可できます。この機能は送信承認と呼ばれ、アイデンティティのコントロールは維持されるため、いつでもアクセス許可を変更または取り消すことができます。たとえば、ビジネスの所有者は、送信承認を使用することで、サードパーティー (E メールマーケティング会社など) の E メールを、自分が所有するドメインから送信できるようにすることができます。

他のユーザーがお客様の代わりに E メールを送信することを承認した場合、お客様がアイデンティティ所有者です。アイデンティティ所有者の場合、次のセクションを読むことをお勧めします。

- [の送信承認の概要 \(p. 155\)](#)
- [送信承認ポリシー \(p. 157\)](#)
- [送信承認ポリシーの例 \(p. 161\)](#)
- [ID 所有者のタスク \(p. 166\)](#)

他のユーザーの代わりに Eメールの送信を承認された場合、お客様は代理送信者です。代理送信者である場合は、次のセクションを読むことをお勧めします。

- [の送信承認の概要 \(p. 155\)](#)
- [代理送信者のタスク \(p. 172\)](#)

### Note

IAM ポリシーを使用して Amazon SES へのアクセスを制御することもできます。IAM ポリシーは、個々の IAM ユーザーが実行できる操作を制限し、送信承認ポリシーは個々の検証済みアイ

デンティティの使用方法を制限します。さらに、クロスアカウントアクセスを許可できるのは送信承認ポリシーのみです。Amazon SES での IAM ポリシーの使用の詳細については、「[Amazon SES へのアクセスのコントロール \(p. 406\)](#)」を参照してください。

## Amazon SES の送信承認の概要

このトピックでは、送信承認プロセスの概要を示し、送信承認と Amazon SES の E メール送信機能 (送信クォータや通知など) がどのように関係するかについて説明します。

このセクションでは以下の用語を使用します。

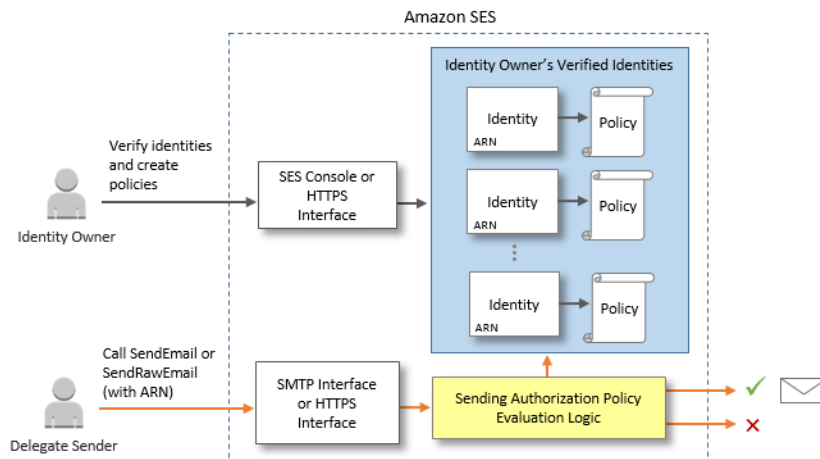
- アイデンティティ – Amazon SES ユーザーが Eメールの送信に使用する Eメールアドレスまたはドメイン。
- アイデンティティ所有者 – 「[ID の検証 \(p. 48\)](#)」の手順を使用して、Eメールアドレスまたはドメインの所有権を検証した Amazon SES ユーザー。
- 代理送信者 – 所有していないアイデンティティから Eメールを送信することを承認されたエンティティ。AWS アカウント、AWS Identity and Access Management (IAM) ユーザー、または AWS のサービスには、このクロスアカウント権限があります。
- 送信承認ポリシー – アイデンティティにアタッチすることで、そのアイデンティティを使用して送信できるユーザーとその条件を指定できるドキュメント。
- Amazon リソースネーム (ARN) – すべての AWS サービス間で AWS リソースを一意に識別する標準化された方法。送信承認の場合、リソースとは、アイデンティティ所有者が代理送信者に使用させるアイデンティティのことです。ARN の例は `arn:aws:ses:us-west-2:123456789012:identity/example.com` などです。

## 送信認証プロセス

送信承認は、送信承認ポリシーに基づいています。代理送信者が代わりに送信できるようにする場合、Amazon SES コンソールまたは Amazon SES API を使用して、送信承認ポリシーを作成してポリシーを ID に関連付けます。代理送信者がお客様の代わりに Amazon SES を通じて Eメールを送信するとき、代理送信者はリクエストまたは Eメールのヘッダーで ID の ARN を渡します。

Amazon SES は、Eメール送信リクエストを受け取ると、ID のポリシーを確認し (存在する場合)、代理送信者がその ID の代わりに送信することが承認されているかどうかを判断します。代理送信者が承認されている場合、Amazon SES は Eメールを受け入れます。承認されていない場合、Amazon SES はエラーメッセージを返します。

次の図は、送信承認の各概念の間の全体的な関係を示しています。



送信承認のプロセスは、以下のステップで構成されています。

1. ID 所有者は、Amazon SES コンソールまたは Amazon SES API を使用して、Amazon SES で ID を検証します。検証手順については、「[ID の検証 \(p. 48\)](#)」を参照してください。
2. 代理送信者が、送信を行うエンティティの AWS アカウント ID、IAM ユーザー ARN、または AWS のサービス名をアイデンティティ所有者に渡します。
3. ID 所有者が、Amazon SES コンソールまたは Amazon SES API を使用して送信承認ポリシーを作成し、ポリシーを ID にアタッチします。
4. ID 所有者が代理送信者に ID の ARN を付与し、代理送信者が E メール送信時に Amazon SES に ARN を提供できるようにします。
5. 代理送信者は、バウンスや苦情の通知を設定します。ID 所有者は、バウンスや苦情に対する Eメールのバウンスや苦情のイベントフィードバック通知をセットアップすることもできます。ID 所有者と代理送信者はどちらも、[イベント発行 \(p. 296\)](#)をセットアップして送信イベントデータをキャプチャすることもできます。

#### Note

ID 所有者が送信イベント通知を無効にする場合、代理送信者は、バウンスイベントと苦情イベントを Amazon SNS トピックまたは Kinesis Data Firehose ストリームに公開するようにイベント発行を設定する必要があります。送信者は、送信する各 Eメールにイベント発行ルールを含む設定セットも適用する必要があります。ID 所有者および代理送信者のいずれも、バウンスイベントと苦情イベントの通知を送信する方法を設定していない場合、または送信者がイベント公開ルールを使用する設定セットを適用しない場合は、ID 所有者が Eメールのフィードバック転送を無効にしても、Amazon SES は、Eメールの Return-Path フィールドのアドレス (または Return-Path アドレスを指定しなかった場合はソースフィールド) に自動的にイベント通知を Eメールで送信します。

6. 代理送信者は、リクエストまたは Eメールのヘッダーで ID 所有者の ID の ARN を渡すことで、ID 所有者の代わりに Amazon SES を通じて Eメールを送信します。代理送信者は、Amazon SES SMTP インターフェイスまたは Amazon SES API を使用して Eメールを送信できます。Amazon SES は、リクエストを受け取ると、ID にアタッチされたポリシーを調べ、代理送信者が指定された「From」アドレスと「Return Path」アドレスの使用を承認されている場合は Eメールを受け入れます。承認されていない場合、Amazon SES はエラーを返し、メッセージを受け入れません。
7. ID 所有者が代理送信者の承認を解除する場合は、ID 所有者が送信承認ポリシーを編集するか、ポリシーを完全に削除します。ID 所有者は、Amazon SES コンソールまたは Amazon SES API を使用してどちらのアクションも実行できます。

アイデンティティ所有者または代理所有者がこれらのタスクを実行する方法の詳細については、それぞれ「[ID 所有者のタスク \(p. 166\)](#)」または「[代理送信者のタスク \(p. 172\)](#)」を参照してください。

## E メール送信機能の属性

日次送信クォータ、バウンスと苦情、DKIM 署名、フィードバック転送などの Amazon SES E メール送信機能に関して、代理送信者とアイデンティティ所有者の役割を理解しておくことが重要です。属性は次のとおりです。

- 送信クォータ – アイデンティティ所有者のアイデンティティから送信される E メールは、代理送信者のクォータにカウントされます。
- バウンスと苦情 – バウンスおよび苦情イベントは、代理送信者の Amazon SES アカウントに記録されます。したがって、代理送信者の評価に影響を及ぼします。
- DKIM 署名 – アイデンティティ所有者がアイデンティティの Easy DKIM 署名を有効にした場合、そのアイデンティティから送信されるすべての E メール (代理送信者が送信した E メールを含む) が DKIM 署名されます。E メールが DKIM 署名されるかどうかをコントロールできるのはアイデンティティ所有者だけです。
- 通知 – アイデンティティ所有者と代理送信者のいずれも、バウンスと苦情に関する通知を設定できます。Eメールのアイデンティティ所有者は、Eメールのフィードバック転送を有効にすることもできます。通知のセットアップについては、「[Amazon SES 送信アクティビティのモニタリング \(p. 267\)](#)」を参照してください。
- 検証 – アイデンティティ所有者には、「[ID の検証 \(p. 48\)](#)」の手順に従って、代理送信者に試用を許可する E メールアドレスとドメインを自分が所有していることを検証する責任があります。代理送信者は、送信許可に指定された E メールアドレスまたはドメインを検証する必要はありません。
- AWS リージョン – 代理送信者は、アイデンティティ所有者のアイデンティティが検証された AWS リージョンから E メールを送信する必要があります。代理送信者にアクセス許可を付与する承認の送信ポリシーは、そのリージョンのアイデンティティにアタッチする必要があります。
- 請求 – 代理送信者のアカウントから送信されるすべてのメッセージ (代理送信者がアイデンティティ所有者のアドレスを使用して送信する E メールを含む) の料金は、代理送信者に請求されます。

## Amazon SES 送信承認ポリシー

別の AWS アカウント、Identity Access and Management (IAM) ユーザー、または AWS サービスがお客様の代わりに Amazon SES を通じて E メールを送信できるようにするには、送信承認ポリシーを作成します。このポリシーは、所有している ID にアタッチする JSON ドキュメントです。このポリシーには、その ID での送信を許可するユーザーとその条件が明示的にリストされます。お客様、およびポリシーでアクセス権限を明示的に付与したエンティティ以外のすべての送信者は、拒否されます。アイデンティティにはポリシーをアタッチしないことも、1 つまたは複数のポリシーをアタッチすることもできます。さらに、複数のステートメントを含む 1 つのポリシーを作成して、複数のポリシーの効果を持たせることもできます。

ポリシーは、シンプルな内容にすることも、細かくコントロールできるよう設定することもできます。たとえば、example.com を所有している場合、シンプルなポリシーを記述してそのドメインから送信するアクセス許可を AWS ID 123456789012 に付与することができます。詳細なポリシーでは、AWS ID 123456789012 が指定された日付範囲に限り、user@example.com からのみ E メールを送信できることを指定できます。

Amazon SES 送信承認ポリシーは、E メール送信 API (SendEmail、SendRawEmail、SendTemplatedEmail、および SendBulkTemplatedEmail) にのみ適用されます。このポリシーによってユーザーが AWS アカウントにアクセスできるようになることはありません。

## ポリシーの構造

各送信承認ポリシーは、アイデンティティにアタッチされる JSON ドキュメントです。各ポリシーには以下のセクションがあります。

- ポリシー全体に関する情報 (ドキュメントの最上部)。
- 1 つ以上の個別のステートメント (それぞれが 1 セットのアクセス許可を説明)。

次の例では、AWS アカウント ID 123456789012 に、検証済みドメイン example.com から送信するアクセス許可を付与します。

```
{
  "Id": "ExampleAuthorizationPolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AuthorizeAccount",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:123456789012:identity/example.com",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      },
      "Action": [
        "ses:SendEmail",
        "ses:SendTemplatedEmail",
        "ses:SendRawEmail",
        "ses:SendBulkTemplatedEmail"
      ]
    }
  ]
}
```

他の送信承認ポリシーの例は、「[送信承認ポリシーの例 \(p. 161\)](#)」を参照してください。

## ポリシーの要素

このセクションでは、送信承認ポリシーに含まれている要素について説明します。まず、ポリシー全体の要素について説明した後、その要素が含まれるステートメントにのみ適用される要素について説明します。その後、ステートメントに条件を追加する方法について説明します。

エレメントの構文の詳細については、[IAM ユーザーガイド](#)の「IAM ポリシー言語の文法」を参照してください。

## ポリシー全体に関する情報

ポリシー全体の要素として、Id と Version の 2 つがあります。次の表に、これらの要素についての情報を示します。

名前	説明	必須	有効な値
Id	ポリシーを一意に識別します。	いいえ	任意の文字列
Version	ポリシーアクセス言語のバージョンを指定します。	いいえ	任意の文字列。ベストプラクティスとして、このフィールド

名前	説明	必須	有効な値
			に値 "2012-10-17" を含めることをお勧めします。

## 個別のポリシーに関するステートメント

送信承認ポリシーには、少なくとも 1 つのステートメントが必要です。各ステートメントには、次の表に示す要素を含めることができます。

名前	説明	必須	有効な値
Sid	ステートメントを一意に識別します。	いいえ	任意の文字列。
Effect	評価時にポリシーのステートメントによって返される結果を指定します。	はい	"Allow" または "Deny"。
Resource	ポリシーが適用されるアイデンティティを指定します。これは、アイデンティティ所有者が代理送信者に使用を承認する E メールアドレスまたはドメインです。	はい	E メール ID の Amazon リソースネーム (ARN)。
Principal	ステートメントでアクセス許可を受け取る AWS アカウント、IAM ユーザー、または AWS のサービスを指定します。	はい	有効な AWS アカウント ID、IAM ユーザー ARN、または AWS のサービス。AWS アカウント ID と IAM ユーザー ARN は "AWS" を使用して指定します ("AWS": ["123456789012"], "AWS": ["arn:aws:iam::123456789012:r..."])。AWS サービス名は "Service" を使用して指定します ("Service": ["cognito-idp.amazonaws.com"]) など。  IAM ユーザー ARN の形式の例については、 <a href="#">AWS 一般リファレンス</a> を参照してください。
Action	ステートメントが適用される E メール送信アクションを指定します。	はい	"ses:SendEmail"、"ses:SendRawEmail"  これらのオペレーションは 1 つ以上指定できます。「ses:Send*」

名前	説明	必須	有効な値
			を指定して、これらのオペレーションをすべてを含めることもできます。代理送信者が SMTP インターフェイスを使用して E メールを送信する予定の場合は、「ses:SendRawEmail」を指定するか、「ses:Send*」を使用する必要があります。
Condition	アクセス許可についての制限や詳細を指定します。	いいえ	この表の後の条件についての情報を参照してください。

## Conditions

条件は、ステートメントのアクセス権限に関する制限のことです。ステートメントの中でも、記述が最も詳細になるのが、この条件部分です。キーは、リクエストの日時など、アクセス制限に使用される基本項目です。

制限は、条件とキーの両方を使用して定義します。たとえば、代理送信者が 2019 年 7 月 30 日以降はお客様の代わりに Amazon SES にリクエストを行うことができないように制限する場合、DateLessThan という条件を使用します。キーは aws:CurrentTime を使用し、値を 2019-07-30T00:00:00Z に設定します。

[IAM ユーザーガイド](#) の「利用可能なキー」で説明されている AWS 全体のキーを使用するか、Amazon SES に固有の次のいずれかのキーを使用できます。

条件キー	説明
ses:Recipients	受取人アドレスを制限します。To:、"CC"、"BCC" アドレスが含まれます。
ses:FromAddress	「From」アドレスを制限します。
ses:FromDisplayName	「From」表示名として使用される文字列の内容を制限します ("フレンドリ名" と呼ばれることもあります)。たとえば、"John Doe <johndoe@example.com>" の表示名は John Doe です。
ses:FeedbackAddress	「Return Path」アドレス (Eメールのフィードバック転送によりバウンスや苦情を送信できるアドレス) を制限します。Eメールのフィードバック転送の詳細については、「 <a href="#">Eメールで送信された Amazon SES 通知 (p. 274)</a> 」を参照してください。

Amazon SES キーで StringEquals 条件および StringLike 条件を使用することができます。これらの条件は、大文字小文字を区別する文字列の一致を指定するために使用します。StringLike の場合、値には、複数文字一致のワイルドカード (\*) または 1 文字一致のワイルドカード (?) を指定できます。文字列のどこにでも含めることができます。たとえば、次の条件は、先頭に「invoicing」、末尾に「@example.com」が付いた「From」アドレスからのみ代理送信者が送信できることを指定します。



```
"Condition": {
  "StringLike": {
    "ses:FromAddress": "invoicing*@example.com"
  }
}
```

また、StringNotLike 条件を使用して、代理送信者が特定の E メールアドレスから E メールを送信できないようにすることもできます。たとえば、admin@example.com や同様の E メールアドレス ("admin"@example.com、admin+1@example.com、sender@admin.example.com) からの送信を禁止するには、ポリシーステートメントの条件に以下を含めます。

```
"Condition": {
  "StringNotLike": {
    "ses:FromAddress": "*admin@example.com"
  }
}
```

条件の指定方法の詳細については、IAM ユーザーガイドの「IAM JSON ポリシーエレメント: 条件」を参照してください。

## ポリシーの要件

ポリシーは、以下の要件をすべて満たしている必要があります。

- 各ポリシーには、少なくとも 1 つのステートメントが含まれている必要があります。
- 各ポリシーには、少なくとも 1 つの有効なプリンシパルが含まれている必要があります。
- 各ポリシーには、1 つのリソースを指定する必要があります。またそのリソースは、ポリシーがアタッチされている ID の ARN である必要があります。
- アイデンティティ所有者は、最大 20 のポリシーにそれぞれ一意のアイデンティティを関連付けることができます。
- ポリシーのサイズが 4 キロバイト (KB) を超えることはできません。
- ポリシー名が 64 文字を超えることはできません。また、英数字、ダッシュ、アンダースコアのみを含めることができます。

## Amazon SES 送信承認ポリシーの例

送信承認では、代理送信者に代理送信を許可する際の細かい条件を指定することができます。

次の例は、送信のさまざまな側面をコントロールするポリシーを記述する方法を示しています。

- [代理送信者の指定 \(p. 161\)](#)
- ["From" アドレスの制限 \(p. 163\)](#)
- [代理送信者による E メール送信時間に対する制限 \(p. 164\)](#)
- [E メール送信アクションの制限 \(p. 164\)](#)
- [E メール送信者の表示名の制限 \(p. 165\)](#)
- [複数のステートメントの使用 \(p. 166\)](#)

## 代理送信者の指定

プリンシパル (許可を付与するエンティティ) は AWS アカウント、AWS Identity and Access Management (IAM) ユーザー、または AWS のサービスです。

以下の例は、検証済み ID example.com (AWS アカウント 888888888888 が所有) からのメール送信を AWS ID 123456789012 に許可する、シンプルなポリシーを示しています。このポリシーの Condition ステートメントにより、marketing+.\*@example.com アドレスからメールを送信できるのが代理送信者 (AWS ID 123456789012) に限定されています。\* は、送信者が marketing+ の後に追加する任意の文字列を示しています。

```
{
  "Id": "SampleAuthorizationPolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AuthorizeMarketer",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      },
      "Action": [
        "SES:SendEmail",
        "SES:SendRawEmail"
      ],
      "Condition": {
        "StringLike": {
          "ses:FromAddress": "marketing+.*@example.com"
        }
      }
    }
  ]
}
```

次のポリシー例では、IAM ユーザー 2人に、アイデンティティ example.com から送信するアクセス許可を付与します。IAM ユーザーは、自分の Amazon リソースネーム (ARN) によって指定されます。

```
{
  "Id": "ExampleAuthorizationPolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AuthorizeIAMUser",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:user/John",
          "arn:aws:iam::444455556666:user/Jane"
        ]
      },
      "Action": [
        "SES:SendEmail",
        "SES:SendRawEmail"
      ]
    }
  ]
}
```

次のポリシー例では、Amazon Cognito に、アイデンティティ example.com から送信するアクセス許可を付与します。

```
{
  "Id": "ExampleAuthorizationPolicy",
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AuthorizeService",
    "Effect": "Allow",
    "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
    "Principal": {
      "Service": [
        "cognito-idp.amazonaws.com"
      ]
    },
    "Action": [
      "SES:SendEmail",
      "SES:SendRawEmail"
    ]
  }
]
```

次の例のポリシーでは、AWS Organization 内のすべてのアカウントに、identity example.com から送信するアクセス許可が付与されます。AWS Organization は、[PrincipalOrgID](#) グローバル条件キーにより指定されます。

```
{
  "Id": "ExampleAuthorizationPolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AuthorizeOrg",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
      "Principal": "*",
      "Action": [
        "SES:SendEmail",
        "SES:SendRawEmail"
      ],
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "o-xxxxxxxxxxxx"
        }
      }
    }
  ]
}
```

## "From" アドレスの制限

検証済みドメインを使用する場合、代理送信者だけが特定の E メールアドレスから送信できるようにするポリシーを作成することをお勧めします。"From" アドレスを制限するには、キーで ses:FromAddress と呼ばれる条件を設定します。次のポリシーでは、AWS アカウント ID 123456789012 がアイデンティティ example.com から送信することを許可すると同時に、送信元の E メールアドレスを sender@example.com に限定しています。

```
{
  "Id": "ExamplePolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AuthorizeFromAddress",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
```

```
    "Principal":{
      "AWS":[
        "123456789012"
      ]
    },
    "Action":[
      "SES:SendEmail",
      "SES:SendRawEmail"
    ],
    "Condition":{
      "StringEquals":{
        "ses:FromAddress":"sender@example.com"
      }
    }
  }
}
```

## 代理送信者による E メール送信時間に対する制限

送信者の承認ポリシーを設定することで、代理送信者が E メールを送信できる日付と時間や、日付の範囲を限定することもできます。たとえば、2018 年 9 月内で E メールキャンペーンの送信を予定している場合、以下のポリシーを使用することで、代理送信者による E メール送信を 9 月に限定することができます。

```
{
  "Id":"ExamplePolicy",
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"ControlTimePeriod",
      "Effect":"Allow",
      "Resource":"arn:aws:ses:us-east-1:888888888888:identity/example.com",
      "Principal":{
        "AWS":[
          "123456789012"
        ]
      },
      "Action":[
        "SES:SendEmail",
        "SES:SendRawEmail"
      ],
      "Condition":{
        "DateGreaterThan":{
          "aws:CurrentTime":"2018-08-31T12:00Z"
        },
        "DateLessThan":{
          "aws:CurrentTime":"2018-10-01T12:00Z"
        }
      }
    }
  ]
}
```

## E メール送信アクションの制限

送信者が Amazon SES で E メールを送信するために使用できるアクションは、SendEmail と SendRawEmail の 2 つです。送信者が Eメールの形式をどの程度コントロールするかに応じて決定します。送信承認ポリシーでは、代理送信者をこれらの 2 つのアクションのいずれかに制限できます。ただし、多くのアイデンティティ所有者は、ポリシーで両方のアクションを有効にすることで、Eメール送信呼び出しの詳細を代理送信者に任せています。

## Note

代理送信者が SMTP インターフェイスを介して Amazon SES にアクセスできるようにする場合、少なくとも `SendRawEmail` を選択する必要があります。

アクションを制限する必要がある場合、送信承認ポリシーにどちらかのアクションのみ含めることで制限することができます。次の例は、アクションを `SendRawEmail` に制限する方法を示しています。

```
{
  "Id": "ExamplePolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ControlAction",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      },
      "Action": [
        "SES:SendRawEmail"
      ]
    }
  ]
}
```

## E メール送信者の表示名の制限

E メールクライアントによっては、実際の「From」アドレスではなく、E メール送信者の「フレンドリ」名 (E メールヘッダーで指定されている場合) が表示されます。たとえば、"John Doe <johndoe@example.com>" の表示名は John Doe です。例として、user@example.com から E メールを送信しますが、E メールが user@example.com ではなく「Marketing」から送信されたものとして受信者に表示したいとします。次のポリシーでは、「From」アドレスの表示名に「Marketing」が含まれている場合のみ、AWS アカウント ID 123456789012 がアイデンティティ example.com から送信できるようにします。

```
{
  "Id": "ExamplePolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AuthorizeFromAddress",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      },
      "Action": [
        "SES:SendEmail",
        "SES:SendRawEmail"
      ],
      "Condition": {
        "StringLike": {
          "ses:FromDisplayName": "Marketing"
        }
      }
    }
  ]
}
```

```
}
```

## 複数のステートメントの使用

送信承認ポリシーには複数のステートメントを含めることができます。以下のサンプルポリシーには、2つのステートメントが含まれています。最初のステートメントは、「From」アドレスとフィードバックアドレスの両方がドメイン example.com を使用している場合に限り、2つの AWS アカウントが sender@example.com から送信することを承認します。2番目のステートメントは、受信者の E メールアドレスが example.com ドメインに属している場合に限り、IAM ユーザーが sender@example.com から E メールを送信することを承認します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AuthorizeAWS",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:999999999999:identity/sender@example.com",
      "Principal": {
        "AWS": [
          "111111111111",
          "222222222222"
        ]
      },
      "Action": [
        "SES:SendEmail",
        "SES:SendRawEmail"
      ],
      "Condition": {
        "StringLike": {
          "ses:FromAddress": "*@example.com",
          "ses:FeedbackAddress": "*@example.com"
        }
      }
    },
    {
      "Sid": "AuthorizeInternal",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:999999999999:identity/sender@example.com",
      "Principal": {
        "AWS": "arn:aws:iam::333333333333:user/Jane"
      },
      "Action": [
        "SES:SendEmail",
        "SES:SendRawEmail"
      ],
      "Condition": {
        "ForAllValues:StringLike": {
          "ses:Recipients": "*@example.com"
        }
      }
    }
  ]
}
```

## Amazon SES 送信承認を行うための ID 所有者のタスク

このセクションでは、送信承認の設定時にアイデンティティ所有者が実行する必要がある手順について説明します。

## トピック

- [Amazon SES の送信承認を行うための ID の検証 \(p. 167\)](#)
- [Amazon SES の送信承認に使用する ID 所有者通知の設定 \(p. 167\)](#)
- [Amazon SES の送信承認を行うための代理送信者からの情報の入手 \(p. 167\)](#)
- [Amazon SES の送信承認ポリシーの作成 \(p. 168\)](#)
- [Amazon SES の送信承認に使用する ID 情報の代理送信者への提供 \(p. 171\)](#)
- [Amazon SES の送信承認ポリシーの管理 \(p. 171\)](#)

## Amazon SES の送信承認を行うための ID の検証

送信承認を設定するための最初の手順は、代理送信者が E メールを送信する E メールアドレスまたはドメインを自分が所有していることを証明することです。検証手順については、「[ID の検証 \(p. 48\)](#)」を参照してください。

E メールアドレスまたはドメインが確認されるかどうかは、[Amazon SES コンソール](#)の ID 管理セクションでステータスを確認するか、GetIdentityVerificationAttributes API オペレーションを使用して確認できます。

ユーザーまたは代理送信者が、検証されていない E メールアドレスに E メールを送信するには、アカウントを Amazon SES サンドボックスから削除するリクエストを送信する必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

### Important

ID 所有者と代理送信者の両方の AWS アカウントは、いずれかのアカウントが検証されていないアドレスに E メールを送信する前に、サンドボックスから削除される必要があります。

## Amazon SES の送信承認に使用する ID 所有者通知の設定

代理送信者がお客様の代わりに E メールを送信することを承認する場合、Amazon SES は、それらの E メールが生成するすべてのバウンスと苦情は、お客様ではなく代理送信者のバウンスと苦情の制限に対してカウントします。ただし、代理送信者から送信されたメッセージの結果、IP アドレスがサードパーティーのスパム対策 DNS ベースのブラックホールリスト(DNSBL)になった場合、ユーザーの ID の評判が損なわれる可能性があります。このため、あなたがアイデンティティ所有者である場合は、ID の E メールフィードバック転送を設定する必要があります。詳細については、「[E メールで送信された Amazon SES 通知 \(p. 274\)](#)」を参照してください。

代理送信者は、あなたが使用することを許可した ID に対して、自分自身のバウンスと苦情の通知を設定できます。また、バウンスや苦情のイベントが発生した場合に、通知を送信するように、[イベント発行 \(p. 296\)](#)を設定することもできます。ID 所有者がフィードバック転送を無効にする場合、代理送信者は、バウンスイベントと苦情イベントを Amazon SNS トピックまたは Kinesis Data Firehose ストリームに公開するようにイベント発行を設定する必要があります。ID 所有者および代理送信者のいずれも、バウンスイベントと苦情イベントの通知を送信する方法を設定していない場合、または送信者がイベント公開ルールを使用する設定セットを適用しない場合は、Eメールのフィードバック転送を無効にしているも、Amazon SES は、Eメールの Return-Path フィールドのアドレス (または Return-Path アドレスを指定しなかった場合はソースフィールド) に自動的にイベント通知を Eメールで送信します。次のイメージは、このプロセスを示したものです。

## Amazon SES の送信承認を行うための代理送信者からの情報の入手

送信承認ポリシーは、少なくとも 1 つのプリンシパル (アクセスを許可するエンティティ) を指定する必要があります。Amazon SES 送信承認ポリシーでは、プリンシパルは AWS アカウント、AWS Identity and Access Management (IAM) ユーザー、または AWS のサービスです。

選択するプリンシパルの種類は自由ですが、できる限り細かくコントロールする場合は、代理送信者の AWS アカウントに含まれるすべてのユーザーではなく 1 人の代理送信者だけがお客様の代わりに送信できるように、IAM ユーザーの設定を代理送信者に依頼してください。代理送信者が IAM ユーザーを設定するための情報は、[IAM ユーザーガイド](#)の「AWS アカウントでの IAM ユーザーの作成」にあります。

AWS アカウントまたは IAM ユーザーへのアクセスを許可するかどうかを決定したら、AWS アカウント ID、IAM ユーザー、または AWS のサービスの Amazon リソースネーム (ARN) を代理送信者に尋ねて、送信承認ポリシーに追加できるようにします。代理送信者には、「[ID 所有者への情報提供 \(p. 173\)](#)」にあるこの情報を調べるための手順を参照してもらいます。代理送信者が AWS サービスである場合、そのサービスのドキュメントを参照してサービス名を確認してください。

## Amazon SES の送信承認ポリシーの作成

所有する E メールアドレスまたはドメイン (ID) を使用して E メールを送信することを代理送信者に許可するには、送信承認ポリシーを作成し、そのポリシーを ID にアタッチします。ID には、0、1、または多くのポリシーを含めることができます。ただし、1 つのポリシーは 1 つの ID にのみ関連付けることができます。

送信承認ポリシーは、次のように作成できます。

- Policy Generator の使用 – Amazon SES コンソールで Policy Generator を使用してシンプルなポリシーを作成できます。E メールを送信可能なユーザーを指定するだけでなく、E メールを送信可能な日時範囲、"From" アドレス、"From" の表示名、バウンスと苦情が送信されるアドレス、受信者アドレス、送信元 IP に基づく条件を使用して E メール送信を制限することもできます。さらに、Policy Generator を使用すると、シンプルなポリシーの構造を作成した後、ポリシーを編集することでカスタマイズすることもできます。
- カスタムポリシーの作成 – 詳細な条件を追加したり、AWS のサービスをプリンシパルとして使用したりする場合、Amazon SES コンソールまたは Amazon SES API でカスタムポリシーを作成してアイデンティティにアタッチできます。

このトピックでは、両方の方法について説明します。

### Note

E メールアドレス ID にアタッチする送信承認ポリシーは、対応するドメイン ID にアタッチするポリシーよりも優先されます。たとえば、代理送信者を許可しない example.com のポリシーを作成し、代理送信者を許可する sender@example.com のポリシーを作成した場合、代理送信者は sender@example.com からメールを送信できますが、example.com ドメインの他のアドレスから送信することはできません。

代理送信者を許可する example.com のポリシーを作成し、代理送信者を拒否する sender@example.com のポリシーを作成した場合、代理送信者は example.com ドメインの任意のアドレスからメールを送信できますが、sender@example.com ドメインの他のアドレスから送信することはできません。

## Policy Generator を使用したポリシーの作成

Policy Generator を使用すると、次の手順を使用してシンプルな送信承認ポリシーを作成できます。

Policy Generator を使用してポリシーを作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインの [Identity Management] で、[Domains] または [Email Addresses] を選択します。
3. ID のリストで、ポリシーを作成する ID を選択します。



4. 詳細ペインで、[Identity Policies] を展開し、[Create Policy]、[Policy Generator] の順に選択します。
5. ウィザードで、次のフィールドの値を選択することでポリシーステートメントを作成します。これらのオプションについては、「[送信承認ポリシー \(p. 157\)](#)」を参照してください。
  - Effect – アクセスを許可する場合、[Allow] を選択します。それ以外の場合は [Deny] を選択します。
  - Principals – アクセスを許可または拒否する 12 桁の AWS アカウント ID または IAM ユーザーの ARN を入力し、[Add] を選択します。この手順を繰り返すことにより、さらにプリンシパルを追加できます。AWS アカウント ID の例は 123456789012 です。IAM ユーザー ARN の例は `arn:aws:iam::123456789012:user/John` です。

#### Note

ポリシーの生成ウィザードでは現在、AWS サービスプリンシパルはサポートされていません。AWS サービスプリンシパルを追加するには、カスタムポリシーを作成する必要があります。あるいは、ポリシージェネレーターで AWS アカウントまたは IAM ユーザープリンシパルを追加してから、そのポリシーを[編集 \(p. 170\)](#)する必要があります。

- Actions – このポリシーを適用する E メール送信アクセスを選択します。通常、アイデンティティ所有者は両方のオプションを選択し、代理送信者が E メール送信の実装方法を選択できるようにします。詳細については、「」を参照してください。[個別のポリシーに関するステートメント \(p. 159\)](#)
6. (任意) ポリシーに制限を追加する場合、[Add Conditions (条件の追加)] を選択し、次の情報を選択します。
    - Key – これは、アクセス制限に使用される基本項目です。Policy Generator では、Amazon SES 固有のキーまたは使用頻度の少し低い AWS 全体のキー (現在時刻と送信元 IP) を選択できます。詳細については、「[Conditions \(p. 160\)](#)」を参照してください。「[利用可能なキー](#)」にリストされているより高度な AWS 全体のキーを指定する場合、ポリシーを作成後に編集できます。
    - Condition – これは、指定する条件の種類です。たとえば、文字列条件、数値条件、日時条件などがあります。条件のリストについては、[IAM ユーザーガイド](#)の「[条件のタイプ](#)」を参照してください。
    - Value – これは、条件に対してテストされる値です。例については、「[送信承認ポリシーの例 \(p. 161\)](#)」のポリシーを参照してください。

キー、条件、値を選択した後、[Add Condition] を選択します。条件が [Conditions] リストに表示されます。リストの条件の横にある [Remove] を選択すると、条件を削除できます。[Add Conditions] をもう一度選択すると、別の条件を追加できます。

**Policy Generator**

With this tool, you can create a basic sending authorization policy by generating simple statements. [Learn more](#) about using the Policy Generator. If you want to include more advanced conditions in your policy, you can edit the policy later.

**Identity** example.com ⓘ

**Effect**  Allow  Deny ⓘ

**Principals\***  + ⓘ

**Actions\***

- ses:SendEmail ⓘ
- ses:SendRawEmail ⓘ

Fields marked with asterisk (\*) are required.

[Add Conditions \(optional\)](#)

**Key** ses:FromAddress ⓘ

**Condition** StringEquals ⓘ

**Value** marketing@example.com ⓘ

7. 条件の追加が完了したら、[Add Statement] を選択します。ステートメントが [Statements] リストに表示されます。ここでは、ステートメントを選択して編集または削除することができます。ステップ 5 ~7 を繰り返すことでステートメントを追加できます。
8. ステートメントの追加が完了したら、[Next] を選択します。
9. [Edit Policy] ダイアログボックスで、ポリシーを確認して必要に応じて編集し、[Apply Policy] を選択します。

## カスタムポリシーの作成

カスタムポリシーを作成してアイデンティティにアタッチする場合、次のオプションがあります。

- Amazon SES API の使用 – [Amazon Simple Email Service API リファレンス](#) で説明されている PutIdentityPolicy API を使用し、テキストエディタでポリシーを作成してアイデンティティにアタッチします。
- Amazon SES コンソールの使用 – テキストエディタを使用してポリシーを作成し、それを Amazon SES コンソールのカスタムポリシーエディタに貼り付けることでアイデンティティにアタッチします。以下の手順では、この方法について説明します。

カスタムポリシーエディタを使用してカスタムポリシーを作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインの [Identity Management] で、[Domains] または [Email Addresses] を選択します。
3. ID のリストで、ポリシーを作成する ID を選択します。
4. 詳細ペインで、[Identity Policies] を展開し、[Create Policy]、[Custom Policy] の順に選択します。
5. [Edit Policy] ペインで、ポリシーのテキストを貼り付けます。
6. [Apply Policy] を選択します。

## Amazon SES の送信承認に使用する ID 情報の代理送信者への提供

送信承認ポリシーを作成してアイデンティティにアタッチしたら、代理送信者にアイデンティティの Amazon リソースネーム (ARN) を提供します。代理送信者は、E メール送信操作または E メールヘッダーで ARN を Amazon SES に渡します。次の手順を使用して、アイデンティティの ARN を調べます。

アイデンティティの ARN を調べるには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインの [Identity Management] で、[Domains] または [Email Addresses] を選択します。
3. ID のリストで、送信承認ポリシーをアタッチしたアイデンティティを選択します。
4. 詳細ペインの上部にある [Identity ARN] の後に、アイデンティティの ARN があります。arn:aws:ses:us-east-1:123456789012:identity/user@example.com のような内容です。ARN 全体をコピーし、代理送信者に提供します。

## Amazon SES の送信承認ポリシーの管理

「[ポリシーの作成 \(p. 168\)](#)」で説明したポリシーの作成とアイデンティティへのアタッチに加えて、次のセクションで説明するようにアイデンティティのポリシーを編集、削除、リスト、取得することができます。

### Note

アクセス許可を取り消すには、ポリシーを編集または削除します。

### ポリシーの編集

ポリシーを編集する最も簡単な方法は、Amazon SES コンソールを使用することです。代わりに Amazon SES API を使用する場合は、[GetIdentityPolicies](#) オペレーションを使用してポリシーを取得し、テキストエディタを使用してポリシーを編集した後、[PutIdentityPolicy](#) API を使用して古いポリシーを上書きできます。

以下の手順では、Amazon SES コンソールを使用してポリシーを編集する方法を示します。

Amazon SES コンソールを使用してポリシーを編集するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインの [Identity Management] で、[Domains] または [Email Addresses] を選択します。

3. ID のリストで、編集するポリシーに関連付けられたアイデンティティを選択します。
4. 詳細ペインで、[Identity Policies] を展開します。
5. 編集したいポリシーの隣にある [Edit Policy] を選択します。
6. [Edit Policy] ペインで、ポリシーを編集して [Apply Policy] を選択します。
7. [Overwrite Existing Policy] ダイアログボックスで、[Overwrite] を選択します。

## ポリシーの削除

ポリシーを削除するだけで、アクセス許可をいつでも取り消すことができます。 [DeleteIdentityPolicy](#) API オペレーションを使用して、または以下の手順で説明しているように、Amazon SES コンソールを使用してポリシーを削除できます。

### Important

ポリシーを削除すると、元に戻す方法はありません。ポリシーを削除する前に、テキストファイルにポリシーをコピーアンドペーストして、ポリシーをバックアップすることをお勧めします。

Amazon SES コンソールを使用してポリシーを削除するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインの [Identity Management] で、[Domains] または Email Addresses を選択します。
3. ID のリストで、削除するポリシーに関連付けられた ID を選択します。
4. 詳細ペインで、[Identity Policies] を展開します。削除したいポリシーの隣にある [Remove Policy] を選択します。
5. Remove Policy ダイアログボックスで、[Yes, Remove Policy] を選択します。

## ポリシーのリストと取得

[ListIdentityPolicies](#) API オペレーションを使用して、ID にアタッチされたポリシーをリストできます。 [GetIdentityPolicies](#) API オペレーションを使用することでポリシー自体を取得することもできます。

これらのタスクを実行するために、次の手順で説明するように Amazon SES コンソールを使用することもできます。

Amazon SES コンソールを使用して、アイデンティティにアタッチされているポリシーを一覧表示するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインの [Identity Management] で、[Domains] または [Email Addresses] を選択します。
3. ID のリストで、ポリシーを表示する ID を選択します。
4. 詳細ペインで、[Identity Policies] を展開します。
5. 表示したいポリシーの隣にある [Show Policy] を選択します。

## Amazon SES の送信承認を行うための代理送信者のタスク

代理送信者は、クロスアカウント E メールを送信します。つまり、自分が所有していないが使用を承認されたアイデンティティの代わりに E メールを送信します。アイデンティティ所有者の代わりに送信しま

すが、バウンスと苦情は自分の AWS アカウントのバウンスと苦情のメトリクスに対してカウントされ、メッセージ送信数は自分の送信クォータに対してカウントされます。アイデンティティ所有者の E メールを送信するために送信クォータの増加が必要になった場合は、それをリクエストする責任も自分にあります。

代理送信者として、次のタスクを完了する必要があります。

- [ID 所有者への情報提供 \(p. 173\)](#)
- [代理送信者通知の使用 \(p. 173\)](#)
- [ID 所有者の Eメールの送信 \(p. 176\)](#)

## Amazon SES の送信承認を行うための ID 所有者への情報提供

代理送信者は、自分の AWS アカウント ID または アイデンティティ所有者の代わりに E メールを送信する AWS Identity and Access Management (IAM) ユーザーの Amazon リソースネーム (ARN) をアイデンティティ所有者に提供する必要があります。この情報は、次の手順を使用して確認できます。

AWS アカウント ID を確認するには

1. AWS Management Console (<https://console.aws.amazon.com>) にサインインします。
2. ナビゲーションメニューで自分の名前を選択し、[My Account] を選択します。
3. [Account Settings] を展開します。このセクションでは、AWS アカウント ID が表示されます。

IAM ユーザーの ARN を確認するには

1. AWS Management Console にサインインして、IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [Users] を選択します。
3. ユーザーのリストで、ユーザー名を選択します。[Summary] セクションに ARN が表示されます。ARN は次の例のようになります。arn:aws:iam::123456789012:user/John。

## Amazon SES の送信承認を行うための代理送信者通知の使用

代理送信者の場合、バウンスと苦情は、アイデンティティ所有者ではなく自分のバウンスと苦情のメトリクスに対してカウントされます。アカウントのバウンス率や苦情率が高くなりすぎる場合、アカウントを確認し、アカウントの E メール送信機能を一時停止することがあります。したがって、通知を設定したり、通知を監視するためのプロセスを備えることが重要です。バウンスや苦情が発生しているアドレスをメーリングリストから削除するためのプロセスを備えることも重要です。

代理送信者は、バウンスや苦情のイベントが発生したときに通知を送信するように Amazon SES をセットアップできます。代理送信者は、Amazon SNS または Kinesis Data Firehose にバウンスや苦情の通知を発行するように [イベント発行 \(p. 296\)](#) をセットアップすることもできます。

### Note

Amazon SNS を使用して通知を送信するように Amazon SES を設定すると、受け取る通知に対して Amazon SNS の標準レートが課金されます。詳細については、「[Amazon SNS 料金](#)」ページを参照してください。

### トピック

- [Amazon SES のクロスアカウント ID 通知設定の実行 \(p. 174\)](#)
- [Amazon SES のクロスアカウント通知設定の編集 \(p. 174\)](#)
- [Amazon SES のクロスアカウント ID 通知の参照 \(p. 175\)](#)
- [Amazon SES のクロスアカウント ID 通知設定の削除 \(p. 175\)](#)

## Amazon SES のクロスアカウント ID 通知設定の実行

通知を設定する前に、アイデンティティ所有者によって使用が承認されており、通知を設定するアイデンティティの Amazon リソースネーム (ARN) を知っておく必要があります。たとえば、アイデンティティ `user@example.com` の ARN は `arn:aws:ses:us-east-1:123456789012:identity/user@example.com` のようになります。ID 所有者から ID の ARN が提供されていない場合、「[ID 情報の代理送信者への提供 \(p. 171\)](#)」の手順を参照してもらってください。

通知を設定する最も簡単な方法は、Amazon SES コンソールを使用することです。[SetIdentityNotificationTopic](#) API オペレーションを使用して、アイデンティティの ARN を `Identity` パラメーターとして渡すこともできます。

次の手順は、Amazon SES コンソールを使用して通知をセットアップする方法を示しています。

Amazon SES コンソールを使用して Amazon SNS のバウンス、苦情、または配信の通知を設定するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで、[Cross-Account Notifications] (クロスアカウント通知) を選択します。
3. [Add Notification Config] を選択します。
4. [Add Notification Configuration] (通知設定の追加) ダイアログボックスの、[Identity ARN] (ID の ARN) に設定する ID の ARN を入力します。現在ログインしているアカウントに属するアイデンティティは使用できません。
5. バウンス、苦情、配信に使用する Amazon SNS トピックを選択します。これらの通知に対して新しい Amazon SNS トピックを作成することもできます。

### Important

Amazon SES 通知に使用する Amazon SNS トピックは、Amazon SES を使用して E メールを送信するのと同じ AWS リージョン内である必要があります。

バウンス、苦情、配信の通知を同じ Amazon SNS トピックに発行するか、異なる Amazon SNS トピックに発行するかは選択可能です。所有していない Amazon SNS トピックを使用する場合、そのトピックの所有者が、そのトピックでの `SNS:Publish` アクションの呼び出しをお客様のアカウントに許可する Amazon SNS アクセスポリシーを設定する必要があります。IAM ポリシーを使用して Amazon SNS トピックへのアクセスをコントロールする方法については、「[Amazon SNS トピックへのアクセスの管理](#)」を参照してください。

6. [Save Config] を選択して通知設定を保存します。これらの変更が反映されるまでにわずかな遅延が生じることがあります。

## Amazon SES のクロスアカウント通知設定の編集

通知設定を編集する最も簡単な方法は、Amazon SES コンソールを使用することです。代わりに Amazon SES API を使用する場合は、[SetIdentityNotificationTopic](#) API オペレーションを使用して、アイデンティティの ARN を `Identity` パラメーターとして渡すことができます。

次の手順は、Amazon SES コンソールを使用してクロスアカウント通知設定を編集する方法を示しています。

Amazon SES コンソールを使用してクロスアカウント通知設定を編集するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで、[Cross-Account Notifications] を選択します。

通知を設定したクロスアカウントアイデンティティは、[Cross-Account Notifications] 詳細ペインにリストされます。

3. 通知設定を表示するアイデンティティの ARN を選択します。
4. 通知設定を編集し、[Save Config] を選択します。

## Amazon SES のクロスアカウント ID 通知の参照

通知設定を参照する最も簡単な方法は、Amazon SES コンソールを使用することです。[GetIdentityNotificationAttributes](#) API オペレーションを使用して、アイデンティティの ARN を `Identity` パラメータとして渡すこともできます。

### Note

クロスアカウントアイデンティティリストに表示されるクロスアカウントアイデンティティは、「[通知設定の実行 \(p. 174\)](#)」で説明されている手順を使用して通知を設定したアイデンティティだけです。

Amazon SES コンソールを使用してクロスアカウント通知設定を参照するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで、[Cross-Account Notifications] を選択します。

通知を設定したクロスアカウントアイデンティティは、[Cross-Account Notifications] 詳細ペインにリストされます。

3. アイデンティティの ARN を選択します。

[Edit Configuration Notification] ダイアログボックスにアイデンティティ設定が表示されます。

## Amazon SES のクロスアカウント ID 通知設定の削除

通知設定を削除する最も簡単な方法は、Amazon SES コンソールを使用することです。[SetIdentityNotificationTopic](#) API オペレーションを使用して、アイデンティティの ARN を `Identity` パラメータとして渡し、`SnsTopic` パラメータの `null` を渡すこともできます。通知設定を完全に削除するには、設定された通知の種類 (バウンス、苦情、または配信) ごとにこのオペレーションを実行する必要があります。

### Note

通知設定を削除すると、クロスアカウントアイデンティティの ARN が Amazon SES コンソールのクロスアカウントアイデンティティ ARN のリストから削除されます。これは、そのアイデンティティで送信を続けることができなくなるということではありません。バウンス、苦情、または配信の通知を受け取る設定ではなくなったという意味にすぎません。通知を再度有効にする場合、「[通知設定の実行 \(p. 174\)](#)」で説明されている通知設定手順を繰り返す必要があります。

次の手順は、Amazon SES コンソールを使用してクロスアカウント通知設定を削除する方法を示しています。

Amazon SES コンソールを使用してクロスアカウント通知設定を削除するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで、[Cross-Account Notifications] を選択します。

通知を設定したクロスアカウントアイデンティティは、[Cross-Account Notifications] 詳細ペインにリストされます。

3. 削除するクロスアカウントアイデンティティの左側にあるボックスをオンにし、[Remove] を選択します。

4. Remove Cross-Account Notification Config ダイアログボックスで、Delete Notification config を選択します。

クロスアカウントアイデンティティの ARN が、クロスアカウントアイデンティティ x のリストに表示されなくなります。これは、そのアイデンティティで送信できないことを意味するのではなく、通知の設定が解除されていることを意味するにすぎません。

## Amazon SES の送信承認時の ID 所有者の E メールを送信

代理送信者とは、他の Amazon SES 送信者と同じ方法で E メールを送信しますが、ID 所有者により使用を承認された ID の ARN を指定する点のみ異なります。E メールを送信するために Amazon SES を呼び出すと、Amazon SES は送信を承認するポリシーが指定された ID にあるかどうかを確認します。

E メールを送信するときにアイデンティティの ARN を指定する方法はいくつかあります。使用できる方法は、Eメールの送信に API オペレーションを使用するか、Amazon SES SMTP インターフェイスを使用するかで異なります。

### Important

E メールを正常に送信するには、ID 所有者が ID を検証した AWS リージョンの Amazon SES エンドポイントに接続する必要があります。

さらに、ID 所有者と代理送信者の両方の AWS アカウントは、いずれかのアカウントが検証されていないアドレスに E メールを送信する前に、サンドボックスから削除される必要があります。詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

## Amazon SES API の使用

すべての Amazon SES E メール送信者と同様に、Amazon SES API を介して Amazon SES にアクセスした場合 (HTTPS 経由で直接、または AWS SDK を通じて間接的に)、2 つの E メール送信アクション (SendEmail と SendRawEmail) のいずれかを選択できます。これらの API の詳細は「[Amazon Simple Email Service API Reference](#)」で説明されていますが、ここでは送信承認パラメータの概要を示します。

### SendRawEmail

SendRawEmail を使用して Eメールの形式をコントロールできるようにする場合、次の 2 つの方法のいずれかを使用してクロスアカウント ID を指定できます。

- オプションパラメーターを `SendRawEmail` API に渡す。必須のパラメーターを次の表で説明します。

Parameter	説明
SourceArn	Source の SendRawEmail パラメータで指定された E メールアドレスから送信することを許可する送信承認ポリシーに関連付けられたアイデンティティの ARN。  Note  SourceArn のみを指定すると、Amazon SES は「From」アドレスと「Return Path」アドレスを SourceArn で指定された ID に設定します。
FromArn	raw Eメールのヘッダーで特定の「From」アドレスを指定することを許可する送信承認ポリシーに関連付けられたアイデンティティの ARN。
ReturnPathArn	ReturnPath の SendRawEmail パラメータで指定された E メールアドレスを使用することを許可する送信承認ポリシーに関連付けられた ID の ARN。



- E メールに X ヘッダーを含める。X ヘッダーは、標準的な E メールヘッダーに加えて使用できるカスタムヘッダーです (「From」、「Reply-To」、または「Subject」ヘッダーなど)。Amazon SES が認識する以下の 3 つの X ヘッダーを使用して、送信承認パラメータを指定することができます。

### Important

これらの X ヘッダーは、Eメールの送信前に Amazon SES により削除されるため、DKIM 署名には含めないでください。

X ヘッダー	説明
X-SES-SOURCE-ARN	SourceArn に対応します。
X-SES-FROM-ARN	FromArn に対応します。
X-SES-RETURN-PATH-ARN	ReturnPathArn に対応します。

Amazon SES は送信前に E メールからすべての X ヘッダーを削除します。X ヘッダーのインスタンスを複数含めた場合、Amazon SES は最初のインスタンスだけを使用します。

次の例は、送信認証 X ヘッダーを含む E メールを示しています。

```
X-SES-SOURCE-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com
X-SES-FROM-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com
X-SES-RETURN-PATH-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com

From: sender@example.com
To: recipient@example.com
Return-Path: feedback@example.com
Subject: subject
Content-Type: multipart/alternative;
  boundary="=====_boundary"

-----=_boundary
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

body
-----=_boundary
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: 7bit

body
-----=_boundary--
```

## SendEmail

SendEmail オペレーションを使用する場合、以下のオプションパラメータを渡すことでクロスアカウントアイデンティティを指定できます。SendEmail オペレーションを使用する場合、X ヘッダーは使用できません。

Parameter	説明
SourceArn	Source の SendRawEmail パラメータで指定された E メールアドレスから送信することを許可する送信承認ポリシーに関連付けられたアイデンティティの ARN。

Parameter	説明
ReturnPathArn	ReturnPath の SendRawEmail パラメータで指定された E メールアドレスを使用することを許可する送信承認ポリシーに関連付けられた ID の ARN。

以下の例は、SourceArn および ReturnPathArn 属性が含まれている E メールを、SendEmail オペレーションと [SDK for Python](#) を使用して送信する方法を示しています。

```
import boto3
from botocore.exceptions import ClientError

# Create a new SES resource and specify a region.
client = boto3.client('ses', region_name="us-west-2")

# Try to send the email.
try:
    # Provide the contents of the email.
    response = client.send_email(
        Destination={
            'ToAddresses': [
                'recipient@example.com',
            ],
        },
        Message={
            'Body': {
                'Html': {
                    'Charset': 'UTF-8',
                    'Data': 'This email was sent with Amazon SES.',
                },
            },
            'Subject': {
                'Charset': 'UTF-8',
                'Data': 'Amazon SES Test',
            },
        },
        SourceArn='arn:aws:ses:us-west-2:123456789012:identity/example.com',
        ReturnPathArn='arn:aws:ses:us-west-2:123456789012:identity/example.com',
        Source='sender@example.com',
        ReturnPath='feedback@example.com'
    )
# Display an error if something goes wrong.
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print("Email sent! Message ID:"),
    print(response['ResponseMetadata']['RequestId'])
```

## Amazon SES SMTP インターフェイスの使用

クロスアカウント送信で Amazon SES SMTP インターフェイスを使用する場合は、X-SES-SOURCE-ARN、X-SES-FROM-ARN、および X-SES-RETURN-PATH-ARN の各ヘッダーをメッセージに含める必要があります。これらのヘッダーは、SMTP 通信で DATA コマンドを発行した後に渡します。

# Amazon SES での専用 IP アドレスの使用

Amazon SES アカウントを新規作成すると、他の Amazon SES ユーザーとの共有 IP アドレスからお客様の E メールが送信されます。[追加の月額料金](#)で、お客様用に予約された専用 IP アドレスをリースできま

す。この 2 種類のアドレスの利点と欠点を次の表にまとめました。[利点] 列の項目をクリックすると、その利点に関する詳細情報が表示されます。

利点	共有 IP アドレス	専用 IP アドレス
<a href="#">追加の設定なしで使用可能 (p. 179)</a>	はい	いいえ
<a href="#">AWS が管理する評価 (p. 179)</a>	はい	いいえ
<a href="#">継続的で予測可能な送信パターンを持つお客様向け (p. 180)</a>	はい	はい
<a href="#">予測可能生が低い送信パターンを持つお客様向け (p. 180)</a>	はい	いいえ
<a href="#">送信量が多い送信者向け (p. 180)</a>	はい	はい
<a href="#">送信量が少ない送信者向け (p. 180)</a>	はい	いいえ
<a href="#">追加の月額料金 (p. 180)</a>	いいえ	はい
<a href="#">送信者の評価に関する完全な制御 (p. 180)</a>	いいえ	はい
<a href="#">メールの種類、受取人、またはその他の要因別の独立した評価 (p. 180)</a>	いいえ	はい
<a href="#">変更されることがない既知の IP アドレスの提供 (p. 181)</a>	いいえ	はい

### Important

大量の E メールを定期的に送信する予定がない場合は、共有 IP アドレスの使用をお勧めします。少量のメールの送信に専用 IP アドレスを使用する場合や、送信パターンが非常に不規則である場合は、配信性能に問題が生じることがあります。

## 容易なセットアップ

共有 IP アドレスを使用する場合、追加設定を行う必要はありません。E メールアドレスを確認してサンドボックス外に移動するとすぐに、Amazon SES アカウントはメールを送信できる状態になります。

専用 IP アドレスをリースする場合は、[リクエストを送信 \(p. 185\)](#)し、オプションで[専用 IP プールを設定す \(p. 181\)](#)必要があります。

## AWS が管理する評価

IP アドレスの評価は、主に送信パターンと送信量の履歴に基づきます。長期にわたって一定量の E メールを送信している IP アドレスは、通常、評価が高くなります。

共有 IP アドレスは何人かの Amazon SES のお客様によって使用されます。これらのお客様は、合わせると大量の E メールを送信しています。AWS では、この送信トラフィックを慎重に管理して、共有 IP アドレスの評価を最大限に高めています。

専用 IP アドレスを使用する場合、予測可能な一定量の E メールを送信して送信者の評価を維持することはお客様の責任となります。

#### Note

専用 IP のスマートネットワークデータサービス (SNDS) データを表示することについての詳細は、「[専用 IP の SNDS メトリクス \(p. 394\)](#)」を参照してください。

## 送信パターンの予測可能性

E メール送信の一貫した履歴を持つ IP アドレスは、以前の送信履歴がなくて急に大量の E メール送信を開始する IP アドレスよりも評価が高くなります。

E メール送信パターンが不規則である (つまり、予測可能なパターンではない) 場合、共有 IP アドレスの方がニーズに合っている可能性があります。共有 IP アドレスを使用すると、状況に応じて E メール送信パターンを増減できます。

専用 IP アドレスを使用する場合は、Eメールの送信量を毎日少しずつ増やして、それらのアドレスをウォームアップする必要があります。新しい IP アドレスをウォームアップするプロセスについては、「[専用 IP アドレスのウォームアップ \(p. 184\)](#)」で説明しています。専用 IP アドレスのウォームアップが完了したら、一貫した送信パターンを維持する必要があります。

## 送信メールのボリューム

専用 IP アドレスは、大量の E メールを送信するお客様に最適です。ほとんどのインターネットサービスプロバイダー (ISP) は、特定の IP アドレスから大量の E メールを受け取った場合にのみ、そのアドレスの評価を追跡します。評価を得る対象の ISP ごとに毎月 1 回以上、24 時間以内に数百通の E メールを送信する必要があります。

状況に応じて、大量の E メールを送信しない場合でも専用 IP アドレスを使用できます。たとえば、正しく定義された小グループの受取人が、IP アドレスの評価ではなく特定の IP アドレスのリストに基づいて E メールを承認または拒否するメールサーバーを利用している場合、これらの受取人に送信するには専用 IP アドレスが適していることがあります。

## 追加料金

共有 IP アドレスの使用には、Amazon SES の標準料金が適用されます。専用 IP アドレスをリースすると、Amazon SES を使用したメール送信に関連する標準料金以外に、追加の月額料金が発生します。専用 IP アドレスの月額料金は、アドレスごとに別々です。料金の詳細については、「[Amazon SES 料金表のページ](#)」を参照してください。

## 送信者の評価に関する制御

専用 IP アドレスを使用する場合、これらのアドレスから E メールを送信できるのは Amazon SES アカウントのみです。したがって、リースした専用 IP アドレスの送信者評価は、E メール送信状況に依存します。

## 送信者評価の分離

専用 IP アドレスを使用すると、E メールプログラムのコンポーネント別に送信者の評価を分離できます。Amazon SES 用に複数の専用 IP アドレスを使用する場合は、専用 IP プール (専用 IP アドレスのグループ) を作成し、特定タイプの Eメールの送信に使用できます。たとえば、マーケティングメールの送信用と取引メールの送信用に専用 IP アドレスのプールをそれぞれ作成できます。詳細については、[専用 IP プールの作成 \(p. 185\)](#)を参照してください。

## 変わることがない既知の IP アドレス

専用 IP アドレスを使用する場合、Amazon SES コンソールの専用 IP アドレスでメールを送信するアドレスの値を見つけることができます。専用 IP アドレスは変更されません。

共有 IP アドレスでは、Amazon SES で E メール送信に使用される IP アドレスは既知ではなく、いつでも変更される場合があります。

## 専用 IP アドレスのリクエストと解放

このセクションでは、[AWS SupportCenter](#)でリクエストを送信し、専用 IP アドレスをリクエストおよび解放する方法について説明します。リースして Amazon SES で使用する専用 IP アドレスごとに月額料金が追加で請求されます。専用 IP アドレスに伴うコストの詳細については、「[Amazon SES 料金表](#)」を参照してください。

## 専用 IP アドレスを使用するためのベストプラクティス

最低契約金はありませんが、Amazon SES を利用する AWS リージョンごとに複数の専用 IP アドレスをリースすることをお勧めします。各 AWS リージョンは、複数のそれぞれ独立した場所で構成されており、これらの場所はアベイラビリティゾーンと呼ばれます。複数の専用 IP アドレスをリースすると、リクエストでそのアドレスが指定された AWS リージョンのアベイラビリティゾーン間で、できるだけ均等に分散されます。このようにしてアベイラビリティゾーン間で専用 IP アドレスを分散することで、専用 IP アドレスの可用性と冗長性が高まります。

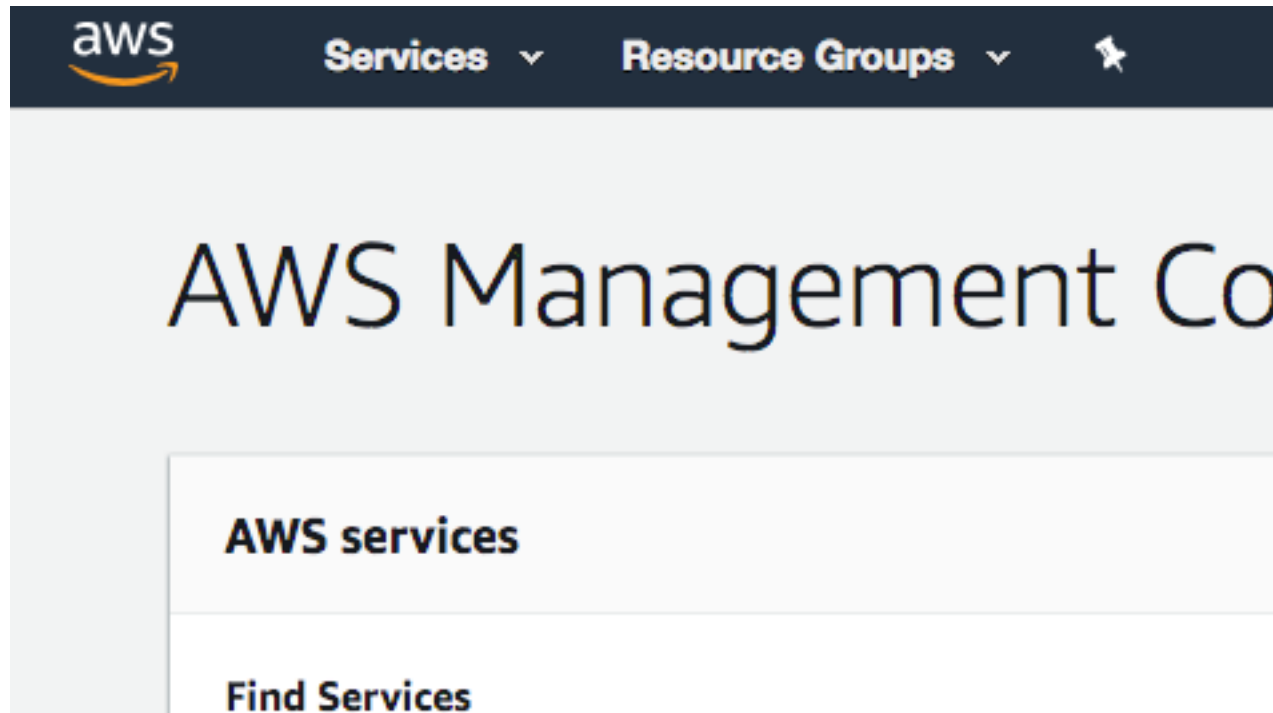
Amazon SES が現在利用可能な リージョンの一覧については[Amazon Web Services 全般のリファレンス](#)の「AWS リージョンとエンドポイント」を参照してください。各リージョンで利用できるアベイラビリティゾーンの数の詳細については、「[AWS グローバルインフラストラクチャ](#)」を参照してください。

## 専用 IP アドレスのリクエスト

次のステップで、AWS Support Center でサービスクォータの引き上げのケースを作成して専用 IP アドレスをリクエストする方法を説明します。このプロセスを使用して、必要な数の専用 IP アドレスをリクエストできます。

専用 IP アドレスをリクエストするには

1. AWS Management Console (<https://console.aws.amazon.com/>) にサインインします。
2. サポートメニューで、次の図のように サポートセンター を選択します。



3. [Create case (ケースの作成)] を選択します。
4. Create case (ケースの作成) で Service limit increase (サービスの上限緩和) を選択します。
5. [Case classification (ケースの分類)] で、以下のセクションに入力します。
  - Limit Type (制限のタイプ) で SES Sending Limits (SES サービスの制限) を選択します。
  - [Mail Type (メールの種類)] で、専用 IP アドレスを使用して送信を予定しているメールのタイプを選択します。複数の値が当てはまる場合は、送信する E メールの大部分に当てはまるオプションを選択します。
  - [Website URL (ウェブサイトの URL)] に、ウェブサイトの URL を入力します。この情報は、お客様が送信を予定しているコンテンツのタイプを AWS が正しく理解するために役立ちます。
  - [My email sending complies with the AWS Service Terms and AUP] で、ユースケースに当てはまるオプションを選択します。
  - [I only send to recipients who have specifically requested my mail (私は明確にリクエストされた受信者のみメールを送信します)] で、ユースケースに当てはまるオプションを選択します。
  - [I have a process to handle bounces and complaints (バウンスや苦情を処理するプロセスがあります)] で、ユースケースに当てはまるオプションを選択します。
6. [Requests (リクエスト)] で、以下のセクションに入力します。
  - [Region (リージョン)] で、リクエストが適用される AWS リージョンを選択します。
  - Limit (制限) で、Desired Maximum Send Rate (希望する最大送信レート) を選択します。
  - [New limit value (新しい制限値)] で、1 秒あたりに送信できる必要があるメッセージの最大数を入力します。指定された値は、ユースケースを実装するために必要な専用 IP アドレス数を計算するために使用されます。そのため、指定する見込み数は正確である必要があります。

#### Note

単一の専用 IP アドレスは、このステップで選択された AWS リージョンでのみ使用できます。他の AWS リージョンで使用する専用 IP アドレスをリクエストする場合は、[Add another request (別のリクエストを追加)] を選択して、別のリージョンについて [Region

(リージョン)、[Limit (制限)]、New limit value (新しい制限値) の各フィールドを入力します。専用 IP アドレスを使用する各リージョンでこのプロセスを繰り返します。

7. Case description (ケースの説明) の [Use case description (ユースケースの説明)] に、専用 IP アドレスをリクエストする旨を記載します。特定の数の専用 IP アドレスを希望する場合は、それも記載します。専用 IP アドレスの数を指定しない場合は、前のステップで指定された送信レートの要件を満たすために必要な数の専用 IP アドレスが提供されます。

次に、Amazon SES を使用して専用 IP アドレスから E メールを送信する方法について説明します。共有 IP アドレスではなく専用 IP アドレスを使用する理由に関する情報も含めます。この情報は、AWS がユースケースを理解する役に立ちます。

8. Contact options (問い合わせ先のオプション) の [Preferred contact language (優先される問い合わせ言語)] で、このケースに関する AWS からの問い合わせは [English (英語)] がいいか Japanese (日本語) がいいかを選択します。
9. 完了したら、[Submit] を選択します。

フォームを送信すると、リクエストの評価が行われます。リクエストが承認された場合は、サポートセンターのケースに、新しい専用 IP アドレスがアカウントに関連付けられることを確認する返信が付きま

## 専用 IP アドレスの解放

アカウントに関連付けられた専用 IP アドレスが不要になった場合は、以下のステップを実行してアドレスを解放することができます。

### Important

専用 IP アドレスの解放プロセスは、元に戻すことはできません。月の途中で専用 IP アドレスを開放した場合、専用 IP の月額使用料を、現在の月の経過日数に基づいて按分します。

専用 IP アドレスを解放するには

1. AWS Management Console (<https://console.aws.amazon.com/>) にサインインします。
2. サポート メニューで [サポートセンター] を選択します。
3. My support cases (サポートケース) タブで、Create case (ケースの作成) を選択します。
4. [Create case (ケースの作成)] で [Service limit increase (サービスの上限緩和)] を選択します。
5. [Case classification (ケースの分類)] で、以下のセクションに入力します。
  - Limit Type (制限のタイプ) で SES Sending Limits (SES サービスの制限) を選択します。
  - [Mail Type (メールの種類)] で、任意の値を選択します。
  - [My email sending complies with the AWS Service Terms and AUP] で、ユースケースに当てはまるオプションを選択します。
  - [I only send to recipients who have specifically requested my mail (私は明確にリクエストされた受信者のみメールを送信します)] で、ユースケースに当てはまるオプションを選択します。
  - [I have a process to handle bounces and complaints (バウンスや苦情を処理するプロセスがあります)] で、ユースケースに当てはまるオプションを選択します。
6. [Requests (リクエスト)] で、以下のセクションに入力します。
  - [Region (リージョン)] で、リクエストが適用される AWS リージョンを選択します。

### Note

専用 IP アドレスは各 AWS リージョンに固有のものです。そのため、専用 IP アドレスが関連付けられる AWS リージョンを選択することが重要です。

- Limit (制限) で、Desired Maximum Send Rate (希望する最大送信レート) を選択します。
- [New limit value (新しい制限値)] で任意の数を入力します。ここで入力する番号は重要ではありません。次のステップで、放棄する専用 IP の数を指定します。

## Note

単一の専用 IP アドレスは、単一の AWS リージョンでのみ使用できます。他の AWS リージョンで使用した専用 IP アドレスを解放する場合は、[Add another request (別のリクエストを追加)] を選択します。次に、そのリージョンについて Region (リージョン)、[Limit (制限)]、New limit value (新しい制限値) の各フィールドに入力します。開放する専用 IP アドレスごとに、このプロセスを繰り返します。

7. Case Description (ケースの説明) の Use case description (ユースケースの説明) に、既存の専用 IP アドレスの解放を希望する旨を記載します。現在複数の専用 IP アドレスをリリースしている場合は、解放する専用 IP アドレスの数を含めます。
8. Contact options (問い合わせ先のオプション) の [Preferred contact language (優先される問い合わせ言語)] で、このケースに関する AWS からの問い合わせは [English (英語)] がいいか Japanese (日本語) がいいかを選択します。
9. 完了したら、[Submit] を選択します。

AWS がリクエストを受信すると、専用 IP アドレスを解放する希望を確認するメッセージが送信されます。IP アドレスを解放することが確認されると、アドレスがアカウントから削除されます。

## 専用 IP アドレスのウォームアップ

E メールサービスプロバイダは、メッセージを受け入れるか拒否するかを判断するとき、その送信元の IP アドレスの評価を考慮します。IP アドレスの評価に影響する要因の 1 つは、アドレスに高品質 Eメールの送信履歴があるかどうかです。E メールプロバイダーが、ほとんどまたはまったく履歴のない新しい IP アドレスからメールを受け入れる可能性はあまりありません。ほとんどまたはまったく履歴のない IP アドレスから送信された E メールは、受信者の迷惑メールフォルダに振り分けられるか、完全にブロックされる可能性があります。

新しい IP アドレスから Eメールの送信を開始するときは、そのアドレスを最大容量まで使用する前に、そのアドレスから送信する Eメールの量を徐々に増やしてください。このプロセスは IP アドレスのウォームアップと呼ばれます。

IP アドレスのウォームアップに必要な時間は、Eメールプロバイダーによって異なります。一部の Eメールプロバイダーでは 2 週間前後で良い評価を確立できますが、最大 6 週間かかるプロバイダもあります。新しい IP アドレスをウォームアップするときは、最もアクティブなユーザーに Eメールを送信して、苦情率が低いことを確認する必要があります。また、バウンスメッセージを慎重に調べて、IP ブロッキングまたはスロットリング通知を多数受け取っている場合は、Eメールの量を減らします。バウンスのモニタリングについては、「[Amazon SES 送信アクティビティのモニタリング \(p. 267\)](#)」を参照してください。

## 専用 IP アドレスの自動ウォームアップ

専用 IP アドレスをリクエストすると、Amazon SES はそれらを自動的にウォームアップして、送信する Eメールの配信を向上させます。IP アドレスの自動ウォームアップ機能はデフォルトで有効化されています。

自動ウォームアップの処理中に発生するステップは、既に専用 IP アドレスがあるかどうかによって異なります。

- 専用 IP アドレスを初めてリクエストすると、Amazon SES は、専用 IP アドレスと、その他の Amazon SES のお客様と共有されたアドレスとの間で Eメールの送信を分散します。Amazon SES は専用 IP アドレスから送信されるメッセージの数を時間と共に徐々に増やします。
- 専用 IP アドレスがすでに存在する場合、Amazon SES は既存の専用 IP (ウォームアップ済み) と新しい専用 IP (ウォームアップされていない) の間で Eメールの送信を分散します。Amazon SES は新しい専用 IP アドレスから送信されるメッセージの数を時間と共に徐々に増やします。



## Note

自動 IP ウォームアップは時間ベースのプロセスです。ウォームアップ率は、送信量とは無関係に 45 日間にわたって着実に増加します。

専用 IP アドレスのウォームアップ後は、良い評判を維持することを希望する E メールプロバイダーごとに、毎日 1,000 通前後の E メールを送信します。このタスクは、Amazon SES で使用する専用 IP アドレスごとに実行します。

ウォームアップ処理の完了直後に、大量の E メールを送信することは避けてください。その代わりに、ターゲットのボリュームに達するまで、送信する E メール数を徐々に増やします。E メールプロバイダーは、IP アドレスから送信される E メール数が急に大量に増えたことを確認すると、そのアドレスからのメッセージの配信をブロックまたはスロットリングする可能性があります。

## 自動ウォームアッププロセスを無効にします。

新しい専用 IP アドレスを購入すると、Amazon SES が自動的にアドレスをウォームアップします。専用の IP アドレスをご自身でウォームアップすることを希望する場合は、自動ウォームアップ機能を無効にすることができます。

### Important

自動ウォームアップ機能を無効にする場合、専用 IP アドレスのウォームアップはお客様の責任となります。ウォームアップされていないアドレスから E メールを送信する場合、配信レートが低下する可能性があります。

自動ウォームアップ機能を無効にするには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションバーで、[Dedicated IPs] を選択します。
3. [Automatic IP warm-up] の横にあるチェックボックスをオフにします。

## 自動ウォームアッププロセスの再開

専用 IP プールに属する一連の IP アドレスの自動 IP ウォームアッププロセスを再開することができます。

自動ウォームアッププロセスを再開するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションバーで、[Dedicated IPs] を選択します。
3. ウォームアッププロセスを再開したい専用 IP プールで、Actions を選択し、次に Restart IP warm up を選択します。

自動ウォームアッププロセスのステータスは、[Warm Up Status (ウォームアップステータス)] 列にあります。ウォームアッププロセスが終了すると、この列は Complete になります。

## 専用 IP プールの作成

Amazon SES で使用する複数の専用 IP アドレスを購入した場合、それらのアドレスのグループを作成できます。これらのグループは専用 IP プールと呼ばれます。一般的なシナリオは、マーケティング E メール送信用と取引 E メール送信用に専用 IP アドレスのプールをそれぞれ作成することです。取引 Eメールの送信者の評価はマーケティング Eメールの送信者の評価とは切り離されています。このシナリオでは、マーケティングキャンペーンで多数の苦情が発生した場合、取引 Eメールの配信には影響しません。

このセクションでは、専用 IP プールを作成する手順について説明します。

#### Note

すべての Amazon SES 顧客が共有する IP アドレスのプールを使用する設定セットを作成することもできます。共有 IP プールは、通常の送信動作と異なる E メールを送信する必要がある場合に便利です。設定セットで共有 IP プールを使用する方法については、「[専用 IP プールの管理 \(p. 264\)](#)」を参照してください。

Amazon SES コンソールを使用して専用 IP プールを作成する

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 画面左側のナビゲーションペインで、Email Sending の下の Dedicated IPs を選択します。
3. [Create a New IP Pool] を選択します。
4. IP Pool Name ページの Pool name で、専用 IP プールを表す名前を入力し、[Next] を選択します。
5. Add Dedicated IPs ページで、プールに追加する各 IP アドレスの横にあるチェックボックスをオンにして、Next を選択します。

#### Note

まだプールに割り当てていない専用 IP アドレスは、ses-default-dedicated-pool に含まれています。送信プールを指定しない設定セットを使用して E メールを送信する場合、または設定セットをまったく指定せずに E メールを送信する場合、Amazon SES は ses-default-dedicated-pool 内のいずれかのアドレスから E メールを送信します。

各専用 IP アドレスが属することができるのは、1 つのプールのみです。別のプールに関連付けられている専用 IP アドレスを選択した場合、Amazon SES はその設定を上書きし、作成するプールにそのアドレスを関連付けます。

6. [Assign to a configuration set] ページで、次のいずれかを実行します。
  - [Add this pool to an existing configuration set] を選択して、専用 IP プールを既存の設定セットに関連付けます。次に、[Existing configuration sets] で、IP プールを関連付ける設定セットを選択します。
  - [Create a new configuration set] を選択して設定セットを作成し、これに専用 IP プールを関連付けます。[Configuration set name] に、設定セットを表す名前を入力します。

終了したら、[Next] を選択します。

7. [Review] ページで、専用 IP プールの設定を確認します。IP プールを作成する準備ができたなら、[Create] を選択します。

## Amazon SES での自分の IP アドレスを使用した Eメールの送信

Amazon SES には、自分の IP アドレスの使用 (BYOIP = Bring Your Own IP) と呼ばれる機能があります。この機能を使用すると、Amazon SES で自分の IP アドレスを使用して E メールを送信できます。Eメールの送信にすでに IP アドレス範囲を使用している場合は、この IP アドレス範囲を Amazon SES での Eメール送信に使用することをリクエストできます。

BYOIP は、社内の Eメール送信システムを使用して IP の良い評価を確立済みであるが、Amazon SES に移行したいという場合などに役立ちます。BYOIP を使用すると、IP アドレスの評価を再確立することなく、Amazon SES を通じてすぐに Eメールの送信を開始できます。

### Requirements

BYOIP を使用するには、IP アドレス範囲が以下の要件を満たしている必要があります。

- アドレス範囲は、American Registry for Internet Numbers (ARIN)、Réseaux IP Européens Network Coordination Centre (RIPE NCC)、Asia-Pacific Network Information Centre (APNIC) などの地域インターネットレジストリ (RIR) に登録している必要があります。アドレス範囲は、企業または組織法人に登録する必要があります。個人に登録することはできません。
- 署名された認証メッセージを送信することで、自分がアドレス範囲を所有していることを証明できる必要があります。
- IP アドレス範囲内のアドレスには、消去履歴が含まれている必要があります。当社は、IP アドレス範囲の評価を調査し、評価が低いか悪意のある行動に関連する IP アドレスが含まれている場合、IP アドレス範囲を拒否する権利を留保できるものとします。

## Considerations

IP 範囲を Amazon SES に転送することを要求する前に、以下の点を考慮する必要があります。

- 指定できる最も具体的なアドレス範囲は /24 です。言い換えると、IP 範囲 203.0.113.0/24 を Amazon SES アカウントに転送した場合、203.0.113.0 から 203.0.113.255 までの合計 256 個のアドレスから送信できます。範囲全体を転送する必要があります。Amazon SES では現在、個々の IP アドレスを転送することはできません。
- 特定の IP アドレス範囲に BYOIP を使用する場合、この範囲にアクセスできる AWS リージョンは 1 つのみです。
- AWS アカウントには、リージョンあたり 5 つのアドレス範囲を登録できます。
- 自分の IP アドレスを使用する場合、共有 Amazon SES IP アドレスのプール内のアドレスは使用できません。これらの共有 IP アドレスを使用する必要がある場合は、別の AWS リージョンで Amazon SES を使用するか、新しい AWS アカウントを作成します。
- BYOIP で使用する IP アドレスごとに、月額料金が発生します。詳細については、「[Amazon SES の料金](#)」を参照してください。

## Amazon SES での自分の IP アドレスの使用

迷惑なコンテンツや悪意のあるコンテンツを送信するためにシステムが悪用されないように、当社は各 BYOIP リクエストを慎重に検討する必要があります。

Amazon SES で自分の IP 範囲を使用する場合は、以下の情報を [ses-byoip-request@amazon.com](mailto:ses-byoip-request@amazon.com) 宛に送信してください。

- AWS アカウント ID。
- IP 範囲を使用する AWS リージョン (ap-south-1 など)。
- ユースケースの説明。
- Amazon SES で使用する IP 範囲。
- 範囲が登録されているインターネットレジストリの名前。

リクエストをいただいてから 48 時間以内にご連絡いたします。お客様とのコミュニケーションにおいて、IP 範囲の所有権を証明するドキュメントなど、当社からお客様に追加情報をリクエストする場合があります。

## Amazon SES での E メール送信のテスト

Amazon SES のメールボックスシミュレーターを使用すると、さまざまな E メール送信シナリオに対応するアプリケーションの機能をテストできます。メールボックスシミュレーターは、架空の E メールアドレスを作成せずに E メール送信アプリケーションをテストしたり、1 日あたりの送信クォータに影響を与えずにシステムの最大スループットを確認したりする場合に役立ちます。

## 重要な考慮事項

Amazon SES メールボックスシミュレーターを使用する場合は、以下の特徴と制限に留意してください。

- メールボックスシミュレーターは、アカウントが Amazon SES サンドボックスにある場合でも使用できます。
- メールボックスシミュレーターに送信する E メールは、アカウントの最大送信レートのクォータを受けませんが、毎日の送信クォータには影響しません。たとえば、アカウントが 24 時間あたり 10,000 件のメッセージ送信を許可されていて、100 件のメッセージをメールボックスシミュレーターに送信した場合、依然として最大 10,000 件のメッセージを通常の受取人に送信でき、送信クォータに達することはありません。
- メールボックスシミュレーターに送信する E メールは、Eメールの配信性能や評価のメトリクスには影響しません。たとえば、大量のメッセージをメールボックスシミュレーターのバウンスアドレスに送信しても、バウンス率が高すぎることを警告するメッセージが[評価ダッシュボード \(p. 377\)](#)に表示されることはありません。
- 請求に関しては、Amazon SES メールボックスシミュレーターに送信する E メールは、Amazon SES を使用して送信する他の E メールと同じ扱いになります。つまり、メールボックスシミュレーターに送信したメッセージに対しても、通常の受取人に送信したメッセージと同額が請求されます。
- メールボックスシミュレーターでは、ラベリングがサポートされているため、メールボックスシミュレーターの同じアドレスに複数の方法で E メールを送信したり、アプリケーションで可変エンベロープリターンパス (VERP) を処理する方法を確認したりできます。たとえば、`bounce+label1@simulator.amazonses.com` と `bounce+label2@simulator.amazonses.com` に E メールを送信して、バウンスメッセージとバウンスを生じた E メールアドレスをアプリケーションが照合できるかどうかを確認できます。
- メールボックスシミュレーターを使用して同じ送信リクエストからの複数のバウンスをシミュレートする場合、Amazon SES は複数のバウンスのレスポンスを 1 つのレスポンスにまとめます。

## メールボックスシミュレーターの使用方法

メールシミュレーターを使用するには、以下の表にあるシナリオを見つけ、対応する E メールアドレスに E メールを送信します。

### Note

メールボックスシミュレーターのアドレスに E メールを送信するには、Amazon SES を介して、AWS CLI、AWS SDK、Amazon SES コンソール、Amazon SES SMTP インターフェイス、または Amazon SES API のいずれかを使用します。メールボックスシミュレーターは、外部ソースから送信された E メールには応答しません。

シミュレートシナリオ	E メールアドレス
正常な配信 - 受取人の E メールプロバイダーが E メールを受領します。「 <a href="#">通知を使用した Amazon SES E メール送信のモニタリング (p. 273)</a> 」に記載されているように配信通知をセットアップしている場合、Amazon SES は Amazon Simple Notification Service (Amazon SNS) を通じて配信通知を送信します。	<code>success@simulator.amazonses.com</code>
バウンス - 受取人の E メールプロバイダーは、SMTP 550 5.1.1 レスポンスコード (「不明なユーザー」) レスポンスコードで E メールを拒否します。Amazon SES は、バウンス通知を生成し、アカウントのセットアップ内容に応じて、この通	<code>bounce@simulator.amazonses.com</code>

シミュレートシナリオ	E メールアドレス
<p>知を E メールで送信するか、Amazon SNS トピックに通知を送信します。通常、ハードバウンスが発生した場合は E メールアドレスが Amazon SES サプレッションリストに追加されますが、メールボックスシミュレーターの E メールアドレスは追加されません。メールボックスシミュレーターから受け取るバウンス応答は、<a href="#">RFC 3464</a> に準拠しています。バウンスフィードバックの受け取り方法については、「<a href="#">通知を使用した Amazon SES E メール送信のモニタリング (p. 273)</a>」を参照してください。</p>	
<p>自動応答 – 受取人の E メールプロバイダーが E メールを受領し、受取人の受信トレイに配信します。E メールプロバイダーは、不在 (OOO) メッセージなどの自動応答を Eメールのリターンパスヘッダーのアドレスに送信します。リターンパスヘッダーが存在しない場合は、エンベロープ送信者 (「MAIL FROM」) アドレスに送信します。メールボックスシミュレーターから受け取る自動応答は、<a href="#">RFC 3834</a> に準拠しています。</p>	ooto@simulator.amazonses.com
<p>苦情 – 受取人の E メールプロバイダーが E メールを受領し、受取人の受信トレイに配信します。受取人は、これを未承諾のメッセージであると判断し、E メールクライアントで [Mark as Spam (スパムとしてマーク)] をクリックします。次に、アカウントのセットアップ内容に応じて、Amazon SES が E メールまたは Amazon SNS トピックへの通知により、送信者に苦情通知を転送します。メールボックスシミュレーターから受け取る苦情応答は、<a href="#">RFC 5965</a> に準拠しています。苦情フィードバックの受け取り方法については、「<a href="#">通知を使用した Amazon SES E メール送信のモニタリング (p. 273)</a>」を参照してください。</p>	complaint@simulator.amazonses.com
<p>サプレッションリストの受取人アドレス – 受取人のアドレスがグローバルサプレッションリストにある場合と同じように Amazon SES でハードバウンスが生成されます。</p>	suppressionlist@simulator.amazonses.com

## 拒否イベントのテスト

Amazon SES を介して送信するすべてのメッセージでウイルスがスキャンされます。ウイルスを含むメッセージを送信すると、Amazon SES はメッセージを受け入れ、ウイルスを検出して、そのメッセージ全体を拒否します。Amazon SES でメッセージが拒否されると、メッセージの処理が停止され、受取人のメールサーバーへのメッセージ配信は試行されません。次に、拒否イベントが生成されます。

Amazon SES メールボックスシミュレーターには、拒否イベントをテストするためのアドレスは含まれていません。ただし、拒否イベントは、EICAR テストファイルを使用してテストできます。このファイルは、ウイルス対策ソフトウェアを安全な方法でテストする業界標準の手段です。EICAR テストファイルを作成するには、以下のテキストをファイルに貼り付けます。

```
X50!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

そのファイルを `sample.txt` として保存し、E メールに添付して、そのメールを確認済みアドレスに送信します。E メールに他の問題がない場合、Amazon SES はメッセージを受け入れますが、実際のウイルスが含まれている場合と同じように拒否します。

#### Note

拒否された E メール(上記の手順を使用して送信したものを含む)は、1日あたりの送信クォータに対してカウントされます。送信するメッセージ(拒否されたメッセージを含む)ごとに料金が発生します。

EICAR テストファイルの詳細については、Wikipedia の「[EICAR テストファイル](#)」を参照してください。ファイルを添付したメッセージの送信に使用できるコード例については、「[AWS SDK を使用して raw E メールを送信する \(p. 451\)](#)」を参照してください。

## アカウントレベルのサブプレッショニリストの使用

Amazon SES には、現在の AWS リージョンの AWS アカウントに適用される、アカウントレベルのサブプレッショニリストが含まれています。このサブプレッショニリストを使用すると、以前にバウンスイベントや苦情イベントが発生したアドレスへの E メール送信を防止できます。アカウントレベルのサブプレッショニリストを設定する場合、アドレスがハードバウンスになったとき、苦情が発生したとき、またはその両方の場合に、アドレスをリストに追加するかどうかを指定します。Amazon SES API v2 を使用すると、アカウントレベルのサブプレッショニリストに対してアドレスを手動で追加または削除できます。

#### Note

住所を一括追加または削除するには、本番稼働用アクセスが必要です。サンドボックスの詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

Amazon SES には、グローバルサブプレッショニリストも含まれています。詳細については、「[Amazon SES グローバルサブプレッショニリストの使用 \(p. 199\)](#)」を参照してください。

## アカウントレベルのサブプレッショニリストに関する考慮事項

アカウントレベルのサブプレッショニリストを使用する場合は、次の要素を考慮する必要があります。

- 2019 年 11 月 25 日以降に Amazon SES の使用を開始した場合、アカウントはバウンスと苦情の両方に対してアカウントレベルのサブプレッショニリストをデフォルトで使用します。この日付より前に Amazon SES の使用を開始した場合は、Amazon SES API の `PutAccountSuppressionAttributes` オペレーションを使用してこの機能を有効にする必要があります。
- アカウントレベルのサブプレッショニリストに含まれているアドレスにメッセージを送信しようとする場合、Amazon SES はメッセージを受け付けませんが、送信はしません。
- Amazon SES は、アカウントレベルのサブプレッショニリストのアドレスに送信したメッセージを、アカウントのバウンスレートに対してカウントしません。
- Amazon SES は、アカウントレベルのサブプレッショニリストのアドレスに送信したメッセージを、毎日の送信クォータに対してカウントします。
- アカウントレベルのサブプレッショニリストの E メールアドレスは、Amazon SES API v2 の `DeleteSuppressedDestination` オペレーションを使用して削除するまで、リストに残ります。
- アカウントのメール送信機能が一時停止されている場合、Amazon SES はアカウントレベルのサブプレッショニリストのアドレスを 90 日後に自動的に削除します。この 90 日の期間が終了する前にアカウントのメール送信機能が復元された場合、アカウントレベルのサブプレッショニリストのアドレスは削除されません。
- Gmail では、Amazon SES に苦情データが提供されません。受取人が Gmail ウェブクライアントの [迷惑メール] ボタンを使用して、受信したメールを迷惑メールとして報告した場合、アカウントレベルのサブプレッショニリストには追加されません。

- アカウントが Amazon SES サンドボックスにある場合は、アカウントレベルのサブプレッジョンリストを有効にすることができます。ただし、アカウントがサンドボックスから削除されるまでは、[PutSuppressedDestination](#)、または [CreateImportJob](#) オペレーションを使用できません。サンドボックスの詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。
- アカウントレベルのサブプレッジョンリストを使用すると、Amazon SES はハードバウンスを引き起こすアドレスもグローバルサブプレッジョンリストに追加します。

## アカウントレベルのサブプレッジョンリストの有効化

Amazon SES API v2 の [PutAccountSuppressionAttributes](#) オペレーションを使用して、アカウントレベルのサブプレッジョンリストを有効にして設定します。AWS CLI を使用すると、この設定をすばやく簡単に設定できます。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用してアカウントレベルのサブプレッジョンリストを設定するには

- コマンドラインで以下のコマンドを入力します。

Linux, macOS, or Unix

```
aws sesv2 put-account-suppression-attributes \  
--suppressed-reasons BOUNCE COMPLAINT
```

Windows

```
aws sesv2 put-account-suppression-attributes \  
--suppressed-reasons BOUNCE COMPLAINT
```

アカウントレベルのサブプレッジョンリストを有効にするには、`suppressed-reasons` パラメータに少なくとも 1 つの理由を指定する必要があります。前の例に示すように、`BOUNCE` もしくは `COMPLAINT`、または両方を指定することもできます。

## 設定セットに応じたアカウントレベルのサブプレッジョンリストの有効化

特定の [設定セット \(p. 257\)](#) にのみ適用されるように、アカウントレベルのサブプレッジョンを設定することもできます。この操作を行うと、バウンスイベントまたは苦情イベントの原因となった E メールを送信したときに設定セットを指定した場合にのみ、アドレスがサブプレッジョンリストに追加されます。

### Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用して設定セット用にアカウントレベルのサブプレッジョンリストを設定するには

- コマンドラインで以下のコマンドを入力します。

Linux, macOS, or Unix

```
aws sesv2 put-configuration-set-suppression-options \  

```

```
--configuration-set-name configSet \  
--suppressed-reasons BOUNCE COMPLAINT
```

#### Windows

```
aws sesv2 put-configuration-set-suppression-options \  
--configuration-set-name configSet \  
--suppressed-reasons BOUNCE COMPLAINT
```

前の例では、*configSet* を、アカウントレベルのサブプレッジョンリストを使用する設定セットの名前に置き換えます。

## アカウントレベルのサブプレッジョンリストに個人の E メールアドレスを手動で追加する

Amazon SES API v2の [PutSuppressedDestination](#) オペレーションを使用して、アカウントレベルのサブプレッジョンリストに個別のアドレスを手動で追加できます。アカウントレベルのサブプレッジョンリストに追加できるアドレスの数の制限はありません。

#### Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用してアカウントレベルのサブプレッジョンリストにアドレスを手動で追加するには

- コマンドラインで以下のコマンドを入力します。

#### Linux, macOS, or Unix

```
aws sesv2 put-suppressed-destination \  
--email-address recipient@example.com \  
--reason BOUNCE
```

#### Windows

```
aws sesv2 put-suppressed-destination \  
--email-address recipient@example.com \  
--reason BOUNCE
```

前の例で、*recipient@example.com* を、アカウントレベルのサブプレッジョンリストに追加する E メールアドレスに置き換え、*BOUNCE* をサブプレッジョンリストにアドレスを追加する理由に置き換えます (許容値は BOUNCE と COMPLAINT)。

## アカウントレベルのサブプレッジョンリストに E メールアドレスを一括で追加

アドレスを手動で一括追加するには、最初に、連絡先リストを、アクセスする権限がある Amazon S3 オブジェクトにアップロードしてから、Amazon SES API v2の [CreateImportJob](#) オペレーションを使用します。



## Note

アカウントレベルのサブプレッジョンリストに追加できるアドレスの数に制限はありませんが、API コールごとの Amazon S3 オブジェクトでのアドレス一括追加には 1,000,000 という制限があります。

アカウントレベルのサブプレッジョンリストにメールアドレスを一括追加するには、次の手順を実行します。

- アドレス一覧を CSV 形式または JSON 形式の Amazon S3 オブジェクトにアップロードします。

住所を追加するための CSV 形式の例：

```
recipient1@example.com,BOUNCE
```

```
recipient2@example.com,COMPLAINT
```

newline-delimited JSON ファイルのみがサポートされています。この形式で、各行は個別のアドレス定義を含む完全な JSON オブジェクトです。

アドレスを追加するための JSON 形式の例：

```
{"emailAddress":"recipient1@example.com","reason":"BOUNCE"}
```

```
{"emailAddress":"recipient2@example.com","reason":"COMPLAINT"}
```

前述の例では、`recipient1@example.com` と `recipient2@example.com` をアカウントレベルのサブプレッジョンリストに追加する E メールアドレスに置き換えます。サブプレッジョンリストにアドレスを追加する理由として受け入れられるのは、`BOUNCE` および `COMPLAINT` です。

- `CreateImportJob` オペレーションを Amazon SES API v2 で使用します。

## Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用してアカウントレベルのサブプレッジョンリストにバルクアドレスを手動で追加するには

- コマンドラインで以下のコマンドを入力します。

Linux, macOS, or Unix

```
aws sesv2 create-import-job \  
--import-destination "{\"SuppressionListDestination\  
{\"SuppressionListImportAction\": \"PUT\"}}\" \  
--import-data-source "{\"S3Url\": \"s3://s3bucket/s3object\", \"DataFormat\": \"CSV  
\"}"
```

Windows

```
aws sesv2 create-import-job \  
--import-destination "{\"SuppressionListDestination\  
{\"SuppressionListImportAction\": \"PUT\"}}\" \  
--import-data-source "{\"S3Url\": \"s3://s3bucket/s3object\", \"DataFormat\": \"CSV  
\"}"
```

前述の例で、`s3bucket` と `s3object` を Amazon S3 バケット名と Amazon S3 オブジェクト名で置き換えます。

## アカウントレベルのサブプレッションリストにあるアドレスの一覧表示

Amazon SES API v2の `ListSuppressedDestinations` オペレーションを使用して、アカウントのアカウントレベルのサブプレッションリストにある E メールアドレスを一覧表示することができます。

### Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

アカウントレベルのサブプレッションリストにあるすべての E メールアドレス一覧表示するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 list-suppressed-destinations
```

上記のコマンドは、アカウントのアカウントレベルのサブプレッションリストにあるすべての E メールアドレスを返します。出力は以下の例のようになります。

```
{
  "SuppressedDestinationSummaries": [
    {
      "EmailAddress": "recipient2@example.com",
      "Reason": "COMPLAINT",
      "LastUpdateTime": 1586552585.077
    },
    {
      "EmailAddress": "recipient0@example.com",
      "Reason": "COMPLAINT",
      "LastUpdateTime": 1586552666.613
    },
    {
      "EmailAddress": "recipient1@example.com",
      "Reason": "BOUNCE",
      "LastUpdateTime": 1586556479.141
    }
  ]
}
```

`StartDate` オプションを使用すると、特定の日付の後にリストに追加された E メールアドレスのみを表示できます。

特定の日付の後にアカウントレベルのサブプレッションリストに追加されたアドレスを一覧表示するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 list-suppressed-destinations --start-date 1604394130
```

上記のコマンドで、`1604394130` を開始日の Unix タイムスタンプに置き換えます。

また、`EndDate` オプションを使用すると、特定の日付の前にリストに追加された E メールアドレスのみを表示することもできます。

特定の日付の前にアカウントレベルのサブプレッションリストに追加されたアドレスを一覧表示するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 list-suppressed-destinations --end-date 1611126000
```

上記のコマンドで、`1611126000` を終了日の Unix タイムスタンプに置き換えます。

Linux、macOS、または Unix コマンドラインで、組み込みの `grep` ユーティリティを使用して、特定のアドレスまたはドメインを検索することもできます。

アカウントレベルのサブプレッションリストで特定のアドレスを検索するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 list-suppressed-destinations | grep -A2 'example.com'
```

上記のコマンドで、`example.com` を、検索するテキストの文字列 (アドレスやドメインなど) に置き換えます。

## アカウントレベルのサブプレッションリストから E メールアドレスを削除する

アカウントのサブプレッションリストに、入っているべきではないアドレスがある場合、Amazon SES API v2 の `DeleteSuppressedDestination` オペレーションを使用して、手動で削除することができます。

### Note

次の手順では、AWS CLI がインストール済みであるものとしてします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用してアカウントレベルのサブプレッションリストからアドレスを削除するには

- コマンドラインで以下のコマンドを入力します。

Linux, macOS, or Unix

```
aws sesv2 delete-suppressed-destination \  
--email-address recipient@example.com
```

Windows

```
aws sesv2 delete-suppressed-destination \  
--email-address recipient@example.com
```

前の例で、`recipient@example.com` を、アカウントレベルのサブプレッジョンリストから削除する E メールアドレスに置き換えます。

## アカウントレベルのサブプレッジョンリストから E メールアドレスを一括削除する

アドレスを手動で一括削除するには、最初に、連絡先リストを、アクセスする権限がある Amazon S3 オブジェクトにアップロードしてから、Amazon SES API v2の `ListImportJobs` オペレーションを使用します。

### Note

アカウントレベルのサブプレッジョンリストから削除できるアドレスの数に制限はありませんが、API コールごとの Amazon S3 オブジェクトでのアドレス一括削除には 10,000 という制限があります。

アカウントレベルのサブプレッジョンリストからメールアドレスを一括で削除するには、次の手順を実行します。

- アドレス一覧を CSV 形式または JSON 形式の Amazon S3 オブジェクトにアップロードします。

アドレスを削除する CSV 形式の例：

```
recipient3@example.com
```

newline-delimited JSON ファイルのみがサポートされています。この形式で、各行は個別のアドレス定義を含む完全な JSON オブジェクトです。

アドレスを追加するための JSON 形式の例：

```
{"emailAddress": "recipient3@example.com"}
```

前述の例で、`recipient3@example.com` を、アカウントレベルのサブプレッジョンリストから削除する E メールアドレスに置き換えます。

- `CreateImportJob` オペレーションを Amazon SES API v2 で使用します。

### Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用してアカウントレベルのサブプレッジョンリストからバルクアドレスを手動で削除するには

- コマンドラインで以下のコマンドを入力します。

Linux, macOS, or Unix

```
aws sesv2 create-import-job \  
--import-destination "{\"SuppressionListDestination\": \  
  {\"SuppressionListImportAction\": \"DELETE\"}}\" \  
--import-data-source "{\"S3Url\": \"s3://s3bucket/s3object\", \"DataFormat\": \"CSV \  
\"}"
```

## Windows

```
aws sesv2 create-import-job `
--import-destination "{\"SuppressionListDestination\":
{\"SuppressionListImportAction\": \"DELETE\"}}" `
--import-data-source "{\"S3Url\": \"s3://s3bucket/s3object\", \"DataFormat\": \"CSV
\"}\""
```

前述の例で、`s3bucket` と `s3object` を Amazon S3 バケット名と Amazon S3 オブジェクト名で置き換えます。

## アカウントのインポートジョブのリストの表示

Amazon SES API v2の `ListImportJobs` オペレーションを使用して、アカウントのアカウントレベルのサブレッションリストにある E メールアドレスを一覧表示することができます。

### Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

アカウントのすべてのインポートジョブのリストを表示するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 list-import-jobs
```

上記のコマンドは、アカウントのすべてのインポートジョブを返します。出力は以下の例のようになります。

```
{
  "ImportJobs": [
    {
      "CreatedTimestamp": 1596175615.804,
      "ImportDestination": {
        "SuppressionListDestination": {
          "SuppressionListImportAction": "PUT"
        }
      },
      "JobStatus": "COMPLETED",
      "JobId": "755380d7-fbdb-4ed2-a9a3-06866220f5b5"
    },
    {
      "CreatedTimestamp": 1596134732.398,
      "ImportDestination": {
        "SuppressionListDestination": {
          "SuppressionListImportAction": "DELETE"
        }
      },
      "JobStatus": "COMPLETED",
      "JobId": "076683bd-a7ee-4a40-9754-4ad1161ba8b6"
    },
    {
      "CreatedTimestamp": 1596645918.134,
      "ImportDestination": {
        "SuppressionListDestination": {
```

```
        "SuppressionListImportAction": "PUT"
      }
    },
    "JobStatus": "COMPLETED",
    "JobId": "6e261869-bd30-4b33-b1f2-9e035a83a395"
  }
]
}
```

## アカウントのインポートジョブに関する情報を取得する

Amazon SES API v2の [GetImportJob](#) オペレーションを使用して、アカウントのインポートジョブに関する情報を取得できます。

### Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

アカウントのインポートジョブに関する情報を取得するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 get-import-job --job-id JobId
```

前述のコマンドは、アカウントのインポートジョブに関する情報を返します。出力は以下の例のようになります。

```
{
  "ImportDataSource": {
    "S3Url": "s3://bucket/object",
    "DataFormat": "CSV"
  },
  "ProcessedRecordsCount": 2,
  "FailureInfo": {
    "FailedRecordsS3Url": "s3presignedurl"
  },
  "JobStatus": "COMPLETED",
  "JobId": "jobid",
  "CreatedTimestamp": 1597251915.243,
  "FailedRecordsCount": 1,
  "ImportDestination": {
    "SuppressionListDestination": {
      "SuppressionListImportAction": "PUT"
    }
  },
  "CompletedTimestamp": 1597252002.583
}
```

## アカウントレベルのサプレッションリストの無効化

Amazon SES API v2 の [PutAccountSuppressionAttributes](#) オペレーションを使用して、`suppressed-reasons` 属性から値を削除することでアカウントレベルのサプレッションリストを効果的に無効にすることができます。

## Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用してアカウントレベルのサブプレッションリストを無効にするには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 put-account-suppression-attributes --suppressed-reasons
```

## Amazon SES グローバルサブプレッションリストの使用

Amazon SES には、グローバルサブプレッションリストが含まれています。Amazon SES のユーザーから送信された E メールがハードバウンスを起こすと、Amazon SES はバウンスを起こした E メールアドレスをグローバルサブプレッションリストに追加します。グローバルサブプレッションリストは、すべての Amazon SES のお客様に適用されるという意味で、グローバルです。つまり、別のカスタマーがグローバルサブプレッションリストに登録されているアドレスに E メールを送信しようとした場合、Amazon SES はメッセージを受け付けますが、送信しません。これは、E メールアドレスが抑制されているためです。グローバルサブプレッションリストの利点は、デフォルトですべての Amazon SES アカウントに適用されることです。つまり、使用するために追加の構成を実行する必要がありません。欠点は、グローバルサブプレッションリストには他の Amazon SES ユーザーのアカウントに関連付けられている E メールアドレスが含まれているため、クエリを実行できないことです。また、グローバルサブプレッションリストに手動でアドレスを追加することはできません。できるのは、Amazon SES コンソールを使用してグローバルサブプレッションリストからアドレスを削除することのみです。

Amazon SES には、アカウントレベルのサブプレッションリストも含まれています。詳細については、「[アカウントレベルのサブプレッションリストの使用 \(p. 190\)](#)」を参照してください。

## グローバルサブプレッションリストの考慮事項

グローバルサブプレッションリストを使用する場合は、次の要素を考慮する必要があります。

- グローバルサブプレッションリストは、すべての Amazon SES アカウントに対してデフォルトで有効になります。無効にすることはできません。
- Amazon SES はグローバルサブプレッションリストをすべてのユーザーに適用するため、グローバルサブプレッションリストにクエリを実行したり、このリストに手動でアドレスを追加したりすることはできません。
- E メールアドレスがハードバウンスを生成すると、Amazon SES はそのアドレスを短時間のグローバルサブプレッションリストに追加します。この期間が経過すると、Amazon SES はリストからアドレスを削除します。アドレスが別のハードバウンスを生成した場合、Amazon SES はそれをより長い期間グローバルサブプレッションリストに追加し、その期間終了時に削除します。アドレスがグローバルサブプレッションリストに残っている時間は、アドレスがハードバウンスを生成するたびに長くなります。アドレスは、グローバルサブプレッションリストに最大 14 日間残ります。
- グローバルサブプレッションリストに含まれているアドレスにメッセージを送信しようとする、Amazon SES はメッセージを受け付けますが、送信はしません。Amazon SES はバウンス通知を生成します。通知には bounceType の Permanent 値および bounceSubType の Suppressed 値が含まれています。E メールアドレスがグローバルサブプレッションリストに含まれているかどうかを確認するには、このタイプのバウンス通知を受信する方法しかありません。グローバルサブプレッションリストをクエリすることはできません。

- Amazon SES は、グローバルサブプレッションリストのアドレスに送信したメッセージをアカウントのパウンスレートにカウントします。
- Amazon SES は、グローバルサブプレッションリストのアドレスに送信したメッセージを、毎日の送信クォータにカウントします。
- ハードバウンスを起こしたすべての E メールアドレスと同様、サブプレッションリストのパウンスを起こしたアドレスは、それが有効なアドレスであることが確実である場合を除き、メーリングリストから削除してください。サブプレッションリストのパウンスは、アカウントのパウンス率に反映されます。パウンス率が高すぎる場合、当社はお客様のアカウントを確認し、アカウントの E メール送信機能を一時停止することがあります。

## グローバルサブプレッションリストからのアドレスの削除

グローバルサブプレッションリスト内のアドレスが実際に有効な受取人であることが確実である場合は、次の手順を使用してそのアドレスを削除できます。1つのリージョンのグローバルサブプレッションリストからアドレスを削除すると、削除はすべてのリージョンのすべての AWS アカウントに適用されます。

グローバルサブプレッションリストから E メールアドレスを削除するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ウィンドウの右上隅にあるリージョンセレクトタを使用して、米国東部 (バージニア北部)、米国西部 (オレゴン)、またはヨーロッパ (アイルランド) のいずれかの AWS リージョンを選択します。サブプレッションリストがすべてのリージョンに等しく適用されるため、どのリージョンを選択するかは重要ではありません。

### Note

この手順を完了するには、前の段落に示した AWS リージョンの 1 つを使用する必要があります。この手順は、他の AWS リージョンでは実行できません。

3. ナビゲーションペインで、[Suppression List Removal (サブプレッションリストの削除)] を選択します。
4. [Email Address] フィールドに、グローバルサブプレッションリストから削除する E メールアドレスを入力します。
5. [Type characters] に、上のイメージに示された文字を入力します。
6. [Submit] (送信) をクリックします。

指定した E メールアドレスがグローバルサブプレッションリストに登録されている場合は、すぐに削除されます。

## リスト管理の使用

Amazon SES にはリスト管理機能があります。つまり、カスタマーは連絡先リストと呼ばれる独自のメーリングリストを管理できます。連絡先リストは、特定の 1 つ以上のトピックを購読したすべての連絡先を保存できるリストです。連絡先は、E メールを受信しているエンドユーザーです。トピックは、リスト内の関心グループ、テーマ、またはラベルです。リストには、複数のトピックを含めることができます。

Amazon SES API v2 の [ListContacts](#) オペレーションを使用して、特定のトピックを購読している連絡先のリストを取得し、[SendEmail](#) オペレーションを使用して、連絡先に E メールを送信することができます。

Amazon SES API v2 を使用すると、アカウントレベルのサブプレッションリストに対してアドレスを手動で追加または削除できます。



## Note

住所を一括追加または削除するには、本番稼働用アクセスが必要です。サンドボックスの詳細については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

サブスクリプション管理の詳細については、「[サブスクリプションの使用 \(p. 206\)](#)」を参照してください。

## リスト管理の概要

リスト管理を使用する場合は、次の要素を考慮する必要があります。

- リストの作成中にリストトピックを指定できます。
- AWS アカウントごとに許可される連絡先リストは 1 つだけです。
- リストには、最大 20 個のトピックを設定できます。
- 既存の連絡先リストを更新できます。これには、リストへの新しいトピックの追加、リストからの連絡先の追加と削除、リストまたはトピックの連絡先設定の更新が含まれます。
- トピックの表示名や説明などのトピックメタデータを更新できます。
- 連絡先リストの連絡先、トピックを購読した連絡先、1 つのトピックの購読を取り消した連絡先、およびリスト内のすべてのトピックの購読を取り消した連絡先のリストを取得できます。
- 既存の連絡先リストを Amazon SES にインポートするには、[CreateImportJob API](#)を使用します。
- Amazon SES は、お客様の連絡先リストに登録されていない連絡先にメールを送信した場合、E メールをバウンスします。詳細については、「[サブスクリプションの使用 \(p. 206\)](#)」を参照してください。
- 各連絡先には、その連絡先に関する情報を格納するための属性を関連付けることができます。

## リスト管理の設定

次の操作を使用して、リスト管理機能を設定できます。連絡先リストおよび連絡先操作の完全なリストは、[Amazon SES API v2 リファレンス](#)を参照してください。

### 連絡先リストを作成する

Amazon SES API v2 の [CreateContactList](#) オペレーションを使用して、連絡先リストを作成できます。AWS CLI を使用すると、この設定をすばやく簡単に設定できます。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用してコンタクトリストを作成するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 create-contact-list --cli-input-json file://CONTACT-LIST-JSON
```

上記のコマンドで、**CONTACT-LIST-JSON** を [CreateContactList](#) リクエストの JSON ファイルへのパスに置き換えます。

リクエスト用の [CreateContactList](#) 入力 JSON ファイルの例を次に示します。

```
{
  "ContactListName": "ExampleContactListName",
  "Description": "Creating a contact list example",
  "Topics": [
    {
```

```
    "TopicName": "Sports",
    "DisplayName": "Sports Newsletter",
    "Description": "Sign up for our free newsletter to receive updates on all
sports.",
    "DefaultSubscriptionStatus": "OPT_OUT"
  },
  {
    "TopicName": "Cycling",
    "DisplayName": "Cycling newsletter",
    "Description": "Never miss a cycling update by subscribing to our
newsletter.",
    "DefaultSubscriptionStatus": "OPT_IN"
  },
  {
    "TopicName": "NewProducts",
    "DisplayName": "New products",
    "Description": "Hear about new products by subscribing to this mailing list.",
    "DefaultSubscriptionStatus": "OPT_IN"
  },
  {
    "TopicName": "DailyUpdates",
    "DisplayName": "Daily updates",
    "Description": "Start your day with sport updates, Monday through Friday.",
    "DefaultSubscriptionStatus": "OPT_OUT"
  }
]
}
```

## 連絡先を作成

Amazon SES API v2 の [CreateContact](#) オペレーションを使用して、連絡先を作成できます。AWS CLI を使用すると、この設定をすばやく簡単に設定できます。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用して接続を作成するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 create-contact --cli-input-json file://CONTACT-JSON
```

上記のコマンドで、**CONTACT-JSON** を [CreateContact](#) リクエストの JSON ファイルへのパスに置き換えます。

リクエストの [CreateContact](#) input JSON ファイルの例を次に示します。

```
{
  "ContactListName": "ExampleContactListName",
  "EmailAddress": "example@amazon.com",
  "UnsubscribeAll": false,
  "TopicPreferences": [
    {
      "TopicName": "Sports",
      "SubscriptionStatus": "OPT_IN"
    }
  ],
  "AttributesData": "{\"Name\": \"John\", \"Location\": \"Seattle\"}"
}
```

上の例で、UnsubscribeAll 値が false の場合、連絡先がすべてのトピックの購読を解除していないことを示し、true の場合、連絡先がすべてのトピックの購読を解除したことを意味します。

TopicPreferences には、連絡先のトピック購読ステータスに関する情報が含まれています。前述の例では、連絡先は「スポーツ」トピックにオプトインしており、「スポーツ」トピックへのすべての E メールを受信します。

AttributesData は JSON フィールドで、連絡先に関するメタデータを格納できます。有効な JSON オブジェクトである必要があります。

## 連絡先リストへの連絡先の一括インポート

アドレスを手動で一括追加するには、最初に、連絡先を、アクセスする権限がある Amazon S3 オブジェクトにアップロードしてから、Amazon SES API v2 の [CreateImportJob](#) オペレーションを使用します。

連絡先をインポートする前に、連絡先リストを作成する必要があります。

### Note

ImportJob ごとに、最大 100 万件の連絡先を連絡先リストに追加できます。

連絡先リストに連絡先を一括で追加するには、次の手順を実行します。

- 連絡先を CSV 形式または JSON 形式で Amazon S3 オブジェクトにアップロードします。

### CSV 形式

Amazon S3 にアップロードされるファイルの最初の行は、ヘッダー行である必要があります。

TopicPreferences オブジェクトは、CSV 形式用にフラット化する必要があります。TopicPreferences の各トピックには、個別のヘッダーフィールドがあります。

連絡先リストに一括で連絡先を追加するための CSV 形式の例：

```
emailAddress,unsubscribeAll,attributesData,topicPreferences.Sports,topicPreferences.Cycling
example1@amazon.com,false,{"Name": "John"},OPT_IN,OPT_OUT
example2@amazon.com,true,,OPT_OUT,OPT_OUT
```

### JSON 形式

newline-delimited JSON ファイルのみがサポートされています。この形式で、各行は 1 つの連絡先情報を含む完全な JSON オブジェクトです。

連絡先リストに一括で連絡先を追加するための JSON 形式の例：

```
{
  "emailAddress": "example1@amazon.com",
  "unsubscribeAll": false,
  "attributesData": "{\"Name\": \"John\"}",
  "topicPreferences": [
    {
      "topicName": "Sports",
      "subscriptionStatus": "OPT_IN"
    },
    {
      "topicName": "Cycling",
      "subscriptionStatus": "OPT_OUT"
    }
  ]
}
```

```

}
{
  "emailAddress": "example2@amazon.com",
  "unsubscribeAll": true,
  "topicPreferences": [
    {
      "topicName": "Sports",
      "subscriptionStatus": "OPT_OUT"
    },
    {
      "topicName": "Cycling",
      "subscriptionStatus": "OPT_OUT"
    }
  ]
}

```

前述の例では `example1@amazon.com` および `example2@amazon.com` を、連絡先リストに追加する E メールアドレスに置き換えます。attributesData の値を、連絡先の特定の値に置き換えます。また、`###` およびサイクリングを、連絡先に適用する topicName に置き換えます。使用可能な topicPreferences は `OPT_IN` および `OPT_OUT` です。

連絡先を CSV 形式または JSON 形式で Amazon S3 オブジェクトにアップロードする場合、次の属性がサポートされます。

属性	説明
emailAddress	連絡先の E メールアドレスです。これは必須フィールドです。
unsubscribeAll	連絡先がすべての連絡先リストのトピックの購読を解除しているかどうかを通知するブール値のステータス。
topicPreferences	トピックへのオプトインまたはオプトアウトに関する連絡先の設定。
attributesData	連絡先に添付された属性データ。

- `CreateImportJob` オペレーションを Amazon SES API v2 で使用します。

#### Note

次の手順では、AWS CLI がインストール済みであるものとします。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用して、連絡先リストに手動で連絡先を一括追加するには

- コマンドラインで以下のコマンドを入力します。

Linux, macOS, or Unix

```

aws sesv2 create-import-job \
--import-destination "{\"ContactListDestination\": {\"ContactListName\": \"ExampleContactListName\", \"ContactListImportAction\": \"PUT\"}}\" \
--import-data-source "{\"S3Url\": \"s3://s3bucket/s3object\", \"DataFormat\": \"CSV\"}"

```

## Windows

```
aws sesv2 create-import-job `
--import-destination "{\"ContactListDestination\": {\"ContactListName\":
\\\"ExampleContactListName\\\", \\\"ContactListImportAction\": \\\"PUT\\\"}}" `
--import-data-source "{\"S3Url\": \"s3://s3bucket/s3object\\\", \"DataFormat\": \\\"CSV
\\\"}\""
```

前述の例で、*s3bucket* と *s3object* を Amazon S3 バケット名と Amazon S3 オブジェクト名で置き換えます。

## 連絡先をリストしてメールを送信する

[ListContacts](#) オペレーションを使用して、特定のトピックを購読している連絡先リストを取得すると共に、[SendEmail](#) オペレーションで、リストメンバーに E メールを送信できます。

AWS CLI を使用して連絡先を一覧表示するには

1. コマンドラインで以下のコマンドを入力します。

```
aws sesv2 list-contacts --cli-input-json file://LIST-CONTACTS-JSON
```

上記のコマンドで、*LIST-CONTACTS-JSON* を [ListContacts](#) リクエストの JSON ファイルへのパスに置き換えます。

リクエストのための ListContacts input JSON ファイルの例を次に示します。

```
{
  "ContactListName": "ExampleContactListName",
  "Filter": {
    "FilteredStatus": "OPT_IN",
    "TopicFilter": {
      "TopicName": "Cycling",
      "UseDefaultIfPreferenceUnavailable": true
    }
  },
  "PageSize": 50
}
```

FilteredStatus には、フィルタリングする購読ステータス (OPT\_IN または OPT\_OUT) が表示されます。

TopicFilter は、結果が欲しいトピックを指定するオプションのフィルターです。上記の例では「Cycling」です。

UseDefaultIfPreferenceUnavailable の値には true または false があります。true の場合、連絡先にトピックに対する明示的な設定がなければ、トピックのデフォルト設定が使用されます。false の場合、明示的に定めた設定を持つ連絡先のみがフィルタリングの対象となります。

2. 上記の [ListContacts](#) オペレーションを使用して、連絡先をリスト化した後、[SendEmail](#) オペレーションを使用して、各連絡先に E メールを送信します。[SendEmail](#) オペレーションを使用している場合、ListManagementOptions を指定して、Amazon SES が E メールに配信停止リンクを追加できるようにします。詳細については、「[サブスクリプションの使用 \(p. 206\)](#)」を参照してください。

連絡先リストにない受信者の E メールアドレスへの [SendEmail](#) リクエストに ListManagementOptions を含めると、連絡先がリストに自動的に作成されます。

連絡先リストにある登録解除済み連絡先にメールを送信した場合、Amazon SES は E メールをバウンズします。つまり、登録解除した連絡先に送信されないようにするために SendEmail リクエストを更新する必要はありません。

SendEmail オペレーションを使用するには、次のように、E メールが属する contactListName と topicName を含めます。TopicName はオプションです。

```
ListManagementOptions:  
  String contactListName  
  String topicName
```

3. または、SMTP インターフェイスで E メールを送信しながら、X-SES-LIST-MANAGEMENT-OPTIONS ヘッダーを使用してリストとトピック名を指定します。

SMTP インターフェイスを使用して E メールを送信するときにリストとトピック名を指定するには、次の E メールヘッダーをメッセージに追加します。

```
X-SES-LIST-MANAGEMENT-OPTIONS: {contactListName}; topic={topicName}
```

## サブスクリプションの使用

Amazon SES には購読管理機能があります。この機能では、SendEmail オペレーションリクエストの ListManagementOptions 内で contactListName と topicName を指定すると、Amazon SES はすべての送信 E メールで購読取り消しリンクを自動的に有効にします

連絡先が特定のトピックまたはリストから登録を解除した場合、Amazon SES はそのトピックまたはリストに関連して連絡先に E メールを送信することを許可しません。

### Note

[Amazon SES の Easy DKIM \(p. 135\)](#) を使用しているユーザーは購読管理を利用できませんが、Amazon SES は、Amazon SES を呼び出す前に E メールに署名している送信者に対して、購読取り消しリンクを E メールに追加することはできません。

リスト管理の詳細については、「[リスト管理の使用 \(p. 200\)](#)」を参照してください。

## 購読管理の概要

購読管理を使用する場合は、次の要素を考慮する必要があります。

- 購読管理は Amazon SES によってすべて管理されます。つまり、Amazon SES は、購読取り消しのウェブページから購読取り消しのメールとリクエストを受信し、リストの連絡先の設定を更新します。設定セット通知を使用して、購読取り消し通知を受信できます。設定セットの詳細については、「[Amazon SES 設定セットの使用 \(p. 257\)](#)」を参照してください。
- E メール送信時に連絡先リストを指定する必要があります。Link-Unsubscribe ヘッダーとフッターのリンクによる購読管理は、それに応じて処理されます。
- Amazon SES では、List-Unsubscribe ヘッダー標準がサポートされます。これにより、E メールクライアントおよび受信トレイプロバイダーは、Eメールの上部に購読取り消しリンクを表示できるようになります。
- List-Unsubscribe ヘッダーは、次の動作に従います。

連絡先リストとトピックの両方が指定されている Eメールの [List-Unsubscribe] ヘッダーリンクをクリックすると、連絡先は特定トピックからのみの購読が解除されます。

トピックが指定されていない場合、連絡先はリスト内のすべてのトピックから配信停止になります。

- 連絡先が、メールフッター内の購読取り消しリンクをクリックすると、購読取り消しのランディングページに移動します。
- 購読取り消しのランディングページでは、連絡先の設定を更新するオプションが表示されます。特定のリスト内のすべてのトピックについて `OPT_IN` または `OPT_OUT` を設定します。ランディングページには、リスト内のすべてのトピックの購読を解除するオプションもあります。
- Amazon SES で購読取り消し URL を挿入すべき場所を示すプレースホルダを E メールに含める必要があります。プレースホルダは、最大 2 倍含めることができます。2 回以上使用した場合、最初の 2 つのオカレンスのみが置き換えられます。
- [List-Unsubscribe] ヘッダーとフッターのリンクは、E メールが 1 人の受信者に送信されている場合にのみ追加され、購読取り消しリンクを追加するにはプレースホルダが必要です。
- 取引先担当者が登録を解除できないようにするメールの場合、[SendEmail](#) リクエストで [ListManagementOptions] フィールドを排除できます。

## 購読取り消しフッターリンクの追加

テンプレートおよびテンプレート無しの E メールで `{{amazonSESUnsubscribeUrl}}` プレースホルダを使用して、Amazon SES が購読取り消し URL を挿入する必要がある場所を指定します。

プレースホルダの置き換えは、HTML および TEXT コンテンツタイプでのみサポートされます。

プレースホルダは、最大 2 倍含めることができます。2 回以上使用した場合、最初の 2 つのオカレンスのみが置き換えられます。

## 購読取り消しヘッダーの考慮事項

Amazon SES が追加する次の 2 つのヘッダーによって、購読取り消しヘッダーによる購読管理が有効になります。これらのヘッダーを Amazon SES に送信する前に自分の E メールに追加する場合、ListManagementOptions が有効になっていれば、Amazon SES はヘッダーを上書きします。

List-Unsubscribe

List-Unsubscribe-Post

購読取り消しヘッダーによるリンクをクリックして購読を解除した連絡先は、E メールクライアントまたは受信トレイのプロバイダーによって異なる体験をすることになります。ヘッダーのリンクを介して購読を取り消したユーザーは、どのトピックから購読を解除するかを選択することはできません。また、E メールが送信されていたトピックから購読が解除されます。

メッセージヘッダーの詳細については、[RFC 2369](#) および [RFC 8058](#) を参照してください。

# Amazon SES およびセキュリティプロトコル

このトピックでは、Amazon SES に接続する際、そして Amazon SES が受信者に E メールを送信する際に使用できるセキュリティプロトコルについて説明します。

## E メール送信者から Amazon SES へ

Amazon SES に接続するために使われるセキュリティプロトコルは、以下に説明されるように、Amazon SES API あるいは Amazon SES SMTP インターフェイスの使用によって異なります。

## HTTPS

Amazon SES API を使用している場合は (直接または AWS SDK 経由で)、すべての通信は Amazon SES の HTTPS エンドポイントによる TLS で暗号化されます。Amazon SES の HTTPS エンドポイントは、TLS 1.2、TLS 1.1、TLS 1.0 のみをサポートします。

## SMTP インターフェイス

SMTP インターフェイスを介して Amazon SES にアクセスする場合、Transport Layer Security (TLS) を使って接続を暗号化する必要があります。TLS は、以前のプロトコルの名前である「Secure Sockets Layer (SSL)」と呼ばれることが多いことに注意してください。

Amazon SES は、TLS で暗号化された接続を確立するために、STARTTLS および TLS ラッパーという 2 つのメカニズムをサポートしています。

- STARTTLS – STARTTLS とは、暗号化されていない接続を暗号化された接続にアップグレードする方法です。STARTTLS には、様々なプロトコルに対応したバージョンがあります。SMTP バージョンは、「RFC 3207」に定義されています。STARTTLS 接続の場合、Amazon SES は TLS 1.2、TLS 1.1、TLS 1.0、および SSLv2Hello をサポートします。
- TLS Wrapper – TLS Wrapper (SMTPS またはハンドシェイクプロトコルとも呼ばれる) は、最初に暗号化されていない接続を確立するのではなく、最初から暗号化された接続を開始する方法です。TLS ラッパーを使用する場合、Amazon SES SMTP エンドポイントは TLS ネゴシエーションを実行しません。TLS を使用してエンドポイントに接続し、通信全体で TLS の使用を継続するのはクライアントの役割です。TLS ラッパーは古いプロトコルですが、数多くのクライアントが今もサポートしています。TLS ラッパー接続の場合、Amazon SES は TLS 1.2、TLS 1.1、および TLS 1.0 をサポートします。

これらの方法を使用した、Amazon SES SMTP インターフェイスへの接続について詳しくは、「[Amazon SES SMTP エンドポイントへの接続 \(p. 94\)](#)」を参照してください。

## Amazon SES から受信者へ

Amazon SES では、TLS 接続に TLS 1.2、TLS 1.1 および TLS 1.0 をサポートしています。

デフォルトで、Amazon SES は便宜的 TLS を使用します。つまり、Amazon SES は常に受信メールサーバーへの安全な接続を確立しようとします。安全な接続を確立できない場合、Amazon SES は平文メッセージを送信します。

この動作は、設定セットを使用することで変更できます。[PutConfigurationSetDeliveryOptions](#) API オペレーションを使用して、設定セットの `TlsPolicy` プロパティを `Require` に設定します。この変更には、[AWS CLI](#) を使用できます。

設定セットに TLS 接続を要求するよう Amazon SES を設定するには

- コマンドラインで以下のコマンドを入力します。

```
aws ses put-configuration-set-delivery-options --configuration-set-name MyConfigurationSet --delivery-options TlsPolicy=Require
```

前述の例では、*MyConfigurationSet* をお客様の設定セットの名前に置き換えます。

この設定セットを使用して E メールを送信する場合、Amazon SES は、安全な接続を確立できる受信 E メールサーバーに対してのみメッセージを送信します。受信 E メールサーバーへの安全な接続が確立できない場合、Amazon SES はメッセージを破棄します。



## エンドツーエンドの暗号化

Amazon SES を使用して、S/MIME または PGP を使用して暗号化されたメッセージを送信できます。これらのプロトコルを使用するメッセージは、送信者によって暗号化されます。これらのコンテンツは、メッセージを復号化するために必要なパブリックキーを有する受信者のみが表示できます。

Amazon SES では以下の MIME タイプがサポートされ、S/MIME で暗号化された E メールを送信するために使用できます。

- `application/pkcs7-mime`
- `application/pkcs7-signature`
- `application/x-pkcs7-mime`
- `application/x-pkcs7-signature`

Amazon SES では以下の MIME タイプ もサポートされており、PGP で暗号化された E メールを送信するために使用できます。

- `application/pgp-encrypted`
- `application/pgp-keys`
- `application/pgp-signature`

# Amazon SES を使用した E メールの受信

Amazon Simple Email Service (Amazon SES) は、メールを送受信できるメールサーバーです。Amazon SES を使用してメールを受信するときは、次のような基礎となるメール受信操作を Amazon SES が処理します。

- 他のメールサーバーとの通信
- スпамおよびウイルスのスキャン
- 信頼されないソースからのメールの拒否
- ドメイン内の受信者宛のメールの承認

E メールを受け取る場合、Amazon SES はそれを指示された手順に従って処理します。たとえば、Amazon SES は Amazon S3 バケットへの受信メールの配信、Amazon SNS トピックへの公開、または Amazon WorkMail への送信を実行できます。また、特定の IP アドレス範囲からのすべてのメッセージを明示的にブロックまたは許可するルールを作成することも、メッセージが特定の E メールアドレスに送信されたときに自動的にバウンスメッセージを送信するルールを作成することもできます。

## Note

Amazon SES には、E メールを受信するための POP サーバーや IMAP サーバーは含まれていません。つまり、Microsoft Outlook などの E メールクライアントを使用して E メールを受信することはできません。E メールクライアントを使用して送信と受信の両方が可能なソリューションが必要な場合は、[Amazon WorkMail](#) の使用を検討してください。

Amazon SES は、特定の AWS リージョンでのみ E メール受信をサポートしています。リージョンの完全なリストを使用した E メール受信については、「AWS 全般のリファレンス」の[Amazon Simple Email Service のエンドポイントとクォータ](#)を参照してください。

このセクションのトピック:

- [Amazon SES メール受信の概念 \(p. 210\)](#)
- [Amazon SES による E メール受信の開始方法 \(p. 212\)](#)
- [Amazon SES E メール受信の設定 \(p. 218\)](#)
- [Amazon SES による E メール受信の管理 \(p. 239\)](#)

## Amazon SES メール受信の概念

メール受信のために Amazon SES を使用するときは、メールにどのような処理を実行するのかをサービスに対して指示する必要があります。メールをきめ細かく制御する方法として主に使用されるのは、受取人に基づいて実行するアクションを指定する方法です。別の方法として、発信元 IP アドレスに基づいてメールをブロックまたは許可する方法があります。このトピックでは、両方の方法について説明します。

### 受信者ベースの制御

受信メールを制御する主な方法として、受取人に基づいてメールの処理方法を指定します。たとえば、example.com というドメインを所有している場合、user@example.com 宛てのメールはバウンスさ

せ、example.com およびそのサブドメイン宛でのそれ以外のすべてのメールを配信するように指定できます。このとき指定する受取人リストを条件と呼びます。

受信ルールをセットアップして、条件が満たされたときのメールの処理方法を指定します。受信ルールは、条件と、順序どおりに並べたアクションのリストで構成されます。受信メールの宛先となっている受信者が、条件に指定された受信者と一致する場合に、ルールに指定されたアクションを Amazon SES が実行します。以下のアクションを使用できます。

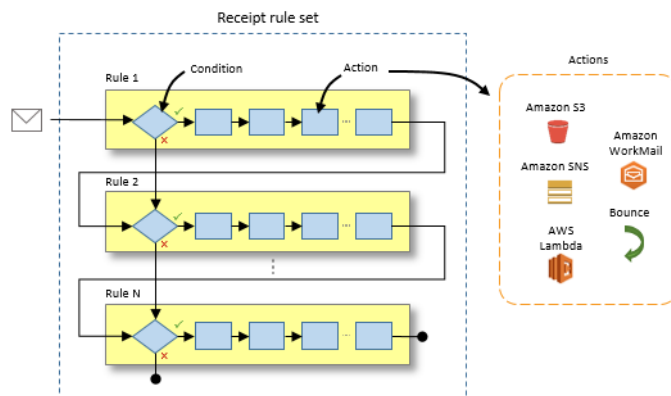
- S3 アクション - メールを Amazon S3 バケットに配信し、オプションで Amazon SNS を通じて通知します。
- SNS アクション - メールを Amazon SNS トピックに公開します。

#### Note

この SNS アクションでは、メールのコンテンツの完全なコピーが Amazon SNS 通知に含まれます。ここで言及されるそれ以外の Amazon SNS 通知は、単純にメールの配信を通知するだけであり、メールのコンテンツそのものではなくメールに関する情報が含まれます。

- Lambda アクション - Lambda 関数を通じてコードを呼び出し、オプションで Amazon SNS を通じて通知します。
- バウンスアクション - 送信者にバウンス応答を返すことによって E メールを拒否し、オプションで Amazon SNS を通じて通知します。
- 停止アクション - 受信ルールセットの評価を終了し、オプションで、Amazon SNS を通じて通知します。
- ヘッダー追加アクション - 受信 E メールにヘッダーを追加します。このアクションは通常、他のアクションとの組み合わせでのみ使用します。
- WorkMail アクション - メールを Amazon WorkMail で処理します。Amazon WorkMail がセットアップを処理するため、このアクションを直接使用することは通常はありません。

複数の受信ルールがグループ化されたものが受信ルールセットです。ご使用の AWS アカウントで複数の受信ルールを定義することはできますが、有効にできる受信ルールセットは常に 1 つだけです。次の図は、受信ルール、受信ルールセット、アクションが相互にどのように関連するのかを示しています。



## IP アドレスベースの制御

IP アドレスフィルタをセットアップすることで、より広範なレベルでメールの流れを制御できます。IP アドレスフィルタはオプションであり、特定の IP アドレスまたは IP アドレス範囲から送信されるメールを受け入れるか拒否するかを指定できます。IP アドレスフィルタには、ブロックリスト (そこから受信するメールをブロックする IP アドレス) と 許可リスト (常にメールを受け入れる IP アドレス) を含めることができます。IP アドレスフィルタは、スパムをブロックするのに便利です。Amazon SES は、スパムの送信元として知られる IP アドレスの独自のブロックリストを保持していますが、該当する IP アドレスを許可リストに追加することで、その IP アドレスからのメールを受信するよう選択できます。

## Note

Amazon EC2 の IP アドレスから送信されるメールを許可する場合は、そのアドレスを許可リストに追加する必要があります。Amazon EC2 から送信されるすべてのメールは、デフォルトではブロックされています。

## E メール受信プロセス

ご使用のドメイン宛での E メールを Amazon SES が受信するときは、以下のイベントが発生します。

1. Amazon SES は最初に送信者の IP アドレスを検索します。Amazon SES は、次の場合を除き、メールがこの段階を通過することを許可します。
  - IP アドレスがブロックリストに含まれている。
  - IP アドレスが Amazon SES のブロックリストに含まれており、指定した許可リストには含まれない。
2. Amazon SES は、設定された有効な受信ルールセットを調査して、受信する E メールの中のいずれかの受信者と一致する条件が受信ルールに含まれているかどうかを判定します。
3. 一致がない場合、Amazon SES は当該メールを拒否します。そうでない場合、Amazon SES はメールを受け入れます。
4. メールを受け入れる場合、Amazon SES は有効な受信ルールセットを評価します。アクションまたは受信ルールによって受信ルールセットの評価が明示的に終了されない限り、受取人の条件のうち少なくとも 1 つと一致するすべての受信ルールが、定義された順序で適用されます。

プロセスの概要を理解したところで、「[E メール受信のセットアップ \(p. 218\)](#)」に進んで利用を開始できます。

## Amazon SES による E メール受信の開始方法

このチュートリアルでは、AWS アカウントを作成し、Amazon Route 53 を使用してドメインを登録した後、ドメインに送信されるすべての E メールが Amazon Simple Storage Service バケットに配信されるように Amazon Simple Email Service を設定します。

## Note

Amazon SES は、特定の AWS リージョンでのみ E メール受信をサポートしています。リージョンの完全なリストを使用した Eメールの受信については、「[AWS 全般のリファレンス](#)」の [Amazon Simple Email Service のエンドポイントとクォータ](#) を参照してください。

## トピック

- [ステップ 1: 開始する前に \(p. 212\)](#)
- [ステップ 2: ドメインの検証 \(p. 213\)](#)
- [ステップ 3: 受信ルールの設定 \(p. 214\)](#)
- [ステップ 4: テスト Eメールの送信 \(p. 216\)](#)
- [ステップ 5: 受信した Eメールの表示 \(p. 217\)](#)
- [ステップ 6: クリーンアップ \(p. 217\)](#)

## ステップ 1: 開始する前に

このチュートリアルをスタートする前に、AWS アカウントにサインアップし (アカウントをお持ちでない場合)、[Amazon Route 53](#) を使用して Eメールの受信に使用するドメインを登録します。

## Note

Amazon SES は、特定の AWS リージョンでのみ E メール受信をサポートしています。リージョンの完全なリストを使用した Eメールの受信については、「AWS 全般のリファレンス」の [Amazon Simple Email Service のエンドポイントとクォータ](#) を参照してください。

## AWS にサインアップ

AWS アカウントが既にある場合、このセクションは省略できます。

AWS アカウントを作成するには

1. <https://console.aws.amazon.com/ses/> へ行き、[Amazon SES の使用開始] を選択します。
2. [Create an AWS Account] ページで、必須のフィールドに入力し、画面の手順に従って新しいアカウントを作成します。

## Route 53 を使用してドメインを登録する。

このチュートリアルは、Route 53 を使って登録したドメインを使用していることを前提としています。別のサービスを使用して登録したドメインを使用することもできますが、ドメインの検証手順はこのチュートリアルに示すものとは異なります。Route 53 を使用してドメインを登録する方法の詳細については、Amazon Route 53 デベロッパーガイドの「[新しいドメインを登録する](#)」を参照してください。

既存のドメインを Route 53 に移管することもできます。Route 53 へのドメインの移管の詳細については、Amazon Route 53 デベロッパーガイドの「[Route 53 で登録を移管する](#)」を参照してください。

次のステップ: [ステップ 2: ドメインの検証 \(p. 213\)](#)

## ステップ 2: ドメインの検証

ドメインの E メールを受信できるよう Amazon SES を設定する前に、そのドメインを所有していることを証明する必要があります。所有しているどのドメインでも検証できますが、Route 53 を使って登録したドメインを検証の方が簡単です。

## Note

アカウントがまだ Amazon SES サンドボックスにある場合は、このセクションの手順を完了する前に、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。

Amazon SES でドメインを検証するには

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

## Note

このセクションの手順を完了するには、Route 53 にドメインを登録したときに使用したのと同じ AWS アカウントを使用して AWS マネジメントコンソールにサインインします。

2. ナビゲーションペインで、[Identity Management] の [Domains] を選択します。
3. [Verify a New Domain] を選択します。
4. [Verify a New Domain (新しいドメインの検証)] ダイアログボックスの [Domain (ドメイン)] に、Route 53 を使用して登録したドメインの名前を入力し、[Verify This Domain (このドメインを検証)] を選択します。
5. [Verify a New Domain] ダイアログボックスで、[Use Route 53] を選択します。


#### Note

[Use Route 53 (Route 53 を使用)] ボタンが表示されない場合、ドメインが Route 53 に登録されていない可能性があります。別のサービスを使用してドメインを登録した場合、「[Amazon SES でのドメインの検証 \(p. 62\)](#)」の手順を完了することで、ドメインを検証できます。

6. [Use Route 53] ダイアログボックスで、[Domain Verification Record] と [Email Receiving Record] を選択します。次に、[Hosted Zones] で、使用するホストゾーンの名前を選択します。Route 53 を使用して登録したドメインに変更を加えていない場合、[Hosted Zones (ホストゾーン)] セクションにあるオプションは 1 つだけです。

#### Important

既にドメインの E メールエクスチェンジャ (MX) レコードを設定している場合は、次のステップでそれらのレコードが新しいレコードに置き換えられます。

7. [Create Record Sets] を選択します。ドメインのリストに戻ります。
8. 5 分待つてから、[refresh (更新)] () ボタンを選択します。[Status] 列の値が [verified] であることを確認します。ステータスが [pending verification] の場合、数分待つてからリストをもう一度更新します。ドメインのステータスが [verified] になるまでこのプロセスを繰り返します。

次のステップ: [ステップ 3: 受信ルールの設定 \(p. 214\)](#)

## ステップ 3: 受信ルールの設定

Amazon SES を E メールレシーバーとして使用するには、有効な受信ルールセットが必要です。受信ルールセットは、検証済みドメインで受信したメールを Amazon SES で処理する方法を定義する受信ルールのコレクションです。Amazon SES での E メール受信を初めて設定する場合は、デフォルトの受信ルールセットが Amazon SES で自動的に作成されます。このセクションで作成した受信ルールは、デフォルトの受信ルールセットに属します。

#### Note

このセクションの手順では、受信ルールセットを作成していることを前提としています。アカウントに受信ルールセットが既に含まれている場合、Amazon SES がそのルールセットをドメインの受信 E メールに適用する前に、このセクションで作成した受信ルールをアクティブにする必要があります。受信ルールセットの有効化と無効化の詳細については、「[受信ルールセットの有効化/無効化 \(p. 240\)](#)」を参照してください。

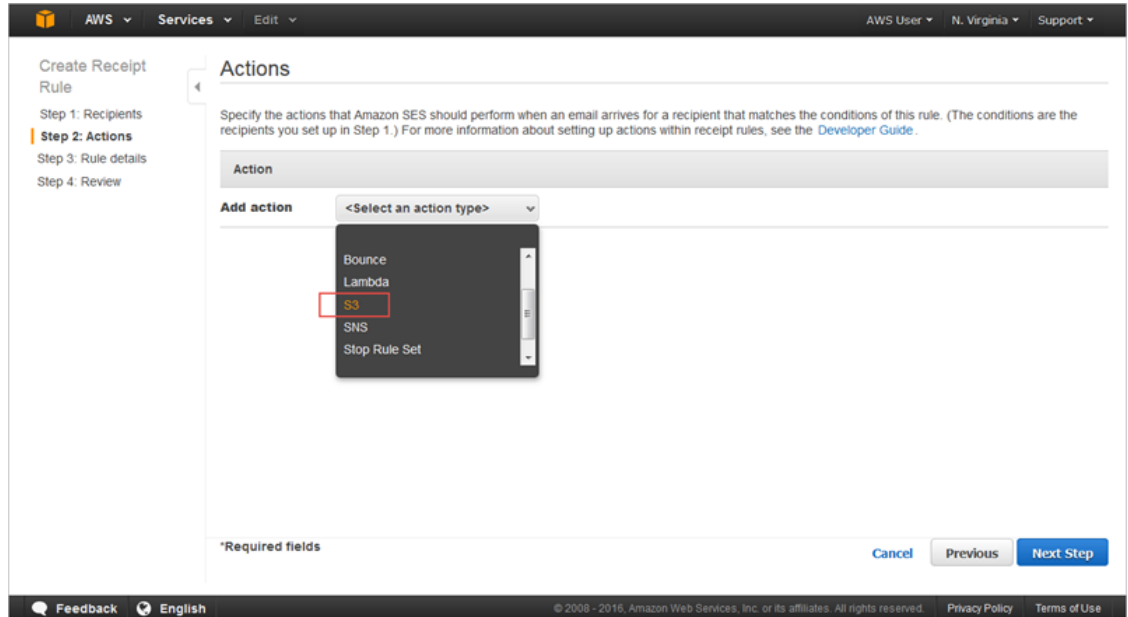
受信ルールを作成するには

1. ナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
2. [Create a Receipt Rule] を選択します。
3. [Recipients] ページで、[Next Step] を選択します。

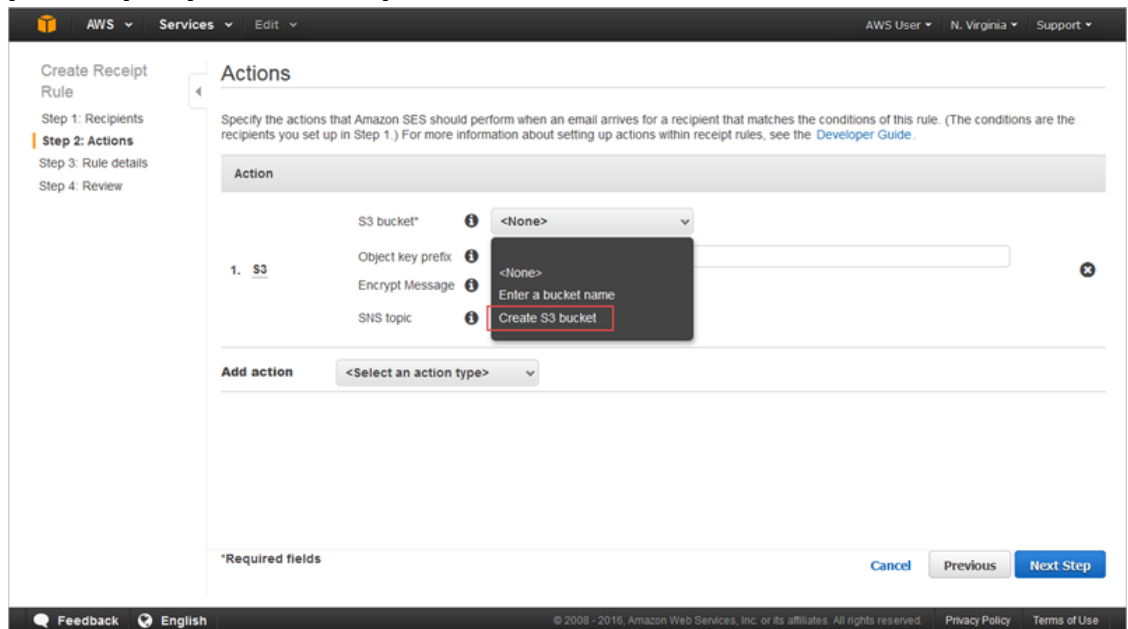
#### Note

受信者を追加していないため、Amazon SES はすべての検証済みドメインのすべての受信者にこのルールを適用します。

4. [Add action] で、[S3] を選択します。



5. [S3 bucket] で、[Create S3 bucket] を選択します。



6. [Bucket Name] に Amazon S3 バケットの名前を入力します。入力したバケット名は次の要件を満たしている必要があります。

- 小文字、数字、ピリオド (.)、ハイフン (-) のみ使用できます。
- AWS 全体で一意である必要があります。
- 先頭または末尾には数字または小文字を使用する必要があります。
- 3~63 文字以内にする必要があります。
- IP アドレスの形式にすることはできません (例: 192.168.5.4)。
- ピリオドを連続して使用したり (..)、ピリオドの横にダッシュを使用したりすることはできません (-. または .-)。

完了したら、[Create Bucket] を選択します。

#### Note

Amazon SES コンソールを使用して Amazon S3 バケットを作成するため、Amazon SES はバケットに書き込むアクセス許可を付与するポリシーを自動的に作成して適用します。ただし、既存の Amazon S3 バケットを選択する場合は、バケットに書き込むアクセス許可を Amazon SES に付与する必要があります。そのためには、Amazon S3 コンソールまたは API を使用して [バケットにポリシーを添付します \(p. 222\)](#)。

7. [Next Step] を選択します。
8. [Rule Details (ルールの詳細)] ページの [Rule name (ルール名)] に、**my-rule** と入力します。[Enabled] の横にあるチェックボックスをオンにし、[Next Step] を選択します。

The screenshot shows the 'Rule Details' page in the AWS Management Console. The 'Rule name' field is set to 'my-rule'. The 'Enabled' checkbox is checked, 'Require TLS' is unchecked, and 'Enable spam and virus scanning' is checked. The 'Rule set' is set to 'default-rule-set' and the 'Rule position' is set to '<Beginning>'. The 'Next Step' button is highlighted with a red box.

9. [Review] ページで、[Create Rule] を選択します。

次のステップ: [ステップ 4: テスト Eメールの送信 \(p. 216\)](#)

## ステップ 4: テスト Eメールの送信

これで、ドメインを検証して設定したため、Eメールを送信してドメインがEメールを受信できることをテストできます。

テスト Eメールを送信するには、個人の Eメールアドレスなど、Eメールを送信できることがわかっている Eメールアカウントを使用します。検証済みドメインで任意の Eメールアドレス宛てにテストメッセージを送信します。例えば、ドメインが example.com の場合、Eメールを test@example.com または abc123@example.com (または example.com ドメインの他のアドレス) に送信できます。

#### Note

ドメインで個々の Eメールアドレスを作成するために追加のステップを実行する必要はありません。Amazon SES は、検証済みドメインに送信されたすべての Eメールを受信し、「[ステップ 3: 受信ルールの設定 \(p. 214\)](#)」で作成した受信ルールを適用します。

次のステップ: [ステップ 5: 受信した Eメールの表示 \(p. 217\)](#)



## ステップ 5: 受信した Eメールの表示

ドメインのアドレスにテストメッセージを送信した後、Amazon S3 バケットから取得してその内容を表示することができます。

Amazon SES 経由で受信したメッセージを表示するには

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. Amazon S3 コンソールで、「[ステップ 3: 受信ルールの設定 \(p. 214\)](#)」で作成したバケットを選択します。
3. Amazon S3 バケット内で、受信したメールを見つけます。Eメールの名前は、文字と数字の一意的な文字列です。

### Note

バケットには、AMAZON\_SES\_SETUP\_NOTIFICATION というファイルも含まれていることがあります。このファイルは無視または削除できます。

4. ファイルの名前の横にあるチェックボックスを選択します。[Actions] メニューで、[Download] を選択します。
5. コンピュータで、前述のステップでダウンロードしたファイルが保存されているフォルダを開きます。ダウンロードしたメッセージを表示するには、次のようないくつかの方法があります。
  - テキストエディタでファイルを開き、その内容を直接読みます。Eメールの送信に使用した方法によっては、メッセージの一部がエンコードされる可能性があります。メッセージの一部がエンコードされている場合、手動でデコードする必要があります (たとえば、base64 デコーダを使用)。
  - ファイル名の末尾に .eml 拡張子を追加し、Microsoft Outlook や Mozilla Thunderbird などの Eメールクライアントを使用してファイルを開きます。ほとんどの Eメールクライアントでは、メッセージのエンコードされた部分が自動的にデコードされ、HTML 形式や添付ファイルなどが表示されます。

次のステップ: [ステップ 6: クリーンアップ \(p. 217\)](#)

## ステップ 6: クリーンアップ

このチュートリアルを完了したら、追加料金が発生しないように、作成したリソースをクリーンアップできます。

### Amazon SES 受信ルールセット

今後 Amazon SES でドメイン宛ての Eメールを受信しない場合は、[有効な受信ルールセットを無効 \(p. 240\)](#)にできます。

### Amazon S3 バケット

作成した Amazon S3 バケットが不要になった場合は、削除できます。バケットを削除するには、まず、そのコンテンツを削除する必要があります。フォルダーとバケットの削除の詳細については、Amazon Simple Storage Service スタートガイドの「[オブジェクトとバケットの削除](#)」を参照してください。

### Route 53 ドメイン

Route 53 を使用してドメインを登録する必要がなくなった場合は、[登録を削除するか、別のレジストラにドメインを転送](#)できます。

## Amazon SES E メール受信の設定

このセクションでは、メールを受信するように Amazon SES を設定するために必要な操作を説明します。例えば、メールの受信、フィルタリング、処理をどのように実行するかは Amazon SES の設定方法に影響するため、これらを最初に検討する必要があります。また、Amazon SES でドメインを確認して、ドメインを所有していることを証明し、受信メールを Amazon SES に送信するようにドメインを設定する必要があります。次に、AWS リソースへのアクセス許可を Amazon SES に付与します。最後に、受信ルールセット、受信ルール、および IP アドレスフィルタを作成して、E メール受信を設定します。

これらのステップについて、これ以降のトピックで説明します。

- [Amazon SES による E メール受信のユースケースの検討 \(p. 218\)](#)
- [Amazon SES による E メール受信のためのドメインの検証 \(p. 220\)](#)
- [Amazon SES による E メール受信のための MX レコードの公開 \(p. 220\)](#)
- [Amazon SES に対する E メール受信に関するアクセス許可の付与 \(p. 222\)](#)
- [Amazon SES E メール受信の IP アドレスフィルタの作成 \(p. 225\)](#)
- [Amazon SES E メール受信の受信ルールセットの作成 \(p. 226\)](#)
- [Amazon SES による E メール受信の受信ルールの作成 \(p. 226\)](#)

これらのタスクが E メール受信プロセス全体のどこに当てはまるかについては、「[E メール受信の概念 \(p. 210\)](#)」を参照してください。

## Amazon SES による E メール受信のユースケースの検討

メールを受信するように Amazon SES を設定する前に、以下の質問を検討することが有用です。

### 一般的な考慮事項

Amazon SES アカウントはまだサンドボックスにありますか？

まだの場合は、アカウントを Amazon SES サンドボックスから削除するために [Amazon SES サンドボックス外への移動 \(p. 74\)](#) の手順を完了してください。

### E メールのコンテンツ

どのような場合に Amazon SES で E メールのコンテンツを合格にしますか。

Amazon SES では、E メールのコンテンツを 2 通りの方法で提供できます。指定された Amazon S3 バケットに E メールを保存する方法か、E メールのコピーを含む Amazon SNS 通知を送信する方法です。Amazon SES では、変更を加えていない E メールを多目的インターネットメール拡張 (MIME) 形式で配信します。MIME 形式の詳細については、[RFC 2045](#) を参照してください。

受信するメールの大きさはどれくらいですか？

Amazon S3 バケットに E メールを保存する場合、E メールの最大サイズ (ヘッダーを含む) は 30 MB です。Amazon SNS を通じて E メールを受信する場合、E メールの最大サイズ (ヘッダーを含む) は 150 KB です。

E メールの処理をどのようにトリガーしますか。

メールが配信された後で、独自のコードで処理する場合があります。たとえば、アプリケーションを使用して Base64 エンコードの E メールを表示可能な形式に変換してから、エンドユーザーが E メールクライアントで使えるようにする場合があります。このプロセスを開始する方法はいくつかあります。

- E メールが Amazon S3 に配信される場合、アプリケーションは、S3 アクションによって生成される Amazon SNS 通知をリッスンし、通知から Eメールのメッセージ ID を抽出して、そのメッセージ ID を使用して Amazon S3 から Eメールを取得します。

または、Lambda 関数を記述することで、受信ルールに Eメール処理を組み込むこともできます。この場合、受信ルールでは、まず Eメールを Amazon S3 に書き込んでから Lambda 関数をトリガーする必要があります。Lambda 関数が他のアクションの実行方法に影響する結果を返す必要があるかどうかに応じて、受信ルールから複数の Lambda アクションを同期または非同期で実行することができます。ユースケースで同期実行が絶対に必要とされるのでない限りは、非同期実行を使用することをお勧めします。AWS Lambda の詳細については、[AWS Lambda 開発者ガイド](#)を参照してください。

- SNS アクションを使用して Eメールを Amazon SNS 通知を介して配信する場合は、アプリケーションは、Amazon SNS 通知をリッスンし、通知から Eメールメッセージを抽出することができます。

Eメールを暗号化しますか。

Amazon SES は AWS Key Management Service (AWS KMS) と統合され、オプションで Amazon S3 バケットに書き込むメールを暗号化します。Amazon SES は、クライアント側の暗号化を使用して、Amazon S3 に書き込む前にメールを暗号化します。つまり、Amazon S3 からメールを取得した後にご自身でコンテンツを復号する必要があります。- [AWS SDK for Java](#) および [AWS SDK for Ruby](#) は、復号化を処理できるクライアントを提供します。Amazon SES では、Eメールを Amazon S3 バケットに配信するよう選択した場合にのみ利用できます。

## 不要なメール

メール受信プロセスのどの時点で、不要なメールを拒否しますか。

送信者が受信者に Eメールを送信しようとする時、送信者の Eメールサーバーは一連のコマンドを受信者のサーバーと交換します。このシーケンスは SMTP 対話と呼ばれます。

Eメール受信プロセスの 2 つの時点 (SMTP 対話中と SMTP 対話後) で、受信 Eメールを拒否できます。SMTP 対話中にメッセージを拒否するために IP アドレスフィルターを使用し、SMTP 対話後に Eメールを拒否するために受信ルールを使用します。

IP アドレスフィルターを使用して、特定の IP アドレスから送信された Eメールを拒否することができます。迷惑メールを拒否するために IP アドレスフィルターを使用する利点は、SMTP 対話中に拒否されたメッセージに対しては請求されないことです。IP アドレスフィルターを使用することの欠点は、実際のメッセージの内容を分析することなく、指定した IP アドレスからの Eメールを拒否してしまうことです。IP アドレスフィルターの詳細については、「[Amazon SES Eメール受信の IP アドレスフィルタの作成 \(p. 225\)](#)」を参照してください。

受信ルールを使用して、メッセージの送信先アドレス (ドメイン、またはサブドメイン) に基づいて、バウンス通知を Eメールの送信者に送信できます。受信ルールを使用する利点は、送信者にバウンス通知を送信する前に、受信メッセージに対して追加の分析を実行できることです。たとえば、メッセージが DKIM 認証に失敗した場合やスパムとして識別された場合にのみバウンス通知を送信するために AWS Lambda を使用できます。受信ルールを使用することの欠点は、受信ルールが SMTP 対話の後に処理されるため、受信したメッセージごとに請求されることです。Lambda を使用して着信メッセージの内容を分析した場合も、料金が発生する可能性があります。受信ルールの詳細については、「[Amazon SES による Eメール受信の受信ルールの作成 \(p. 226\)](#)」を参照してください。Lambda を使用した受信 Eメール分析の詳細については、「[Lambda 関数の例 \(p. 234\)](#)」を参照してください。

## 他の AWS のサービスの使用

適切なアクセス権限をセットアップしましたか。

メールを Amazon S3 バケットに配信する場合、自身が所有していない Amazon SNS トピックに公開する場合、Lambda 関数をトリガーする場合、またはカスタムのマスター AWS KMS キーを使用する場合は、それらのリソースへのアクセス許可を Amazon SES に付与する必要があります。Amazon SES にアクセ

ス権限を付与するには、それらの AWS のサービスに対応するコンソールまたは API から、リソースに対するポリシーを作成します。詳細については [アクセス許可の付与 \(p. 222\)](#)。

## メールのストリーミング

メールのストリームをどのように分割しますか。

ご使用のドメインでは、さまざまなクラスのメールを受信する可能性があります。たとえば、`user@example.com` 宛てのメールのように、ドメインのメールの一部は個人の受信箱に配信されます。それ以外に、`unsubscribe@example.com` 宛てのメールのように、自動化されたシステムに送信されるべきものもあります。それぞれ異なる処理を行えるように、受信ルールを使用して受信メールを分割することができます。受信ルールをセットアップする方法については、「[受信ルールの作成 \(p. 226\)](#)」を参照してください。

## リージョナルな可用性

ご利用のリージョンで Amazon SES は E メール受信をサポートしていますか？

Amazon SES は、特定の AWS リージョンでのみ E メール受信をサポートしています。リージョンの完全なリストを使用した E メール受信については、「[AWS 全般のリファレンス](#)」の [Amazon Simple Email Service のエンドポイントとクォータ](#) を参照してください。

## Amazon SES による E メール受信のためのドメインの検証

Amazon SES での Eメールの送受信に使用するドメインに関しては、まず、ご自身がそのドメインを所有していることを証明する必要があります。検証の手順には、Amazon SES でのドメイン検証の開始と、ご使用の DNS サーバーに対する TXT レコードの公開が含まれますが、それについては「[Amazon SES でのドメインの検証 \(p. 61\)](#)」で説明しています。

### Note

Amazon SES では個別の E メールアドレスを検証できますが、Amazon SES で E メールを受信する場合はドメインを検証する必要があります。

「[受信ルールの作成 \(p. 226\)](#)」で受信ルールをセットアップするときに、ドメイン検証プロセスを開始することもできます。受取人リストは、未検証の受取人を示し、検証を開始できるようにします。いずれの場合も、「[Amazon SES ドメイン検証 TXT レコード \(p. 65\)](#)」の説明に従って、DNS サーバーに対して TXT レコードを発行することによってドメイン検証を完了する必要があります。

E メールアドレスまたはドメインが確認されるかどうかは、Amazon SES コンソール内の E メールアドレスの ID またはドメインの ID リストを見るか、Amazon SES `GetIdentityVerificationAttributes` API を利用します。

## Amazon SES による E メール受信のための MX レコードの公開

メールエクスチェンジャレコード (MX レコード#) は、ドメインに送信された E メールを受け入れることができるメールサーバーを指定する設定です。

Amazon SES で受信メールを管理するには、MX レコードをドメインの DNS 設定に追加する必要があります。作成した MX レコードは、Amazon SES を使用している AWS リージョンの E メールを受信するエンドポイントを表します。たとえば、米国西部 (オレゴン) リージョンのエンドポイントは `inbound-smtp.us-west-2.amazonaws.com` です。エンドポイントの詳細なリストについては、「[Amazon SES のリージョンとエンドポイント \(p. 467\)](#)」を参照してください。

#### Note

E メールを受信する Amazon SES のエンドポイントは、IMAP または POP3 E メールサーバーではありません。これらの URL を E メールクライアントの受信メールサーバーとして使用することはできません。

E メールクライアントを使用して送信と受信の両方が可能なソリューションが必要な場合は、[Amazon WorkMail](#) の使用を検討してください。

次の手順には、MX レコードを作成するための一般的なステップが含まれています。MX レコードを作成する具体的な手順は、DNS またはホスティングプロバイダによって異なります。ドメインの DNS 設定への MX レコードの追加の詳細については、プロバイダのドキュメントを参照してください。

#### Note

次の手順を完了するには、自分のドメインの DNS レコードを変更できる必要があります。ドメインの DNS レコードにアクセスできない場合、またはアクセスに問題がある場合は、システム管理者に連絡してください。

ドメインの DNS 設定に MX レコードを追加するには

1. DNS プロバイダーの管理コンソールにサインインします。
2. 新しい MX レコードを作成します。
3. MX レコードの [名前] に、ドメイン、続いてピリオドを入力します。たとえば、Amazon SES でドメイン `example.com` に送信される E メールを管理する場合は、次のように入力します。

```
example.com.
```

#### Note

一部の DNS プロバイダーは名前フィールドをホスト、ドメイン、またはメールドメインとして参照します。

4. [Type (タイプ)] で [MX] を選択します。

#### Note

DNS プロバイダの中には、[Type (タイプ)] フィールドを [Record Type (レコードタイプ)] などの名前として参照するものがあります。

5. [値] に以下の値を入力します。

```
10 inbound-smtp.regionInboundUrl.amazonaws.com
```

上記の例では、*regionInboundUrl* を、Amazon SES で使用している AWS リージョンの E メールを受信するエンドポイントのアドレスに置き換えます。例えば、米国東部 (バージニア北部) リージョンを使用している場合は、*#####* を `us-east-1` に置き換えます。メール受信用のエンドポイントの完全なリストについては、「[Amazon SES のリージョンとエンドポイント \(p. 467\)](#)」を参照してください。

#### Note

一部の DNS プロバイダーの管理コンソールには、レコードの [値] とレコードの [Priority (優先度)] に別々のフィールドがあります。DNS プロバイダーがこのような場合は、優先度値に 10 を入力し、受信メールのエンドポイント URL を値に入力します。

## さまざまなプロバイダーの MX レコードを作成するための手順

ドメインの MX レコードを作成する手順は、使用している DNS プロバイダーによって異なります。このセクションには、いくつかの一般的な DNS プロバイダーに関するドキュメントへのリンクが含まれてい

ます。これは、プロバイダーの完全なリストではありません。以下にプロバイダーが示されていない場合は、Amazon SES で使用することができると考えられます。このリストに含まれているからといって、他社の製品やサービスを支持または推奨するものではありません。

DNS/ホスティングプロバイダー名	ドキュメントのリンク
Amazon Route 53	<a href="#">Amazon Route 53 コンソールを使用して、レコードを作成</a>
GoDaddy	<a href="#">MX レコードを追加する</a>
DreamHost	<a href="#">MX レコードを変更するにはどうすればよいですか?</a>
Cloudflare	<a href="#">メールまたは MX レコードを追加または編集するにはどうすればよいですか?</a>
HostGator	<a href="#">MX レコードを変更する - Windows</a>
Namecheap	<a href="#">メールサービスに必要な MX レコードを設定するには、どうすればよいですか?</a>
Names.co.uk	<a href="#">ドメインの DNS 設定の変更</a>
Wix	<a href="#">Wix アカウントの MX レコードの追加または更新</a>

## Amazon SES に対する E メール受信に関するアクセス許可の付与

Amazon S3 バケットへの Eメールの送信や Lambda 関数の呼び出しなど、Amazon SES で Eメールを受信したときに実行できるタスクには、特別なアクセス許可が必要です。このセクションには、いくつかの一般的なユースケースのサンプルポリシーを含みます。

このセクションのトピック:

- [Amazon S3 バケットへの書き込みができる Amazon SES への許可 \(p. 222\)](#)
- [AWS KMS マスターキーを使用する許可を Amazon SES に与える \(p. 223\)](#)
- [Lambda 関数を呼び出すためのアクセス許可を Amazon SES に与えます。 \(p. 224\)](#)
- [異なる AWS アカウントに属する Amazon SNS トピックに発行するための Amazon SES のアクセス許可を付与します。 \(p. 224\)](#)

## Amazon S3 バケットへの書き込みができる Amazon SES への許可

Amazon S3 バケットに以下のポリシーを適用すると、Amazon SES にそのバケットに書き込む許可を与えます。入力メールを Amazon S3 に転送する受信ルールを作成することに関する詳細については、「[S3 アクション \(p. 236\)](#)」を参照してください。

Amazon S3 バケットの添付ポリシーの詳細については、Amazon Simple Storage Service デベロッパーガイドの「[バケットポリシーとユーザーポリシーの使用](#)」を参照してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Sid": "AllowSESPuts",
  "Effect": "Allow",
  "Principal": {
    "Service": "ses.amazonaws.com"
  },
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::myBucket/*",
  "Condition": {
    "StringEquals": {
      "aws:Referer": "111122223333"
    }
  }
}
]
```

上のポリシー例に、以下の変更を加えます。

- `myBucket` を書き込みたい Amazon S3 バケットの名前に置き換えます。
- `111122223333` は自分の AWS アカウント ID に置き換えます。

## AWS KMS マスターキーを使用する許可を Amazon SES に与える

Amazon SES で E メールを暗号化するためには、受信ルールをセットアップしたときに指定した AWS KMS キーを使用するためのアクセス許可を付与する必要があります。ご使用のアカウントのデフォルトのマスターキー (`aws/ses`) を使用するか、ご自身で作成するカスタムのマスターキーを使用することができます。デフォルトのマスターキーを使用する場合は、そのキーを使用するためのアクセス許可を Amazon SES に付与する追加の手順を特に実行する必要はありません。カスタムのマスターキーを使用する場合は、キーのポリシーにステートメントを追加することで、使用するためのアクセス許可を Amazon SES に付与する必要があります。

ご使用のドメインで Amazon SES が E メールを受信するときに Amazon SES がカスタムのマスターキーを使用することを許可するために、次のポリシーステートメントを使用します。

```
{
  "Sid": "AllowSESToEncryptMessagesBelongingToThisAccount",
  "Effect": "Allow",
  "Principal": {
    "Service": "ses.amazonaws.com"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*"
}
```

### Note

Amazon SES は Amazon S3 [マルチパートアップロード API](#) を使用して、大きなメッセージ (5 MB 以上) を Amazon S3 バケットに送信します。AWS KMS を使用して暗号化されたメッセージを Amazon S3 バケットに送信し、5 MB を超えるメッセージを受信する計画の場合は、前の例のステートメントではなく、次のポリシーステートメントを使用する必要があります。

```
{
  "Sid": "AllowSESToEncryptMessagesBelongingToThisAccount",
```

```
"Effect": "Allow",
"Principal": {
  "Service": "ses.amazonaws.com"
},
"Action": [
  "kms:Encrypt",
  "kms:Decrypt",
  "kms:ReEncrypt*",
  "kms:GenerateDataKey*",
  "kms:DescribeKey"
],
"Resource": "*"
}
```

Amazon S3 におけるマルチパートアップロードの詳細については、Amazon Simple Storage Service デベロッパーガイドの「[マルチパートアップロード API の許可](#)」を参照してください。AWS KMS キーの添付ポリシーの詳細については、[AWS KMS デベロッパーガイド](#)の「[AWS Key Management Service のキーポリシーの使用](#)」を参照してください。

## Lambda 関数を呼び出すためのアクセス許可を Amazon SES に与えます。

Amazon SES が Lambda 関数を呼び出せるようにするには、Amazon SES コンソールで受信ルールを作成するときに、その関数を選択します。これを行うと、Amazon SES は自動的に必要なアクセス許可を関数に追加します。

または、AddPermission API の AWS Lambda オペレーションを使用して、関数へのポリシーを添付します。次の AddPermission API 呼び出しでは、Lambda 関数を呼び出すためのアクセス許可を Amazon SES に付与しています。次の例では、**111122223333**をご自分の AWS アカウント ID に置き換えます。Lambda 関数へのポリシーの添付に関する詳細については、[AWS Lambda デベロッパーガイド](#)の「[AWS Lambda のアクセス許可](#)」を参照ください。

```
{
  "Action": "lambda:InvokeFunction",
  "Principal": "ses.amazonaws.com",
  "SourceAccount": "111122223333",
  "StatementId": "GiveSESPermissionToInvokeFunction"
}
```

## 異なる AWS アカウントに属する Amazon SNS トピックに発行するための Amazon SES のアクセス許可を付与します。

使用する Amazon SNS トピックが、Amazon SES で使用すると同じ AWS アカウントによって所有されている場合、Amazon SES は追加のセットアップステップなしに、そのトピックに公開できます。別の AWS アカウントのトピックに通知を公開する場合は、Amazon SNS トピックにポリシーを添付する必要があります。

次のポリシーでは、Amazon SES トピックに別の AWS アカウントの Amazon SNS トピックに発行するためのアクセス許可を Amazon SES に与えます。

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ses.amazonaws.com"
      },

```



```
"Action": "SNS:Publish",
"Resource": "arn:aws:sns:us-west-2:SNS-TOPIC-ACCOUNT-ID:myTopic",
"Condition": {
  "StringEquals": {
    "AWS:SourceOwner": "SES-RECEIVING-ACCOUNT-ID"
  }
}
}
```

上のポリシー例に、以下の変更を加えます。

- `us-west-2` を Amazon SNS トピックがある AWS リージョンに置き換えます。
- `SNS-TOPIC-ACCOUNT-ID` を Amazon SNS トピックがある AWS アカウント ID に置き換えます。
- `myTopic` を通知を発行する Amazon SNS トピックの名前に置き換えます。
- `SES-RECEIVING-ACCOUNT-ID` を E メールを受信するように設定された AWS アカウント ID に置き換えます。

## Amazon SES E メール受信の IP アドレスフィルタの作成

IP アドレスフィルタを使用すると、オプションで、特定の IP アドレスまたは IP アドレス範囲から送信されるメールを受け入れるか拒否するかを指定できます。

IP アドレスフィルタを作成するには、Amazon SES コンソールまたは `CreateReceiptFilter` API を使用できます。

### Note

既知の IP アドレスで構成される有限のリストからのメールのみを受信する場合は、`0.0.0.0/0` を含むブロックリストをセットアップし、信頼する IP アドレスを含む許可リストをセットアップします。この設定では、デフォルトではすべての IP アドレスがブロックされ、明示的に指定する IP アドレスからのメールのみが許可されます。

IP アドレスフィルタを作成するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[IP Address Filters] の順に選択します。
3. コンテンツペインで、[Create Filter] を選択します。
4. [Filter Name] に、IP アドレスフィルタの名前を入力します。名前は 64 文字未満で、英数字、ハイフン (-)、アンダースコア (\_)、ピリオド (.) を使用できます。名前の最初と最後は、文字または数字でなければなりません。
5. [IP Address Range] には、ブロックまたは許可する 1 つの IP アドレスまたは IP アドレス範囲を、クラスレスドメイン間ルーティング (CIDR) 表記で入力します。1 つの IP アドレスの例は `10.0.0.1` です。IP アドレス範囲の例は `10.0.0.1/24` となります。CIDR 表記の詳細については、[RFC 2317](#) を参照してください。
6. [Policy Type] では、[Allow] または [Block] を選択します。
7. [Create Filter] を選択します。

`CreateReceiptFilter` API を使用して IP アドレスフィルタを作成する方法については、[Amazon Simple Email Service の API リファレンス](#) をご覧ください。

## Amazon SES E メール受信の受信ルールセットの作成

受信ルールセットは、すべてのドメインで受信したメールを Amazon SES で処理する方法を定義する受信ルールのコレクションです。Amazon SES を E メールサーバーとして使用するには、アカウントに受信ルールセットを作成する必要があります。E メール受信プロセスにおける受信ルールセットの役割の詳細については、「[E メール受信の概念 \(p. 210\)](#)」を参照してください。

一度にアクティブにできる受信ルールセットは 1 つだけです。ただし、複数の受信ルールセットを作成することができます。たとえば、過去に使用した受信ルールのレコードを保持する必要がある場合や、テスト目的で受信ルールを迅速に変更する必要がある場合は、複数の受信ルールセットがあると役立ちます。

### Note

Eメールの受信機能として Amazon SES を使用しない場合は、単純にすべての受信ルールセットを無効にします。受信ルールセットを無効にする方法の詳細については、「[受信ルールセットの管理 \(p. 239\)](#)」を参照してください。

受信ルールセットを作成するには、Amazon SES コンソールまたは API を使用できます。

- Amazon SES コンソールの使用
  - 受信ルールは受信ルールセット内のみ存在するため、受信ルールセットを作成するには、まず受信ルールの作成から開始します。詳細については、「[受信ルールの作成 \(p. 226\)](#)」を参照してください。この手順を最後まで実行すると、新しい受信ルールセットを作成できるようになります。
  - 「[受信ルールセットの管理 \(p. 239\)](#)」の説明に従って、既存の受信ルールセットをコピーします。
  - 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets]、[Create a New Rule Set] の順に選択します。
- Amazon SES API の使用 — CreateReceiptRuleSet API を使用して、空の受信ルールセットを作成します。[Amazon Simple Email Service API リファレンス](#)に記載しています。その後で、Amazon SES コンソールまたは CreateReceiptRule API を使用して、そのセットに受信ルールを追加します。

## Amazon SES による E メール受信の受信ルールの作成

受信ルールを使用すると、所有する E メールアドレスやドメインで受信した E メールを Amazon SES で処理する方法を指定できます。受信ルールは、条件および順序指定されたアクションのリストで構成されます。受信メールの受信者が受信ルールの条件に指定された受信者と一致すると、その受信ルールに指定されたアクションが Amazon SES で実行されます。E メール受信プロセスにおける受信ルールの役割の詳細については、「[E メール受信の概念 \(p. 210\)](#)」を参照してください。

### Important

受信ルールを設定するには、まずドメインを検証し、そのドメインに対して MX レコードを発行します。ドメインの検証の詳細については、「[Amazon SES でのドメインの検証 \(p. 61\)](#)」を参照してください。MX レコードの発行の詳細については、「[the section called “MX レコードの公開” \(p. 220\)](#)」を参照してください。

受信ルールは、Amazon SES コンソールまたは CreateReceiptRule API オペレーションを使用して作成できます。このセクションでは、コンソールで新しい受信ルールを作成する手順を示します。この手順では、Amazon SES アカウントに既存の受信ルールが含まれていないものとします。

## 受信ルールのセットアップ

受信ルールは、Amazon SES コンソールまたは CreateReceiptRule API オペレーションを使用して作成できます。

コンソールを使用して受信ルールを作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. [Create a Receipt Rule] を選択します。
4. 次の手順を使用して、1 人以上の受取人を追加します。これらの受取人を総称して条件と呼びます。受信ルールごとに 100 人の受取人を指定できます。
  - a. [Recipients] で、受信ルールをセットアップする対象の受信メールアドレスまたはドメインを指定します。次の表では、アドレス `user@example.com` を使用して受取人を指定する方法を示しています。

目的	指定する受取人	コメント
特定の E メールアドレスとマッチさせます。	<code>user@example.com</code>	また、ラベルが含まれるアドレスのバリエーション ( <code>user+123@example.com</code> や <code>user+xyz@example.com</code> など) ともマッチさせます。ただし、ラベルが含まれるアドレスを指定すると、その特定のアドレスのみがマッチされます。
ドメイン内のすべてのアドレスとマッチさせるが、そのサブドメイン内のアドレスとはマッチさせません。	<code>example.com</code>	
特定のサブドメイン内のすべてのアドレスとマッチさせるが、親ドメイン内のアドレスとはマッチさせません。	<code>subdomain.example.com</code>	
すべてのサブドメイン内のすべてのアドレスとマッチさせるが、親ドメイン内のアドレスとはマッチさせません。	<code>.example.com</code>	ドメイン名の前のピリオド (.) に注意してください。
ドメイン内のすべてのアドレスとマッチさせて、さらにそのすべてのサブドメイン内のすべてのアドレスとマッチさせます。	<code>example.com</code> <code>.example.com</code>	2 つの受取人を別個に作成して、1 つにはドメイン名を使用し、別の 1 つにはピリオド (.) に続けてドメイン名を使用します。
すべての検証済みドメイン内のすべての受信者とマッチさせます。	[なし]	受信者フィールドを空にします。

#### Important

複数の Amazon SES アカウントが共通のドメインで E メールを受信する場合 (同じ会社の複数のチームが個別の Amazon SES アカウントを使用する場合など)、Amazon SES では、これらのアカウント間で一致するすべての受信ルールが同時に処理されます。こ

の動作により、あるアカウントではバウンスが生成され、別のアカウントでは E メールが承認されるという状況が発生する場合があります。  
組織内で Amazon SES を使用するチーム間で調整し、各アカウントで固有の受信ルールを使用すること、それらのルールは重複していないことを確認することをお勧めします。このような場合、グループやチームに固有のメールアドレスやサブドメインのみを使用するように受信ルールを設定するのが最適です。

- b. [Add Recipient] を選択します。
  - c. 追加する受取人ごとにステップ a とステップ b を繰り返します。受取人の追加が完了したら、[Next Step] を選択します。
5. 次の手順を使用して、受信ルールに 1 つ以上のアクションを追加します。
- a. メニューからアクションを選択します。
  - b. アクションの設定を選択します。各アクションのオプションについては、「[アクションのオプション \(p. 228\)](#)」を参照してください。
  - c. 必要に応じてアクションを追加してから、[Next Step] を選択します。
6. [Rule Details] については、次の手順を使用して設定を選択します。
- a. [Rule Name] に、受信ルールの名前を入力します。名前は 64 文字未満で、英数字、ハイフン (-)、アンダースコア (\_)、ピリオド (.) を使用できます。名前の最初と最後は、文字または数字でなければなりません。
  - b. その受信ルールを有効にする場合は、[Enabled] オプションを選択したままにします。
  - c. Transport Layer Security (TLS) を使用して暗号化された接続を介して送信されていない受信メールを Amazon SES で拒否するには、[TLS] を選択します。
  - d. Amazon SES で受信メールのスパムおよびウイルスのスキャンを実行するには、[Enable Spam and Virus Scanning] を選択します。
7. [Rule Set] では、既存の受信ルールセットを選択するか、[Create New Rule Set] をクリックします。
8. [Rule Position] では、この受信ルールを、順序どおりに並べた受信ルールのリストのどの位置に配置するかを選択します。受信ルールは順序どおりに評価されます。
9. [Next Step] を選択し、さらに [Create Rule] を選択します。

CreateReceiptRule API を使用してルールを作成する方法については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## アクションのオプション

Amazon SES による E メール受信に関する受信ルールにはそれぞれ、順序どおりに並べられたアクションのリストが含まれます。受信ルールの全体的なセットアップ手順は、「[Amazon SES による E メール受信の受信ルールの作成 \(p. 226\)](#)」で説明しています。このセクションでは、各アクションタイプに固有のオプションについて説明します。

アクションタイプには以下のものがあります。

- [ヘッダー追加アクション \(p. 229\)](#)
- [バウンスアクション \(p. 229\)](#)
- [Lambda アクション \(p. 229\)](#)
- [S3 アクション \(p. 236\)](#)
- [SNS アクション \(p. 238\)](#)
- [停止アクション \(p. 238\)](#)
- [WorkMail アクション \(p. 238\)](#)

## ヘッダー追加アクション

ヘッダー追加アクションは、受信した E メールにカスタムヘッダーを追加します。このアクションは通常、他のアクションとの組み合わせでのみ使用します。このアクションには以下のオプションがあります。

- Header name - 追加するヘッダーの名前。1 ~ 50 文字で、英数字 (a-z、A-Z、0-9) とダッシュのみ使用できます。
- Header value - 追加するヘッダーの値。2048 文字未満で、改行文字 ("r" または "n") を含めることはできません。

## バウンスアクション

バウンスアクションは、送信者にバウンス応答を返すことによって E メールを拒否し、オプションで Amazon SNS を通じて通知します。このアクションには以下のオプションがあります。

- SMTP Reply Code - [RFC 5321](#) で定義される SMTP 応答コード。
- SMTP Status Code - [RFC 3463](#) で定義される SMTP 拡張ステータスコード。
- Message - バウンス E メールに含める、人間が読み取れるテキスト。
- Reply Sender - バウンス Eメールの送信者の E メールアドレス。これは、バウンス Eメールの送信元アドレスです。Amazon SES で検証されている必要があります。
- SNS Topic - バウンス Eメールが送信されたときにオプションで通知するための Amazon SNS トピックの名前または ARN。Amazon SNS トピックの ARN の例は、arn:aws:sns:us-west-2:123456789012:MyTopic となります。SNS トピックを作成するを選んでアクションを設定するときに、Amazon SNS トピックを作成することもできます。Amazon SNS トピックの詳細については、[Amazon Simple Notification Service デベロッパーガイド](#)を参照してください。

### Note

選択する Amazon SNS トピックは、E メールを受け取る Amazon SES エンドポイントと同じ AWS リージョン内に存在する必要があります。

これらのフィールドに独自の値を入力するか、バウンスの理由に基づいて SMTP Reply Code、SMTP Status Code、Message の各フィールドに値を入力するテンプレートを選択することができます。次のテンプレートを使用できます。

- Mailbox Does Not Exist - SMTP 応答コード = 550、SMTP ステータスコード = 5.1.1
- Message Too Large - SMTP 応答コード = 552、SMTP ステータスコード = 5.3.4
- Message Full - SMTP 応答コード = 552、SMTP ステータスコード = 5.2.2
- Message Content Rejected - SMTP 応答コード = 500、SMTP ステータスコード = 5.6.1
- Unknown Failure - SMTP 応答コード = 554、SMTP ステータスコード = 5.0.0
- Temporary Failure - SMTP 応答コード = 450、SMTP ステータスコード = 4.0.0

フィールドにカスタムの値を入力して使用できる可能性があるその他のバウンスコードについては、[RFC 3463](#) を参照してください。

## Lambda アクション

Lambda アクションは、Lambda 関数を通じてコードを呼び出し、オプションで Amazon SNS を通じて通知します。このアクションには以下のオプションがあります。

- Lambda 関数 — Lambda 関数の ARN。Lambda 関数の ARN の例は、arn:aws:lambda:us-west-2:account-id:function:MyFunction となります。AWS Lambda の詳細については、「[AWS Lambda デベロッパーガイド](#)」を参照してください。

- Invocation type - Lambda 関数の呼び出しタイプ。呼び出しタイプが RequestResponse の場合は、関数の実行により即座に応答が得られることを意味します。また、値が Event の場合は、関数が非同期で呼び出されることを意味します。ユースケースで同期実行が絶対に必要とされる場合を除き、呼び出しタイプには Event を使用することをお勧めします。

#### Note

RequestResponse の呼び出しには 30 秒のタイムアウトがあります。

AWS Lambda の呼び出しタイプの詳細については、「[AWS Lambda 開発者ガイド](#)」を参照してください。

- SNS Topic - 指定された Lambda 関数がトリガーされたときに通知するための Amazon SNS トピックの名前または ARN。Amazon SNS トピックの ARN の例は、arn:aws:sns:us-west-2:123456789012:MyTopic となります。SNS トピックを作成するを選んでアクションを設定するときに、Amazon SNS トピックを作成することもできます。Amazon SNS トピックの詳細については、「[Amazon Simple Notification Service デベロッパーガイド](#)」を参照してください。

#### Note

選択する Amazon SNS トピックは、E メールを受け取る Amazon SES エンドポイントと同じ AWS リージョン内に存在する必要があります。

## Lambda 関数の記述

E メールを処理するために、Lambda 関数を (Event 呼び出しタイプを使用して) 非同期で呼び出すことができます。Lambda 関数に渡されるイベントオブジェクトには、インバウンド E メールイベントに関するメタデータが格納されます。このメタデータを使用して、Amazon S3 バケットからメッセージコンテンツにアクセスすることもできます。

メールフローを実際に制御するには、Lambda 関数を同期的に呼び出す (RequestResponse 呼び出しタイプを使用する) 必要があります。Lambda 関数では、2 つの引数 (1 つ目の引数は callback、2 つ目の引数は null、disposition、あるいは STOP\_RULE に設定された STOP\_RULE\_SET プロパティ) を指定して CONTINUE メソッドを呼び出す必要があります。2 つ目の引数が null であるか、有効な disposition プロパティが指定されていない場合は、CONTINUE の場合と同様に、メールフローが継続し、後続のアクションとルールが処理されます。

例えば、Lambda 関数のコードの末尾に次の行を記述することで、受信ルールを停止することができます。

```
callback( null, { "disposition" : "STOP_RULE_SET" } );
```

AWS Lambda のコードの例については、「[Lambda 関数の例 \(p. 234\)](#)」を参照してください。概要レベルのユースケースの例については、「[ユースケースの例 \(p. 231\)](#)」を参照してください。

## 入力形式

Amazon SES は、Lambda 関数に JSON 形式で情報を渡します。最上位レベルのオブジェクトには Records 配列が格納され、この配列には eventSource、eventVersion、および ses のプロパティが入力されます。ses オブジェクトには receipt オブジェクトと mail オブジェクトが格納されますが、これらは「[通知の内容 \(p. 246\)](#)」で説明する Amazon SNS 通知とまったく同じ形式です。

#### Note

Amazon SES が Lambda に渡すデータには、メッセージに関するメタデータと、複数の E メールヘッダーが含まれます。ただし、メッセージの本文は含まれません。

以下に、Amazon SES が Lambda 関数に提供する入力の構造を概要レベルで示します。

```
{  
  "Records": [  
    {  
      "eventSource": "SES",  
      "eventVersion": "1.0",  
      "ses": {  
        "receipt": {  
          "action": "VIEW",  
          "timestamp": "2014-10-01T12:01:02.000Z",  
          "headers": {  
            "Content-Type": "text/html",  
            "From": "Example Sender",  
            "Subject": "Example Subject",  
            "To": "Example Recipient",  
            "X-SES-Header": "Example Value" }  
        },  
        "mail": {  
          "contentType": "HTML",  
          "headers": {  
            "Content-Type": "text/html",  
            "From": "Example Sender",  
            "Subject": "Example Subject",  
            "To": "Example Recipient",  
            "X-SES-Header": "Example Value" }  
        },  
        "text": "Example Content",  
        "html": "Example Content" }  
      }  
    }  
  ]  
}
```

```
{
  "eventSource": "aws:ses",
  "eventVersion": "1.0",
  "ses": {
    "receipt": {
      <same contents as SNS notification>
    },
    "mail": {
      <same contents as SNS notification>
    }
  }
}
]
```

## 戻り値

Lambda 関数では、次のいずれかの値を返すことにより、メールフローを制御できます。

- `STOP_RULE` - 現在の受信ルールでの後続のアクションは処理されませんが、後続の受信ルールは処理可能です。
- `STOP_RULE_SET` - 後続のアクションまたは受信のルールは処理されません。
- `CONTINUE` またはそれ以外の無効な値 - 後続のアクションおよび受信ルールが処理可能であることを意味します。

## ユースケースの例

以下の例では、Lambda 関数の結果を使用してメールフローを制御するように設定できるルールの概要をいくつか示します。これらの例の多くでは、デモの目的で、S3 アクションを結果として使用します。

### ユースケース 1: すべてのドメインにわたってスパムを削除する

この例では、すべてのドメインにわたってスパムを削除するグローバルなルールを示します。ルール 2 と 3 は、すべてのドメインにわたってスパムを削除した後で、ドメイン固有のルールを適用できることを示すために含めてあります。

#### ルール 1

受信者リスト: 空。したがって、このルールは、検証済みのすべてのドメインに含まれるすべての受取人に適用されます。

#### アクション

1. E メールがスパムである場合に `STOP_RULE_SET` を返す Lambda アクション (非同期)。そうでない場合は `CONTINUE` を返します。「[Lambda 関数の例 \(p. 234\)](#)」で示した、スパムの削除に関する Lambda 関数の例を参照してください。

#### ルール 2

受信者リスト: example1.com

#### アクション

1. 任意のアクション。

#### ルール 3

受信者リスト: example2.com

#### アクション

1. 任意のアクション。

### ユースケース 2: すべてのドメインにわたってスパムをバウンスする

この例では、すべてのドメインにわたってスパムをバウンスするグローバルなルールを示します。ルール 2 と 3 は、すべてのドメインにわたってスパムをバウンスした後で、ドメイン固有のルールを適用できることを示すために含めてあります。

#### ルール 1

受信者リスト: 空。したがって、このルールは、検証済みのすべてのドメインに含まれるすべての受取人に適用されます。

#### アクション

1. E メールがスパムである場合に CONTINUE を返す Lambda アクション (非同期)。そうでない場合は STOP\_RULE を返します。
2. バウンスアクション ("500 5.6.1. Message content rejected")。
3. 停止アクション。

#### ルール 2

受信者リスト: example1.com

#### アクション

1. 任意のアクション

#### ルール 3

受信者リスト: example2.com

#### アクション

1. 任意のアクション

### ユースケース 3: 固有のルールを適用する

この例では、停止アクションを使用して、E メールが複数のルールによって処理されるのを防ぐ方法を示します。この例では、特定のアドレスに対して 1 つのルールを設定し、ドメインに含まれるすべての E メールアドレスに対して別のルールを設定します。停止アクションを使用することで、特定の E メールアドレスに関するルールと一致するメッセージは、ドメインに適用される汎用的なルールでは処理されません。

#### ルール 1

受信者リスト: user@example.com

#### アクション

1. Lambda アクション (非同期)。
2. 停止アクション。

#### ルール 2

受信者リスト: example.com



## アクション

1. 任意のアクション。

### ユースケース 4: CloudWatch にメールイベントをログ付けする

この例では、メールを Amazon SES に保存する前に、システムを通過するすべてのメールの監査ログを保持する方法を示します。

#### ルール 1

受信者リスト: example.com

#### アクション

1. CloudWatch log にイベントオブジェクトを書き込む Lambda アクション (非同期)。CloudWatch への [Lambda 関数の例 \(p. 234\)](#) ログ内のサンプル Lambda 関数。
2. S3 アクション。

### ユースケース 5: DKIM に適合しないメールを削除する

この例では、すべての受信メールを Amazon S3 バケットに保存するが、特定の E メールアドレスに宛てた E メールのうち DKIM に合格したもののみを、自動化された E メールアプリケーションに送信する方法を示します。

#### ルール 1

受信者リスト: example.com

#### アクション

1. S3 アクション。
2. メッセージが DKIM に合格しない場合に `STOP_RULE_SET` を返す Lambda アクション (非同期)。そうでない場合は `CONTINUE` を返します。

#### ルール 2

受信者リスト: support@example.com

#### アクション

1. 自動化されたアプリケーションをトリガーする Lambda アクション (非同期)。

### ユースケース 6: 件名の行に基づいてメールをフィルタリングする

この例では、ドメインの受信メールのうち、件名行に "discount" という語を含むすべてのメールを削除してから、自動化されたシステム宛てのメールについては特定の処理を実行し、ドメイン内のその他のすべての受信者宛のメールは別の方法で処理する方法を示します。

#### ルール 1

受信者リスト: example.com

#### アクション

1. 件名行に "discount" という語が含まれる場合に `STOP_RULE_SET` を返す Lambda アクション (非同期)。そうでない場合は `CONTINUE` を返します。

## ルール 2

受信者リスト: support@example.com

アクション

1. バケット 1 を使用する S3 アクション。
2. 自動化されたアプリケーションをトリガーする Lambda アクション (非同期)。
3. 停止アクション。

## ルール 3

受信者リスト: example.com

アクション

1. バケット 2 を使用する S3 アクション。
2. ドメインのそれ以外の受取人宛の E メールを処理する Lambda アクション (非同期)。

## Lambda 関数の例

このトピックでは、メールフローを制御する Lambda 関数の例を示します。

### 例 1: スパムを削除する

この例では、少なくとも 1 つのスパムインジケータがあるメッセージの処理を停止します。

```
exports.handler = function(event, context, callback) {
  console.log('Spam filter');

  var sesNotification = event.Records[0].ses;
  console.log("SES Notification:\n", JSON.stringify(sesNotification, null, 2));

  // Check if any spam check failed
  if (sesNotification.receipt.spfVerdict.status === 'FAIL'
      || sesNotification.receipt.dkimVerdict.status === 'FAIL'
      || sesNotification.receipt.spamVerdict.status === 'FAIL'
      || sesNotification.receipt.virusVerdict.status === 'FAIL') {
    console.log('Dropping spam');
    // Stop processing rule set, dropping message
    callback(null, {'disposition':'STOP_RULE_SET'});
  } else {
    callback(null, null);
  }
};
```

### 例 2: 特定のヘッダーが見つかった場合に続行する

この例では、E メールに特定のヘッダー値が含まれる場合にのみ、現在のルールの処理を続行します。

```
exports.handler = function(event, context, callback) {
  console.log('Header matcher');

  var sesNotification = event.Records[0].ses;
  console.log("SES Notification:\n", JSON.stringify(sesNotification, null, 2));

  // Iterate over the headers
  for (var index in sesNotification.mail.headers) {
    var header = sesNotification.mail.headers[index];

    // Examine the header values
```

```
        if (header.name === 'X-Header' && header.value === 'X-Value') {
            console.log('Found header with value.');
            callback(null, null);
            return;
        }
    }

    // Stop processing the rule if the header value wasn't found
    callback(null, {'disposition':'STOP_RULE'});
};
```

### 例 3: Amazon S3 からの E メールの取得

この例では、Amazon S3 から raw Eメールを取得して処理します。

#### Note

先に S3 アクションを使用して Amazon S3 に E メールを書き込む必要があります。

```
var AWS = require('aws-sdk');
var s3 = new AWS.S3();

var bucketName = '<YOUR BUCKET GOES HERE>';

exports.handler = function(event, context, callback) {
    console.log('Process email');

    var sesNotification = event.Records[0].ses;
    console.log("SES Notification:\n", JSON.stringify(sesNotification, null, 2));

    // Retrieve the email from your bucket
    s3.getObject({
        Bucket: bucketName,
        Key: sesNotification.mail.messageId
    }, function(err, data) {
        if (err) {
            console.log(err, err.stack);
            callback(err);
        } else {
            console.log("Raw email:\n" + data.Body);

            // Custom email processing goes here

            callback(null, null);
        }
    });
};
```

### 例 4: DMARC 認証に失敗するメッセージのバウンス

この例では、受信する E メールが DMARC 認証に失敗した場合にバウンスメッセージを送信します。

#### Note

この例を使用する場合は、E メール受信ドメインに `emailDomain` 環境変数の値を設定します。

```
'use strict';

const AWS = require('aws-sdk');

// Assign the emailDomain environment variable to a constant.
const emailDomain = process.env.emailDomain;

exports.handler = (event, context, callback) => {
    console.log('Spam filter starting');
```

```
const sesNotification = event.Records[0].ses;
const messageId = sesNotification.mail.messageId;
const receipt = sesNotification.receipt;

console.log('Processing message:', messageId);

// If DMARC verdict is FAIL and the sending domain's policy is REJECT
// (p=reject), bounce the email.
if (receipt.dmarcVerdict.status === 'FAIL'
    && receipt.dmarcPolicy.status === 'REJECT') {
    // The values that make up the body of the bounce message.
    const sendBounceParams = {
        BounceSender: `mailer-daemon@${emailDomain}`,
        OriginalMessageId: messageId,
        MessageDsn: {
            ReportingMta: `dns; ${emailDomain}`,
            ArrivalDate: new Date(),
            ExtensionFields: [],
        },
    },
    // Include custom text explaining why the email was bounced.
    Explanation: "Unauthenticated email is not accepted due to the sending domain's
DMARC policy.",
    BouncedRecipientInfoList: receipt.recipients.map((recipient) => ({
        Recipient: recipient,
        // Bounce with 550 5.6.1 Message content rejected
        BounceType: 'ContentRejected',
    })),
    });

    console.log('Bouncing message with parameters:');
    console.log(JSON.stringify(sendBounceParams, null, 2));
    // Try to send the bounce.
    new AWS.SES().sendBounce(sendBounceParams, (err, data) => {
        // If something goes wrong, log the issue.
        if (err) {
            console.log(`An error occurred while sending bounce for message:
${messageId}`, err);
            callback(err);
        }
        // Otherwise, log the message ID for the bounce email.
        } else {
            console.log(`Bounce for message ${messageId} sent, bounce message ID:
${data.MessageId}`);
            // Stop processing additional receipt rules in the rule set.
            callback(null, {
                disposition: 'stop_rule_set',
            });
        }
    });
    // If the DMARC verdict is anything else (PASS, QUARANTINE or GRAY), accept
    // the message and process remaining receipt rules in the rule set.
    } else {
        console.log('Accepting message:', messageId);
        callback();
    }
};
```

## S3 アクション

S3 アクションは、メールを Amazon S3 バケットに配信し、オプションで Amazon SNS を通じて通知します。このアクションには以下のオプションがあります。

- S3 Bucket - 受信した E メールを保存する Amazon S3 バケットの名前。S3 バケットを作成するを選択してアクションを設定するときに、新しい Amazon S3 トピックを作成することもできます。Amazon

SES は 通常は多目的インターネットメール拡張 (MIME) 形式の、変更を加えていない raw E メールを格納する文字列を提供します。MIME 形式の詳細については、[RFC 2045](#) を参照してください。

#### Important

E メールを Amazon S3 バケットに保存するときの Eメールの最大サイズ (ヘッダーを含む) は 30 MB です。

- Object Key Prefix - Amazon S3 バケット内で使用するキー名のプレフィックス。キー名のプレフィックスを使用して、フォルダー構造の Amazon S3 バケットを組織付けできます。例えば、[Email (E メール)] を [Object Key Prefix (オブジェクトキープレフィックス)] として使用すると、Amazon S3 バケットの [Email] という名前のフォルダーに E メールが表示されます。
- KMS Key ( コンソールで "Encrypt Message" が選択されている場合)- Eメールを Amazon S3 バケットに保存する前に暗号化するために Amazon SES が使用する、お客様のマスターキー。デフォルトのマスターキーが、AWS KMS で作成したカスタムのマスターキーを使用できます。

#### Note

選択するマスターキーは、Eメールを受け取る Amazon SES エンドポイントと同じ AWS リージョン内に存在する必要があります。

- デフォルトのマスターキーを使用するには、aws/sesを Amazon SES コンソールで受信ルールを設定するときを使用します。Amazon SES API を使用する場合、arn:aws:kms:REGION:AWSACCOUNTID:alias/aws/ses 形式で ARN を指定することでデフォルトのマスターキーを指定できます。たとえば、AWS アカウント ID が 123456789012 であり、米国西部 (オレゴン) リージョンのデフォルトのマスターキーを使用する場合は、デフォルトのマスターキーの ARN は arn:aws:kms:us-west-2:123456789012:alias/aws/ses となります。デフォルトのマスターキーを使用する場合は、そのキーを使用するためのアクセス許可を Amazon SES に付与する手順を特に実行する必要はありません。
- AWS KMS で作成したカスタムのマスターキーを使用するには、マスターキーの ARN を指定し、そのキーを使用するためのアクセス許可を Amazon SES に付与するステートメントをキーのポリシーに追加します。アクセス権限の付与の詳細については、「[Amazon SES に対する Eメール受信に関するアクセス許可の付与 \(p. 222\)](#)」を参照してください。

Amazon SESで AWS KMS の使用の詳細については、『[AWS Key Management Service開発者ガイド](#)』を参照してください。コンソールまたは API でマスターキーを指定しない場合、Amazon SES は Eメールを暗号化しません。

#### Important

Eメールは、Amazon S3 に送信されて保存される前に、Amazon SES 暗号化クライアントを使用して Amazon SES で暗号化されます。Amazon S3 サーバー側での暗号化を使用して暗号化されるわけではありません。つまり、サービスは復号のために AWS KMS キーを使用するためのアクセス権限を持たないため、Amazon S3 から Eメールを取得した後で、Amazon S3 暗号化クライアントを使用して復号する必要があります。この暗号化クライアントは、[AWS SDK for Java](#) および [AWS SDK for Ruby](#) でのみ使用できます。AWS KMS マスターキーを使用したクライアント側での暗号化の詳細については、「[Amazon Simple Storage Service デベロッパーガイド](#)」を参照してください。

- SNS Topic - Eメールが Amazon S3 バケットに保存されたときに通知するための Amazon SNS トピックの名前または ARN。Amazon SNS トピックの ARN の例は、arn:aws:sns:us-west-2:123456789012:MyTopic となります。SNS トピックを作成するを選んでアクションを設定するとき、Amazon SNS トピックを作成することもできます。Amazon SNS トピックの詳細については、[Amazon Simple Notification Service デベロッパーガイド](#)を参照してください。

#### Note

選択する Amazon SNS トピックは、Eメールを受け取る Amazon SES エンドポイントと同じ AWS リージョン内に存在する必要があります。

## SNS アクション

SNS アクションは、Amazon SNS 通知を使用してメールを公開します。通知には、完全なメールコンテンツが含まれます。このアクションには以下のオプションがあります。

- SNS Topic - E メールを公開する Amazon SNS トピックの名前または ARN。Amazon SNS 通知には、変更を加えていない raw E メールのコピーが含まれます。これは通常においては多目的インターネットメール拡張 (MIME) 形式です。MIME 形式の詳細については、[RFC 2045](#) を参照してください。

### Important

Amazon SNS の通知を通じて E メールを受信することを選択した場合、Eメールの最大サイズ (ヘッダーを含む) は 150 KB です。それよりも大きいメールはバウンスします。このサイズよりも大きい E メールが予想される場合は、代わりに Amazon S3 バケットに E メールを保存してください。

Amazon SNS トピックの ARN の例は、arn:aws:sns:us-west-2:123456789012:MyTopic となります。SNS トピックを作成するを選んでアクションを設定するときに、Amazon SNS トピックを作成することもできます。Amazon SNS トピックの詳細については、[Amazon Simple Notification Service デベロッパーガイド](#)を参照してください。

### Note

選択する Amazon SNS トピックは、E メールを受け取る Amazon SES エンドポイントと同じ AWS リージョン内に存在する必要があります。

- エンコーディング - Amazon SNS 通知内で E メールに使用するエンコーディング。UTF-8 は使用が容易ですが、別のエンコード形式でメッセージがエンコードされた場合にすべての特殊文字が保存されないことがあります。Base64 ではすべての特殊文字が保存されます。UTF-8 および Base64 の詳細については、それぞれ [RFC 3629](#) および [RFC 4648](#) を参照してください。

## 停止アクション

停止アクションは、受信ルールセットの評価を終了し、オプションで、Amazon SNS を通じて通知します。このアクションには以下のオプションがあります。

- SNS Topic - 停止アクションが実行されたときに通知するための Amazon SNS トピックの名前または ARN。Amazon SNS トピックの ARN の例は、arn:aws:sns:us-west-2:123456789012:MyTopic となります。SNS トピックを作成するを選んでアクションを設定するときに、Amazon SNS トピックを作成することもできます。Amazon SNS トピックの詳細については、[Amazon Simple Notification Service デベロッパーガイド](#)を参照してください。

### Note

選択する Amazon SNS トピックは、E メールを受け取る Amazon SES エンドポイントと同じ AWS リージョン内に存在する必要があります。

## WorkMail アクション

[WorkMail] アクションは、Amazon WorkMail と連携します。Amazon WorkMail ですべての E メール処理を実行する場合は、Amazon WorkMail がこの設定を担当するため、通常このアクションを直接使用することはありません。このアクションには以下のオプションがあります。

- Organization ARN — Amazon WorkMail の組織の ARN。Amazon WorkMail の組織の ARN は、arn:aws:workmail:**region**:**account\_ID**:organization/**organization\_ID** となります。ここで、
  - **region** は、Amazon SES および Amazon WorkMail を使用しているリージョンです。(これらは同じリージョンから使用する必要があります)。例: 「us-west-2」。

- `account_ID` は AWS アカウント ID です。AWS アカウント ID は、AWS マネジメントコンソールの「[アカウント](#)」ページを参照してください。
- `organization_ID` は、組織を作成したときに Amazon WorkMail によって生成される一意の識別子です。組織 ID は、Amazon WorkMail コンソールで、組織の [Organization 設定] ページから確認できます。

Amazon WorkMail 組織の完全な ARN の例は、`arn:aws:workmail:us-west-2:123456789012:organization/m-68755160c4cb4e29a2b2f8fb58f359d7` となります。Amazon WorkMail 組織の詳細については、[Amazon WorkMail 管理者ガイド](#)を参照してください。

- SNS Topic - Amazon WorkMail アクションが実行されたときに通知するための Amazon SNS トピックの名前または ARN。Amazon SNS トピックの ARN の例は、`arn:aws:sns:us-west-2:123456789012:MyTopic` となります。SNS トピックを作成するを選んでアクションを設定するときに、Amazon SNS トピックを作成することもできます。Amazon SNS トピックの詳細については、[Amazon Simple Notification Service デベロッパーガイド](#)を参照してください。

#### Note

選択する Amazon SNS トピックは、E メールを受け取る Amazon SES エンドポイントと同じ AWS リージョン内に存在する必要があります。

## Amazon SES による E メール受信の管理

受信ルールセット、受信ルール、IP アドレスフィルタを作成したら、Amazon SES コンソールまたは API を使用して、編集、削除、その他の操作を実行できます。また、受け取った Amazon SNS 通知を確認することや、Amazon CloudWatch を使用してエラーメトリクスを表示することもできます。

このセクションのトピック:

- [Amazon SES による E メール受信の受信ルールセットの管理 \(p. 239\)](#)
- [Amazon SES E メール受信のための受信ルールの管理 \(p. 242\)](#)
- [Amazon SES によるメール受信の IP アドレスフィルタの管理 \(p. 244\)](#)
- [Amazon SES によるメール受信のメトリクスの表示 \(p. 245\)](#)
- [Amazon SES E メール受信に関する通知の使用 \(p. 246\)](#)

## Amazon SES による E メール受信の受信ルールセットの管理

「[受信ルールセットの作成 \(p. 226\)](#)」の説明に従って受信ルールセットを作成した後は、必要に応じて更新することができます。受信ルールセットの編集では通常、「[受信ルールの管理 \(p. 242\)](#)」で説明するように個別の受信ルールを編集しますが、受信ルールセットの削除、有効化、無効化、コピーも行うことができます。さらに、受信ルールセット内の受信ルールの順序を変更することもできます。これらの操作について、これ以降のセクションで説明します。

このセクションのトピック:

- [受信ルールセットの削除 \(p. 240\)](#)
- [受信ルールセットの有効化/無効化 \(p. 240\)](#)
- [受信ルールセットのコピー \(p. 241\)](#)
- [受信ルールセットの順序の変更 \(p. 241\)](#)

## 受信ルールセットの削除

受信ルールセットを削除するには、Amazon SES コンソールまたは `DeleteReceiptRuleSet` API を使用できます。

### Note

現在有効になっている受信ルールセットは削除できません。

受信ルールセットを削除するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. [Inactive Rule Sets] リストで、削除する受信ルールセットを選択します。
4. [Actions] メニューから [Delete] を選択し、受信ルールセットの削除を確認します。

`DeleteReceiptRuleSet` API を使用して受信ルールセットを削除する方法については、[Amazon Simple Email Service API Reference](#) を参照してください。

## 受信ルールセットの有効化/無効化

個々の受信ルールセットは、有効または無効のいずれかの状態です。有効にできる受信ルールセットは常に 1 つだけです。受信ルールセットの無効化は、有効な受信ルールセットに変更を加えたいが、変更内容が適切であることを確認できるまではそれらの変更を有効にたくない場合に有用です。その場合、有効な受信ルールセットをコピーし、コピーした無効な受信ルールセットに対して変更を加えます。変更内容を確認したら、コピーした受信ルールセットを有効にすることができます。1 つの受信ルールセットをアクティブにすると、それ以外のすべての受信ルールセットは自動的に無効化されます。

### Note

Amazon SES を使用した E メール受信を完全に無効にするには、すべての受信ルールセットを無効にします。

Amazon SES コンソールまたは `SetActiveReceiptRuleSet` API を使用して、どのルールセットをアクティブにするかを制御できます。

無効な受信ルールセットを有効にするには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. [Inactive Rule Sets] リストで、有効にする受信ルールセットを選択します。
4. [Set as Active Rule Set] を選択します。

有効な受信ルールセットを無効にするには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. [Active Rule Set]、[Disable Active Rule Set] の順に選択し、受信ルールセットを無効にすることを確認します。

`SetActiveReceiptRuleSet` API を使用してルールセットを有効化または無効化する方法については、[Amazon Simple Email Service API Reference](#) を参照してください。



## 受信ルールセットのコピー

受信ルールセットをコピーするには、Amazon SES コンソールまたは `CloneReceiptRuleSet` API を使用できます。Amazon SES コンソールを使用する場合、コピーする受信ルールセットがアクティブが無効かによって、手順が若干異なります。

有効な受信ルールセットをコピーするには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. コンテンツペインで、[Copy Active Rule Set] を選択します。
4. [Copy Rule Set] ダイアログボックスで、コピーされた受信ルールセットに割り当てる名前を入力します。
5. [Copy Rule Set] を選択します。コピーされた受信ルールセットは [Inactive Rule Sets] リストに表示されます。

無効な受信ルールセットをコピーするには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. [Inactive Rule Sets] リストで、コピーする受信ルールセットを選択します。
4. [Actions] メニューから [Copy] を選択します。
5. [Copy Rule Set] ダイアログボックスで、コピーされた受信ルールセットに割り当てる名前を入力します。
6. [Copy Rule Set] を選択します。コピーされた受信ルールセットは [Inactive Rule Sets] リストに表示されます。

`CloneReceiptRuleSet` API を使用して受信ルールセットをコピーする方法については、[Amazon Simple Email Service API リファレンス](#) を参照してください。

## 受信ルールセットの順序の変更

受信ルールセット内の受信ルールの順序を変更するには、Amazon SES コンソールまたは `ReorderReceiptRuleSet` API を使用できます。Amazon SES コンソールを使用する場合、受信ルールセットが有効が無効かによって、手順が若干異なります。

有効な受信ルールセット内の受信ルールの順序を変更するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. コンテンツペインで、[View Active Rule Set] を選択します。
4. [Reorder Rules] を選択します。
5. 受信ルールの名前の横にある上向き矢印と下向き矢印を使用して受信ルールの順序を変更し、[Save Order] を選択します。

無効な受信ルールセット内の受信ルールの順序を変更するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。

2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. [Inactive Rule Sets] リストで、受信ルールセットを選択します。
4. [Reorder Rules] を選択します。
5. 受信ルールの名前の横にある上向き矢印と下向き矢印を使用して受信ルールの順序を変更し、[Save Order] を選択します。

ReorderReceiptRuleSet API を使用して受信ルールセット内の受信ルールの順序を変更するには、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## Amazon SES E メール受信のための受信ルールの管理

「[受信ルールの作成 \(p. 226\)](#)」で説明する受信ルールの作成に加えて、以下のセクションで説明するように、受信ルールの編集、削除、有効化、無効化、コピー、受信ルールセット内での受信ルールの位置の設定を行うことができます。

### Note

このセクションの手順では、受信ルールが有効な受信ルールセットに含まれていることを前提としています。無効な受信ルールセット内の受信ルールを編集するには、[Inactive Rule Sets] リストから受信ルールセットを選択します。それ以降の受信ルールの編集手順は、有効な受信ルールセットの場合と同じです。

このセクションのトピック:

- [受信ルールの編集 \(p. 242\)](#)
- [受信ルールの削除 \(p. 242\)](#)
- [受信ルールの有効化と無効化 \(p. 243\)](#)
- [受信ルールのコピー \(p. 243\)](#)
- [受信ルールの位置の設定 \(p. 244\)](#)

## 受信ルールの編集

受信ルールを編集するには、Amazon SES コンソールまたは Amazon SES API を使用できます。Amazon SES コンソールを使用するほうが簡単です。

受信ルールを編集するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. コンテンツペインで、[View Active Rule Set] を選択するか、[Inactive Rule Sets] リストから受信ルールセットを選択します。
4. 詳細ペインで、編集する受信ルールを選択します。
5. [Edit Rule] ペインで、ポリシーを編集して [Save Rule] を選択します。

コンソールではなく Amazon SES API を使用する場合は、DescribeReceiptRule API を使用してルールを取得し、テキストエディターを使用してルールを編集してから、UpdateReceiptRule API を使用して旧バージョンのルールを上書きします。詳細については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## 受信ルールの削除

受信ルールを削除するには、Amazon SES コンソールまたは DeleteReceiptRule API を使用できます。

### 受信ルールを削除するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. コンテンツペインで、[View Active Rule Set] を選択するか、[Inactive Rule Sets] リストから受信ルールセットを選択します。
4. 詳細ペインで、受信ルールを選択します。
5. [Actions] メニューから [Delete] を選択し、受信ルールの削除を確認します。

DeleteReceiptRule API を使用してルールを削除する方法については、[Amazon Simple Email Service API リファレンス](#)を参照してください。。

## 受信ルールの有効化と無効化

受信ルールを有効化または無効化するには、Amazon SES コンソールまたは Amazon SES API を使用できます。Amazon SES コンソールを使用するほうが簡単です。

### 受信ルールを有効または無効にするには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. コンテンツペインで、[View Active Rule Set] を選択するか、[Inactive Rule Sets] リストから受信ルールセットを選択します。
4. 詳細ペインで、編集する受信ルールを選択します。
5. [Edit Rule] ペインで、[Enabled] を選択または選択解除して、[Save Rule] を選択します。

コンソールではなく Amazon SES API DescribeReceiptRule を使用する場合は、API を使用して受信ルールを取得し、テキストエディターを使用して受信ルールの [Enabled] フィールドを編集してから、UpdateReceiptRule API を使用して受信ルールの旧バージョンを上書きします。詳細については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## 受信ルールのコピー

受信ルールをコピーするには、Amazon SES コンソールまたは Amazon SES API を使用できます。Amazon SES コンソールを使用するほうが簡単です。

### 受信ルールをコピーするには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. コンテンツペインで、[View Active Rule Set] を選択するか、[Inactive Rule Sets] リストから受信ルールセットを選択します。
4. 詳細ペインで、受信ルールを選択します。
5. [Actions] メニューから [Copy Rule] を選択します。
6. [Copy Rule] ダイアログボックスで、新しい受信ルールの名前を入力し、追加先の受信ルールセットを選択します。新しい受信ルールは受信ルールセットの冒頭に挿入され、初期状態では無効になっています。

Amazon SES API を代わりに使用する場合は、DescribeReceiptRule API を使用して受信ルールを取得し、テキストエディターを使用して受信ルールの名前と (必要な場合は) 受信ルールセットを編集してから、その受信ルールを CreateReceiptRule API に渡します。詳細については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## 受信ルールの位置の設定

受信ルールセット内での受信ルールの位置を変更するには、Amazon SES コンソールまたは SetReceiptRulePosition API を使用できます。

受信ルールの位置を設定するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[Rule Sets] の順に選択します。
3. コンテンツペインで、[View Active Rule Set] を選択するか、[Inactive Rule Sets] リストから受信ルールセットを選択します。
4. コンテンツペインで、[Reorder Rules] を選択します。
5. 受信ルールの名前の横にある上向き矢印と下向き矢印を使用して受信ルールの順序を変更し、[Save Order] を選択します。

SetReceiptRulePosition API を使用して受信ルールセット内の受信ルールの位置を変更する方法については、[Amazon Simple Email Service](#) を参照してください。

## Amazon SES によるメール受信の IP アドレスフィルタの管理

「[IP アドレスフィルタの作成 \(p. 225\)](#)」で説明する IP アドレスフィルタの作成に加えて、以下のセクションで説明するように、フィルタの表示と削除を行うことができます。

### IP アドレスフィルタの表示

IP アドレスフィルタのリストを取得するには、Amazon SES コンソールまたは ListReceiptFilters API を使用できます。

IP アドレスフィルタを表示するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左側のナビゲーションペインで、[Email Receiving]、[IP Address Filters] の順に選択します。IP アドレスフィルタのリストが表示されます。

ListReceiptFilters API を使用して IP アドレスフィルタのリストを取得する方法については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

### IP アドレスフィルタの削除

IP アドレスフィルタを削除するには、Amazon SES コンソールまたは DeleteReceiptFilter API を使用できます。

IP アドレスフィルタを削除するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。

2. 左側のナビゲーションペインで、[Email Receiving]、[IP Address Filters] の順に選択します。
3. 詳細ペインで、IP アドレスフィルタを選択します。
4. [Delete] を選択し、IP アドレスフィルタを削除することを確認します。

DeleteReceiptFilter API を使用して IP アドレスフィルタを削除する方法については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## Amazon SES によるメール受信のメトリクスの表示

Amazon CloudWatch (CloudWatch) を使用して、受信ルールの障害に関するメトリクスを表示できます。メトリクスは [SES/Rule Metrics] で確認できます。

失敗に関するメトリクスは 2 つあります。

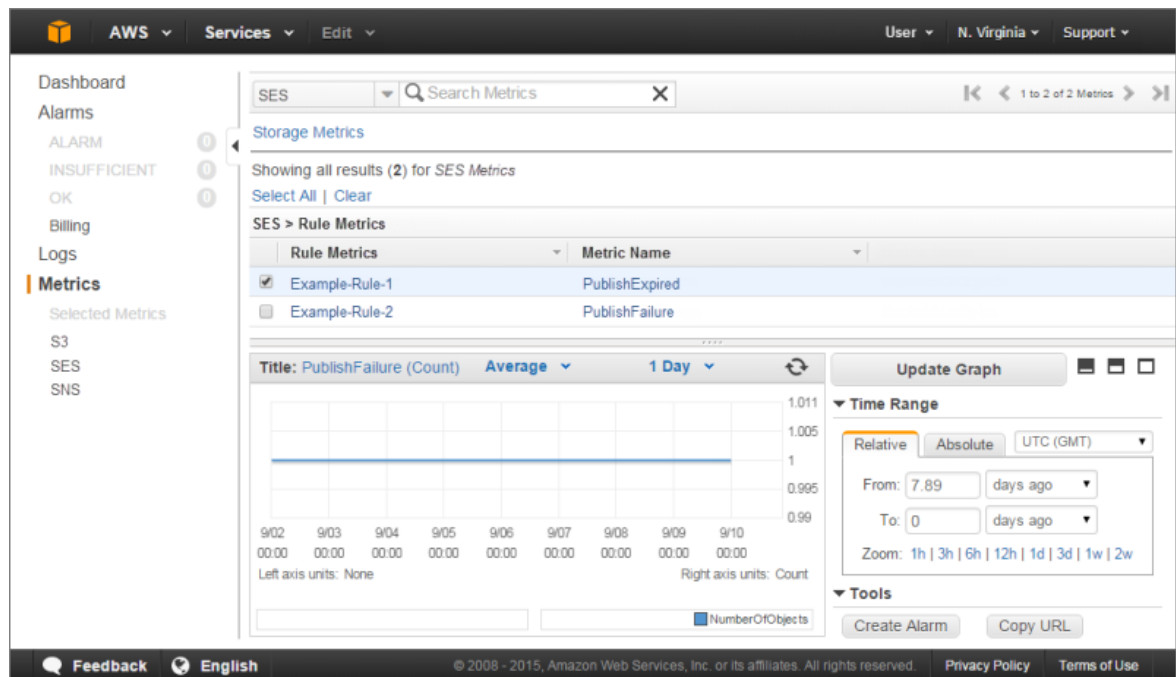
- PublishFailure - 構成されたアクションを Amazon SES が実行しようとしたときにエラーが発生しました。
- PublishExpired - 構成されたアクションを が実行しようとしたときに Amazon SES にエラーが発生し、Amazon SES は当該 E メール の配信をそれ以上再試行しません。この障害は、永続的または一時的である可能性があります。Amazon SES は 4 時間以内にアクションが成功しなかったため、これ以上再試行しません。

これらのエラーは、たとえば、いずれかの受信ルールのアクションで使用するように設定されている Amazon S3 バケット、Amazon SNS トピック、Lambda 関数に対するアクセス許可を削除したか取り消した場合に、発生することがあります。

### Important

受信ルールセットを修正するために行う変更は、更新後に Amazon SES が受信する E メールにのみ適用されます。Eメールの評価は常に、その Eメールが受信された時点で有効な受信ルールセットに対して行われます。

次の図は、CloudWatch コンソールに表示されたメトリクスを示します。



## Amazon SES E メール受信に関する通知の使用

E メールを受信すると、Amazon SES はアクティブな受信ルールセットのルールを実行します。Amazon SNS を使用して通知を送信するように受信ルールを設定できます。受信ルールでは、次の 2 つの異なるタイプの通知を送信できます。

- SNS アクションから送信される通知 - 受信ルールに [SNS \(p. 238\)](#) アクションを追加すると、E メールに関する情報が送信されます。メッセージが 150KB 以下の場合、この通知タイプには Eメールの完全な MIME 本文も含まれます。
- 他のアクションタイプから送信される通知 - 他のアクションタイプ ([バウンズ \(p. 229\)](#)、[Lambda \(p. 229\)](#)、[ルールセットの停止 \(p. 238\)](#) または [WorkMail \(p. 238\)](#) アクションなど) を受信ルール追加すると、オプションで Amazon SNS トピックを指定することができます。指定した場合は、これらのアクションが実行されると通知が送信されます。これらの通知には Eメールに関する情報が含まれますが、Eメールのコンテンツは含まれません。

このセクションでは、これら通知の内容を説明し、それぞれの通知タイプの例を示します。

このセクションのトピック:

- [Amazon SES の E メール受信通知の内容 \(p. 246\)](#)
- [Amazon SES の E メール受信通知の例 \(p. 252\)](#)

## Amazon SES の E メール受信通知の内容

E メール受信に関するすべての通知は、JavaScript Object Notation (JSON) 形式で Amazon Simple Notification Service (Amazon SNS) トピックに公開されます。

### トップレベル JSON オブジェクト

最上位の JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
notificationType	通知タイプ。このタイプの通知の場合、この値は常に Received です。
<a href="#">receipt (p. 247)</a>	E メール配信に関する情報を格納するオブジェクト。
<a href="#">mail (p. 251)</a>	通知が関係する E メールについての情報を格納するオブジェクト。
content	通常は多目的インターネットメール拡張 (MIME) 形式の、変更を加えていない raw E メールを格納する文字列。MIME 形式の詳細については、 <a href="#">RFC 2045</a> を参照してください。  Note  このフィールドは、通知が SNS アクションによってトリガーされたときだけ使用されます。その他のすべてのアクションによってトリガーされる通知には、このフィールドは含まれません。

## receipt オブジェクト

receipt オブジェクトには以下のフィールドがあります。

フィールド名	説明
<a href="#">action (p. 248)</a>	実行されたアクションに関する情報をカプセル化するオブジェクト。可能な値の一覧については、 <a href="#">アクションオブジェクト (p. 248)</a> を参照してください。
<a href="#">dkimVerdict (p. 249)</a>	ドメインキーアイデンティファイドメール (DKIM) のチェックに合格したどうかを示すオブジェクト。可能な値の一覧については、 <a href="#">dkimVerdict オブジェクト (p. 249)</a> を参照してください。
<code>dmARCPolicy</code>	送信元ドメインの Domain-based Message Authentication, Reporting & Conformance (DMARC) の設定を示します。このフィールドは、メッセージが DMARC 認証に失敗した場合にのみ表示されます。  このフィールドに可能な値は次のとおりです。 <ul style="list-style-type: none"> <li>• <code>none</code>: 送信元ドメインの所有者は、DMARC 認証に失敗するメッセージに対して特定のアクションをリクエストしません。</li> <li>• <code>quarantine</code>: 送信元ドメインの所有者は、DMARC 認証に失敗するメッセージが受信者によって疑わしいものとして扱われるようリクエストします。</li> <li>• <code>reject</code>: 送信元ドメインの所有者は、DMARC 認証に失敗するメッセージが拒否されるようリクエストします。</li> </ul>
<a href="#">dmARCVerdict (p. 249)</a>	Domain-based Message Authentication, Reporting & Conformance (DMARC) のチェックに合格したかどうかを示すオブジェクト。可能な値の一覧については、 <a href="#">dmARCVerdict オブジェクト (p. 249)</a> を参照してください。
<code>processingTimeMillis</code>	Amazon SES がメッセージを受信した時点からアクションをトリガーした時点までの期間をミリ秒で表す文字列。
<code>recipients</code>	アクティブな受信ルール ( <a href="#">p. 226</a> ) によって一致した受信者のリスト (具体的には、エンベロープの RCPT TO アドレス)。ここにリストされるアドレスは、「destination」の [ <a href="#">the section called "Mail オブジェクト" (p. 251)</a> ] フィールドに記載されているものとは異なる場合があります。
<a href="#">spamVerdict (p. 250)</a>	メッセージがスパムであるかどうかを示すオブジェクト。可能な値の一覧については、 <a href="#">spamVerdict オブジェクト (p. 250)</a> を参照してください。

フィールド名	説明
<a href="#">spfVerdict (p. 250)</a>	Sender Policy Framework (SPF) のチェックに合格したかどうかを示すオブジェクト。可能な値の一覧については、 <a href="#">spfVerdict オブジェクト (p. 250)</a> を参照してください。
timestamp	<a href="#">ISO 8601</a> フォーマットでアクションのトリガーが認定された日時を指定する文字列。
<a href="#">virusVerdict (p. 250)</a>	メッセージにウイルスが含まれているかどうかを示すオブジェクト。可能な値の一覧については、 <a href="#">virusVerdict オブジェクト (p. 250)</a> を参照してください。

## アクションオブジェクト

action オブジェクトには以下のフィールドがあります。

フィールド名	説明
type	実行されたアクションのタイプを示す文字列。使用できる値は、S3、SNS、Bounce、Lambda、Stop、および WorkMail です。
topicArn	通知が公開された Amazon SNS トピックの Amazon リソースネーム (ARN) を格納する文字列。
bucketName	メッセージが公開された Amazon S3 バケットの名前を格納する文字列。S3 アクションタイプでのみ使用されます。
objectKey	Amazon S3 バケット内の E メールを一意に識別する名前を格納する文字列。これは、「messageId」の <a href="#">the section called “Mail オブジェクト” (p. 251)</a> と同じです。S3 アクションタイプでのみ使用されます。
smtpReplyCode	<a href="#">RFC 5321</a> で定義されている SMTP 応答コードを格納する文字列。バウンスアクションタイプでのみ使用されます。
statusCode	<a href="#">RFC 3463</a> で定義されている SMTP 拡張ステータスコードを格納する文字列。バウンスアクションタイプでのみ使用されます。
message	バウンスメッセージに含める、人間が読み取れるテキストを格納する文字列。バウンスアクションタイプでのみ使用されます。
sender	バウンスした Eメールの送信元 Eメールアドレスを格納する文字列。これは、バウンスメッセージの送信元のアドレスです。バウンスアクションタイプでのみ使用されます。



フィールド名	説明
functionArn	トリガーされた Lambda 関数の ARN を格納する文字列。Lambda アクションタイプでのみ使用されます。
invocationType	Lambda 関数の呼び出しタイプを格納する文字列。指定できる値は RequestResponse および Event です。Lambda アクションタイプでのみ使用されます。
organizationArn	Amazon WorkMail 組織の ARN を格納する文字列。WorkMail アクションタイプでのみ使用されます。

### dkimVerdict オブジェクト

dkimVerdict オブジェクトには以下のフィールドがあります。

フィールド名	説明
status	DKIM 判定を格納する文字列。想定される値は次のとおりです。 <ul style="list-style-type: none"> <li>• PASS: メッセージは DKIM 認証に合格しています。</li> <li>• FAIL: メッセージは DKIM 認証に失敗しました。</li> <li>• GRAY: メッセージが DKIM 署名されていません。</li> <li>• PROCESSING_FAILED: Amazon SES で DKIM 署名をチェックできない問題が存在します。たとえば、DNS クエリに失敗しているか、DKIM 署名ヘッダーが正しくフォーマットされていません。</li> </ul>

### dmarcVerdict オブジェクト

dmarcVerdict オブジェクトには以下のフィールドがあります。

フィールド名	説明
status	DMARC 判定を格納する文字列。想定される値は次のとおりです。 <ul style="list-style-type: none"> <li>• PASS: メッセージは DMARC 認証に合格しています。</li> <li>• FAIL: メッセージは DMARC 認証に失敗しました。</li> <li>• GRAY: メッセージは DMARC 認証に失敗しました。送信元ドメインに DMARC ポリシーが存在しないか、p=none ポリシーを使用しています。</li> </ul>

フィールド名	説明
	<ul style="list-style-type: none"> <li>PROCESSING_FAILED: Amazon SES で DMARC の判定を提供できない問題が存在します。</li> </ul>

### spamVerdict オブジェクト

spamVerdict オブジェクトには以下のフィールドがあります。

フィールド名	説明
status	<p>スパムスキャンの結果を格納する文字列。想定される値は次のとおりです。</p> <ul style="list-style-type: none"> <li>PASS: スパムスキャンにより、メッセージにスパムが含まれている可能性は低いと判定されました。</li> <li>FAIL: スパムスキャンにより、メッセージにスパムが含まれている可能性が高いと判定されました。</li> <li>GRAY: Amazon SES でメールをスキャンしましたが、スパムかどうかを断定できませんでした。</li> <li>PROCESSING_FAILED: Amazon SES でメールの内容をスキャンできませんでした。たとえば、メールが有効な MIME メッセージではありません。</li> </ul>

### spfVerdict オブジェクト

spfVerdict オブジェクトには以下のフィールドがあります。

フィールド名	説明
status	<p>SPF 判定を格納する文字列。想定される値は次のとおりです。</p> <ul style="list-style-type: none"> <li>PASS: メッセージは SPF 認証に合格しています。</li> <li>FAIL: メッセージは SPF 認証に失敗しました。</li> <li>GRAY: MAIL FROM コマンドで使用されているドメインに SPF ポリシーがありません。</li> <li>PROCESSING_FAILED: Amazon SES で SPF レコードをチェックできない問題が存在します。たとえば、DNS クエリに失敗しています。</li> </ul>

### virusVerdict オブジェクト

virusVerdict オブジェクトには以下のフィールドがあります。

フィールド名	説明
status	<p>ウイルススキャンの結果を格納する文字列。想定される値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• PASS: メッセージにウイルスは含まれていません。</li> <li>• FAIL: メッセージにウイルスが含まれています。</li> <li>• GRAY: Amazon SES でメールをスキャンしましたが、ウイルスが含まれているかどうかを断定できませんでした。</li> <li>• PROCESSING_FAILED: Amazon SES でメールの内容をスキャンできません。たとえば、メールが有効な MIME メッセージではありません。</li> </ul>

## Mail オブジェクト

mail オブジェクトには以下のフィールドがあります。

フィールド名	説明
destination	受信 E メール MIME ヘッダーからのすべての受信者アドレス (To: および CC: 受信者を含む) の完全なリスト。
messageId	Amazon SES によって E メールに割り当てられる一意の ID を格納する文字列。E メールが Amazon S3 に配信された場合、このメッセージ ID は、Amazon S3 バケットにメッセージを書き込むために使用される Amazon S3 オブジェクトキーにもなります。
source	Eメールの送信元の E メールアドレス (具体的には、エンベロープ MAIL FROM アドレス) を格納する文字列。
timestamp	Eメールを受信した時刻を ISO8601 形式で格納する文字列。
headers	Amazon SES ヘッダーおよびカスタムヘッダーのリスト。リスト内の各ヘッダーには、name フィールドと value フィールドがあります。
<a href="#">commonHeaders (p. 252)</a>	すべての E メールに共通するヘッダーのリスト。リスト内の各ヘッダーは名前と値で構成されます。
headersTruncated	通知でヘッダーが切り捨てられたかどうかを示す文字列。切り捨ては、ヘッダーが 10 KB を超える場合に発生します。指定できる値は true および false です。

## commonHeaders オブジェクト

commonHeaders オブジェクトには、次の表に示すフィールドを含めることができます。このオブジェクトに存在するフィールドは、受信 E メールにどのフィールドが存在するかに応じて異なります。

フィールド名	説明
messageId	元のメッセージの ID。
date	Amazon SES がメッセージを受信した日時。
to	Eメールの To ヘッダーの値。
cc	Eメールの CC ヘッダーの値。
bcc	Eメールの BCC ヘッダーの値。
from	Eメールの From ヘッダーの値。
sender	Eメールの Sender ヘッダーの値。
returnPath	Eメールの Return-Path ヘッダーの値。
reply-to	Eメールの Reply-To ヘッダーの値。
subject	Eメールの Subject ヘッダーの値。

## Amazon SES の E メール受信通知の例

このセクションでは、以下のタイプの通知の例を紹介します。

- [SNS アクションの結果として送信される通知](#) (p. 252)
- [別のタイプのアクションの結果として送信された通知](#) (p. 254) (アラート通知)。

### SNS アクションの通知

このセクションでは、SNS アクションの通知の例を示します。前述のアラート通知とは異なり、E メールを格納する content セクションが含まれています。これは通常、多目的インターネットメール拡張 (MIME) 形式です。

```
{
  "notificationType": "Received",
  "receipt": {
    "timestamp": "2015-09-11T20:32:33.936Z",
    "processingTimeMillis": 222,
    "recipients": [
      "recipient@example.com"
    ],
    "spamVerdict": {
      "status": "PASS"
    },
    "virusVerdict": {
      "status": "PASS"
    },
    "spfVerdict": {
      "status": "PASS"
    },
    "dkimVerdict": {
      "status": "PASS"
    }
  }
}
```

```
    },
    "action":{
      "type":"SNS",
      "topicArn":"arn:aws:sns:us-east-1:012345678912:example-topic"
    }
  },
  "mail":{
    "timestamp":"2015-09-11T20:32:33.936Z",
    "source":"61967230-7A45-4A9D-BEC9-87CBCE2211C9@example.com",
    "messageId":"d6iitobk75ur44p8kdnnp7g2n800",
    "destination":[
      "recipient@example.com"
    ],
    "headersTruncated":false,
    "headers":[
      {
        "name":"Return-Path",

"value":"<0000014fbc1c09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-000000@amazonses.com>"
      },
      {
        "name":"Received",
        "value":"from a9-183.smtp-out.amazonses.com (a9-183.smtp-out.amazonses.com
[54.240.9.183]) by inbound-smtp.us-east-1.amazonaws.com with SMTP id
d6iitobk75ur44p8kdnnp7g2n800 for recipient@example.com; Fri, 11 Sep 2015 20:32:33 +0000
(UTC)"
      },
      {
        "name":"DKIM-Signature",
        "value":"v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple;
s=ug7nbtff4gccmlpwj322ax3p6ow6yfsug; d=amazonses.com; t=1442003552;
h=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Date:Message-
ID:Feedback-ID; bh=DWr3IOmYWoXCA9ARqGC/UaODfghffiwFNRIb2Mckyt4=;
b=p4ukUDSFqhqiub+zPRODW1kp7oJZakrzupr6LBe6sUuvqpBkig56UzUwc29rFbJF
h1X3Ov7DeYVNoN38stqwsF8ivcajXpQsXRC1cW9z8x875J041rClAjV7EGbLmudVpPX
4hHst1XPYX5wmgdHIhmUuh8oZKpVqGi6bHGzzf7g="
      },
      {
        "name":"From",
        "value":"sender@example.com"
      },
      {
        "name":"To",
        "value":"recipient@example.com"
      },
      {
        "name":"Subject",
        "value":"Example subject"
      },
      {
        "name":"MIME-Version",
        "value":"1.0"
      },
      {
        "name":"Content-Type",
        "value":"text/plain; charset=UTF-8"
      },
      {
        "name":"Content-Transfer-Encoding",
        "value":"7bit"
      },
      {
        "name":"Date",
        "value":"Fri, 11 Sep 2015 20:32:32 +0000"
      },
      {
```

```

    "name": "Message-ID",
    "value": "<61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com>"
  },
  {
    "name": "X-SES-Outgoing",
    "value": "2015.09.11-54.240.9.183"
  },
  {
    "name": "Feedback-ID",
    "value": "1.us-east-1.Krv2FKpFdWV+KUYw3Qd6wcpPJ4Sv/pOPpEPShn2u2o4=:AmazonSES"
  }
],
"commonHeaders": {
  "returnPath": "0000014fbe1c09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-000000@amazonses.com",
  "from": [
    "sender@example.com"
  ],
  "date": "Fri, 11 Sep 2015 20:32:32 +0000",
  "to": [
    "recipient@example.com"
  ],
  "messageId": "<61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com>",
  "subject": "Example subject"
}
},
"content": "Return-Path: <61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com>\r\nReceived: from a9-183.smtp-out.amazonses.com (a9-183.smtp-out.amazonses.com [54.240.9.183])\r\n by inbound-smtp.us-east-1.amazonaws.com with SMTP id d6iitobk75ur44p8kdnnp7g2n800\r\n for recipient@example.com;\r\n Fri, 11 Sep 2015 20:32:33 +0000 (UTC)\r\nDKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple;\r\n\t s=ug7nbt4gccmlpwj322ax3p6ow6yfsug; d=amazonses.com; t=1442003552;\r\n\t h=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Date:Message-ID:Feedback-ID;\r\n\t bh=DWr3IOmYWoXCA9ARqGC/UaODfghffiwFNRIb2Mckyt4=;\r\n\t b=p4ukUDSFqhqiuB+zPR0DW1kp7oJZakrzupr6LBe6sUuvqpBkig56UzUwc29rFbJF\r\n\t h1X30v7DeYVNoN38stqwsF8ivcajXpQsXRC1cW9z8x875J041rClAjV7EGbLmudVpPX\r\n\t 4hHst1XPyX5wmgdHIhmUuh8oZKpVqG16bHGzsf7g=\r\nFrom: sender@example.com\r\nTo: recipient@example.com\r\nSubject: Example subject\r\nMIME-Version: 1.0\r\nContent-Type: text/plain; charset=UTF-8\r\nContent-Transfer-Encoding: 7bit\r\nDate: Fri, 11 Sep 2015 20:32:32 +0000\r\nMessage-ID: <61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com>\r\nX-SES-Outgoing: 2015.09.11-54.240.9.183\r\nFeedback-ID: 1.us-east-1.Krv2FKpFdWV+KUYw3Qd6wcpPJ4Sv/pOPpEPShn2u2o4=:AmazonSES\r\n\r\nExample content\r\n"
}

```

## アラート通知

このセクションでは、S3 アクションによってトリガーできる Amazon SNS 通知の例を示します。Lambda アクション、バウンスアクション、停止アクション、および WorkMail アクションによってトリガーされる通知は類似しています。通知には E メールに関する情報が含まれますが、Eメールのコンテンツそのものは含まれません。

```

{
  "notificationType": "Received",
  "receipt": {
    "timestamp": "2015-09-11T20:32:33.936Z",
    "processingTimeMillis": 406,
    "recipients": [
      "recipient@example.com"
    ],
    "spamVerdict": {
      "status": "PASS"
    },
    "virusVerdict": {
      "status": "PASS"
    }
  }
}

```

```

},
"spfVerdict": {
  "status": "PASS"
},
"dkimVerdict": {
  "status": "PASS"
},
"action": {
  "type": "S3",
  "topicArn": "arn:aws:sns:us-east-1:012345678912:example-topic",
  "bucketName": "my-S3-bucket",
  "objectKey": "\email"
}
},
"mail": {
  "timestamp": "2015-09-11T20:32:33.936Z",
  "source": "0000014fbc09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-000000@amazonses.com",
  "messageId": "d6iitobk75ur44p8kdnp7g2n800",
  "destination": [
    "recipient@example.com"
  ],
  "headersTruncated": false,
  "headers": [
    {
      "name": "Return-Path",
      "value": "<0000014fbc09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-000000@amazonses.com>"
    },
    {
      "name": "Received",
      "value": "from a9-183.smtp-out.amazonses.com (a9-183.smtp-out.amazonses.com [54.240.9.183]) by inbound-smtp.us-east-1.amazonaws.com with SMTP id d6iitobk75ur44p8kdnp7g2n800 for recipient@example.com; Fri, 11 Sep 2015 20:32:33 +0000 (UTC)"
    },
    {
      "name": "DKIM-Signature",
      "value": "v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple; s=ug7nbt4gcccmlpwj322ax3p6ow6yfsug; d=amazonses.com; t=1442003552; h=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Date:Message-ID:Feedback-ID; bh=DWr3IOmYWoXCA9ARqGC/UaODfghffiwFNRIb2Mcky4=; b=p4ukUDSFqhqiub+zPRODW1kp7oJZakrzupr6LBe6sUuvqpBkig56UzUwc29rFbJFhlX3Ov7DeYVNoN38stqwsF8ivcajXpQsXRC1cW9z8x875J041rClAjV7EGbLmudVpPX4hHst1XPYX5wmgdHIhmUuh8oZKpVqGi6bHGzzf7g="
    },
    {
      "name": "From",
      "value": "sender@example.com"
    },
    {
      "name": "To",
      "value": "recipient@example.com"
    },
    {
      "name": "Subject",
      "value": "Example subject"
    },
    {
      "name": "MIME-Version",
      "value": "1.0"
    },
    {
      "name": "Content-Type",
      "value": "text/plain; charset=UTF-8"
    },
    {
      "name": "Content-Transfer-Encoding",

```

```
    "value": "7bit"
  },
  {
    "name": "Date",
    "value": "Fri, 11 Sep 2015 20:32:32 +0000"
  },
  {
    "name": "Message-ID",
    "value": "<61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com>"
  },
  {
    "name": "X-SES-Outgoing",
    "value": "2015.09.11-54.240.9.183"
  },
  {
    "name": "Feedback-ID",
    "value": "1.us-east-1.Krv2FKpFdWV+KUYw3Qd6wcpPJ4Sv/pOPpEPSHn2u2o4=:AmazonSES"
  }
],
"commonHeaders": {
  "returnPath":
  "0000014fbelc09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-000000@amazonses.com",
  "from": [
    "sender@example.com"
  ],
  "date": "Fri, 11 Sep 2015 20:32:32 +0000",
  "to": [
    "recipient@example.com"
  ],
  "messageId": "<61967230-7A45-4A9D-BEC9-87CBCF2211C9@example.com>",
  "subject": "Example subject"
}
}
}
```



# Amazon SES 設定セットの使用

設定セットは、Amazon SES で送信する E メールに適用できるルールのグループです。Eメールのヘッダーに設定セットへの参照を含めることで、Eメールに設定セットを適用します。Eメールに設定セットを適用すると、その設定セットに含まれるすべてのルールが Eメールに適用されます。Eメールの設定セットの指定の詳細については、[メールの送信時に設定セットを指定する \(p. 264\)](#) を参照してください。

設定セットを使用して以下のタイプのルールを Eメールに適用できます。

- イベントの発行 – Amazon SES は送信される各 Eメールの送信、配信、開封、クリック、バウンス、苦情イベントの数を追跡できます。イベントの発行を使用して、他の AWS のサービスにこれらのイベントについての情報を送信できます。例えば Eメールのメトリクスを to Amazon Kinesis Data Firehose destination に送り、それから Amazon Kinesis Data Analytics を使用しているものを分析できます。または、Amazon SNS にバウンスや苦情の情報を送信し、これらのイベントが発生したときに即座に通知を受け取ることができます。
- IP プールの管理 - 専用の IP アドレスをリースして Amazon SES で使用する場合、専用 IP プールと呼ばれるこれらのアドレスのグループを作成できます。その後、これらの専用 IP プールと設定セットを関連付けることができます。一般的なユースケースは、マーケティング Eメール送信用と取引 Eメール送信用に専用 IP アドレスのプールをそれぞれ作成することです。取引 Eメールの送信者の評価はマーケティング Eメールの送信者の評価とは切り離されています。

設定セットには、これらのいずれかまたは両方のタイプのルールを含めることも、いずれも含めないこともできます。

設定セットと関連コンポーネントの詳細については、以下のトピックを参照してください。

- [Amazon SES の設定セットの管理 \(p. 257\)](#)
- [Amazon SES でのイベント送信先の管理 \(p. 262\)](#)
- [専用 IP プールの管理 \(p. 264\)](#)

## Amazon SES の設定セットの管理

このセクションでは、設定セットの作成、既存の設定セットの一覧表示、個々の設定セットの詳細表示、および設定セットの削除の手順を示します。

### 設定セットの作成

Amazon SES コンソールまたは `CreateConfigurationSet` API を使用して、新しい設定セットを作成できます。

Amazon SES コンソールを使用して設定セットを作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで [設定セット] を選択します。

3. [Create Configuration Set] を選択します。
4. [Configuration set name (設定セットの名前)] に、設定セットの名前を入力します。

#### Note

この名前には 64 文字までの英数字を含めることができます。また、ハイフン (-) とアンダースコア (\_) も使用できます。名前にスペース、アクセント文字、またはその他の特殊文字を含めることはできません。

また、設定セットの作成には CreateConfigurationSet API を使用することもできます。この API を呼び出す一般的な方法は、AWS CLI を使用することです。

AWS CLI を使用して設定セットを作成するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses create-configuration-set --configuration-set Name=ConfigSet
```

前述のコマンドで、*ConfigSet* を設定セットに付ける名前置き換えます。

#### Note

この名前には 64 文字までの英数字を含めることができます。また、ハイフン (-) とアンダースコア (\_) も使用できます。名前にスペース、アクセント文字、またはその他の特殊文字を含めることはできません。

CreateConfigurationSet API を使用して設定セットを作成する方法についての詳細は、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。

## 設定セットの一覧表示

Amazon SES コンソールまたは ListConfigurationSets API を使用して、設定セットを一覧表示できます。

Amazon SES コンソールで設定セットを表示するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。

詳細ペインに、設定セットが一覧表示されます。

また、設定セットの一覧表示には ListConfigurationSets API を使用することもできます。この API を呼び出す一般的な方法は、AWS CLI を使用することです。

AWS CLI を使用して設定セットを一覧表示するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses list-configuration-sets
```

ListConfigurationSets API を使用して設定セットを一覧表示する方法についての詳細は、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。

## 設定セットの詳細の表示

Amazon SES コンソールを使用して設定セットの詳細を表示、または DescribeConfigurationSet API を使用して設定セットを記述できます。

Amazon SES コンソールを使用した設定セットの詳細の表示

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. 詳細ペインで、設定セットの横の展開アイコンを選択します。

設定セットの詳細が表示されます。

また、DescribeConfigurationSet API を使用して設定セットの詳細を表示することもできます。この API を呼び出す一般的な方法は、AWS CLI を使用することです。

AWS CLI を使用して設定セットの詳細情報を取得するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses describe-configuration-set --configuration-set-name ConfigSet
```

前述のコマンドで、`ConfigSet` を詳細を表示する設定セットの名前で置き換えます。

DescribeConfigurationSet API を使用して設定セットを記述する方法についての詳細は、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。

## 設定セットの削除

Amazon SES コンソールまたは DeleteConfigurationSet API を使用して設定セットを削除できます。

Amazon SES コンソールを使用して設定セットを削除するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. 詳細ペインで、設定セットを選択します。
4. [Actions] メニューから [Delete] を選択し、設定セットの削除を確認します。

また、設定セットの削除には DeleteConfigurationSet API を使用することもできます。この API を呼び出す一般的な方法は、AWS CLI を使用することです。

AWS CLI を使用して設定セットを削除するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses delete-configuration-set --configuration-set ConfigSet
```

前述のコマンドで、`ConfigSet` を、削除する設定セットの名前で置き換えます。

DeleteConfigurationSet API を使用して設定セットを削除する方法についての詳細は、「[Amazon Simple Email Service API リファレンス](#)」を参照してください。

## Amazon SES デフォルト設定セットの管理

ID を作成するときに、電子メール ID にデフォルト設定セットを追加したり、既存の ID にデフォルト設定セットを追加したりできます。デフォルト設定セットは、E メール ID とドメイン ID の両方に適用できません。

デフォルト設定セットは、その設定セットに関連付けられた電子メール ID から送信するすべてのメッセージに、その規則を自動的に適用します。

デフォルト設定セットに関する考慮事項

- ID に関連付ける前に、まず設定セットを作成する必要があります。
- デフォルト設定セットは、ID が検証された場合のみ適用されます。
- E メール ID は、一度に 1 つの設定セットにのみ関連付けることができます。ただし、同じ設定セットを複数の ID に適用することができます。
- 電子メールアドレスレベルのデフォルトの設定セットは、ドメインレベルのデフォルトの設定セットよりも優先されます。たとえば、joe@example.com に関連するデフォルト設定セットが、example.com のドメインの設定セットをオーバーライドします。
- ドメインレベルで設定されたデフォルト設定は、そのドメインのすべての電子メールアドレスに適用されます (ドメインの特定アドレスを確認しない限り)。
- ID のデフォルト設定セットとして指定されている設定セットを削除し、その ID を使用して E メールを送信しようとすると、Amazon SES への呼び出しが「bad request」エラーで失敗します。

## Amazon SES API を使用したデフォルト設定セットの ID の作成と検証

Amazon SES API v2 で、[CreateEmailIdentity](#) オペレーションを使用して、新しい E メール ID を作成し、同時にデフォルト設定を設定します。

### Note

このセクションの手順を完了する前に、まず AWS CLI をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

を使用してデフォルト設定セットを設定するには AWS CLI

- コマンドラインで以下のコマンドを入力して、[CreateEmailIdentity](#) オペレーションを使用します。

```
aws sesv2 create-email-identity --email-identity ADDRESS-OR-DOMAIN --configuration-set-name CONFIG-SET
```

上記のコマンドで、**ADDRESS-OR-DOMAIN** を、検証する E メール ID に置き換えます。**CONFIG-SET** を、ID のデフォルト設定セットとして設定する設定セット名に置き換えます。

コマンドが正常に実行された場合、何の出力もなく終了します。

E メールアドレスを検証するには

1. 受信箱で、検証するメールアドレスをチェックします。「アマゾン ウェブ サービス - Email Address Verification Request in region **RegionName**」という件名のメッセージが表示されます。ここで言う、**RegionName** は、E メールアドレスを確認しようとした AWS リージョンの名前です。

メッセージを開き、メッセージ内のリンクをクリックします。

#### Note

検証メッセージ内のリンクは、メッセージが送信されてから 24 時間で期限切れになります。確認メールを受信してから 24 時間が経過した場合は、手順 1〜5 を繰り返して、有効なリンクが設定された確認メールを受信してください。

2. Amazon SES コンソールの [Identity Management] で [Email Addresses] を選択します。E メールアドレスのリストで、検証する E メールアドレスを探します。E メールアドレスが検証された場合は、[Status] 列の値が「verified」になります。

ドメインを検証するには

ドメインの検証の詳細については、「[Amazon SES でのドメインの検証 \(p. 62\)](#)」を参照してください。

## Amazon SES API を使用して既存の E メール ID にデフォルト設定セットを追加する

`PutEmailIdentityConfigurationSetAttributes` オペレーションを使用して、既存の電子メール ID からデフォルト設定セットを追加または削除できます。

#### Note

このセクションの手順を完了する前に、まず AWS CLI をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

を使用してデフォルト設定セットを追加するには AWS CLI

- コマンドラインで以下のコマンドを入力して `PutEmailIdentityConfigurationSetAttributes` オペレーションを使用します。

```
aws sesv2 put-email-identity-configuration-set-attributes --email-identity ADDRESS-OR-DOMAIN --configuration-set-name CONFIG-SET
```

上記のコマンドで、*ADDRESS-OR-DOMAIN* を、検証する E メール ID に置き換えます。*CONFIG-SET* を、ID のデフォルト設定セットとして設定する設定セット名に置き換えます。

コマンドが正常に実行された場合、何の出力もなく終了します。

を使用してデフォルト設定セットを削除するには AWS CLI

- コマンドラインで以下のコマンドを入力して `PutEmailIdentityConfigurationSetAttributes` オペレーションを使用します。

```
aws sesv2 put-email-identity-configuration-set-attributes --email-identity ADDRESS-OR-DOMAIN
```

上記のコマンドで、*ADDRESS-OR-DOMAIN* を、検証する E メール ID に置き換えます。

コマンドが正常に実行された場合、何の出力もなく終了します。

## Amazon SES API を使用したデフォルト設定セットを返す

`GetEmailIdentity` オペレーションを使用して、電子メール ID のデフォルト設定セットを返します (該当する場合)。

## Note

このセクションの手順を完了する前に、まず AWS CLI をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

を使用してデフォルト設定セットを返すには AWS CLI

- コマンドラインで以下のコマンドを入力して `GetEmailIdentity` オペレーションを使用します。

```
aws sesv2 get-email-identity --email-identity ADDRESS-OR-DOMAIN
```

上記のコマンドで、*ADDRESS-OR-DOMAIN* を、デフォルト設定セットが知りたい電子メールIDに置き換えます (存在する場合)。

コマンドが正常に実行された場合、E メール ID の詳細を含む JSON オブジェクトが提供されます。

## Amazon SES API を使用したデフォルト設定セットのオーバーライド

`SendEmail` オペレーションを使用して、異なる設定セットで電子メールを送信できます。そうした場合、指定した設定セットが、ID のデフォルト設定セットをオーバーライドします。

## Note

このセクションの手順を完了する前に、まず AWS CLI をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

を使用してデフォルト設定セットをオーバーライドするには AWS CLI

- コマンドラインで以下のコマンドを入力して、`SendEmail` オペレーションを使用します。

```
aws sesv2 send-email --destination file://DESTINATION-JSON --content file://CONTENT-JSON --from-email-address ADDRESS-OR-DOMAIN --configuration-set-name CONFIG-SET
```

上記のコマンドで、*DESTINATION-JSON* を宛先の JSON ファイルに置き換え、*CONTENT-JSON* をコンテンツ JSON ファイルに置き換え、*ADDRESS-OR-DOMAIN* を FROM メールアドレスに置き換え、*CONFIG-SET* を、ID のデフォルト設定セットではなく、使用する設定セットの名前に置き換えます。

コマンドが正常に実行された場合、MessageId を出力します。

## Amazon SES でのイベント送信先の管理

イベント送信先を使用すると、E メール送信のメトリクス (送信、配信、オープン、クリック、バウンス、苦情の数など) を他の AWS 製品に発行できます。イベント発行の詳しい設定方法については、「[the section called “イベント発行を使用した送信のモニタリング” \(p. 296\)](#)」を参照してください。

## イベント送信先の更新

Amazon SES コンソールまたは `UpdateConfigurationSetEventDestination` API を使用して、イベント送信先を更新できます。

イベント送信先を更新するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. 設定セットリストで、更新するイベント送信先が含まれている設定セットを選択します。
4. [Destination (送信先)] リストで、編集する送信先の右にある [edit (編集)] アイコン (✎) を選択します。
5. イベント送信先の詳細を編集し、[Save] を選択します。
6. [Edit Configuration Set] ページを終了するには、ブラウザの戻るボタンを使用します。

UpdateConfigurationSetEventDestination API を使用してイベントの発行先を更新する方法については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## イベント送信先の削除

Amazon SES コンソールまたは DeleteConfigurationSetEventDestination API を使用して、イベント送信先を削除できます。

イベント送信先を削除するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. 設定セットリストで、削除するイベント送信先が含まれている設定セットを選択します。
4. [Destination (送信先)] リストで、[delete (削除)] アイコン (✖) を選択します。
5. 設定セットの削除を確認します。
6. [Edit Configuration Set] ページを終了するには、ブラウザの戻るボタンを使用します。

DeleteConfigurationSetEventDestination API を使用してイベントの発行先を削除する方法については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## イベント送信先の有効化または無効化

Amazon SES コンソールまたは UpdateConfigurationSetEventDestination API を使用してイベント送信先を有効化/無効化できます。

イベント送信先を有効化または無効化するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. 設定セットリストで、有効化または無効化するイベント送信先が含まれている設定セットを選択します。
4. [Destination] リストで、編集する発行先の右側の編集アイコン (鉛筆) を選択します。
5. [Enabled] をオンまたはオフにし、[Save] を選択します。
6. [Edit Configuration Set] ページを終了するには、ブラウザの戻るボタンを使用します。

UpdateConfigurationSetEventDestination API を使用してイベントの有効化または無効化を更新する方法については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## 専用 IP プールの管理

IP プールを使用すると、特定タイプの E メールを送信するための専用 IP アドレスのグループを作成できます。また、すべての Amazon SES の顧客が共有する IP アドレスのプールを使用できます。

### IP プールを既存の設定セットに割り当てる

Amazon SES コンソールを使用して、IP プールを既存の設定セットに関連付けることができます。

IP プールを設定セットに割り当てるには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. 設定セットのリストで、IP プールを関連付ける設定セットを選択します。
4. [Sending IP pool] タブの [Pool name] で、次のいずれかのオプションを選択します。
  - 特定の専用 IP プール – 既存の専用 IP プールを選択すると、そのプールに属する専用 IP アドレスのみを使用して、設定セットを使用する E メールが送信されます。新しい IP のプールの作成手順については、[専用 IP プールの作成 \(p. 185\)](#) を参照してください。
  - ses-default-dedicated-pool – このプールには、まだ IP プールに属していない、アカウントの専用 IP アドレスがすべて含まれています。プールに関連付けられていない設定セットを使用して E メールを送信する場合、または設定セットを全く指定せずに E メールを送信する場合、E メールはデフォルトプール内にあるいずれかのアドレスから送信されます。
  - ses-shared-pool – このプールには、すべての Amazon SES ユーザー間で共有されている多数の IP アドレスが含まれています。このオプションは、通常の送信動作と異なる E メールを送信する必要がある場合に便利です。

完了したら、[Assign] を選択します。

### IP プールの割り当てを変更する

Amazon SES コンソールを使用すると、既にプールに関連付けられている設定セットに、別のプールを割り当てることができます。同じ設定セットに別のプールを割り当てると、以前の関連付けが上書きされます。

IP プールの割り当てを編集するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. 設定セットのリストで、変更する対象の設定セットを選択します。
4. [Sending IP pool (IP プール送信)] タブの [Assign an IP pool (IP プールの割り当て)] の、[edit (編集)] アイコン (🔗) を選択します。
5. [Pool name] で、使用するプールを選択し、[Assign] を選択します。

## メールの送信時に設定セットを指定する

Eメールの送信時に設定セットを使用するには、Eメールのヘッダー内で設定セットの名前を渡す必要があります。送信する Eメールのヘッダー内で設定セットを渡すことは、すべての Amazon SES Eメール



送信手段 ([AWS CLI](#)、[AWS SDK](#)、[Amazon SES SMTP インターフェイス \(p. 88\)](#)など) で許可されています。

[SMTP インターフェイス \(p. 88\)](#)または [SendRawEmail API オペレーション](#)を使用している場合、E メールに以下のヘッダーを含めることで設定セットを指定できます (`ConfigSet` は使用する設定セットの名前に置き換えます)。

```
X-SES-CONFIGURATION-SET: ConfigSet
```

このガイドでは、Postfix、AWS SDK、Amazon SES SMTP のインターフェイスを使用して E メールを送信するためのコード例を紹介します。それぞれの例には、設定セットの指定方法が含まれています。設定セットへの参照を含む E メールを送信するための手順を確認するには、以下を参照してください。

- [Amazon SES と Postfix の統合 \(p. 97\)](#)
- [AWS SDK を使用して Amazon SES から E メールを送信する \(p. 31\)](#)
- [SMTP を使用して Amazon SES から E メールを送信する \(p. 21\)](#)

追加のコード例については、「[Amazon SES コード例 \(p. 425\)](#)」のセクションを参照してください。

## CloudWatch への設定セットの評価メトリクスのエクスポート

Amazon SES は、アカウント全体のバウンス率や苦情率についての情報を Amazon CloudWatch に自動的にエクスポートします。これらのメトリクスを使用して、CloudWatch でアラームを作成したり、Lambda 関数を使用して E メール送信を自動的に一時停止したりすることができます。

個々の設定セットの評価メトリクスを CloudWatch にエクスポートすることもできます。設定セットレベルで評価データをエクスポートすると、送信者の評価をより詳細に制御できます。

このセクションでは、Amazon SES API を使用して、個々の設定セットの評価データを CloudWatch にエクスポートする手順について説明します。

### 設定セットの評価メトリクスのエクスポートの有効化

設定セット評価メトリクスのエクスポートを開始するには、`UpdateConfigurationSetReputationMetricsEnabled` API オペレーションを使用します。Amazon SES API にアクセスするには、AWS CLI またはいずれかの AWS SDK を使用することをお勧めします。

この手順では、AWS CLI がコンピュータにインストールされ、正しく設定されていることを前提としています。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

設定セットの評価メトリクスのエクスポートを有効化するには

- コマンドラインから、以下のコマンドを入力します。aws ses update-configuration-set-reputation-metrics-enabled --configuration-set-name *ConfigSet* --enabled

前のコマンドの *ConfigSet* を、評価メトリクスのエクスポートを開始する設定セットの名前に置き換えます。

## 設定セットの評価メトリクスのエクスポートの無効化

また、UpdateConfigurationSetReputationMetricsEnabled API オペレーションを使用して、設定セットの評価メトリクスのエクスポートを無効にすることもできます。

設定セットの評価メトリクスのエクスポートを無効化するには

- コマンドラインから、以下のコマンドを入力します。aws ses update-configuration-set-reputation-metrics-enabled --configuration-set-name **ConfigSet** --no-enabled

前のコマンドの **ConfigSet** を、評価メトリクスのエクスポートを無効にする設定セットの名前に置き換えます。

# Amazon SES 送信アクティビティのモニタリング

Amazon SES には、送信アクティビティをモニタリングするための方法が用意されています。これらの方法を実装し、アカウントのバウンス率、苦情率、拒否率などの重要な指標を追跡することをお勧めします。バウンス率や苦情率が高すぎると、Amazon SES での E メール送信に支障が生じる場合があります。

これらの方法を使用して、送信した E メールに対する顧客のエンゲージメント率を測定することもできます。たとえば、これらの送信メトリクスに基づいてオープン率やクリックスルー率を確認できます。

Amazon SES で測定できるメトリクスは、E メール送信イベントと呼ばれます。モニタリングできるメール送信イベントは、次のとおりです。

- 送信 – Amazon SES に対する呼び出しが成功しました。Amazon SES はメールの配信を試行します。
- 拒否 – Amazon SES は E メールを受け取りましたが、ウイルスが含まれていると判断して拒否しました。Amazon SES は、受信者のメールサーバーに E メールを配信しようとしませんでした。
- バウンス – 受信者のメールサーバーにより、メールが完全に拒否されました。このイベントはハードバウンスに該当します。ソフトバウンスは、Amazon SES が一定期間にわたって再試行してもメールを配信できなかった場合に限りです。
- 苦情 – E メールは受信者に正常に配信されました。受取人がメールをスパムとしてマークしました。
- 配信 – Amazon SES が受信者のメールサーバーに E メールを正常に送信しました。
- オープン – 受信者がメッセージを受け取り、E メールクライアントで開きました。
- クリック – 受信者はメール内の 1 つ以上のリンクをクリックしました。
- レンダリング失敗 – テンプレートレンダリングの問題により E メールが送信されませんでした。このイベントタイプは、[SendTemplatedEmail](#) または [SendBulkTemplatedEmail](#) API オペレーションを使用してテンプレートメールを送信する場合にのみ発生します。このイベントタイプは、テンプレートデータが見つからない場合や、テンプレートのパラメータとデータが一致しない場合に発生します。
- 配信の遅延 – 一時的な問題が発生したため、メールを受信者に配信できませんでした。配信の遅延は、受信者の受信トレイがいっぱいになった場合や、受信側の電子メールサーバーで一時的な問題が発生した場合などに発生します。

E メール送信イベントはいくつかの方法でモニタリングできます。どのモニタリング方法を選択するかは、モニタリングするイベントのタイプ、モニタリングする詳細の範囲と深度、Amazon SES でデータを発行する場所によって異なります。バウンスと苦情のイベントを追跡するには、フィードバック通知またはイベント発行のいずれかを使用する必要があります。複数のモニタリング方法の使用を選択することもできます。各方法の特徴を次の表に示します。

モニタリング方法	モニタリング可能なイベント	データへのアクセス方法	詳細レベル	細分性
Amazon SES コンソール	配信、拒否	Amazon SES コンソールでの <a href="#">送信統計ページ</a> (p. 269)	カウントのみ	AWS アカウント全体
Amazon SES コンソール	バウンス率、苦情率	Amazon SES コンソールでの <a href="#">評価ダッシュボードページ</a> (p. 377)	計算された率のみ	AWS アカウント全体

モニタリング方法	モニタリング可能なイベント	データへのアクセス方法	詳細レベル	細分性
Amazon SES API	配信、バウンス、苦情、拒否	GetSendStatistics API オペレーション	カウントのみ	AWS アカウント全体
Amazon CloudWatch コンソール	送信、配信、開封数、クリック、バウンス、バウンス率、苦情、苦情率、拒否	CloudWatch コンソール  Note  一部のメトリクスは、関連付けられたイベントが発生するまで CloudWatch に表示されません。たとえば、バウンスのメトリクスは、送信した Eメールの少なくとも 1つがバウンスするまでは、またはメールボックスシミュレーター (p. 187) を使用して、シミュレートされたバウンスイベントを作成するまでは、CloudWatch に表示されません。。	カウントのみ	AWS アカウント全体

モニタリング方法	モニタリング可能なイベント	データへのアクセス方法	詳細レベル	細分性
フィードバック通知	配信、バウンス、および苦情	Amazon SNS 通知 (配信、バウンス、苦情) または E メール (バウンスと苦情のみ)	各イベントの詳細	AWS アカウント全体
イベントの発行	送信、配信、オープン、クリック、バウンス、苦情、拒否、レンダーリングの失敗。	Amazon CloudWatch または Amazon Kinesis Data Firehose、または Amazon SNS 通知による	各イベントの詳細	細かい (ユーザー定義可能なメールの特性に基づく)

#### Note

E メール送信イベントで測定したメトリクスは、送信クォータと完全には一致しない場合があります。この不一致は、Eメールのバウンスや拒否、または Amazon SES インボックスシミュレーターの使用に伴って発生する場合があります。どれだけ送信クォータに近付いているかを確認するには、「[送信クォータのモニタリング \(p. 150\)](#)」を参照してください。

各モニタリング方法を使用する方法については、以下のトピックを参照してください。

- [Amazon SES コンソールを使用した送信統計情報のモニタリング \(p. 269\)](#)
- [Amazon SES API を使用した使用統計のモニタリング \(p. 270\)](#)
- [通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)
- [Amazon SES イベント発行を使用して E メール送信をモニタリングする \(p. 296\)](#)

## Amazon SES コンソールを使用した送信統計情報のモニタリング

アカウントから配信されたメールの数および拒否されたメッセージの数を Amazon SES コンソールから直接モニタリングできます。アカウントの配信率と拒否率が [Sending Statistics] ページに表示されます。

バウンスおよび苦情に関する情報は、評価ダッシュボードに表示されます。詳細については、「[Amazon SES 送信者評価のモニタリング \(p. 377\)](#)」を参照してください。

配信と拒否に関するメトリクスを表示するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで、[Sending Statistics] を選択します。使用統計は、Amazon SES メトリクスに表示されます。
3. メトリクスの動向データを表示するには、当該グラフをダブルクリックします。グラフをダブルクリックする際に、分析期間を変更することもできます。

# Amazon SES API を使用した使用統計のモニタリング

Amazon SES API は、`GetSendStatistics` オペレーションを使用してサービスの使用に関する情報を返します。送信統計を定期的にチェックし、必要に応じて調整を行うようお勧めします。

`GetSendStatistics` オペレーションを呼び出すと、過去 2 週間の送信アクティビティを示すデータポイントのリストが返されます。このリストの各データポイントは、15 分間のアクティビティを示し、その期間に関する以下の情報が含まれています:

- ハードバウンス数
- 苦情数
- 配信試行回数 (送信した E メールの数に相当します)
- 送信試行の拒否数
- 分析期間のタイムスタンプ

`GetSendStatistics` オペレーションの詳細については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

このセクションでは、以下のトピックについて説明します。

- [the section called “GetSendStatistics を使用した AWS CLI API オペレーションの呼び出し” \(p. 270\)](#)
- [the section called “プログラムを使用した GetSendStatistics オペレーションの呼び出し” \(p. 270\)](#)

## GetSendStatistics を使用した AWS CLI API オペレーションの呼び出し

`GetSendStatistics` API オペレーションを最も簡単に呼び出す方法は [AWS Command Line Interface \(AWS CLI\)](#) を使用することです。

**GetSendStatistics** を使用して AWS CLI API オペレーションを呼び出すには

1. AWS CLI をインストールします (まだの場合)。詳細については、AWS Command Line Interface ユーザーガイドの「[AWS Command Line Interface のインストール](#)」を参照してください。
2. AWS セキュリティ認証情報を使用するように AWS CLI を設定します (まだの場合)。詳細については、『AWS Command Line Interface ユーザーガイド』の「[AWS CLI の設定](#)」を参照してください。
3. コマンドラインで、`aws ses get-send-statistics` と入力します

AWS CLI を正常に設定すると、送信統計が JSON 形式で一覧表示されます。JSON オブジェクトごとに 15 分間の集約された送信統計が含まれています。

## プログラムを使用した GetSendStatistics オペレーションの呼び出し

AWS SDK を使用して `GetSendStatistics` オペレーションを呼び出すこともできます。このセクションでは、AWS SDK のコード例を Go、PHP、Python および Ruby で示します。以下のいずれかのリンクを選択すると、その言語でコード例が表示されます。

- [のコード例 AWS SDK for Go \(p. 271\)](#)
- [のコード例 AWS SDK for PHP \(p. 272\)](#)
- [のコード例 AWS SDK for Python \(Boto\) \(p. 272\)](#)
- [のコード例 AWS SDK for Ruby \(p. 272\)](#)

#### Note

これらのコード例では、AWS アクセスキー ID、AWS シークレットアクセスキー、および使用する AWS リージョンを含む、AWS 共有認証情報ファイルが作成済みであるものとします。詳細については、「[」を参照してください](#)[認証情報の共有ファイルの作成 \(p. 31\)](#)

## GetSendStatistics を使用した AWS SDK for Go の呼び出し

```
package main

import (
    "fmt"

    //go get github.com/aws/aws-sdk-go/...
    "github.com/aws/aws-sdk-go/aws"
    "github.com/aws/aws-sdk-go/aws/session"
    "github.com/aws/aws-sdk-go/service/ses"
    "github.com/aws/aws-sdk-go/aws/awserr"
)

const (
    // Replace us-west-2 with the AWS Region you're using for Amazon SES.
    AwsRegion = "us-west-2"
)

func main() {

    // Create a new session and specify an AWS Region.
    sess, err := session.NewSession(&aws.Config{
        Region:aws.String(AwsRegion)},
    )

    // Create an SES client in the session.
    svc := ses.New(sess)
    input := &ses.GetSendStatisticsInput{}

    result, err := svc.GetSendStatistics(input)

    // Display error messages if they occur.
    if err != nil {
        if aerr, ok := err.(awserr.Error); ok {
            switch aerr.Code() {
            default:
                fmt.Println(aerr.Error())
            }
        } else {
            // Print the error, cast err to awserr.Error to get the Code and
            // Message from an error.
            fmt.Println(err.Error())
        }
        return
    }

    fmt.Println(result)
}
```

## GetSendStatistics を使用した AWS SDK for PHP の呼び出し

```
<?php

// Replace path_to_sdk_inclusion with the path to the SDK as described in
// http://docs.aws.amazon.com/aws-sdk-php/v3/guide/getting-started/basic-usage.html
define('REQUIRED_FILE', 'path_to_sdk_inclusion');

// Replace us-west-2 with the AWS Region you're using for Amazon SES.
define('REGION', 'us-west-2');

require REQUIRED_FILE;

use Aws\Ses\SesClient;

$client = SesClient::factory(array(
    'version' => 'latest',
    'region' => REGION
));

try {
    $result = $client->getSendStatistics([]);
    echo($result);
} catch (Exception $e) {
    echo($e->getMessage()."\n");
}

?>
```

## GetSendStatistics を使用した AWS SDK for Python (Boto) の呼び出し

```
import boto3 #pip install boto3
import json
from botocore.exceptions import ClientError

client = boto3.client('ses')

try:
    response = client.get_send_statistics(
    )
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print(json.dumps(response, indent=4, sort_keys=True, default=str))
```

## GetSendStatistics を使用した AWS SDK for Ruby の呼び出し

```
require 'aws-sdk' # gem install aws-sdk
require 'json'

# Replace us-west-2 with the AWS Region you're using for Amazon SES.
awsregion = "us-west-2"

# Create a new SES resource and specify a region
ses = Aws::SES::Client.new(region: awsregion)
```



```
begin

  resp = ses.get_send_statistics({
  })
  puts JSON.pretty_generate(resp.to_h)

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => error
  puts error
end
```

## 通知を使用した Amazon SES E メール送信のモニタリング

Amazon SES を使用して電子メールを送信するには、バウンスと苦情を管理するためのシステムが必要です。Amazon SES は、通知メールを送信する、Amazon SNS トピックを通知する、送信イベントを公開する、の 3 つのいずれかの方法でバウンスや苦情イベントを通知します。このセクションでは、E メールまたは Amazon SNS トピックに通知することによって、特定の種類の通知を送信するように Amazon SES を設定する方法について説明します。送信イベントを公開する方法の詳細については、「[Amazon SES イベント発行を使用して E メール送信をモニタリングする \(p. 296\)](#)」を参照してください。

Amazon SES コンソールまたは Amazon SES API を使用して通知をセットアップできます。

### トピック

- [重要な考慮事項 \(p. 273\)](#)
- [E メールで送信された Amazon SES 通知 \(p. 274\)](#)
- [Amazon SNS 経由で送信された Amazon SES 通知 \(p. 276\)](#)

## 重要な考慮事項

通知を送信するように Amazon SES を設定する際に考慮すべきいくつかの重要な点があります。

- E メールと Amazon SNS 通知は、個々の ID (Eメールの送信に使用する検証済みの電子メールアドレスまたはドメイン) に適用されます。ID の通知を有効にすると、Amazon SES はその ID から送信された Eメールにのみ通知を送信し、通知を設定した AWS リージョンでのみ通知を送信します
- バウンスや苦情の通知を受け取る方法を有効にする必要があります。バウンスや苦情を生成したドメインや電子メールアドレス、または Amazon SNS トピックに通知を送信できます。また [イベント発行 \(p. 296\)](#) を使用して、いくつかの異なる種類のイベントに関する通知 (バウンス、苦情、配信) を Amazon SNS トピックまたは Kinesis Data Firehose ストリーミングに送信できます。

バウンスや苦情の通知を受け取るこれらの方法の 1 つを設定しない場合は、Eメールのフィードバック転送を無効にしても、Amazon SES はバウンスや苦情のイベントの原因となった Eメールの Return-Path アドレス (または Return-Path アドレスを指定しなかった場合はソースアドレス) にバウンスや苦情の通知を自動的に転送します。

Eメールのフィードバック転送を無効にし、イベント発行を有効にする場合は、送信するすべての Eメールにイベント発行ルールを含む設定セットを適用する必要があります。この状況で、設定セットを使用しない場合は、Amazon SES はバウンスや苦情のイベントの原因となった Eメールのリターンパスアドレスまたは出典アドレスにバウンスや苦情の通知を自動的に転送します。

- 1 つ以上の方法 (Eメール通知を送信する、イベントの送信を使用するなど) を使用してバウンスや苦情を送信するように Amazon SES を設定する場合は、同じイベントに対して 1 つ以上の通知を受け取る場合があります。

## E メールで送信された Amazon SES 通知

Amazon SES は、ユーザーがバウンスや苦情を受信すると、Eメールのフィードバック転送というプロセスを使用して、ユーザーにEメールを送信します。

Amazon SES を使用して E メールを送信するには、次のいずれかの方法を使用して、バウンスや苦情の通知を送信するように設定する必要があります。

- Eメールのフィードバック転送を有効にする。このタイプの通知を設定する手順は、このセクションに含まれています。
- Amazon SNS トピックに通知を送信する。詳細については、「[Amazon SNS 経由で送信された Amazon SES 通知 \(p. 276\)](#)」を参照してください。
- イベント通知を発行する。詳細については、「[Amazon SES イベント発行を使用して E メール送信をモニタリングする \(p. 296\)](#)」を参照してください。

### Important

通知についてのいくつかの重要なポイントについては、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。

### トピック

- [Eメールのフィードバック転送を有効にする \(p. 274\)](#)
- [Eメールのフィードバック転送を無効にする \(p. 275\)](#)
- [Eメールのフィードバック転送先 \(p. 275\)](#)

## Eメールのフィードバック転送を有効にする

Eメールのフィードバック転送はデフォルトで有効です。以前に無効にしている場合、このセクションの以下の手順に従って有効にできます。

Amazon SES コンソールを使用して E メールによるバウンスや苦情の転送を有効にする

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. E メールアドレスに対してバウンスや苦情の通知を設定する場合、ナビゲーションペインの [Identity Management] で、[Email Addresses] を選択します。また、ドメインに対してバウンスや苦情の通知を設定する場合、[Domains] を選択します。
3. 確認済みの E メールアドレスまたはドメインで、バウンスと苦情の通知を設定する E メールアドレスまたはドメインを選択します。
4. 詳細ペインで、[Notifications] セクションを展開します。
5. [Edit Configuration] を選択します。
6. [E メール Feedback Forwarding] で、[Enabled] を選択します。

### Note

このページで行った変更は、反映されるまでに数分かかる場合があります。

また、[SetIdentityFeedbackForwardingEnabled](#) API オペレーションを使用して、バウンスや苦情の通知を有効にできます。

## Eメールのフィードバック転送を無効にする

バウンスや苦情の通知を提供する別の方法を設定する場合は、バウンスや苦情のイベントが発生したときに複数の通知を受け取らないように、Eメールのフィードバック転送を無効にすることができます。

Amazon SES コンソールを使用して E メールを介したバウンスや苦情の転送を無効にする

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. E メールアドレスに対してバウンスや苦情の通知を設定する場合、ナビゲーションペインの [Identity Management] で、[Email Addresses] を選択します。また、ドメインに対してバウンスや苦情の通知を設定する場合、[Domains] を選択します。
3. 確認済みの E メールアドレスまたはドメインで、バウンスと苦情の通知を設定する E メールアドレスまたはドメインを選択します。
4. 詳細ペインで、[Notifications] セクションを展開します。
5. [Edit Configuration] を選択します。
6. [E メール Feedback Forwarding] で、[Disabled] を選択します。

### Note

Amazon SES を介して E メールを送信するには、バウンスや苦情の通知を受け取る 1 つの方法を設定する必要があります。Eメールのフィードバック転送を無効にする場合は、Amazon SNS によって送信される通知を有効にするか、[イベント発行 \(p. 296\)](#)を使用して、バウンスと苦情のイベントを Amazon SNS トピックまたは Kinesis Data Firehose ストリーミングに発行する必要があります。イベント発行を使用する場合は、送信する各 E メールにイベント発行ルールを含む設定セットも適用する必要があります。バウンスや苦情の通知を受け取る方法を設定しない場合は、Amazon SES はバウンスや苦情のイベントの原因となったメッセージのリターンパス フィールド (またはリターンパス アドレスを指定しなかった場合は出典フィールド) のアドレスに、Eメールによるフィードバック通知を自動的に転送します。この場合、Eメールのフィードバック通知を無効にしても、Amazon SES はバウンス通知や苦情の通知を転送します。

7. [Save Config] を選択して通知設定を保存します。

### Note

このページで行った変更は、反映されるまでに数分かかる場合があります。

また、[SetIdentityFeedbackForwardingEnabled](#) API オペレーションを使用して、バウンスや苦情の通知を無効にできます。

## Eメールのフィードバック転送先

Eメールで通知を受け取る場合、Amazon SES は From ヘッダーを書き換えて通知を送信します。Amazon SES が通知を転送するアドレスは、元のメッセージの送信方法によって異なります。

SMTP インターフェイスを使用してメッセージを送信すると、MAIL FROM コマンドで指定されたアドレスに通知が送信されます。

SendEmail API オペレーションを使用してメッセージを送信すると、通知は以下の規則に従って配信されます。

- ReturnPath API を呼び出す際にオプションの SendEmail パラメータを指定すると、通知はそのアドレスに送信されます。
- 指定しない場合、Source の必須パラメータである SendEmail で指定されたアドレスに通知が送信されます。

SendRawEmail API オペレーションを使用してメッセージを送信すると、通知は以下の規則に従って配信されます。

- Source API を呼び出す際に SendRawEmail パラメータを指定すると、通知はそのアドレスに送信されます。これは、E メール本文で Return-Path ヘッダーを指定した場合であっても同様です。
- 指定しない場合、raw メッセージで Return-Path ヘッダーを指定すると、そのアドレスに通知が送信されます。
- その他の場合、raw メッセージの From ヘッダーにおいて指定されたアドレスに通知が送信されます。

#### Note

E メールで Return-Path アドレスを指定すると、通知はそのアドレスに送信されます。ただし、受信者が受信するメッセージのバージョンには、匿名化された電子メールアドレス (a0b1c2d3e4f5a6b7-c8d9e0f1-a2b3-c4d5-e6f7-a8b9c0d1e2f3-000000@amazonses.com) を含む Return-Path ヘッダー含まれています。この匿名化は、Eメールの送信方法にかかわらず実行されます。

## Amazon SNS 経由で送信された Amazon SES 通知

バウンスや苦情を受け取ったとき、または E メールが配信されたときに、Amazon SES トピックを通知するように Amazon SNS を設定できます。Amazon SNS 通知は [JavaScript Object Notation \(JSON\)](#)形式になっており、プログラムで処理できます。

Amazon SES を使用して E メールを送信するには、次のいずれかの方法を使用して、バウンスや苦情の通知を送信するように設定する必要があります。

- Amazon SNS トピックに通知を送信する。このタイプの通知を設定する手順は、このセクションに含まれています。
- Eメールのフィードバック転送を有効にする。詳細については、「[Eメールで送信された Amazon SES 通知 \(p. 274\)](#)」を参照してください。
- イベント通知を発行する。詳細については、「[Amazon SES イベント発行を使用して Eメール送信をモニタリングする \(p. 296\)](#)」を参照してください。

#### Important

通知に関する重要な情報については、「[通知を使用した Amazon SES Eメール送信のモニタリング \(p. 273\)](#)」を参照してください。

#### トピック

- [Amazon SES の Amazon SNS 通知の設定 \(p. 276\)](#)
- [Amazon SES の Amazon SNS 通知コンテンツ \(p. 279\)](#)
- [Amazon SES の Amazon SNS 通知の例 \(p. 290\)](#)

## Amazon SES の Amazon SNS 通知の設定

[Amazon Simple Notification Service \(Amazon SNS\)](#)を通して、Amazon SES は、バウンス、苦情、配信を通知します。

通知は Amazon SES コンソールで設定するか、Amazon SES API を使用して設定できます。

このセクションのトピック:

- [Prerequisites \(p. 277\)](#)

- [Amazon SES コンソールを使用した通知の設定 \(p. 277\)](#)
- [Amazon SES API を使用した通知の設定 \(p. 278\)](#)
- [フィードバック通知のトラブルシューティング \(p. 279\)](#)

## Prerequisites

Amazon SES で Amazon SNS 通知を設定する前に、次のステップを完了します。

1. Amazon SNS トピックを作成します。詳細については、『Amazon Simple Notification Service デベロッパーガイド』の「[トピックの作成](#)」を参照してください。

### Important

Amazon SNS を使用してトピックを作成する場合、タイプはスタンダードを選択します。  
(SES は FIFO タイプのトピックをサポートしていません。)

2. 少なくとも 1 つのエンドポイントをトピックにサブスクライブします。たとえば、テキストメッセージで通知を受け取る場合は、トピックに SMS エンドポイント (携帯電話番号) をサブスクライブします。E メールで通知を受信するには、E メールエンドポイント (E メールアドレス) をトピックにサブスクライブします。

詳細については、『Amazon Simple Notification Service デベロッパーガイド』の「[作業スタート](#)」を参照してください。

3. (オプション) Amazon SNS トピックでサーバー側の暗号化に AWS Key Management Service (AWS KMS) を使用する場合は、AWS KMS キーポリシーにアクセス許可を追加する必要があります。アクセス権限を追加するには、次のポリシーを AWS KMS キーポリシーにアタッチします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSESToUseKMSKey",
      "Effect": "Allow",
      "Principal": {
        "Service": "ses.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*"
    }
  ]
}
```

## Amazon SES コンソールを使用した通知の設定

Amazon SES コンソールを使用して通知を設定するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインの [Identity Management] で、[Domains] または [Email Addresses] を選択します。
3. 確認済み送信者のリストで、通知を設定する E メールアドレスまたはドメインを選択します。

### Important

確認済みドメイン通知設定は、同様に確認済みの E メールアドレスを除き、そのドメインの E メールアドレスから送信されるすべてのメールに適用されます。

4. [Notifications] で、[Edit Configuration] を選択します。
5. [SNS トピックの設定] で、Amazon SNS トピックの設定を次のように変更します。
  - a. 通知を受け取るために使用する Amazon SNS トピックを選択します。複数のイベントタイプの通知と同じ Amazon SNS トピックに発行するか、異なる Amazon SNS トピックに発行することができます。

#### Important

バウンス、苦情、配信の通知に使用する Amazon SNS トピックは、Amazon SES を使用するのと同じ AWS リージョンに含まれている必要があります。さらに、通知を受け取るには、1 つ以上のエンドポイントをトピックにサブスクライブしている必要があります。たとえば、E メールアドレスに通知を送信する場合は、E メールエンドポイントをそのトピックにサブスクライブする必要があります。詳細については、『Amazon Simple Notification Service デベロッパーガイド』の「[作業スタート](#)」を参照してください。

自分が所有していない Amazon SNS トピックを使用する場合は、[AWS Identity and Access Management \(IAM\) ポリシーの設定](#)で、その Amazon SNS トピックの Amazon リソースネーム (ARN) からの発行を許可する必要があります。

- b. Amazon SES に転送した Eメールの元のヘッダーを Amazon SNS 通知に含める場合は、[Include original headers (元のヘッダーを含める)] を選択します。このオプションは、その通知タイプに Amazon SNS トピックを割り当てている場合にのみ使用できます。元の Eメールヘッダーの内容については、mail の [Amazon SNS 通知コンテンツ \(p. 279\)](#) オブジェクトを参照してください。
6. (オプション) バウンスと苦情の両方で Amazon SNS トピックを選択すると、Eメール通知を完全に無効にできます。バウンスや苦情の Eメール通知を無効にするには、[Email Feedback Forwarding] で [Disable] を選択します。配信の通知は、Amazon SNS からのみ利用できます。
7. [Save Config] を選択します。通知設定に対する変更は、反映されるまでに数分かかる場合があります。

設定の完了後は、バウンス、苦情、配信の通知が Amazon SNS トピックに送られるようになります。これらの通知は JavaScript Object Notation (JSON) 形式になっており、「[Amazon SNS 通知コンテンツ \(p. 279\)](#)」で説明されている構造に従っています。

バウンス、苦情、配信の通知には、Amazon SNS のスタンダードレートが課金されます。詳細については、「[Amazon SNS 料金ページ](#)」を参照してください。

#### Note

トピックが削除されたか、AWS アカウントにそのトピックに発行するアクセス許可がないために、Amazon SNS トピックへの発行が失敗した場合、Amazon SES はそのトピックの設定を削除します。さらに、Amazon SES が ID のバウンスと苦情の Eメール通知を再度有効にし、ユーザーは変更の通知を Eメールで受け取ります。トピックを使用するように複数の ID が設定されている場合、各 ID でトピックへの発行に失敗すると、各 ID のトピック設定が変更されます。

## Amazon SES API を使用した通知の設定

バウンス、苦情、配信の通知は、Amazon SES API を使用して設定することもできます。次のオペレーションを使用して通知を設定します。

- [SetIdentityNotificationTopic](#)
- [SetIdentityFeedbackForwardingEnabled](#)
- [GetIdentityNotificationAttributes](#)
- [SetIdentityHeadersInNotificationsEnabled](#)

これらの API アクションを使用して、通知用にカスタマイズしたフロントエンドアプリケーションを作成することができます。通知に関連する API アクションの詳細な説明については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

## フィードバック通知のトラブルシューティング

通知が送られてこない

通知を受け取っていない場合は、通知が送信されるトピックにエンドポイントをサブスクライブしていることを確認します。E メールエンドポイントをトピックにサブスクライブすると、サブスクリプションの確認を求める E メールが届きます。E メール通知の受信を開始するには、サブスクリプションを確認する必要があります。詳細については、『Amazon Simple Notification Service デベロッパガイド』の「[作業スタート](#)」を参照してください。

**InvalidParameterValue** トピックを選択する際にエラーが発生する

**InvalidParameterValue** エラーが発生したことを示すエラーを受け取った場合は、Amazon SNS トピックが AWS KMS を使用して暗号化されているかどうかをチェックします。AWS KMS を使って暗号化されている場合は、その AWS KMS キーのポリシーを変更する必要があります。サンプルポリシーについては、「[Prerequisites \(p. 277\)](#)」を参照してください。

## Amazon SES の Amazon SNS 通知コンテンツ

バウンス、苦情、および配信の通知は、JavaScript Object Notation (JSON) 形式で、[Amazon Simple Notification Service \(Amazon SNS\)](#) トピックに発行されます。トップレベル JSON オブジェクトには、`notificationType` 文字列と `mail` オブジェクトに加え、`bounce` オブジェクト、`complaint` オブジェクト、または `delivery` オブジェクトのいずれかが含まれます。

オブジェクトのタイプごとの詳細については以下のセクションを参照してください。

- [トップレベル JSON オブジェクト \(p. 279\)](#)
- [mail オブジェクト \(p. 280\)](#)
- [bounce オブジェクト \(p. 283\)](#)
- [complaint オブジェクト \(p. 287\)](#)
- [delivery オブジェクト \(p. 289\)](#)

以下は、Amazon SES の Amazon SNS 通知の内容に関する重要な注意事項です。

- 該当する通知タイプにより、複数の受信者に対応する 1 つの Amazon SNS 通知を受け取ることもあれば、各受信者に 1 つの Amazon SNS 通知を受け取ることもあります。コードでは Amazon SNS 通知を解析して、どちらの場合にも対応できる必要があります。Amazon SNS を使用して送信された通知に関しては、Amazon SES では順序付けや一括処理が保証されません。ただし、タイプの異なる Amazon SNS 通知 (バウンスと苦情など) が 1 つの通知にまとめられることは決してありません。
- 1 人の受信者に対して複数のタイプの Amazon SNS 通知を受け取ることがあります。たとえば、受信メールサーバーは、E メールを受信した場合でも (配信の通知をトリガーします)、そのメールの処理後に、そのメールは実際にはバウンスであると判定する場合があります (バウンスの通知をトリガーします)。ただし、通知のタイプが異なるため、これらは常に個別に通知されます。
- Amazon SES には、通知にフィールドを追加する権限があります。そのため、これらの通知を解析するアプリケーションには、不明なフィールドを処理できるだけの十分な柔軟性が必要です。
- Amazon SES では、Eメールの送信時にメッセージのヘッダーを上書きします。mail オブジェクトの `headers` および `commonHeaders` フィールドから元のメッセージのヘッダーを取得できます。

## トップレベル JSON オブジェクト

Amazon SES 通知のトップレベル JSON オブジェクトには、以下のフィールドが含まれています。

フィールド名	説明
notificationType	通知のタイプを格納する文字列は、JSON オブジェクトによって表されます。想定される値は、Bounce、Complaint、および Delivery です。
mail	通知に関連する元のメールについての情報を含む JSON オブジェクト。詳細については、「 <a href="#">Mail オブジェクト (p. 280)</a> 」を参照してください。
bounce	このフィールドは notificationType が Bounce である場合のみ存在し、バウンスに関する情報を持つ JSON オブジェクトが含まれます。詳細については、「 <a href="#">Bounce オブジェクト (p. 283)</a> 」を参照してください。
complaint	このフィールドは notificationType が Complaint である場合のみ存在し、苦情に関する情報を持つ JSON オブジェクトが含まれます。詳細については、「 <a href="#">苦情のオブジェクト (p. 287)</a> 」を参照してください。
delivery	このフィールドは notificationType が Delivery である場合のみ存在し、配信に関する情報を持つ JSON オブジェクトが含まれます。詳細については、「 <a href="#">配信オブジェクト (p. 289)</a> 」を参照してください。

## Mail オブジェクト

バウンス、苦情、または配信の通知にはそれぞれ、mail オブジェクト内の元の E メールについての情報が含まれます。mail オブジェクトについての情報を含む JSON オブジェクトには次のフィールドが含まれます。

フィールド名	説明
timestamp	元のメッセージが送信された日時 (ISO 8601 形式)。
messageId	Amazon SES がメッセージに割り当てた一意の ID。ユーザーがメッセージを送信すると、Amazon SES がこの値を返します。  Note  このメッセージ ID は Amazon SES によって割り当てられます。元の Eメールのメッセージ ID は、headers オブジェクトの commonHeaders および mail フィールドにあります。
source	元のメッセージが送信された E メールアドレス (エンベロープ MAIL FROM アドレス)。
sourceArn	Eメールの送信に使用された ID の Amazon リソースネーム (ARN)。送信認可の場合、sourceArn



フィールド名	説明
	は、代理送信者が E メール送信に使用することを ID 所有者により承認された ID の ARN です。送信承認の詳細については、「 <a href="#">送信承認の使用 (p. 154)</a> 」を参照してください。
sourceIp	Amazon SES にリクエストを送信する E メールを実行するクライアントの送信側パブリック IP アドレス。
sendingAccountId	E メール送信に使用されたアカウントの AWS アカウント ID。送信承認の場合、sendingAccountId は代理送信者のアカウント ID です。
destination	元のメールの受取人の E メールアドレスのリスト。
headersTruncated	元の E メールからヘッダーを含めるように通知設定を構成した場合にのみ、このオブジェクトが表示されます。  ヘッダーが通知で切り詰められるかどうかを示します。Amazon SES では、通知のヘッダーが 10 KB 以上のサイズである場合に、通知のヘッダーが切り詰められます。指定できる値は true および false です。
headers	元の E メールからヘッダーを含めるように通知設定を構成した場合にのみ、このオブジェクトが表示されます。  Eメールの元のヘッダーの一覧。リスト内の各ヘッダーには、name フィールドと value フィールドがあります。  Note  headers オブジェクト内のメッセージ ID は、Amazon SES に渡した元のメッセージのものです。その後 Amazon SES がメッセージに割り当てるメッセージ ID は、messageId オブジェクトの mail フィールドのものです。

フィールド名	説明
commonHeaders	<p>元の E メールからヘッダーを含めるように通知設定を構成した場合にのみ、このオブジェクトが表示されます。</p> <p>元の E メールからの一般的な E メールヘッダーに関する情報 (送信元、宛先、件名フィールドなど) が含まれます。このオブジェクト内では、各ヘッダーはキーとなります。送信元フィールドと宛先フィールドは、複数の値を含むことができる配列で表されます。</p> <p>Note</p> <p>commonHeaders オブジェクト内のメッセージ ID は、Amazon SES に渡した元のメッセージのもので、その後 Amazon SES がメッセージに割り当てるメッセージ ID は、messageId オブジェクトの mail フィールドのもので、</p>

以下は、元の E メールヘッダーを含む mail オブジェクトの例です。この通知タイプが元の E メールヘッダーを含めるように設定されていない場合は、mail オブジェクトに headersTruncated、headers および commonHeaders フィールドが含まれません。

```
{
  "timestamp": "2018-10-08T14:05:45 +0000",
  "messageId": "000001378603177f-7a5433e7-8edb-42ae-af10-f0181f34d6ee-000000",
  "source": "sender@example.com",
  "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",
  "sourceIp": "127.0.3.0",
  "sendingAccountId": "123456789012",
  "destination": [
    "recipient@example.com"
  ],
  "headersTruncated": false,
  "headers": [
    {
      "name": "From",
      "value": "\"Sender Name\" <sender@example.com>"
    },
    {
      "name": "To",
      "value": "\"Recipient Name\" <recipient@example.com>"
    },
    {
      "name": "Message-ID",
      "value": "custom-message-ID"
    },
    {
      "name": "Subject",
      "value": "Hello"
    },
    {
      "name": "Content-Type",
      "value": "text/plain; charset=UTF-8"
    },
    {
      "name": "Content-Transfer-Encoding",
      "value": "base64"
    }
  ]
}
```

```

    },
    {
      "name": "Date",
      "value": "Mon, 08 Oct 2018 14:05:45 +0000"
    }
  ],
  "commonHeaders": {
    "from": [
      "Sender Name <sender@example.com>"
    ],
    "date": "Mon, 08 Oct 2018 14:05:45 +0000",
    "to": [
      "Recipient Name <recipient@example.com>"
    ],
    "messageId": " custom-message-ID",
    "subject": "Message sent using Amazon SES"
  }
}

```

## Bounce オブジェクト

バウンスに関する情報を含む JSON オブジェクトには以下のフィールドがあります。

フィールド名	説明
bounceType	バウンスのタイプで、Amazon SES によって決定されます。詳細については、「 <a href="#">バウンスのタイプ (p. 285)</a> 」を参照してください。
bounceSubType	バウンスのサブタイプで、Amazon SES によって決定されます。詳細については、「 <a href="#">バウンスのタイプ (p. 285)</a> 」を参照してください。
bouncedRecipients	バウンスとなった元のメールの受取人についての情報を含むリスト。詳細については、「 <a href="#">バウンスとなった受取人 (p. 284)</a> 」を参照してください。
timestamp	バウンスが送信された日時 (ISO 8601 形式)。この時刻は、ISP によって通知が送信された時刻であり、Amazon SES が通知を受け取った時刻ではないことに注意してください。
feedbackId	バウンスの一意の ID。

Amazon SES がリモートの Message Transfer Authority (MTA) に接続できた場合は、次のフィールドも表示されます。

フィールド名	説明
remoteMtaIp	Amazon SES で E メール配信を試みた先の MTA の IP アドレス。

配信状態通知 (DSN) がバウンスに添付されている場合は、次のフィールドも表示されます。

フィールド名	説明
reportingMTA	DSN の Reporting-MTA フィールドの値。これは、DSN に記述された配信、リレー、またはゲートウェイのオペレーションを試みた MTA の値です。

以下は、bounce オブジェクトの例です。

```
{
  "bounceType": "Permanent",
  "bounceSubType": "General",
  "bouncedRecipients": [
    {
      "status": "5.0.0",
      "action": "failed",
      "diagnosticCode": "smtp; 550 user unknown",
      "emailAddress": "recipient1@example.com"
    },
    {
      "status": "4.0.0",
      "action": "delayed",
      "emailAddress": "recipient2@example.com"
    }
  ],
  "reportingMTA": "example.com",
  "timestamp": "2012-05-25T14:59:38.605Z",
  "feedbackId": "000001378603176d-5a4b5ad9-6f30-4198-a8c3-b1eb0c270a1d-000000",
  "remoteMtaIp": "127.0.2.0"
}
```

### バウンスとなった受取人

バウンスの通知には、1 人の受信者に関するものと複数の受信者に関するものがあります。bouncedRecipients フィールドはオブジェクトのリスト (バウンスの通知が関係する受取人ごとに 1 つのオブジェクト) を保持し、常に次のフィールドが含まれます。

フィールド名	説明
emailAddress	受取人の E メールアドレス。DSN が利用できる場合、これが DSN の Final-Recipient フィールドの値です。

オプションで、DSN がバウンスに添付されている場合、以下のフィールドも表示される場合があります。

フィールド名	説明
action	DSN の Action フィールドの値。このフィールドには、Reporting-MTA により実行された、この受取人に対してメッセージを送信しようとした結果のアクションが示されます。
status	DSN の Status フィールドの値。これは、メッセージの配信状態を示す、受取人ごとに個別の、トランスポート独立型ステータスコードです。

フィールド名	説明
diagnosticCode	ステータスコードは、Reporting-MTA により発行されます。これは、DSN の Diagnostic-Code フィールドの値です。このフィールドが DSN に存在しない場合もあります (その場合 JSON オブジェクトにも表示されません)。

以下は、bouncedRecipients のリストに示されるオブジェクトの例です。

```
{
  "emailAddress": "recipient@example.com",
  "action": "failed",
  "status": "5.0.0",
  "diagnosticCode": "X-Postfix; unknown user"
}
```

### バウンスのタイプ

バウンスオブジェクトには、バウンスタイプとして Undetermined、Permanent、または Transient が含まれます。Permanent バウンスタイプと Transient バウンスタイプには、いくつかあるバウンスサブタイプの1つが含まれます。

バウンスタイプが Transient のバウンス通知を受信した場合は、メッセージのバウンスを起こした問題が解決されたときに、この受取人に対して将来 E メールを送信できる可能性があります。

バウンスタイプが Permanent のバウンス通知を受信した場合、この受取人に将来 E メールを送信できる可能性はありません。このため、バウンスを生じたアドレスを持つ受取人はメーリングリストから即座に削除してください。

#### Note

ソフトバウンス (受信者の受信トレイが満杯であるなどの一時的な問題に伴うバウンス) が発生すると、Amazon SES は一定期間にわたり、Eメールの再配信を試行します。この期間の終了時に、まだ E メールを送信できない場合、Amazon SES は試行を停止します。

Amazon SES は、ハードバウンスの通知に加えて、配信の試行を停止したソフトバウンスの通知を提供します。ソフトバウンスが発生するたびに通知を受信する場合は、[イベントの公開を有効 \(p. 298\)](#)にし、配信遅延イベントが発生したときに通知を送信するように設定します。

bounceType	bounceSubType	説明
Undetermined	Undetermined	受取人の E メールプロバイダーはバウンスメッセージを送信しました。バウンスメッセージには、Amazon SES がバウンスの理由を判断できるだけの十分な情報が含まれていませんでした。バウンスを生じた Eメールのリターンパスヘッダーのアドレスに送信されたバウンス Eメールには、Eメールのバウンスを起こした問題について追加情報が含まれている可能性があります。
Permanent	General	受取人の E メールプロバイダーは、ハードバウンスメッセージを送信しましたが、ハードバウンスの原因を示していません。  <b>Important</b>  このタイプのバウンス通知を受信した場合は、受取人の E メールアドレスをメー

bounceType	bounceSubType	説明
		<p>リングリストから即座に削除してください。ハードバウンスを生じたアドレスにメッセージを送信すると、送信者としての評価に悪影響を及ぼす可能性があります。ハードバウンスを生じたアドレスに E メールを送信し続けると、追加の Eメールの送信機能が一時停止される場合があります。</p>
Permanent	NoEmail	<p>対象の受取人の E メールプロバイダーから、E メールアドレスが存在しないというバウンスメッセージが送信されました。</p> <p><b>Important</b></p> <p>このタイプのバウンス通知を受信した場合は、受取人の E メールアドレスをメーリングリストから即座に削除してください。存在しないアドレスにメッセージを送信すると、送信者としての評価に悪影響を及ぼす可能性があります。存在しないアドレスに E メールを送信し続けると、追加の Eメールの送信機能が一時停止される場合があります。</p>
Permanent	Suppressed	<p>受信者の E メールアドレスは、最近の履歴でハードバウンスを生じているため、Amazon SES サプレッションリストに追加されています。Amazon SES サプレッションリストからアドレスを削除する方法については、「<a href="#">Amazon SES グローバルサプレッションリストの使用 (p. 199)</a>」を参照してください。</p>
Permanent	OnAccountSuppressionList	<p>Amazon SES は、アドレスが<a href="#">アカウントレベルのサプレッションリスト (p. 190)</a>にあるので、このアドレスへの送信を抑制しました。</p>
Transient	General	<p>受取人の E メールプロバイダーは一般的なバウンスメッセージを送信しました。メッセージのバウンスを生じた問題が解決された場合、将来、同じ受取人にメッセージを送信できる可能性があります。</p> <p><b>Note</b></p> <p>アクティブな自動応答ルール (不在メッセージなど) が設定されている受取人に E メールを送信すると、このタイプの通知を受け取る場合があります。レスポンスの通知タイプが Bounce であっても、Amazon SES は自動応答をカウントすることなくアカウントのバウンス率を計算します。</p>

bounceType	bounceSubType	説明
Transient	MailboxFull	受取人の E メールプロバイダーは、受取人の受信トレイが満杯であるために、バウンスメッセージを送信しました。メールボックスが満杯でなくなった場合、将来、同じ受取人に送信できる可能性があります。
Transient	MessageTooLarge	受取人の E メールプロバイダーは、受信したメッセージが大きすぎるために、バウンスメッセージを送信しました。メッセージのサイズを小さくすることで、同じ受取人にメッセージを送信できる可能性があります。
Transient	ContentRejected	受取人の E メールプロバイダーは、受信したメッセージにプロバイダーが許可しないコンテンツが含まれていたために、バウンスメッセージを送信しました。メッセージのコンテンツを変更することで、同じ受取人にメッセージを送信できる可能性があります。
Transient	AttachmentRejected	受取人の E メールプロバイダーは、メッセージ内に許容されないコンテンツが含まれていたために、バウンスメッセージを送信しました。たとえば、一部の E メールプロバイダーは特定のファイルタイプのファイルが添付されたメッセージや、非常に大きなファイルが添付されたメッセージを拒否する場合があります。添付ファイルのコンテンツを削除または変更することで、同じ受取人にメッセージを送信できる可能性があります。

## 苦情のオブジェクト

苦情に関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
complainedRecipients	苦情の原因である可能性がある受取人についての情報を含むリスト。詳細については、「 <a href="#">苦情を申告した受取人 (p. 288)</a> 」を参照してください。
timestamp	ISP が苦情通知を ISO 8601 形式で送信した日時。このフィールドの日は、Amazon SES が通知を受信した日時とは同じでない可能性があります。
feedbackId	苦情に関連付けられた一意の ID。
complaintSubType	complaintSubType フィールドの値は、null または OnAccountSuppressionList のいずれかになります。値が OnAccountSuppressionList の場合、Amazon SES はメッセージを受け入れますが、 <a href="#">アカウントレベルのサブレッションリスト (p. 190)</a> に含まれているため、送信を試みません。

また、フィードバックレポートが苦情に添付されている場合、以下のフィールドが示される場合があります。

フィールド名	説明
userAgent	フィードバックレポートの User-Agent フィールドの値。これは、レポートを生成したシステムの名前とバージョンを示します。
complaintFeedbackType	ISP から受け取ったフィードバックレポートの Feedback-Type フィールドの値。これには、フィードバックのタイプが含まれます。
arrivalDate	ISO 8601 形式のフィードバックレポートの Arrival-Date フィールドまたは Received-Date フィールドの値。このフィールドがレポートにない場合もあります (その場合、JSON オブジェクトにも表示されません)。

以下は、complaint オブジェクトの例です。

```
{
  "userAgent": "ExampleCorp Feedback Loop (V0.01)",
  "complainedRecipients": [
    {
      "emailAddress": "recipient1@example.com"
    }
  ],
  "complaintFeedbackType": "abuse",
  "arrivalDate": "2009-12-03T04:24:21.000-05:00",
  "timestamp": "2012-05-25T14:59:38.623Z",
  "feedbackId": "000001378603177f-18c07c78-fa81-4a58-9dd1-fedc3cb8f49a-000000"
}
```

### 苦情を申告した受取人

complainedRecipients フィールドには、苦情の送信元と思われる受信者のリストが含まれます。この情報を使用して、苦情の送信元の受取人を特定し、この受取人をメーリングリストから即座に削除します。

#### Important

ほとんどの ISP は、苦情の送信元である受取人の E メールアドレスを苦情通知から削除します。このため、このリストに含まれている苦情の送信元と思われる受信者に関する情報は、元のメッセージの受信者と、苦情の送信元の ISP に基づくものです。Amazon SES では、この受信者のリストを確認するために元のメッセージを参照します。

このリストの JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
emailAddress	受取人の E メールアドレス。

以下は、苦情を申告した受取人のオブジェクトの例です。

```
{ "emailAddress": "recipient1@example.com" }
```



## Note

受信者 1 人につき 1 つのメッセージが送信されるように制限している (BCC 行に 30 個の異なる E メールアドレスを指定して送信しない) 場合は、この動作を利用して、メッセージに関する苦情を伝えている E メールアドレスをより確実に特定することができます。

## 苦情のタイプ

complaintFeedbackType フィールドには以下の苦情のタイプが示されます ([Internet Assigned Numbers Authority ウェブサイト](#) に基づいて、報告する ISP により割り当てられます)。

- abuse — 未承諾またはその他種類の不正使用メールを示します。
- auth-failure — E メール認証の障害を示します。
- fraud — なんらかの詐欺またはフィッシング行為を示します。
- not-spam — レポートの提供者がこのメッセージをスパムではないと見なしていることを示します。このタイプは、誤ってスパムとしてタグ付けまたは分類されたメッセージを修正するために使用される場合があります。
- other — その他の登録されたタイプに該当しないフィードバックを示します。
- virus — 元のメッセージでウイルスが見つかったことを示します。

## 配信オブジェクト

配信に関する情報を含む JSON オブジェクトには常に以下のフィールドがあります。

フィールド名	説明
timestamp	Amazon SES が E メールを受信者のメールサーバーに配信した時間 (ISO8601 形式)。
processingTimeMillis	Amazon SES で送信者からのリクエストを受理したときから受信者のメールサーバーにそのメッセージが渡されたときまでのミリ秒単位での時間。
recipients	配信通知を適用する Eメールの対象となる受信者のリスト。
smtpResponse	Amazon SES から Eメールを受け取ったりリモート ISP による SMTP 応答メッセージ。このメッセージは、Eメール、受信メールサーバー、および受信 ISP ごとに異なります。
reportingMTA	メールを送信する Amazon SES メールサーバーのホスト名。
remoteMtaIp	Amazon SES が Eメールを配信した先の MTA の IP アドレス。

以下は、delivery オブジェクトの例です。

```
{
  "timestamp": "2014-05-28T22:41:01.184Z",
  "processingTimeMillis": 546,
  "recipients": ["success@simulator.amazonses.com"],
  "smtpResponse": "250 ok: Message 64111812 accepted",
  "reportingMTA": "a8-70.smtp-out.amazonses.com",
  "remoteMtaIp": "127.0.2.0"
}
```

```
}
```

## Amazon SES の Amazon SNS 通知の例

以下のセクションでは、3 種類の通知の例を紹介します。

- バウンス通知の例については、「[Amazon SNS バウンス通知の例 \(p. 290\)](#)」を参照してください。
- 苦情通知の例については、「[Amazon SNS 苦情通知の例 \(p. 292\)](#)」を参照してください。
- 配信通知の例については、「[Amazon SNS 配信通知の例 \(p. 295\)](#)」を参照してください。

### Amazon SNS バウンス通知の例

このセクションには、フィードバックを送信した E メール受取人によって提供される配信ステータス通知 (DSN) を含むバウンス通知と含まないバウンス通知の例が記載されています。

#### DSN を含むバウンス通知

以下は、DSN および元の E メールヘッダーを含むバウンス通知の例です。バウンス通知が元の E メールヘッダーを含めるように設定されていない場合は、通知内の `mail` オブジェクトに `headersTruncated`、`headers` および `commonHeaders` フィールドが含まれません。

```
{
  "notificationType": "Bounce",
  "bounce": {
    "bounceType": "Permanent",
    "reportingMTA": "dns; email.example.com",
    "bouncedRecipients": [
      {
        "emailAddress": "jane@example.com",
        "status": "5.1.1",
        "action": "failed",
        "diagnosticCode": "smtp; 550 5.1.1 <jane@example.com>... User"
      }
    ],
    "bounceSubType": "General",
    "timestamp": "2016-01-27T14:59:38.237Z",
    "feedbackId": "00000138111222aa-33322211-cccc-cccc-cccc-ddddaaaa068a-000000",
    "remoteMtaIp": "127.0.2.0"
  },
  "mail": {
    "timestamp": "2016-01-27T14:59:38.237Z",
    "source": "john@example.com",
    "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",
    "sourceIp": "127.0.3.0",
    "sendingAccountId": "123456789012",
    "messageId": "00000138111222aa-33322211-cccc-cccc-cccc-ddddaaaa0680-000000",
    "destination": [
      "jane@example.com",
      "mary@example.com",
      "richard@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "\"John Doe\" <john@example.com>"
      },
      {
        "name": "To",
        "value": "\"Jane Doe\" <jane@example.com>, \"Mary Doe\" <mary@example.com>, \"Richard Doe\" <richard@example.com>"
      }
    ]
  }
}
```

```
{
  "name": "Message-ID",
  "value": "custom-message-ID"
},
{
  "name": "Subject",
  "value": "Hello"
},
{
  "name": "Content-Type",
  "value": "text/plain; charset=UTF-8"
},
{
  "name": "Content-Transfer-Encoding",
  "value": "base64"
},
{
  "name": "Date",
  "value": "Wed, 27 Jan 2016 14:05:45 +0000"
}
],
"commonHeaders": {
  "from": [
    "John Doe <john@example.com>"
  ],
  "date": "Wed, 27 Jan 2016 14:05:45 +0000",
  "to": [
    "Jane Doe <jane@example.com>, Mary Doe <mary@example.com>, Richard Doe <richard@example.com>"
  ],
  "messageId": "custom-message-ID",
  "subject": "Hello"
}
}
```

### DSN を含まないバウンス通知

以下は、元の E メールヘッダーを含むが DSN を含まないバウンス通知の例です。バウンス通知が元の E メールヘッダーを含めるように設定されていない場合は、通知内の mail オブジェクトに headersTruncated、headers および commonHeaders フィールドが含まれません。

```
{
  "notificationType": "Bounce",
  "bounce": {
    "bounceType": "Permanent",
    "bounceSubType": "General",
    "bouncedRecipients": [
      {
        "emailAddress": "jane@example.com"
      },
      {
        "emailAddress": "richard@example.com"
      }
    ],
    "timestamp": "2016-01-27T14:59:38.237Z",
    "feedbackId": "00000137860315fd-869464a4-8680-4114-98d3-716fe35851f9-000000",
    "remoteMtaIp": "127.0.2.0"
  },
  "mail": {
    "timestamp": "2016-01-27T14:59:38.237Z",
    "messageId": "00000137860315fd-34208509-5b74-41f3-95c5-22c1edc3c924-000000",
    "source": "john@example.com",
    "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",
  }
}
```

```
"sourceIp": "127.0.3.0",
"sendingAccountId": "123456789012",
"destination": [
  "jane@example.com",
  "mary@example.com",
  "richard@example.com"
],
"headersTruncated": false,
"headers": [
  {
    "name": "From",
    "value": "\"John Doe\" <john@example.com>"
  },
  {
    "name": "To",
    "value": "\"Jane Doe\" <jane@example.com>, \"Mary Doe\" <mary@example.com>,
\"Richard Doe\" <richard@example.com>"
  },
  {
    "name": "Message-ID",
    "value": "custom-message-ID"
  },
  {
    "name": "Subject",
    "value": "Hello"
  },
  {
    "name": "Content-Type",
    "value": "text/plain; charset=UTF-8"
  },
  {
    "name": "Content-Transfer-Encoding",
    "value": "base64"
  },
  {
    "name": "Date",
    "value": "Wed, 27 Jan 2016 14:05:45 +0000"
  }
],
"commonHeaders": {
  "from": [
    "John Doe <john@example.com>"
  ],
  "date": "Wed, 27 Jan 2016 14:05:45 +0000",
  "to": [
    "Jane Doe <jane@example.com>, Mary Doe <mary@example.com>, Richard Doe
<richard@example.com>"
  ],
  "messageId": "custom-message-ID",
  "subject": "Hello"
}
}
```

## Amazon SNS 苦情通知の例

このセクションには、フィードバックを送信した E メール受取人によって提供されるフィードバックレポートを含む苦情通知と含まない苦情通知の例が記載されています。

### フィードバックレポートを含む苦情通知

以下は、フィードバックレポートおよび元の E メールヘッダーを含む苦情通知の例です。苦情通知が元の E メールヘッダーを含めるように設定されていない場合は、通知内の mail オブジェクトに headersTruncated、headers および commonHeaders フィールドが含まれません。

```
{
  "notificationType": "Complaint",
  "complaint": {
    "userAgent": "AnyCompany Feedback Loop (V0.01)",
    "complainedRecipients": [
      {
        "emailAddress": "richard@example.com"
      }
    ],
    "complaintFeedbackType": "abuse",
    "arrivalDate": "2016-01-27T14:59:38.237Z",
    "timestamp": "2016-01-27T14:59:38.237Z",
    "feedbackId": "000001378603177f-18c07c78-fa81-4a58-9dd1-fedc3cb8f49a-000000"
  },
  "mail": {
    "timestamp": "2016-01-27T14:59:38.237Z",
    "messageId": "000001378603177f-7a5433e7-8edb-42ae-af10-f0181f34d6ee-000000",
    "source": "john@example.com",
    "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",
    "sourceIp": "127.0.3.0",
    "sendingAccountId": "123456789012",
    "destination": [
      "jane@example.com",
      "mary@example.com",
      "richard@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "\"John Doe\" <john@example.com>"
      },
      {
        "name": "To",
        "value": "\"Jane Doe\" <jane@example.com>, \"Mary Doe\" <mary@example.com>, \"Richard Doe\" <richard@example.com>"
      },
      {
        "name": "Message-ID",
        "value": "custom-message-ID"
      },
      {
        "name": "Subject",
        "value": "Hello"
      },
      {
        "name": "Content-Type",
        "value": "text/plain; charset=UTF-8"
      },
      {
        "name": "Content-Transfer-Encoding",
        "value": "base64"
      },
      {
        "name": "Date",
        "value": "Wed, 27 Jan 2016 14:05:45 +0000"
      }
    ],
    "commonHeaders": {
      "from": [
        "John Doe <john@example.com>"
      ],
      "date": "Wed, 27 Jan 2016 14:05:45 +0000",
      "to": [

```

```
    "Jane Doe <jane@example.com>, Mary Doe <mary@example.com>, Richard Doe  
<richard@example.com>"  
  ],  
  "messageId": "custom-message-ID",  
  "subject": "Hello"  
}  
}  
}
```

### フィードバックレポートを含まない苦情通知

以下は、元の E メールヘッダーを含むがフィードバックレポートを含まない苦情通知の例です。苦情通知が元の E メールヘッダーを含めるように設定されていない場合は、通知内の mail オブジェクトに headersTruncated、headers および commonHeaders フィールドが含まれません。

```
{  
  "notificationType": "Complaint",  
  "complaint": {  
    "complainedRecipients": [  
      {  
        "emailAddress": "richard@example.com"  
      }  
    ],  
    "timestamp": "2016-01-27T14:59:38.237Z",  
    "feedbackId": "0000013786031775-fea503bc-7497-49e1-881b-a0379bb037d3-000000"  
  },  
  "mail": {  
    "timestamp": "2016-01-27T14:59:38.237Z",  
    "messageId": "0000013786031775-163e3910-53eb-4c8e-a04a-f29debf88a84-000000",  
    "source": "john@example.com",  
    "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",  
    "sourceIp": "127.0.3.0",  
    "sendingAccountId": "123456789012",  
    "destination": [  
      "jane@example.com",  
      "mary@example.com",  
      "richard@example.com"  
    ],  
    "headersTruncated": false,  
    "headers": [  
      {  
        "name": "From",  
        "value": "\"John Doe\" <john@example.com>"  
      },  
      {  
        "name": "To",  
        "value": "\"Jane Doe\" <jane@example.com>, \"Mary Doe\" <mary@example.com>,  
\"Richard Doe\" <richard@example.com>"  
      },  
      {  
        "name": "Message-ID",  
        "value": "custom-message-ID"  
      },  
      {  
        "name": "Subject",  
        "value": "Hello"  
      },  
      {  
        "name": "Content-Type",  
        "value": "text/plain; charset=UTF-8"  
      },  
      {  
        "name": "Content-Transfer-Encoding",  
        "value": "base64"  
      }  
    ]  
  }  
}
```

```
    },
    {
      "name": "Date",
      "value": "Wed, 27 Jan 2016 14:05:45 +0000"
    }
  ],
  "commonHeaders": {
    "from": [
      "John Doe <john@example.com>"
    ],
    "date": "Wed, 27 Jan 2016 14:05:45 +0000",
    "to": [
      "Jane Doe <jane@example.com>, Mary Doe <mary@example.com>, Richard Doe <richard@example.com>"
    ],
    "messageId": "custom-message-ID",
    "subject": "Hello"
  }
}
}
```

## Amazon SNS 配信通知の例

以下は、元の E メールヘッダーを含む配信通知の例です。配信通知が元の E メールヘッダーを含めるように設定されていない場合は、通知内の mail オブジェクトに headersTruncated、headers および commonHeaders フィールドが含まれません。

```
{
  "notificationType": "Delivery",
  "mail": {
    "timestamp": "2016-01-27T14:59:38.237Z",
    "messageId": "0000014644fe5ef6-9a483358-9170-4cb4-a269-f5dcdf415321-000000",
    "source": "john@example.com",
    "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",
    "sourceIp": "127.0.3.0",
    "sendingAccountId": "123456789012",
    "destination": [
      "jane@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "\"John Doe\" <john@example.com>"
      },
      {
        "name": "To",
        "value": "\"Jane Doe\" <jane@example.com>"
      },
      {
        "name": "Message-ID",
        "value": "custom-message-ID"
      },
      {
        "name": "Subject",
        "value": "Hello"
      },
      {
        "name": "Content-Type",
        "value": "text/plain; charset=UTF-8"
      },
      {
        "name": "Content-Transfer-Encoding",
        "value": "base64"
      }
    ]
  }
}
```

```
    },
    {
      "name": "Date",
      "value": "Wed, 27 Jan 2016 14:58:45 +0000"
    }
  ],
  "commonHeaders": {
    "from": [
      "John Doe <john@example.com>"
    ],
    "date": "Wed, 27 Jan 2016 14:58:45 +0000",
    "to": [
      "Jane Doe <jane@example.com>"
    ],
    "messageId": "custom-message-ID",
    "subject": "Hello"
  }
},
"delivery": {
  "timestamp": "2016-01-27T14:59:38.237Z",
  "recipients": ["jane@example.com"],
  "processingTimeMillis": 546,
  "reportingMTA": "a8-70.smtp-out.amazonses.com",
  "smtpResponse": "250 ok: Message 64111812 accepted",
  "remoteMtaIp": "127.0.2.0"
}
}
```

## Amazon SES イベント発行を使用して E メール送信をモニタリングする

メール送信をきめ細かく追跡できるようにするには、Amazon SES を設定して、お客様が定義した特性に基づいて、E メール送信イベントを、Amazon CloudWatch、Amazon Kinesis Data Firehose、または Amazon Simple Notification Service に発行します。

送信、配信、オープン、クリック、バウンス、苦情、拒否、レンダリングの失敗、配信遅延など、さまざまな種類の E メール送信イベントを追跡できます。この情報は、運用と分析の目的で役立ちます。例えば、E メール送信データを CloudWatch に公開し、E メールキャンペーンのパフォーマンスを追跡するダッシュボードを作成できます。または、Amazon SNS を使用して、特定のイベントが発生したときに通知を送信できます。

### イベント発行のしくみ

イベント発行を使用するには、まず 1 つ以上の設定セットをセットアップします。設定セットは、イベントの発行先の場所と発行するイベントを指定します。その後、メールを送信するたびに、設定セットの名前と 1 つ以上のメッセージタグを名前と値のペアの形式で指定し、メールを分類します。たとえば、書籍を宣伝する場合、関連するキャンペーンのメールを送信するときに、メッセージタグに genre という名前を付け、値 sci-fi または western を割り当てることができます。使用する E メール送信インターフェイスに応じて、API コールへのパラメータとして、または Amazon SES 固有の E メールヘッダーとして、メッセージタグを指定します。設定セットの詳細については、「[Amazon SES 設定セットの使用 \(p. 257\)](#)」を参照してください。

ユーザーが指定するメッセージタグに加え、送信するメッセージには Amazon SES によって自動タグも追加されます。自動タグを使用するために追加のステップを実行する必要はありません。

次の表は、Amazon SES を使用して送信するメールに自動的に適用される自動タグの一覧です。



## Amazon SES の自動タグ

自動タグ名	説明
ses:configuration-set	E メールに関連付けられた設定セットの名前。
ses:caller-identity	E メールを送信した Amazon SES ユーザーの IAM ID。
ses:from-domain	「From」アドレスのドメイン。
ses:source-ip	呼び出し元がメールの送信に使用した IP アドレス。
ses:outgoing-ip	Amazon SES で Eメールの送信に使用した IP アドレス。

## イベント発行を使用する方法

以下のセクションには、Amazon SES イベント発行のセットアップと使用に必要な情報が含まれています。

- [イベント発行のセットアップ \(p. 298\)](#)
- [イベントデータの使用 \(p. 307\)](#)
- [チュートリアル \(p. 348\)](#)

## イベント発行の用語

次のリストでは、Amazon SES イベント発行に関連する条件を定義します。

### メール送信イベント

Amazon SES に送信したメールの結果に関連付けられた情報。以下に示しているのは、送信イベントです。

- **送信** – Amazon SES に対する呼び出しが成功しました。Amazon SES はメールの配信を試行します。
- **拒否** – Amazon SES は E メールを受け取りましたが、ウイルスが含まれていると判断して拒否しました。Amazon SES は、受信者のメールサーバーに E メールを配信しようとしませんでした。
- **バウンス** – 受信者のメールサーバーにより、メールが完全に拒否されました。このイベントはハードバウンスに該当します。ソフトバウンスは、Amazon SES が一定期間にわたって再試行してもメールを配信できなかった場合に限り発生します。
- **苦情** – E メールは受信者に正常に配信されました。受取人がメールをスパムとしてマークしました。
- **配信** – Amazon SES が受信者のメールサーバーに E メールを正常に送信しました。
- **オープン** – 受信者がメッセージを受け取り、E メールクライアントで開きました。
- **クリック** – 受信者はメール内の 1 つ以上のリンクをクリックしました。
- **レンダリング失敗** – テンプレートレンダリングの問題により E メールが送信されませんでした。このイベントタイプは、[SendTemplatedEmail](#) または [SendBulkTemplatedEmail](#) API オペレーションを使用してテンプレートメールを送信する場合にのみ発生します。このイベントタイプは、テンプレートデータが見つからない場合や、テンプレートのパラメータとデータが一致しない場合に発生します。
- **配信の遅延** – 一時的な問題が発生したため、メールを受信者に配信できませんでした。配信の遅延は、受信者の受信トレイがいっぱいになった場合や、受信側の電子メールサーバーで一時的な問題が発生した場合などに発生します。

## 設定セット

Amazon SES が発行する E メール送信イベントの送信先と、発行する E メール送信イベントのタイプを定義する一連のルール。イベント発行で使用するメールを送信するとき、メールに関連付ける設定セットを指定します。

### イベントの発行先

Amazon SES E メール送信イベントを発行する AWS のサービス。セットアップする各イベントの宛先は、設定セット 1 つだけに帰属しています。

### メッセージタグ

イベント発行の目的でメールの分類に使用する名前と値のペア。たとえば、campaign/book や campaign/clothing などです。メールを送信するときに、API コールのパラメータとして、または Amazon SES 固有のメールヘッダーとして、メッセージタグを指定します。

### 自動タグ

イベント発行レポートに自動的に含まれるメッセージタグ。自動タグとして、設定セット名、「From」アドレスのドメイン、発信者の送信 IP アドレス、Amazon SES の送信 IP アドレス、および発信者の IAM ID があります。

## Amazon SES でのイベント発行のセットアップ

このセクションでは、Amazon SES を設定して、Amazon CloudWatch、Amazon Kinesis Data Firehose、または Amazon Simple Notification Service (Amazon SNS) に E メール送信イベントを発行するために必要な操作について説明します。

最初に、Amazon SES コンソールまたは API を使用して、作成する設定セットを作成します。設定セットを作成したら、1 つ以上のイベント送信先 (CloudWatch、Kinesis Data Firehose、または Amazon SNS) を設定セットに追加し、イベント送信先に一意のパラメータを設定します。次に、E メールを送信するたびに、使用する設定セットを指定します。

### このセクションのトピック

- [ステップ 1: 設定セットを作成する \(p. 298\)](#)
- [ステップ 2: イベント送信先を追加する \(p. 299\)](#)
- [ステップ 3: E メールを送信するときに設定セットを指定する \(p. 305\)](#)

## ステップ 1: 設定セットを作成する

設定セットを使用して、E メール送信イベント (バウンス、苦情、配信、送信済み E メール、拒否された E メール、レンダリングの失敗、および配信遅延) を Amazon CloudWatch または Amazon Kinesis Data Firehose に発行できます。

また、Amazon SES コンソールまたは Amazon SES API V2 の [CreateConfigurationSet](#) オペレーションを使用して、設定セットを作成できます。

Amazon SES コンソールを使用して設定セットを作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで [設定セット] を選択します。
3. コンテンツペインで、[設定セットを作成] を選択します。
4. 設定セットの名前を入力し、[設定セットを作成] を選択します。
5. [終了] を選択します。

## ステップ 2: イベント送信先を追加する

イベントの送信先は、Amazon SES イベントを発行する場所です。セットアップする各イベントの宛先は、設定セット 1 つだけに帰属しています。Amazon SES を使用してイベント送信先をセットアップする場合、発行先の AWS のサービスを選択し、その発行先に関連付けられたパラメータを指定します。

イベント送信先を設定する場合、Amazon CloudWatch、Amazon Kinesis Data Firehose、および Amazon Simple Notification Service (Amazon SNS) のいずれかの AWS のサービスにイベントを送信することを選択できます。どのイベント送信先を選択するかは、イベントに関して必要な詳細のレベルとイベント情報を受け取る方法によって異なります。各タイプのイベントの合計数のみ必要な場合は (たとえば、合計数が高くなりすぎた場合にアラームを設定できるようにするなど)、CloudWatch を使用します。分析のために Amazon Elasticsearch Service や Amazon Redshift などの別のサービスに出力できる詳細なイベントレコードが欲しい場合は、Kinesis Data Firehose を使用できます。特定のイベントが発生したときに通知を受け取る場合は、Amazon SNS を選択します。

このセクションは、以下のトピックで構成されます。

- [イベント発行の CloudWatch Event 送信先のセットアップ \(p. 299\)](#)
- [Amazon SES イベント公開用の Kinesis Data Firehose イベント送信先を設定する \(p. 301\)](#)
- [イベント発行の Amazon SNS イベント送信先のセットアップ \(p. 304\)](#)

### イベント発行の CloudWatch Event 送信先のセットアップ

Amazon CloudWatch Event 送信先を使用して、Amazon SES E メール送信イベントを CloudWatch に発行することができます。CloudWatch Event 送信先は設定セット内にのみ存在するため、まず [設定セットを作成 \(p. 298\)](#) し、次にその設定セットにイベント送信先を追加する必要があります。

CloudWatch Event 送信先を設定セットに追加したら、Eメールの送信時に使用するメッセージタグに対応する CloudWatch デイメンションを 1 つ以上選択する必要があります。メッセージタグと同様、CloudWatch デイメンションは、メトリクスを一意に識別できるようにする名前と値のペアです。

たとえば、メッセージタグと、メールキャンペーンの識別に使用する `campaign` と呼ばれるデイメンションを選択できます。メール送信イベントを CloudWatch に発行するとき、メッセージタグとデイメンションを選択することが重要です。その選択内容によって、CloudWatch の請求に影響が及び、CloudWatch でメール送信イベントデータのフィルタ方法が決まるからです。

このセクションでは、デイメンションの選択に役立つ情報を示し、CloudWatch Event 送信先を設定セットに追加する方法を示します。

このセクションのトピック

- [CloudWatch Event 送信先の追加 \(p. 299\)](#)
- [CloudWatch デイメンションの選択 \(p. 301\)](#)

### CloudWatch Event 送信先の追加

このセクションの手順では、作成した設定セットに CloudWatch Event 送信先を追加する方法を示します。

また、Amazon SES API V2 で、[UpdateConfigurationSetEventDestination](#) オペレーションを使用して、イベントの送信先を作成および変更します。

コンソールを使用して CloudWatch Event 送信先を設定セットに追加するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。

2. ナビゲーションペインで [設定セット] を選択します。
3. 設定セットのリストで、CloudWatch Event の送信先を作成する設定セットを選択します。リストが空の場合、まず [設定セットを作成 \(p. 298\)](#) する必要があります。
4. [イベント送信先] タブの [送信先の追加] で、[送信先タイプの選択] を選択して、[CloudWatch] を選択します。
5. [CloudWatch 送信先] ダイアログボックスで、[有効] を選択します。
6. [名前] に、イベント発行先の名前を入力します。
7. [Event types] で、イベント送信先に発行するイベントタイプを選択します。次のイベントタイプを使用できます。
  - 送信 – Amazon SES に対する呼び出しが成功しました。Amazon SES はメールの配信を試行します。
  - 拒否 – Amazon SES は E メールを受け取りましたが、ウイルスが含まれていると判断して拒否しました。Amazon SES は、受信者のメールサーバーに E メールを配信しようとしませんでした。
  - バウンス – 受信者のメールサーバーにより、メールが完全に拒否されました。このイベントはハードバウンスに該当します。ソフトバウンスは、Amazon SES が一定期間にわたって再試行してもメールを配信できなかった場合に限り発生します。
  - 苦情 – E メールは受信者に正常に配信されました。受取人がメールをスパムとしてマークしました。
  - 配信 – Amazon SES が受信者のメールサーバーに E メールを正常に送信しました。
  - オープン – 受信者がメッセージを受け取り、E メールクライアントで開きました。
  - クリック – 受信者はメール内の 1 つ以上のリンクをクリックしました。
  - レンダリング失敗 – テンプレートレンダリングの問題により E メールが送信されませんでした。このイベントタイプは、[SendTemplatedEmail](#) または [SendBulkTemplatedEmail](#) API オペレーションを使用してテンプレートメールを送信する場合にのみ発生します。このイベントタイプは、テンプレートデータが見つからない場合や、テンプレートのパラメータとデータが一致しない場合に発生します。
  - 配信の遅延 – 一時的な問題が発生したため、メールを受信者に配信できませんでした。配信の遅延は、受信者の受信トレイがいっぱいになった場合や、受信側の電子メールサーバーで一時的な問題が発生した場合などに発生します。
8. [値ソース] で、Amazon SES が CloudWatch に渡すデータを取得する方法を指定します。次の値ソースを使用できます。
  - メッセージタグ – Amazon SES は、X-SES-MESSAGE-TAGS ヘッダーまたは Tags API のパラメータを使用して、指定するタグからディメンション名と値を取得します。メッセージタグの使用の詳細については、「[the section called “ステップ 3: 送信時に設定セットを指定する” \(p. 305\)](#)」を参照してください。

#### Note

メッセージタグには、0~9 の数字、A~Z の文字 (大文字と小文字)、ハイフン (-)、およびアンダースコア (\_) を使用できます。

[メッセージタグ] 値ソースを使用し、Amazon SES 自動タグに基づいてディメンションを作成することもできます。自動タグを使用するには、自動タグの完全な名前を [Dimension Name] として入力します。例えば、設定セット自動タグに基づいてディメンションを作成するには、[ディメンション名] に `ses:configuration-set` を使用し、[デフォルト値] に設定セットの名前を使用します。自動タグの詳細一覧については、「[イベント発行のしくみ \(p. 296\)](#)」を参照してください。

- Email Header – Amazon SES が、E メール内のヘッダーからディメンション名と値を取得します。

#### Note

以下の E メールヘッダーをディメンション名として使用することはできません：  
Received、To、From、DKIM-Signature、CC、message-id、または Return-Path。

- リンクタグ - Amazon SES は、リンクで指定したタグからディメンション名と値を取得します。リンクへのタグの追加の詳細については、「[リンクに一意的識別子をタグ付けできますか? \(p. 521\)](#)」を参照してください。
9. [ディメンション名] で、CloudWatch に渡すディメンションの名前を入力します。[Default Value] に、ディメンションの値を入力します。

#### Note

ディメンション名と値には、文字 A~Z、数字 0~9、アンダースコア (\_)、アットマーク (@)、ハイフン (-)、およびピリオド (.) のみを含めることができます。スペース、アクセント文字、非ラテン文字、およびその他の特殊文字は使用できません。

10. さらにディメンションを追加する場合は、[Add Dimension] を選択します。問題がなければ、[保存] を選択します。

### CloudWatch ディメンションの選択

CloudWatch ディメンションとして使用する名前と値を選択するときは、次の要素を考慮してください。

- メトリクスあたりの料金 - CloudWatch のベーシック Amazon SES メトリクスを無料で表示できます。ただし、イベント発行を使用してメトリクスを収集する場合は、CloudWatch でカスタムメトリクスを作成します。イベントタイプ、ディメンション名、ディメンション値のそれぞれの固有の組み合わせによって、CloudWatch で異なるカスタムメトリクスが作成されます。CloudWatch を使用する場合は、作成したカスタムメトリクスごとに課金されます。このため、多くの異なる値を取得する可能性があるディメンションの選択を回避できます。たとえば、「From」ドメインごとに E メール送信イベントを追跡することが特に必要な場合を除き、Amazon SES 自動タグ `ses:from-domain` のディメンションを定義する必要はありません。多くの異なる値が使用されるためです。詳細については、「[CloudWatch 料金表](#)」をご覧ください。
- メトリクスフィルタリング - メトリクスに複数のディメンションがある場合、CloudWatch では各ディメンションに基づいて別個にメトリクスにアクセスすることはできません。そのため、1 つの CloudWatch Event 送信先に複数のディメンションを追加する場合は、事前によく検討してください。たとえば、campaign あたりのメトリクスと、campaign と genre の組み合わせあたりのメトリクスが必要な場合、2 つのイベント送信先を追加する必要があります。ディメンションとして campaign のみ使用するイベント送信先と、campaign および genre の両方をディメンションとして使用するイベント送信先です。
- ディメンション値ソース - Amazon SES 固有のヘッダーまたは API へのパラメータを使用してディメンション値を指定する代わりに、Amazon SES が独自の MIME メッセージヘッダーからディメンション値を取得するように選択することもできます。このオプションは、すでにカスタムヘッダーを使用しており、ヘッダー値に基づいてメトリクスを収集するために、メールやメール送信 API 呼び出しを変更したくない場合に使用できます。Amazon SES イベント発行に独自の MIME メッセージヘッダーを使用する場合、Amazon SES イベント発行に使用するヘッダー名と値には、文字 A~Z、数字 0~9、アンダースコア (\_)、アットマーク (@)、ハイフン (-)、およびピリオド (.) のみを含めることができます。その他の文字を含む名前または値を指定した場合、Eメールの送信には成功しますが、イベントメトリクスは Amazon CloudWatch に出力されません。

CloudWatch コンセプトの詳細については、[Amazon CloudWatch ユーザーガイド](#)の「Amazon CloudWatch の概念」を参照してください。

### Amazon SES イベント公開用の Kinesis Data Firehose イベント送信先を設定する

Amazon Kinesis Data Firehose イベント送信先は、特定の Amazon SES Eメール送信イベントを Kinesis Data Firehose に発行するエンティティを表します。Kinesis Data Firehose イベントの送信先は設定セット内のみ存在するため、まず[設定セットを作成 \(p. 298\)](#)する必要があります。次に、設定セットにイベントの送信先を追加します。

ここでは、Amazon SES コンソールを使用してイベントの送信先を作成する手順について説明します。また、Amazon SES API V2 で、[UpdateConfigurationSetEventDestination](#) オペレーションを使用して、イベントの送信先を作成および更新できます。

Kinesis Data Firehose イベント送信先を設定セットに追加するには (コンソール)

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. 設定セットリストから設定セットを選択します。リストが空の場合、まず [設定セットを作成 \(p. 298\)](#) する必要があります。
4. [送信先の追加] で、[送信先タイプの選択] を選択して [Kinesis Data Firehose] を選択します。
5. [<名前>] に、イベント発行先の名前を入力します。
6. [イベントタイプ] で、イベント送信先に発行するイベントタイプを少なくとも 1 つ選択します。
  - 送信 – Amazon SES に対する呼び出しが成功しました。Amazon SES はメールの配信を試行します。
  - 拒否 – Amazon SES は E メールを受け取りましたが、ウイルスが含まれていると判断して拒否しました。Amazon SES は、受信者のメールサーバーに E メールを配信しようとしませんでした。
  - バウンス – 受信者のメールサーバーにより、メールが完全に拒否されました。このイベントはハードバウンスに該当します。ソフトバウンスは、Amazon SES が一定期間にわたって再試行してもメールを配信できなかった場合に限ります。
  - 苦情 – E メールは受信者に正常に配信されました。受取人がメールをスパムとしてマークしました。
  - 配信 – Amazon SES が受信者のメールサーバーに E メールを正常に送信しました。
  - オープン – 受信者がメッセージを受け取り、E メールクライアントで開きました。
  - クリック – 受信者はメール内の 1 つ以上のリンクをクリックしました。
  - レンダリング失敗 – テンプレートレンダリングの問題により E メールが送信されませんでした。このイベントタイプは、[SendTemplatedEmail](#) または [SendBulkTemplatedEmail](#) API オペレーションを使用してテンプレートメールを送信する場合にのみ発生します。このイベントタイプは、テンプレートデータが見つからない場合や、テンプレートのパラメータとデータが一致しない場合に発生します。
  - 配信の遅延 – 一時的な問題が発生したため、メールを受信者に配信できませんでした。配信の遅延は、受信者の受信トレイがいっぱいになった場合や、受信側の電子メールサーバーで一時的な問題が発生した場合などに発生します。
7. [Enabled] を選択します。
8. [ストリーミング] で、Kinesis Data Firehose 配信ストリームを選択するか、ストリーミングを新規作成をクリックし、Kinesis Data Firehose コンソールで新しいものを作成します。

Kinesis Data Firehose コンソールを使用したストリーミング作成の詳細については、[Amazon Kinesis Data Firehose デベロッパーガイド](#) の「Amazon Kinesis Firehose 配信ストリームの作成」を参照してください。

9. [IAM ロール] で、Amazon SES がお客様の代わりに Kinesis Data Firehose に発行するアクセス許可を持つ IAM ロールを選択します。ロールは、既存のものを選択するか、Amazon SES で自動作成するか、独自に作成することができます。

既存のロールを選択するか、独自のロールを作成する場合、ロールのポリシーを手動で変更し、Kinesis Data Firehose 配信ストリームへのアクセス許可をロールに付与して、ロールを引き受けるアクセス許可を Amazon SES に付与する必要があります。ポリシーの例については、「[Kinesis Data Firehose 配信ストリームへの公開に関する Amazon SES の許可 \(p. 303\)](#)」を参照してください。

10. [Save] を選択します。

UpdateConfigurationSetEventDestinationAPI を使用して、Kinesis Data Firehose イベントの宛先を追加する方法については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

### Kinesis Data Firehose 配信ストリームへの公開に関するAmazon SES の許可

Amazon SES が Kinesis Data Firehose 配信ストリームにレコードを発行できるようにするには、AWS Identity and Access Management (IAM) [ロール](#)を使用して、ロールのアクセス許可ポリシーと信頼ポリシーを添付または変更する必要があります。アクセス許可ポリシーではロールでレコードを Kinesis Data Firehose 配信ストリームに発行でき、信頼ポリシーでは Amazon SES でロールを引き受けることができます。

このセクションでは、両方のポリシーの例を示します。IAM ロールへの添付ポリシーについての詳細は、IAM ユーザーガイドの「[ロールの修正](#)」を参照してください。

#### アクセス許可ポリシー

以下のアクセス許可ポリシーにより、ロールがデータレコードを Kinesis Data Firehose 配信ストリームに発行できるようになります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "firehose:PutRecordBatch"
      ],
      "Resource": [
        "arn:aws:firehose:REGION:ACCOUNT-ID:deliverystream/DELIVERY-STREAM-NAME "
      ]
    }
  ]
}
```

#### 信頼ポリシー

次の信頼ポリシーにより、Amazon SES はロールを引き受けることができます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ses.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "ACCOUNT-ID"
        }
      }
    }
  ]
}
```

## イベント発行の Amazon SNS イベント送信先のセットアップ

イベント送信先では、Amazon SNS を使用して特定のメール送信イベントに関して通知します。Amazon SNS イベント送信先は設定セット内にのみ存在するため、まず[設定セットを作成 \(p. 298\)](#)してからその設定セットにイベント送信先を追加する必要があります。

ここでは、Amazon SES コンソールを使用してイベントの送信先を作成する手順について説明します。また、Amazon SES API V2 で、[UpdateConfigurationSetEventDestination](#) オペレーションを使用して、イベントの送信先を作成および更新できます。

### Note

通知は、アカウントレベルで Amazon SNS を通じて受け取ることもできます。つまり、Amazon SES アカウント全体で送信イベントが発生するたびに、Amazon SNS 通知を受け取ることができます。アカウントレベルの通知ではなくイベント発行を使用することで、特定のイベントタイプに関する通知のみ、または特定の設定セットを使用して送信されたメールに関する通知のみを、Amazon SES で送信するように設定できます。アカウントレベルの Amazon SNS 通知をセットアップする詳しい方法については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。

Amazon SNS トピックにサブスクライブしているエンドポイントにメッセージを送信するには、追加料金が発生します。詳細については、「[Amazon SNS の料金](#)」を参照してください。

設定セットに Amazon SNS イベント送信先を追加するには

1. まだ行っていない場合は、Amazon SNS トピックを作成し、そのトピックにサブスクライブします。詳細については、[Amazon Simple Notification Service デベロッパーガイド](#)の「開始方法」を参照してください。

### Important

Amazon SNS を使用してトピックを作成する場合、タイプはスタンダードを選択します。  
(SES は FIFO タイプのトピックをサポートしていません。)

2. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
3. ナビゲーションペインで [設定セット] を選択します。
4. 設定セットリストから設定セットを選択します。リストが空の場合、まず[設定セットを作成 \(p. 298\)](#)する必要があります。
5. [Add Destination] で、[Select a destination type] を選択して [SNS] を選択します。
6. [名前] に、イベント発行先の名前を入力します。
7. [イベントタイプ] で、イベント送信先に発行するイベントタイプを少なくとも 1 つ選択します。
  - 送信 – Amazon SES に対する呼び出しが成功しました。Amazon SES はメールの配信を試行します。
  - 拒否 – Amazon SES は E メールを受け取りましたが、ウイルスが含まれていると判断して拒否しました。Amazon SES は、受信者のメールサーバーに E メールを配信しようとしませんでした。
  - バウンス – 受信者のメールサーバーにより、メールが完全に拒否されました。このイベントはハードバウンスに該当します。ソフトバウンスは、Amazon SES が一定期間にわたって再試行してもメールを配信できなかった場合に限りです。
  - 苦情 – E メールは受信者に正常に配信されました。受取人がメールをスパムとしてマークしました。
  - 配信 – Amazon SES が受信者のメールサーバーに E メールを正常に送信しました。
  - オープン – 受信者がメッセージを受け取り、E メールクライアントで開きました。
  - クリック – 受信者はメール内の 1 つ以上のリンクをクリックしました。
  - レンダリング失敗 – テンプレートレンダリングの問題により E メールが送信されませんでした。このイベントタイプは、[SendTemplatedEmail](#) または [SendBulkTemplatedEmail](#) API オペレーション



を使用してテンプレートメールを送信する場合にのみ発生します。このイベントタイプは、テンプレートデータが見つからない場合や、テンプレートのパラメータとデータが一致しない場合に発生します。

- 配信の遅延 – 一時的な問題が発生したため、メールを受信者に配信できませんでした。配信の遅延は、受信者の受信トレイがいっぱいになった場合や、受信側の電子メールサーバーで一時的な問題が発生した場合などに発生します。

8. [Enabled] を選択します。
9. [トピック] で、既存の Amazon SNS トピックを選択するか、[新しいトピックの作成] を選択して新しいトピックを作成します。

トピックの作成の詳細については、Amazon Simple Notification Service デベロッパーガイドの「[トピックの作成](#)」を参照してください。

10. [Save] を選択します。
11. E メール送信時に設定セットを使用するには、[メールの送信時に設定セットを指定する \(p. 264\)](#) を参照してください。

## ステップ 3: E メールを送信するときに設定セットを指定する

[設定セットを作成 \(p. 298\)](#)し、[イベント発行先を追加 \(p. 299\)](#)したら、イベント発行の最後のステップとして E メールを送信します。

メールに関連付けられているイベントを発行するには、メールに関連付ける設定セットの名前を指定する必要があります。必要に応じて、メールを分類するためのメッセージタグを指定することもできます。

この情報を Amazon SES に提供する方法として、E メール送信 API へのパラメータ、Amazon SES 固有の E メールヘッダー、または MIME メッセージのカスタムヘッダーがあります。選択する方法は、次の表に示すように、使用するメール送信インターフェイスによって決まります。

メール送信インターフェイス	イベント発行方法
SendEmail	API パラメータ
SendRawEmail	API パラメータ、Amazon SES 固有のメールヘッダー、またはカスタム MIME ヘッダー  Important  ヘッダーと API パラメータの両方を使用してメッセージタグを指定すると、Amazon SES では API パラメータで指定されたメッセージタグのみが使用されます。Amazon SES は、API パラメータとヘッダーで指定されたメッセージタグを結合しません。
SMTP インターフェイス	Amazon SES 固有の E メールヘッダー

以下のセクションでは、ヘッダーおよび API パラメータを使用して設定セットとメッセージタグを指定する方法について説明します。

- [Amazon SES API パラメータの使用 \(p. 306\)](#)
- [Amazon SES 固有の E メールヘッダーの使用 \(p. 306\)](#)
- [カスタムメールヘッダーの使用 \(p. 306\)](#)

このガイドには、Amazon SES を使用してプログラムでメールを送信する方法を示す複数のコード例も含まれています。各コード例では、メールの送信時に設定セットを渡す方法が示されています。詳細については、「[Amazon SES コード例 \(p. 425\)](#)」を参照してください。

#### Note

必要に応じて、Eメールのヘッダーにメッセージタグを含めることができます。メッセージタグには、0-9の数字、A-Zの文字(大文字と小文字)、ハイフン(-)、およびアンダースコア(\_)を使用できます。

## Amazon SES API パラメータの使用

イベント発行で `SendEmail` または `SendRawEmail` を使用するには、`ConfigurationSet` および `MessageTag` というデータ構造を API コールに渡すことで設定セットとメッセージタグを指定します。

Amazon SES API の使用については、[Amazon Simple Email Service API リファレンス](#) を参照してください。

## Amazon SES 固有の Eメールヘッダーの使用

`SendRawEmail` または SMTP インターフェイスを使用する場合、Amazon SES 固有のヘッダーをメールに追加することで設定セットとメッセージタグを指定できます。Amazon SES は、Eメールを送信する前にヘッダーを削除します。次の表に、使用するヘッダー名を示します。

イベント発行情報	ヘッダー
設定セット	X-SES-CONFIGURATION-SET
メッセージタグ	X-SES-MESSAGE-TAGS

次の例で、Amazon SES に送信した raw Eメールにヘッダーがどのように表示されるかを示します。

```
X-SES-MESSAGE-TAGS: tagName1=tagValue1, tagName2=tagValue2
X-SES-CONFIGURATION-SET: myConfigurationSet
From: sender@example.com
To: recipient@example.com
Subject: Subject
Content-Type: multipart/alternative;
  boundary="-----=_boundary"

-----=_boundary
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

body
-----=_boundary
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: 7bit

body
-----=_boundary--
```

## カスタムメールヘッダーの使用

設定セット名は Amazon SES 固有のヘッダー `X-SES-CONFIGURATION-SET` を使用して指定する必要がありますが、メッセージタグは独自の MIME ヘッダーを使用して指定できます。

#### Note

Amazon SES イベント発行に使用するヘッダー名と値は、ASCII にする必要があります。Amazon SES イベント発行のために ASCII 以外のヘッダー名または値を指定した場合で

も、E メール送信コールに成功しますが、イベントメトリクスは Amazon CloudWatch に出力されません。

## Amazon SES イベントデータの使用

[イベント発行をセットアップ \(p. 298\)](#)し、メール送信の設定セットを指定すると、メールに関連付けられた設定セットのセットアップ時に指定したイベント送信先からメール送信イベントを取得できます。

このセクションでは、Amazon CloudWatch および Amazon Kinesis Data Firehose から E メール送信イベントを取得する方法と、Amazon SNS によって提供されたイベントデータを解釈する方法について説明します。

- [CloudWatch から Amazon SES イベントデータの取得 \(p. 307\)](#)
- [Kinesis Data Firehose から Amazon SES イベントデータの取得 \(p. 308\)](#)
- [Amazon SNS からの Amazon SES イベントデータの解釈 \(p. 328\)](#)

## CloudWatch から Amazon SES イベントデータの取得

Amazon SES は、Amazon CloudWatch への E メール送信イベントに関するメトリクスを発行することができます。イベントデータを CloudWatch に発行する場合、これらのメトリクスは時系列データのセットとして提供されます。これらのメトリクスを使用して、メール送信のパフォーマンスをモニタリングできます。例えば、苦情メトリクスをモニタリングし、メトリクスが一定の値を超えたときに CloudWatch アラームがトリガーされるように設定できます。

Amazon SES がこれらのイベントを CloudWatch に発行する際の詳細度には、2 つのレベルがあります。

- **AWS アカウント全体** - これらの粗いメトリクス (Amazon SES コンソールおよび `GetSendStatistics` API を使用してモニタリングするメトリクスに対応) は、AWS アカウント全体における合計です。Amazon SES は、これらのメトリクスを自動的に CloudWatch に発行します。
- **細かい** - これらのメトリクスは、メッセージタグを使用して定義する E メール特性によって分類されます。これらのメトリクスを CloudWatch に公開するには、CloudWatch Event 送信先を含めて [イベント発行をセットアップ \(p. 298\)](#)し、E メールを送信時に [設定セットを指定 \(p. 305\)](#)する必要があります。また、メッセージタグを指定したり、Amazon SES が自動的に提供する [自動タグ \(p. 296\)](#)を使用できます。

このセクションでは、使用可能なメトリクスと、CloudWatch でメトリクスを表示する方法について説明します。

### 使用可能なメトリクス

以下の Amazon SES E メール送信メトリクスを CloudWatch に発行することができます。

- **送信** - Amazon SES に対する呼び出しが成功しました。Amazon SES はメールの配信を試行します。
- **拒否** - Amazon SES は E メールを受け取りましたが、ウイルスが含まれていると判断して拒否しました。Amazon SES は、受信者のメールサーバーに E メールを配信しようとしませんでした。
- **バウンス** - 受信者のメールサーバーにより、メールが完全に拒否されました。このイベントはハードバウンスに該当します。ソフトバウンスは、Amazon SES が一定期間にわたって再試行してもメールを配信できなかった場合に限ります。
- **苦情** - E メールは受信者に正常に配信されました。受取人がメールをスパムとしてマークしました。
- **配信** - Amazon SES が受信者のメールサーバーに E メールを正常に送信しました。
- **オープン** - 受信者がメッセージを受け取り、E メールクライアントで開きました。
- **クリック** - 受信者はメール内の 1 つ以上のリンクをクリックしました。
- **レンダリング失敗** - テンプレートレンダリングの問題により E メールが送信されませんでした。このイベントタイプは、[SendTemplatedEmail](#) または [SendBulkTemplatedEmail](#) API オペレーションを使用し

てテンプレートメールを送信する場合にのみ発生します。このイベントタイプは、テンプレートデータが見つからない場合や、テンプレートのパラメータとデータが一致しない場合に発生します。

- 配信の遅延 – 一時的な問題が発生したため、メールを受信者に配信できませんでした。配信の遅延は、受信者の受信トレイがいっぱいになった場合や、受信側の電子メールサーバーで一時的な問題が発生した場合などに発生します。

## 使用できるディメンション

CloudWatch は、Amazon SES でユーザーが CloudWatch Event 送信先を設定セットに追加する際に指定したディメンション名を使用します。詳細については、「[イベント発行の CloudWatch Event 送信先のセットアップ](#) (p. 299)」を参照してください。

## CloudWatch コンソールでの Amazon SES メトリクスの表示

次の手順では、CloudWatch コンソールを使用して Amazon SES イベント発行メトリクスを表示する方法について説明します。

CloudWatch コンソールを使用してメトリクスを表示するには

1. AWS Management Console にサインインして、CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. 必要に応じてリージョンを変更します。ナビゲーションバーから、AWS リソースがあるリージョンを選択します。詳細については、「[リージョンとエンドポイント](#)」を参照してください。
3. ナビゲーションペインで [Metrics (メトリクス)] を選択します。
4. [All metrics] ペインで、[AWS Namespaces] を展開し、[SES] を選択します。
5. Amazon SES によって自動的に発行される AWS アカウント全体のメトリクスを表示するには、[アカウントの送信メトリクス] を選択します。細かい [イベント発行メトリクス](#) (p. 296) を表示するには、[CloudWatch Event 発行のセットアップ](#) (p. 299) 時に指定したディメンションの組み合わせを選択します。
6. 表示するメトリクスを選択します。

グラフには、時間の経過に伴うメトリクスが示されます。

AWS CLI を使用してメトリクスを表示するには

- コマンドプロンプトで、次のコマンドを使用します。

```
aws cloudwatch list-metrics --namespace "AWS/SES"
```

## Kinesis Data Firehose から Amazon SES イベントデータの取得

Amazon SES は、E メール送信イベントを JSON レコードとして Kinesis Data Firehose に発行します。Kinesis Data Firehose は、Kinesis Data Firehose で配信ストリームを設定したときに選択した AWS のサービス宛先にレコードを発行します。Kinesis Data Firehose 配信ストリームの設定についての詳細は、[Amazon Kinesis Data Firehose デベロッパーガイド](#)の「Amazon Kinesis Data Firehose 配信ストリームの作成」を参照してください。

Kinesis Data Firehose を使用して E メール送信イベントを Amazon Redshift と Amazon Elasticsearch Service に発行する方法の例については、[チュートリアル](#) (p. 348) を参照してください。

このセクションのトピック:

- [Amazon SES が Kinesis Data Firehose に発行するイベントデータのコンテンツ](#) (p. 309)

- [Amazon SES が Kinesis Data Firehose に発行するイベントデータの例 \(p. 318\)](#)

## Amazon SES が Kinesis Data Firehose に発行するイベントデータのコンテンツ

Amazon SES は、E メール送信イベントレコードを JSON 形式で Amazon Kinesis Data Firehose に発行します。Amazon SES は、イベントを Kinesis Data Firehose に発行するときに、JSON レコードごとに改行文字を付加します。

トップレベル JSON オブジェクトには、eventType 文字列と mail オブジェクトに加え、イベントのタイプに応じて Bounce オブジェクト、Complaint オブジェクト、Delivery オブジェクト、Send オブジェクト、Reject オブジェクト、Open オブジェクト、Click オブジェクト、Rendering Failure オブジェクトまたは DeliveryDelay オブジェクトが含まれます。

これらのすべての通知タイプのレコード例については、「[Amazon SES が Kinesis Data Firehose に発行するイベントデータの例 \(p. 318\)](#)」を参照してください。

このセクションのトピック

- [トップレベル JSON オブジェクト \(p. 309\)](#)
- [Mail オブジェクト \(p. 310\)](#)
- [Bounce オブジェクト \(p. 311\)](#)
- [苦情のオブジェクト \(p. 313\)](#)
- [配信オブジェクト \(p. 315\)](#)
- [Send オブジェクト \(p. 315\)](#)
- [Reject オブジェクト \(p. 315\)](#)
- [Open オブジェクト \(p. 316\)](#)
- [Click オブジェクト \(p. 316\)](#)
- [レンダリング失敗オブジェクト \(p. 316\)](#)
- [DeliveryDelay オブジェクト \(p. 317\)](#)

### トップレベル JSON オブジェクト

メール送信イベントレコードのトップレベル JSON オブジェクトには、以下のフィールドが含まれています。

フィールド名	説明
eventType	イベントのタイプを説明する文字列。可能な値: Delivery、Send、Reject、Open、Click、Bounce、Complaint、Failure、または DeliveryDelay。
mail	イベントを作成した E メールに関する情報を含む JSON オブジェクト。
bounce	このフィールドは、eventType が Bounce の場合のみ表示されます。このフィールドには、バウンスに関する情報が含まれています。
complaint	このフィールドは、eventType が Complaint の場合のみ表示されます。このフィールドには、苦情に関する情報が含まれています。
delivery	このフィールドは、eventType が Delivery の場合のみ表示されます。このフィールドには、配信に関する情報が含まれています。

フィールド名	説明
send	このフィールドは、eventType が Send の場合のみ表示されます。
reject	このフィールドは、eventType が Reject の場合のみ表示されます。このフィールドには、拒否に関する情報が含まれています。
open	このフィールドは、eventType が Open の場合のみ表示されます。このフィールドには、オープンイベントに関する情報が含まれています。
click	このフィールドは、eventType が Click の場合のみ表示されます。このフィールドには、クリックイベントに関する情報が含まれています。
failure	このフィールドは、eventType が Rendering Failure の場合のみ表示されます。このフィールドには、レンダリング失敗イベントに関する情報が含まれています。
deliveryDelay	このフィールドは、eventType が DeliveryDelay の場合のみ表示されます。Eメールの遅延配信に関する情報が含まれています。

### Mail オブジェクト

各メール送信イベントレコードには、mail オブジェクトの元のメールに関する情報が含まれています。mail オブジェクトについての情報を含む JSON オブジェクトには次のフィールドが含まれます。

フィールド名	説明
timestamp	メッセージが送信された、ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) の日時。
messageId	Amazon SES がメッセージに割り当てた一意の ID。ユーザーがメッセージを送信すると、Amazon SES がこの値を返します。  Note  このメッセージ ID は Amazon SES によって割り当てられます。元の Eメールのメッセージ ID は、headers オブジェクトの commonHeaders および mail フィールドにあります。
source	メッセージの送信元の Eメールアドレス (エンベロープ MAIL FROM アドレス)。
sourceArn	Eメールの送信に使用された ID の Amazon リソースネーム (ARN)。送信認可の場合、sourceArn は、代理送信者が Eメールの送信に使用することを ID 所有者により承認された ID の ARN です。送信承認の詳細については、「 <a href="#">送信承認の使用 (p. 154)</a> 」を参照してください。

フィールド名	説明
sendingAccountId	Eメールの送信に使用されたアカウントの AWS アカウント ID。送信承認の場合、sendingAccountId は代理送信者のアカウント ID です。
destination	元のメールの受取人の E メールアドレスのリスト。
headersTruncated	通知でヘッダーが切り捨てられたかどうかを示す文字列。切り捨ては、ヘッダーが 10 KB を超える場合に発生します。指定できる値は true および false です。
headers	Eメールの元のヘッダーの一覧。リスト内の各ヘッダーには、name フィールドと value フィールドがあります。  Note  headers フィールド内のメッセージ ID は、Amazon SES に渡した元のメッセージのもので、その後 Amazon SES がメッセージに割り当てるメッセージ ID は、messageId オブジェクトの mail フィールドのもので、
commonHeaders	Eメールのよく使用される元のヘッダー一覧。リスト内の各ヘッダーには、name フィールドと value フィールドがあります。  Note  commonHeaders フィールド内のメッセージ ID は、Amazon SES に渡した元のメッセージのもので、その後 Amazon SES がメッセージに割り当てるメッセージ ID は、messageId オブジェクトの mail フィールドのもので、

## Bounce オブジェクト

Bounce イベントに関する情報を含む JSON オブジェクトには常に以下のフィールドが含まれます。

フィールド名	説明
bounceType	バウンスのタイプで、Amazon SES によって決定されます。
bounceSubType	バウンスのサブタイプで、Amazon SES によって決定されます。
bouncedRecipients	バウンスとなった元のメールの受取人についての情報を含むリスト。
timestamp	ISP がバウンス通知を送信した、ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) の日時。

フィールド名	説明
feedbackId	バウンスの一意の ID。
reportingMTA	DSN の Reporting-MTA フィールドの値。これは、DSN に記述された配信、リレー、またはゲートウェイのオペレーションを試みた MTA (Message Transfer Authority) の値です。  Note  このフィールドは、配信ステータス通知 (DSN) がバウンスにアタッチされている場合にのみ表示されます。

### バウンスとなった受取人

バウンスイベントには、1 人の受信者に関するものと複数の受信者に関するものがあります。bouncedRecipients フィールドにはオブジェクトのリストが含まれ (バウンスイベントが関係する受信者ごとに 1 つのオブジェクト)、常に以下のフィールドが含まれています。

フィールド名	説明
emailAddress	受取人の E メールアドレス。DSN が利用できる場合、これが DSN の Final-Recipient フィールドの値です。

オプションで、DSN がバウンスに添付されている場合、以下のフィールドも表示される場合があります。

フィールド名	説明
action	DSN の Action フィールドの値。このフィールドには、Reporting-MTA により実行された、この受取人に対してメッセージを送信しようとした結果のアクションが示されます。
status	DSN の Status フィールドの値。これは、メッセージの配信状態を示す、受取人ごとに個別の、トランスポート独立型ステータスコードです。
diagnosticCode	ステータスコードは、Reporting-MTA により発行されます。これは、DSN の Diagnostic-Code フィールドの値です。このフィールドが DSN に存在しない場合もあります (その場合 JSON オブジェクトにも表示されません)。

### バウンスのタイプ

各バウンスイベントは、以下の表に示すいずれかのタイプになります。

イベント発行システムは、Amazon SES によって再試行されなくなるハードバウンスおよびソフトバウンスのみを発行します。Permanent とマークされたバウンスを受信した場合は、対応する E メールアドレスをメーリングリストから削除する必要があります。これらは今後も送信できません。Transient バウンスは、メッセージに対してソフトバウンスが複数回発生して Amazon SES が再送を停止した場合に送信



されます。最初は `Transient` バウンスとして返されても、将来的には同じアドレスに正常に再送できる場合があります。

bounceType	bounceSubType	説明
Undetermined	Undetermined	Amazon SES はバウンスの理由を特定できませんでした。
Permanent	General	Amazon SES は、一般的なハードバウンスを受け取りました。このタイプのバウンスを受信した場合は、メーリングリストからこの受信者のメールアドレスを削除する必要があります。
Permanent	NoEmail	Amazon SES は、送信先の E メールアドレスが存在しないことが原因である永続的なハードバウンスを受け取りました。このタイプのバウンスを受信した場合は、メーリングリストからこの受信者のメールアドレスを削除する必要があります。
Permanent	Suppressed	Amazon SES は、このアドレスへの送信を抑制しました。このアドレスは無効なアドレスとして最近バウンスとなった履歴があるからです。サブレッションリストからメールアドレスを削除する方法については、「 <a href="#">Amazon SES グローバルサブレッションリストの使用 (p. 199)</a> 」を参照してください。
Permanent	OnAccountSuppressionList	Amazon SES は、アドレスが <a href="#">アカウントレベルのサブレッションリスト (p. 190)</a> にあるので、このアドレスへの送信を抑制しました。
Transient	General	Amazon SES は、一般的なバウンスを受け取りました。今後、この受信者に正常に再送できる可能性があります。
Transient	MailboxFull	Amazon SES は、メールボックスが一杯であることを示すバウンスを受け取りました。今後、この受信者に正常に再送できる可能性があります。
Transient	MessageTooLarge	Amazon SES は、メッセージサイズが大きすぎることを示すバウンスを受け取りました。メッセージサイズを小さくすることで、この受信者に正常に再送できる可能性があります。
Transient	ContentRejected	Amazon SES は、コンテンツが拒否されたことを示すバウンスを受け取りました。メッセージのコンテンツを変更することで、この受信者に正常に再送できる可能性があります。
Transient	AttachmentRejected	Amazon SES は、添付ファイルが拒否されたことを示すバウンスを受け取りました。添付ファイルを削除または変更することで、この受信者に正常に再送できる可能性があります。

### 苦情のオブジェクト

Complaint イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
complainedRecipients	苦情を送信した可能性がある受取人に関する情報を含むリスト。
timestamp	ISP が苦情通知を送信した、ISO8601 形式 (YYYY-MM-DDThh:mm:ss.SZ) の日時。
feedbackId	苦情の一意の ID。
complaintSubType	苦情のサブタイプで、Amazon SES によって決定されます。

また、フィードバックレポートが苦情に添付されている場合、以下のフィールドが示される場合があります。

フィールド名	説明
userAgent	フィードバックレポートの User-Agent フィールドの値。これは、レポートを生成したシステムの名前とバージョンを示します。
complaintFeedbackType	ISP から受け取ったフィードバックレポートの Feedback-Type フィールドの値。これには、フィードバックのタイプが含まれます。
arrivalDate	ISO 8601 形式のフィードバックレポートの Arrival-Date または Received-Date フィールドの値 (YYYY-MM-DDThh:mm:ss.SZ)。このフィールドがレポートにない場合もあります (その場合、JSON オブジェクトにも表示されません)。

### 苦情を申告した受取人

complainedRecipients フィールドには、苦情の送信元と思われる受信者のリストが含まれます。

#### Important

ほとんどの ISP は、苦情の通知に基づいて苦情を送信した受信者の E メールアドレスを編集するため、このリストは、元のメッセージの受信者と苦情を送信した ISP に基づいて想定される苦情送信元の受信者に関する情報を示します。Amazon SES では、この受信者のリストを確認するために元のメッセージを参照します。

このリストの JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
emailAddress	受取人の E メールアドレス。

### 苦情のタイプ

complaintFeedbackType フィールドには以下の苦情のタイプが示されます ([Internet Assigned Numbers Authority ウェブサイト](#) に基づいて、報告する ISP により割り当てられます)。

フィールド名	説明
abuse	迷惑メールまたはその他のメール不正使用。
auth-failure	Eメールの認証障害レポート
fraud	なんらかの詐欺またはフィッシング行為を示します。
not-spam	レポートの提供者がこのメッセージをスパムではないと見なしていることを示します。このタイプは、誤ってスパムとしてタグ付けまたは分類されたメッセージを修正するために使用される場合があります。
other	その他の登録されたタイプに該当しないフィードバックを示します。
virus	元のメッセージでウイルスが見つかったことを示します。

### 配信オブジェクト

Delivery イベントに関する情報を含む JSON オブジェクトには常に以下のフィールドが含まれます。

フィールド名	説明
timestamp	Amazon SES が受信者のメールサーバーにメールを配信した、ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) の日時。
processingTimeMillis	Amazon SES で送信者からリクエストを受理して、Amazon SES が受取人のメールサーバーにメッセージを渡すまでのミリ秒単位の時間。
recipients	配信イベントの適用対象となる受信者のリスト。
smtpResponse	Amazon SES から Eメールを受け取ったリモート ISP による SMTP 応答メッセージ。このメッセージは、Eメール、受信メールサーバー、または受信する ISP によって異なります。
reportingMTA	メールを送信する Amazon SES メールサーバーのホスト名。

### Send オブジェクト

send イベントに関する情報を含む JSON オブジェクトは常に空です。

### Reject オブジェクト

Reject イベントに関する情報を含む JSON オブジェクトには常に以下のフィールドが含まれます。

フィールド名	説明
reason	メールが拒否された理由。このフィールドの唯一の値は Bad content です。Eメール内のウイル

フィールド名	説明
	スを Amazon SES で検出したことを意味します。メッセージが拒否されると、Amazon SES はメッセージの処理を停止し、そのメッセージを受信者のメールサーバーに配信しようとしません。

### Open オブジェクト

Open イベントに関する情報を含む JSON オブジェクトには、常に以下のフィールドが含まれます。

フィールド名	説明
ipAddress	受信側の IP アドレス
timestamp	ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) で表示された、オープンイベントが発生日時。
userAgent	受取人がメールを開くために使用したデバイスまたは E メールクライアントのユーザーエージェント。

### Click オブジェクト

click イベントに関する情報を含む JSON オブジェクトには、常に以下のフィールドが含まれます。

フィールド名	説明
ipAddress	受信側の IP アドレス
timestamp	ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) で表示された、クリックイベントが発生日時。
userAgent	受信者が E メールリンクをクリックするために使用したクライアントのユーザーエージェント。
link	受取人がクリックしたリンクの URL。
linkTags	ses:tags 属性を使用してリンクに追加されたタグのリスト。メール内のリンクにタグを追加する方法については、「 <a href="#">Q5. リンクに一意的識別子をタグ付けできますか? (p. 521)</a> 」の「 <a href="#">Amazon SES E メール送信メトリクスに関するよくある質問 (p. 518)</a> 」を参照してください。

### レンダリング失敗オブジェクト

Rendering Failure イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
templateName	Eメールの送信に使用されるテンプレートの名前。

フィールド名	説明
errorMessage	レンダリング失敗に関する詳細情報を提供するメッセージ。

### DeliveryDelay オブジェクト

DeliveryDelay イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
delayType	<p>遅延のタイプ。想定される値は次のとおりです。</p> <ul style="list-style-type: none"> <li>InternalFailure – 内部的な Amazon SES の問題により、メッセージが遅延しました。</li> <li>一般 – SMTP 通信中に一般的なエラーが発生しました。</li> <li>MailboxFull – 受取人のメールボックスがいっぱいで、追加のメッセージを受信できません。</li> <li>SpamDetected – 受取人のメールサーバーが、お客様のアカウントからの迷惑メールを大量に検出しました。</li> <li>RecipientServerError – 受取人の E メールサーバーの一時的な問題により、メッセージの配信が妨げられています。</li> <li>IPFailure – メッセージを送信する IP アドレスは、受信者の E メールプロバイダーによってブロックまたはスロットリングされています。</li> <li>TransientCommunicationGeneral – 受取人の E メールプロバイダーとの SMTP 通信中に、一時的な通信障害が発生しました。</li> <li>BYOIPHostNameLookupUnavailable – Amazon SES はユーザーの IP アドレスの DNS ホスト名を検索できませんでした。このタイプの遅延は、<a href="#">Bring Your Own IP (p. 186)</a> を使用する場合にのみ発生します。</li> <li>未確定 – Amazon SES は配信遅延の理由を判断できませんでした。</li> </ul>
delayedRecipients	Eメールの受取人に関する情報を含むオブジェクト。
expirationTime	Amazon SES がメッセージの配信を停止する日時。この値は ISO 8601 形式で表示されます。
reportingMTA	遅延を報告したメッセージ転送エージェント (MTA) の IP アドレス。
timestamp	遅延が発生した日時。ISO 8601 形式で表示されます。

### 遅延受取人

delayedRecipients オブジェクトには、以下の値が含まれています。

フィールド名	説明
emailAddress	メッセージの配信が遅れる原因となった E メールアドレス。
status	配信遅延に関連付けられた SMTP ステータスコード。
diagnosticCode	受信側メッセージ転送エージェント (MTA) によって提供される診断コード。

## Amazon SES が Kinesis Data Firehose に発行するイベントデータの例

このセクションでは、Amazon SES が Kinesis Data Firehose に発行するメール送信イベントレコードの例をイベントタイプ別に示します。

このセクションのトピック:

- [バウンスレコード \(p. 318\)](#)
- [苦情レコード \(p. 319\)](#)
- [配信レコード \(p. 321\)](#)
- [Send レコード \(p. 322\)](#)
- [Reject レコード \(p. 323\)](#)
- [Open レコード \(p. 324\)](#)
- [Click レコード \(p. 326\)](#)
- [レンダリング失敗レコード \(p. 327\)](#)
- [DeliveryDelay レコード \(p. 328\)](#)

### バウンスレコード

Amazon SES が Kinesis Data Firehose に発行する Bounce イベントレコードの例を以下に示します。

```
{
  "eventType": "Bounce",
  "bounce": {
    "bounceType": "Permanent",
    "bounceSubType": "General",
    "bouncedRecipients": [
      {
        "emailAddress": "recipient@example.com",
        "action": "failed",
        "status": "5.1.1",
        "diagnosticCode": "smtp; 550 5.1.1 user unknown"
      }
    ]
  },
  "timestamp": "2017-08-05T00:41:02.669Z",
  "feedbackId": "01000157c44f053b-61b59c11-9236-11e6-8f96-7be8aexample-000000",
  "reportingMTA": "dsn; mta.example.com"
},
"mail": {
  "timestamp": "2017-08-05T00:40:02.012Z",
  "source": "Sender Name <sender@example.com>",
  "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
  "sendingAccountId": "123456789012",
  "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
```

```
"destination":[
  "recipient@example.com"
],
"headersTruncated":false,
"headers":[
  {
    "name":"From",
    "value":"Sender Name <sender@example.com>"
  },
  {
    "name":"To",
    "value":"recipient@example.com"
  },
  {
    "name":"Subject",
    "value":"Message sent from Amazon SES"
  },
  {
    "name":"MIME-Version",
    "value":"1.0"
  },
  {
    "name":"Content-Type",
    "value":"multipart/alternative; boundary=\"-----_Part_7307378_1629847660.1516840721503\""
  }
],
"commonHeaders":{
  "from":[
    "Sender Name <sender@example.com>"
  ],
  "to":[
    "recipient@example.com"
  ],
  "messageId":"EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "subject":"Message sent from Amazon SES"
},
"tags":{
  "ses:configuration-set":[
    "ConfigSet"
  ],
  "ses:source-ip":[
    "192.0.2.0"
  ],
  "ses:from-domain":[
    "example.com"
  ],
  "ses:caller-identity":[
    "ses_user"
  ]
}
}
```

## 苦情レコード

Amazon SES が Kinesis Data Firehose に発行する Complaint イベントレコードの例を以下に示します。

```
{
  "eventType":"Complaint",
  "complaint": {
    "complainedRecipients":[
      {
        "emailAddress":"recipient@example.com"
      }
    ]
  }
}
```

```
    }
  ],
  "timestamp":"2017-08-05T00:41:02.669Z",
  "feedbackId":"01000157c44f053b-61b59c11-9236-11e6-8f96-7be8aexample-000000",
  "userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/60.0.3112.90 Safari/537.36",
  "complaintFeedbackType":"abuse",
  "arrivalDate":"2017-08-05T00:41:02.669Z"
},
"mail":{
  "timestamp":"2017-08-05T00:40:01.123Z",
  "source":"Sender Name <sender@example.com>",
  "sourceArn":"arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
  "sendingAccountId":"123456789012",
  "messageId":"EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "destination":[
    "recipient@example.com"
  ],
},
"headersTruncated":false,
"headers":[
  {
    "name":"From",
    "value":"Sender Name <sender@example.com>"
  },
  {
    "name":"To",
    "value":"recipient@example.com"
  },
  {
    "name":"Subject",
    "value":"Message sent from Amazon SES"
  },
  {
    "name":"MIME-Version","value":"1.0"
  },
  {
    "name":"Content-Type",
    "value":"multipart/alternative; boundary=\"-----
_Part_7298998_679725522.1516840859643\""
  }
],
"commonHeaders":{
  "from":[
    "Sender Name <sender@example.com>"
  ],
  "to":[
    "recipient@example.com"
  ],
  "messageId":"EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "subject":"Message sent from Amazon SES"
},
"tags":{
  "ses:configuration-set":[
    "ConfigSet"
  ],
  "ses:source-ip":[
    "192.0.2.0"
  ],
  "ses:from-domain":[
    "example.com"
  ],
  "ses:caller-identity":[
    "ses_user"
  ]
}
}
```



```
}
```

## 配信レコード

Amazon SES が Kinesis Data Firehose に発行する Delivery イベントレコードの例を以下に示します。

```
{
  "eventType": "Delivery",
  "mail": {
    "timestamp": "2016-10-19T23:20:52.240Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "sender@example.com"
      },
      {
        "name": "To",
        "value": "recipient@example.com"
      },
      {
        "name": "Subject",
        "value": "Message sent from Amazon SES"
      },
      {
        "name": "MIME-Version",
        "value": "1.0"
      },
      {
        "name": "Content-Type",
        "value": "text/html; charset=UTF-8"
      },
      {
        "name": "Content-Transfer-Encoding",
        "value": "7bit"
      }
    ],
    "commonHeaders": {
      "from": [
        "sender@example.com"
      ],
      "to": [
        "recipient@example.com"
      ],
      "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
      "subject": "Message sent from Amazon SES"
    },
    "tags": {
      "ses:configuration-set": [
        "ConfigSet"
      ],
      "ses:source-ip": [
        "192.0.2.0"
      ],
      "ses:from-domain": [
        "example.com"
      ]
    }
  }
}
```

```
    "ses:caller-identity": [
      "ses_user"
    ],
    "ses:outgoing-ip": [
      "192.0.2.0"
    ],
    "myCustomTag1": [
      "myCustomTagValue1"
    ],
    "myCustomTag2": [
      "myCustomTagValue2"
    ]
  }
},
"delivery": {
  "timestamp": "2016-10-19T23:21:04.133Z",
  "processingTimeMillis": 11893,
  "recipients": [
    "recipient@example.com"
  ],
  "smtpResponse": "250 2.6.0 Message received",
  "reportingMTA": "mta.example.com"
}
}
```

## Send レコード

Amazon SES が Kinesis Data Firehose に発行する Send イベントレコードの例を以下に示します。

```
{
  "eventType": "Send",
  "mail": {
    "timestamp": "2016-10-14T05:02:16.645Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "sender@example.com"
      },
      {
        "name": "To",
        "value": "recipient@example.com"
      },
      {
        "name": "Subject",
        "value": "Message sent from Amazon SES"
      },
      {
        "name": "MIME-Version",
        "value": "1.0"
      },
      {
        "name": "Content-Type",
        "value": "multipart/mixed; boundary=\"----=_Part_0_716996660.1476421336341\""
      },
      {
        "name": "X-SES-MESSAGE-TAGS",
```

```
        "value": "myCustomTag1=myCustomTagValue1, myCustomTag2=myCustomTagValue2"
    }
  ],
  "commonHeaders": {
    "from": [
      "sender@example.com"
    ],
    "to": [
      "recipient@example.com"
    ],
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "subject": "Message sent from Amazon SES"
  },
  "tags": {
    "ses:configuration-set": [
      "ConfigSet"
    ],
    "ses:source-ip": [
      "192.0.2.0"
    ],
    "ses:from-domain": [
      "example.com"
    ],
    "ses:caller-identity": [
      "ses_user"
    ],
    "myCustomTag1": [
      "myCustomTagValue1"
    ],
    "myCustomTag2": [
      "myCustomTagValue2"
    ]
  }
},
"send": {}
}
```

## Reject レコード

Amazon SES が Kinesis Data Firehose に発行する Reject イベントレコードの例を以下に示します。

```
{
  "eventType": "Reject",
  "mail": {
    "timestamp": "2016-10-14T17:38:15.211Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "sender@example.com"
    ]
  },
  "headersTruncated": false,
  "headers": [
    {
      "name": "From",
      "value": "sender@example.com"
    },
    {
      "name": "To",
      "value": "recipient@example.com"
    },
    {
      "name": "Subject",
      "value": "Message sent from Amazon SES"
    }
  ]
}
```

```
    },
    {
      "name": "MIME-Version",
      "value": "1.0"
    },
    {
      "name": "Content-Type",
      "value": "multipart/mixed; boundary=\"qMm9M+Fa2AknHoGS\""
    },
    {
      "name": "X-SES-MESSAGE-TAGS",
      "value": "myCustomTag1=myCustomTagValue1, myCustomTag2=myCustomTagValue2"
    }
  ],
  "commonHeaders": {
    "from": [
      "sender@example.com"
    ],
    "to": [
      "recipient@example.com"
    ],
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "subject": "Message sent from Amazon SES"
  },
  "tags": {
    "ses:configuration-set": [
      "ConfigSet"
    ],
    "ses:source-ip": [
      "192.0.2.0"
    ],
    "ses:from-domain": [
      "example.com"
    ],
    "ses:caller-identity": [
      "ses_user"
    ],
    "myCustomTag1": [
      "myCustomTagValue1"
    ],
    "myCustomTag2": [
      "myCustomTagValue2"
    ]
  }
},
"reject": {
  "reason": "Bad content"
}
}
```

## Open レコード

Amazon SES が Kinesis Data Firehose に発行する Open イベントレコードの例を次に示します。

```
{
  "eventType": "Open",
  "mail": {
    "commonHeaders": {
      "from": [
        "sender@example.com"
      ],
      "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
      "subject": "Message sent from Amazon SES",
      "to": [
        "recipient@example.com"
      ]
    }
  }
}
```

```
]
},
"destination": [
  "recipient@example.com"
],
"headers": [
  {
    "name": "X-SES-CONFIGURATION-SET",
    "value": "ConfigSet"
  },
  {
    "name": "X-SES-MESSAGE-TAGS",
    "value": "myCustomTag1=myCustomValue1, myCustomTag2=myCustomValue2"
  },
  {
    "name": "From",
    "value": "sender@example.com"
  },
  {
    "name": "To",
    "value": "recipient@example.com"
  },
  {
    "name": "Subject",
    "value": "Message sent from Amazon SES"
  },
  {
    "name": "MIME-Version",
    "value": "1.0"
  },
  {
    "name": "Content-Type",
    "value": "multipart/alternative; boundary=\"XBoundary\""
  }
],
"headersTruncated": false,
"messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
"sendingAccountId": "123456789012",
"source": "sender@example.com",
"tags": {
  "myCustomTag1": [
    "myCustomValue1"
  ],
  "myCustomTag2": [
    "myCustomValue2"
  ],
  "ses:caller-identity": [
    "ses-user"
  ],
  "ses:configuration-set": [
    "ConfigSet"
  ],
  "ses:from-domain": [
    "example.com"
  ],
  "ses:source-ip": [
    "192.0.2.0"
  ]
},
"timestamp": "2017-08-09T21:59:49.927Z"
},
"open": {
  "ipAddress": "192.0.2.1",
  "timestamp": "2017-08-09T22:00:19.652Z",
  "userAgent": "Mozilla/5.0 (iPhone; CPU iPhone OS 10_3_3 like Mac OS X)
AppleWebKit/603.3.8 (KHTML, like Gecko) Mobile/14G60"
```

```
}  
}
```

## Click レコード

Amazon SES が Kinesis Data Firehose に発行する click イベントレコードの例を以下に示します。

```
{  
  "eventType": "Click",  
  "click": {  
    "ipAddress": "192.0.2.1",  
    "link": "http://docs.aws.amazon.com/ses/latest/DeveloperGuide/send-email-smtp.html",  
    "linkTags": {  
      "samplekey0": [  
        "samplevalue0"  
      ],  
      "samplekey1": [  
        "samplevalue1"  
      ]  
    },  
    "timestamp": "2017-08-09T23:51:25.570Z",  
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like  
    Gecko) Chrome/60.0.3112.90 Safari/537.36"  
  },  
  "mail": {  
    "commonHeaders": {  
      "from": [  
        "sender@example.com"  
      ],  
      "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",  
      "subject": "Message sent from Amazon SES",  
      "to": [  
        "recipient@example.com"  
      ]  
    },  
    "destination": [  
      "recipient@example.com"  
    ],  
    "headers": [  
      {  
        "name": "X-SES-CONFIGURATION-SET",  
        "value": "ConfigSet"  
      },  
      {  
        "name": "X-SES-MESSAGE-TAGS",  
        "value": "myCustomTag1=myCustomValue1, myCustomTag2=myCustomValue2"  
      },  
      {  
        "name": "From",  
        "value": "sender@example.com"  
      },  
      {  
        "name": "To",  
        "value": "recipient@example.com"  
      },  
      {  
        "name": "Subject",  
        "value": "Message sent from Amazon SES"  
      },  
      {  
        "name": "MIME-Version",  
        "value": "1.0"  
      },  
      {  
        "name": "X-SES-CONFIGURATION-SET",  
        "value": "ConfigSet"  
      }  
    ]  
  }  
}
```

```
    "name": "Content-Type",
    "value": "multipart/alternative; boundary=\"XBoundary\""
  },
  {
    "name": "Message-ID",
    "value": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000"
  }
],
"headersTruncated": false,
"messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
"sendingAccountId": "123456789012",
"source": "sender@example.com",
"tags": {
  "myCustomTag1": [
    "myCustomValue1"
  ],
  "myCustomTag2": [
    "myCustomValue2"
  ],
  "ses:caller-identity": [
    "ses_user"
  ],
  "ses:configuration-set": [
    "ConfigSet"
  ],
  "ses:from-domain": [
    "example.com"
  ],
  "ses:source-ip": [
    "192.0.2.0"
  ]
},
"timestamp": "2017-08-09T23:50:05.795Z"
}
```

### レンダリング失敗レコード

Amazon SES が Kinesis Data Firehose に発行する Rendering Failure イベントレコードの例を以下に示します。

```
{
  "eventType": "Rendering Failure",
  "mail": {
    "timestamp": "2018-01-22T18:43:06.197Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "tags": {
      "ses:configuration-set": [
        "ConfigSet"
      ]
    }
  },
  "failure": {
    "errorMessage": "Attribute 'attributeName' is not present in the rendering data.",
    "templateName": "MyTemplate"
  }
}
```

## DeliveryDelay レコード

Amazon SES が Kinesis Data Firehose に発行する DeliveryDelay イベントレコードの例を以下に示します。

```
{
  "eventType": "DeliveryDelay",
  "mail": {
    "timestamp": "2020-06-16T00:15:40.641Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "tags": {
      "ses:configuration-set": [
        "ConfigSet"
      ]
    }
  },
  "deliveryDelay": {
    "timestamp": "2020-06-16T00:25:40.095Z",
    "delayType": "TransientCommunicationFailure",
    "expirationTime": "2020-06-16T00:25:40.914Z",
    "delayedRecipients": [
      {
        "emailAddress": "recipient@example.com",
        "status": "4.4.1",
        "diagnosticCode": "smtp; 421 4.4.1 Unable to connect to remote host"
      }
    ]
  }
}
```

## Amazon SNS からの Amazon SES イベントデータの解釈

Amazon SES は、E メール送信イベントを JSON レコードとして Amazon Simple Notification Service (Amazon SNS) に発行します。次に、Amazon SNS は、イベント送信先に関連付けられている Amazon SNS トピックをサブスクライブしているエンドポイントに、通知を送信します。Amazon SNS でのトピックおよびサブスクリプションの設定に関する詳細は、Amazon Simple Notification Service デベロッパーガイドの「[スタート方法](#)」を参照してください。

レコードコンテンツの説明とレコードの例については、以下のセクションを参照してください。

- [イベントレコードのコンテンツ \(p. 328\)](#)
- [イベントレコードの例 \(p. 338\)](#)

## Amazon SES が Amazon SNS に発行するイベントデータのコンテンツ

Amazon SES は、Amazon Simple Notification Service に E メール送信イベントレコードを JSON 形式で発行します。

トップレベル JSON オブジェクトには、eventType 文字列と mail オブジェクトに加え、イベントのタイプに応じて Bounce オブジェクト、Complaint オブジェクト、Delivery オブジェクト、Send オブジェクト、Reject オブジェクト、Open オブジェクト、Click オブジェクト、Rendering Failure オブジェクトまたは DeliveryDelay オブジェクトが含まれます。

これらのすべての通知タイプのレコード例については、「[Amazon SES が Amazon SNS に発行するイベントデータの例 \(p. 338\)](#)」を参照してください。



このセクションのトピック:

- [トップレベル JSON オブジェクト \(p. 329\)](#)
- [Mail オブジェクト \(p. 330\)](#)
- [Bounce オブジェクト \(p. 331\)](#)
- [苦情のオブジェクト \(p. 333\)](#)
- [配信オブジェクト \(p. 335\)](#)
- [Send オブジェクト \(p. 335\)](#)
- [Reject オブジェクト \(p. 335\)](#)
- [Open オブジェクト \(p. 336\)](#)
- [Click オブジェクト \(p. 336\)](#)
- [レンダリング失敗オブジェクト \(p. 336\)](#)
- [DeliveryDelay オブジェクト \(p. 336\)](#)

## トップレベル JSON オブジェクト

メール送信イベントレコードのトップレベル JSON オブジェクトには、以下のフィールドが含まれています。

フィールド名	説明
eventType	イベントのタイプを説明する文字列。可能な値: Delivery、Send、Reject、Open、Click、Bounce、Complaint、Failure、または DeliveryDelay。
mail	イベントを作成した E メールに関する情報を含む JSON オブジェクト。
bounce	このフィールドは、eventType が Bounce の場合のみ表示されます。このフィールドには、バウンスに関する情報が含まれています。
complaint	このフィールドは、eventType が Complaint の場合のみ表示されます。このフィールドには、苦情に関する情報が含まれています。
delivery	このフィールドは、eventType が Delivery の場合のみ表示されます。このフィールドには、配信に関する情報が含まれています。
send	このフィールドは、eventType が Send の場合のみ表示されます。
reject	このフィールドは、eventType が Reject の場合のみ表示されます。このフィールドには、拒否に関する情報が含まれています。
open	このフィールドは、eventType が Open の場合のみ表示されます。このフィールドには、オープンイベントに関する情報が含まれています。
click	このフィールドは、eventType が Click の場合のみ表示されます。このフィールドには、クリックイベントに関する情報が含まれています。
failure	このフィールドは、eventType が Rendering Failure の場合のみ表示されます。このフィールド

フィールド名	説明
	ドには、レンダリング失敗イベントに関する情報が含まれています。
deliveryDelay	このフィールドは、eventType が DeliveryDelay の場合のみ表示されます。Eメールの遅延配信に関する情報が含まれています。

## Mail オブジェクト

各メール送信イベントレコードには、mail オブジェクトの元のメールに関する情報が含まれています。mail オブジェクトについての情報を含む JSON オブジェクトには次のフィールドが含まれます。

フィールド名	説明
timestamp	メッセージが送信された、ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) の日時。
messageId	Amazon SES がメッセージに割り当てた一意の ID。ユーザーがメッセージを送信すると、Amazon SES がこの値を返します。  Note  このメッセージ ID は Amazon SES によって割り当てられます。元の Eメールのメッセージ ID は、headers オブジェクトの commonHeaders および mail フィールドにあります。
source	メッセージの送信元の E メールアドレス (エンベロープ MAIL FROM アドレス)。
sourceArn	Eメールの送信に使用された ID の Amazon リソースネーム (ARN)。送信認可の場合、sourceArn は、代理送信者が Eメールの送信に使用することを ID 所有者により承認された ID の ARN です。送信承認の詳細については、「 <a href="#">送信承認の使用 (p. 154)</a> 」を参照してください。
sendingAccountId	Eメールの送信に使用されたアカウントの AWS アカウント ID。送信承認の場合、sendingAccountId は代理送信者のアカウント ID です。
destination	元のメールの受取人の E メールアドレスのリスト。
headersTruncated	通知でヘッダーが切り捨てられたかどうかを示す文字列。切り捨ては、ヘッダーが 10 KB を超える場合に発生します。指定できる値は true および false です。
headers	Eメールの元のヘッダーの一覧。リスト内の各ヘッダーには、name フィールドと value フィールドがあります。

フィールド名	説明
	<p>Note</p> <p>headers フィールド内のメッセージ ID は、Amazon SES に渡した元のメッセージのもので、その後 Amazon SES がメッセージに割り当てるメッセージ ID は、messageId オブジェクトの mail フィールドのもので。</p>
commonHeaders	<p>E メールによく使用される元のヘッダー一覧。リスト内の各ヘッダーには、name フィールドと value フィールドがあります。</p> <p>Note</p> <p>commonHeaders フィールド内のメッセージ ID は、Amazon SES に渡した元のメッセージのもので、その後 Amazon SES がメッセージに割り当てるメッセージ ID は、messageId オブジェクトの mail フィールドのもので。</p>

## Bounce オブジェクト

Bounce イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
bounceType	バウンスのタイプで、Amazon SES によって決定されます。
bounceSubType	バウンスのサブタイプで、Amazon SES によって決定されます。
bouncedRecipients	バウンスとなった元のメールの受取人についての情報を含むリスト。
timestamp	ISP がバウンス通知を送信した、ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) の日時。
feedbackId	バウンスの一意的 ID。
reportingMTA	<p>DSN の Reporting-MTA フィールドの値。これは、DSN に記述された配信、リレー、またはゲートウェイのオペレーションを試みた MTA (Message Transfer Authority) の値です。</p> <p>Note</p> <p>このフィールドは、配信ステータス通知 (DSN) がバウンスにアタッチされている場合にのみ表示されます。</p>

## バウンスとなった受取人

バウンスイベントには、1人の受信者に関するものと複数の受信者に関するものがあります。bouncedRecipients フィールドにはオブジェクトのリストが含まれ (バウンスを作成した E メールアドレスの受信者ごとに1つのオブジェクト)、以下のフィールドが含まれています。

フィールド名	説明
emailAddress	受取人の E メールアドレス。DSN が利用できる場合、これが DSN の Final-Recipient フィールドの値です。

オプションで、DSN がバウンスに添付されている場合、以下のフィールドも表示される場合があります。

フィールド名	説明
action	DSN の Action フィールドの値。このフィールドには、Reporting-MTA により実行された、この受取人に対してメッセージを送信しようとした結果のアクションが示されます。
status	DSN の Status フィールドの値。これは、メッセージの配信状態を示す、受取人ごとに個別の、トランスポート独立型ステータスコードです。
diagnosticCode	ステータスコードは、Reporting-MTA により発行されます。これは、DSN の Diagnostic-Code フィールドの値です。このフィールドが DSN に存在しない場合もあります (その場合 JSON オブジェクトにも表示されません)。

## バウンスのタイプ

各バウンスイベントは、以下の表に示すいずれかのタイプになります。

イベント発行システムは、Amazon SES によって再試行されなくなるハードバウンスおよびソフトバウンスのみを発行します。Permanent とマークされたバウンスを受信した場合は、対応する E メールアドレスをメーリングリストから削除する必要があります。これらは今後も送信できません。Transient バウンスは、メッセージに対してソフトバウンスが複数回発生して Amazon SES が再送を停止した場合に送信されます。最初は Transient バウンスとして返されても、将来的には同じアドレスに正常に再送できる場合があります。

bounceType	bounceSubType	説明
Undetermined	Undetermined	Amazon SES はバウンスの理由を特定できませんでした。
Permanent	General	Amazon SES は、一般的なハードバウンスを受け取りました。このタイプのバウンスを受信した場合は、メーリングリストからこの受信者のメールアドレスを削除する必要があります。
Permanent	NoEmail	Amazon SES は、送信先の E メールアドレスが存在しないことが原因である永続的なハードバウンス

bounceType	bounceSubType	説明
		スを受け取りました。このタイプのバウンスを受信した場合は、メーリングリストからこの受信者のメールアドレスを削除する必要があります。
Permanent	Suppressed	Amazon SES は、このアドレスへの送信を抑制しました。このアドレスは無効なアドレスとして最近バウンスとなった履歴があるからです。サブレッションリストからメールアドレスを削除する方法については、「 <a href="#">Amazon SES グローバルサブレッションリストの使用 (p. 199)</a> 」を参照してください。
Permanent	OnAccountSuppressionList	Amazon SES は、アドレスが <a href="#">アカウントレベルのサブレッションリスト (p. 190)</a> にあるので、このアドレスへの送信を抑制しました。
Transient	General	Amazon SES は、一般的なバウンスを受け取りました。今後、この受信者に正常に再送できる可能性があります。
Transient	MailboxFull	Amazon SES は、メールボックスが一杯であることを示すバウンスを受け取りました。今後、この受信者に正常に再送できる可能性があります。
Transient	MessageTooLarge	Amazon SES は、メッセージサイズが大きすぎることを示すバウンスを受け取りました。メッセージサイズを小さくすることで、この受信者に正常に再送できる可能性があります。
Transient	ContentRejected	Amazon SES は、コンテンツが拒否されたことを示すバウンスを受け取りました。メッセージのコンテンツを変更することで、この受信者に正常に再送できる可能性があります。
Transient	AttachmentRejected	Amazon SES は、添付ファイルが拒否されたことを示すバウンスを受け取りました。添付ファイルを削除または変更することで、この受信者に正常に再送できる可能性があります。

## 苦情のオブジェクト

Complaint イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
complainedRecipients	苦情を送信した可能性がある受取人に関する情報を含むリスト。
timestamp	ISP が苦情通知を送信した、ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) の日時。
feedbackId	苦情の一意の ID。
complaintSubType	苦情のサブタイプで、Amazon SES によって決定されます。

また、フィードバックレポートが苦情に添付されている場合、以下のフィールドが示される場合があります。

フィールド名	説明
userAgent	フィードバックレポートの User-Agent フィールドの値。これは、レポートを生成したシステムの名前とバージョンを示します。
complaintFeedbackType	ISP から受け取ったフィードバックレポートの Feedback-Type フィールドの値。これには、フィードバックのタイプが含まれます。
arrivalDate	ISO 8601 形式のフィードバックレポートの Arrival-Date または Received-Date フィールドの値 (YYYY-MM-DDThh:mm:ss.SZ)。このフィールドがレポートにない場合もあります (その場合、JSON オブジェクトにも表示されません)。

### 苦情を申告した受取人

complainedRecipients フィールドには、苦情の送信元と思われる受信者のリストが含まれます。

#### Important

ほとんどの ISP は、苦情を送信した受信者の E メールアドレスを編集します。このため、complainedRecipients フィールドには、苦情通知を発行したドメインにアドレスが含まれていて、E メールが送信された全員のリストが含まれます。

このリストの JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
emailAddress	受取人の E メールアドレス。

### 苦情のタイプ

complaintFeedbackType フィールドには以下の苦情のタイプが示されます ([Internet Assigned Numbers Authority ウェブサイト](#) に基づいて、報告する ISP により割り当てられます)。

フィールド名	説明
abuse	迷惑メールまたはその他のメール不正使用。
auth-failure	Eメールの認証障害レポート
fraud	なんらかの詐欺またはフィッシング行為を示します。
not-spam	レポートの提供者がこのメッセージをスパムではないと見なしていることを示します。このタイプは、誤ってスパムとしてタグ付けまたは分類されたメッセージを修正するために使用される場合があります。

フィールド名	説明
other	その他の登録されたタイプに該当しないフィードバックを示します。
virus	元のメッセージでウイルスが見つかったことを示します。

### 苦情のサブタイプ

complaintSubType フィールドの値は、null または OnAccountSuppressionList のいずれかになります。値が OnAccountSuppressionList の場合、Amazon SES はメッセージを受け入れますが、[アカウントレベルのサプレッションリスト \(p. 190\)](#)に含まれているため、送信を試みません。

### 配信オブジェクト

Delivery イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
timestamp	Amazon SES が受信者のメールサーバーにメールを配信した、ISO8601 形式 (YYYY-MM-DDThh:mm:ss.SZ) の日時。
processingTimeMillis	Amazon SES で送信者からリクエストを受理して、Amazon SES が受取人のメールサーバーにメッセージを渡すまでのミリ秒単位の時間。
recipients	配信イベントの適用対象となる受信者のリスト。
smtpResponse	Amazon SES から E メールを受け取ったリモート ISP による SMTP 応答メッセージ。このメッセージは、E メール、受信メールサーバー、または受信する ISP によって異なります。
reportingMTA	メールを送信する Amazon SES メールサーバーのホスト名。

### Send オブジェクト

send イベントに関する情報を含む JSON オブジェクトは常に空です。

### Reject オブジェクト

Reject イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
reason	メールが拒否された理由。このフィールドの唯一の値は Bad content です。E メール内のウイルスを Amazon SES で検出したことを意味します。メッセージが拒否されると、Amazon SES はメッセージの処理を停止し、そのメッセージを受信者のメールサーバーに配信しようとしません。

## Open オブジェクト

Open イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
ipAddress	受信側の IP アドレス
timestamp	ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) で表示された、オープンイベントが発生日時。
userAgent	受取人がメールを開くために使用したデバイスまたは E メールクライアントのユーザーエージェント。

## Click オブジェクト

click イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
ipAddress	受信側の IP アドレス
timestamp	ISO8601 形式 (YYYY-MM-DDThh:mm:ss.sZ) で表示された、クリックイベントが発生日時。
userAgent	受信者が E メールリンクをクリックするために使用したクライアントのユーザーエージェント。
link	受取人がクリックしたリンクの URL。
linkTags	ses:tags 属性を使用してリンクに追加されたタグのリスト。メール内のリンクにタグを追加する方法については、「 <a href="#">Q5. リンクに一意的識別子をタグ付けできますか? (p. 521)</a> 」の「 <a href="#">Amazon SES E メール送信メトリクスに関するよくある質問 (p. 518)</a> 」を参照してください。

## レンダリング失敗オブジェクト

Rendering Failure イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。

フィールド名	説明
templateName	Eメールの送信に使用されるテンプレートの名前。
errorMessage	レンダリング失敗に関する詳細情報を提供するメッセージ。

## DeliveryDelay オブジェクト

DeliveryDelay イベントに関する情報を含む JSON オブジェクトには以下のフィールドが含まれます。



フィールド名	説明
delayType	<p>遅延のタイプ。想定される値は次のとおりです。</p> <ul style="list-style-type: none"> <li>InternalFailure – 内部的な Amazon SES の問題により、メッセージが遅延しました。</li> <li>一般 – SMTP 通信中に一般的なエラーが発生しました。</li> <li>MailboxFull – 受取人のメールボックスがいっぱいで、追加のメッセージを受信できません。</li> <li>SpamDetected – 受取人のメールサーバーが、お客様のアカウントからの迷惑メールを大量に検出しました。</li> <li>RecipientServerError – 受取人の E メールサーバーの一時的な問題により、メッセージの配信が妨げられています。</li> <li>IPFailure – メッセージを送信する IP アドレスは、受信者の E メールプロバイダーによってブロックまたはスロットリングされています。</li> <li>TransientCommunicationGeneral – 受取人の E メールプロバイダーとの SMTP 通信中に、一時的な通信障害が発生しました。</li> <li>BYOIPHostNameLookupUnavailable – Amazon SES はユーザーの IP アドレスの DNS ホスト名を検索できませんでした。このタイプの遅延は、<a href="#">Bring Your Own IP (p. 186)</a> を使用する場合にのみ発生します。</li> <li>未確定 – Amazon SES は配信遅延の理由を判断できませんでした。</li> </ul>
delayedRecipients	E メールを受取人に関する情報を含むオブジェクト。
expirationTime	Amazon SES がメッセージの配信を停止する日時。この値は ISO 8601 形式で表示されます。
reportingMTA	遅延を報告したメッセージ転送エージェント (MTA) の IP アドレス。
timestamp	遅延が発生した日時。ISO 8601 形式で表示されます。

### 遅延受取人

delayedRecipients オブジェクトには、以下の値が含まれています。

フィールド名	説明
emailAddress	メッセージの配信が遅れる原因となった E メールアドレス。
status	配信遅延に関連付けられた SMTP ステータスコード。

フィールド名	説明
diagnosticCode	受信側メッセージ転送エージェント (MTA) によって提供される診断コード。

## Amazon SES が Amazon SNS に発行するイベントデータの例

このセクションでは、Amazon SES から Amazon SNS に発行するメール送信イベントレコードの例をイベントタイプ別に示します。

このセクションのトピック:

- [バウンスレコード \(p. 338\)](#)
- [苦情レコード \(p. 339\)](#)
- [配信レコード \(p. 340\)](#)
- [Send レコード \(p. 342\)](#)
- [Reject レコード \(p. 343\)](#)
- [Open レコード \(p. 344\)](#)
- [Click レコード \(p. 345\)](#)
- [レンダリング失敗レコード \(p. 347\)](#)
- [DeliveryDelay レコード \(p. 347\)](#)

### バウンスレコード

Amazon SES が Amazon SNS に発行する Bounce イベントレコードの例を以下に示します。

```
{
  "eventType": "Bounce",
  "bounce": {
    "bounceType": "Permanent",
    "bounceSubType": "General",
    "bouncedRecipients": [
      {
        "emailAddress": "recipient@example.com",
        "action": "failed",
        "status": "5.1.1",
        "diagnosticCode": "smtp; 550 5.1.1 user unknown"
      }
    ],
    "timestamp": "2017-08-05T00:41:02.669Z",
    "feedbackId": "01000157c44f053b-61b59c11-9236-11e6-8f96-7be8aexample-000000",
    "reportingMTA": "dsn; mta.example.com"
  },
  "mail": {
    "timestamp": "2017-08-05T00:40:02.012Z",
    "source": "Sender Name <sender@example.com>",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "Sender Name <sender@example.com>"
      }
    ]
  }
}
```

```
{
  "name": "To",
  "value": "recipient@example.com"
},
{
  "name": "Subject",
  "value": "Message sent from Amazon SES"
},
{
  "name": "MIME-Version",
  "value": "1.0"
},
{
  "name": "Content-Type",
  "value": "multipart/alternative; boundary=\"-----
_Part_7307378_1629847660.1516840721503\""
}
],
"commonHeaders": {
  "from": [
    "Sender Name <sender@example.com>"
  ],
  "to": [
    "recipient@example.com"
  ],
  "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "subject": "Message sent from Amazon SES"
},
"tags": {
  "ses:configuration-set": [
    "ConfigSet"
  ],
  "ses:source-ip": [
    "192.0.2.0"
  ],
  "ses:from-domain": [
    "example.com"
  ],
  "ses:caller-identity": [
    "ses_user"
  ]
}
}
```

## 苦情レコード

Amazon SES が Amazon SNS に発行する Complaint イベントレコードの例を以下に示します。

```
{
  "eventType": "Complaint",
  "complaint": {
    "complainedRecipients": [
      {
        "emailAddress": "recipient@example.com"
      }
    ],
    "timestamp": "2017-08-05T00:41:02.669Z",
    "feedbackId": "01000157c44f053b-61b59c11-9236-11e6-8f96-7be8aexample-000000",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36",
    "complaintFeedbackType": "abuse",
    "arrivalDate": "2017-08-05T00:41:02.669Z"
  },
  "mail": {
```

```
"timestamp": "2017-08-05T00:40:01.123Z",
"source": "Sender Name <sender@example.com>",
"sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
"sendingAccountId": "123456789012",
"messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
"destination": [
  "recipient@example.com"
],
"headersTruncated": false,
"headers": [
  {
    "name": "From",
    "value": "Sender Name <sender@example.com>"
  },
  {
    "name": "To",
    "value": "recipient@example.com"
  },
  {
    "name": "Subject",
    "value": "Message sent from Amazon SES"
  },
  {
    "name": "MIME-Version", "value": "1.0"
  },
  {
    "name": "Content-Type",
    "value": "multipart/alternative; boundary=\"-----
_Part_7298998_679725522.1516840859643\""
  }
],
"commonHeaders": {
  "from": [
    "Sender Name <sender@example.com>"
  ],
  "to": [
    "recipient@example.com"
  ],
  "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "subject": "Message sent from Amazon SES"
},
"tags": {
  "ses:configuration-set": [
    "ConfigSet"
  ],
  "ses:source-ip": [
    "192.0.2.0"
  ],
  "ses:from-domain": [
    "example.com"
  ],
  "ses:caller-identity": [
    "ses_user"
  ]
}
}
```

## 配信レコード

Amazon SES が Amazon SNS に発行する Delivery イベントレコードの例を以下に示します。

```
{
  "eventType": "Delivery",
  "mail": {
```

```
"timestamp": "2016-10-19T23:20:52.240Z",
"source": "sender@example.com",
"sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
"sendingAccountId": "123456789012",
"messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
"destination": [
  "recipient@example.com"
],
"headersTruncated": false,
"headers": [
  {
    "name": "From",
    "value": "sender@example.com"
  },
  {
    "name": "To",
    "value": "recipient@example.com"
  },
  {
    "name": "Subject",
    "value": "Message sent from Amazon SES"
  },
  {
    "name": "MIME-Version",
    "value": "1.0"
  },
  {
    "name": "Content-Type",
    "value": "text/html; charset=UTF-8"
  },
  {
    "name": "Content-Transfer-Encoding",
    "value": "7bit"
  }
],
"commonHeaders": {
  "from": [
    "sender@example.com"
  ],
  "to": [
    "recipient@example.com"
  ],
  "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "subject": "Message sent from Amazon SES"
},
"tags": {
  "ses:configuration-set": [
    "ConfigSet"
  ],
  "ses:source-ip": [
    "192.0.2.0"
  ],
  "ses:from-domain": [
    "example.com"
  ],
  "ses:caller-identity": [
    "ses_user"
  ],
  "ses:outgoing-ip": [
    "192.0.2.0"
  ],
  "myCustomTag1": [
    "myCustomTagValue1"
  ],
  "myCustomTag2": [
    "myCustomTagValue2"
  ]
}
```

```
    ]
  }
},
"delivery": {
  "timestamp": "2016-10-19T23:21:04.133Z",
  "processingTimeMillis": 11893,
  "recipients": [
    "recipient@example.com"
  ],
  "smtpResponse": "250 2.6.0 Message received",
  "reportingMTA": "mta.example.com"
}
}
```

## Send レコード

Amazon SES が Amazon SNS に発行する send イベントレコードの例を以下に示します。

```
{
  "eventType": "Send",
  "mail": {
    "timestamp": "2016-10-14T05:02:16.645Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "sender@example.com"
      },
      {
        "name": "To",
        "value": "recipient@example.com"
      },
      {
        "name": "Subject",
        "value": "Message sent from Amazon SES"
      },
      {
        "name": "MIME-Version",
        "value": "1.0"
      },
      {
        "name": "Content-Type",
        "value": "multipart/mixed; boundary=\"----=_Part_0_716996660.1476421336341\""
      },
      {
        "name": "X-SES-MESSAGE-TAGS",
        "value": "myCustomTag1=myCustomTagValue1, myCustomTag2=myCustomTagValue2"
      }
    ],
    "commonHeaders": {
      "from": [
        "sender@example.com"
      ],
      "to": [
        "recipient@example.com"
      ],
      "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    }
  }
}
```

```
    "subject": "Message sent from Amazon SES"
  },
  "tags": {
    "ses:configuration-set": [
      "ConfigSet"
    ],
    "ses:source-ip": [
      "192.0.2.0"
    ],
    "ses:from-domain": [
      "example.com"
    ],
    "ses:caller-identity": [
      "ses_user"
    ],
    "myCustomTag1": [
      "myCustomTagValue1"
    ],
    "myCustomTag2": [
      "myCustomTagValue2"
    ]
  }
},
"send": {}
}
```

## Reject レコード

Amazon SES が Amazon SNS に発行する Reject イベントレコードの例を以下に示します。

```
{
  "eventType": "Reject",
  "mail": {
    "timestamp": "2016-10-14T17:38:15.211Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "sender@example.com"
    ],
    "headersTruncated": false,
    "headers": [
      {
        "name": "From",
        "value": "sender@example.com"
      },
      {
        "name": "To",
        "value": "recipient@example.com"
      },
      {
        "name": "Subject",
        "value": "Message sent from Amazon SES"
      },
      {
        "name": "MIME-Version",
        "value": "1.0"
      },
      {
        "name": "Content-Type",
        "value": "multipart/mixed; boundary=\"qMm9M+Fa2AknHoGS\""
      },
      {
        "name": "X-SES-MESSAGE-TAGS",
```

```
    "value": "myCustomTag1=myCustomTagValue1, myCustomTag2=myCustomTagValue2"
  }
],
"commonHeaders": {
  "from": [
    "sender@example.com"
  ],
  "to": [
    "recipient@example.com"
  ],
  "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "subject": "Message sent from Amazon SES"
},
"tags": {
  "ses:configuration-set": [
    "ConfigSet"
  ],
  "ses:source-ip": [
    "192.0.2.0"
  ],
  "ses:from-domain": [
    "example.com"
  ],
  "ses:caller-identity": [
    "ses_user"
  ],
  "myCustomTag1": [
    "myCustomTagValue1"
  ],
  "myCustomTag2": [
    "myCustomTagValue2"
  ]
}
},
"reject": {
  "reason": "Bad content"
}
}
```

## Open レコード

次に示すのが、Amazon SES が Amazon SNS に発行した Open イベントレコードの例です。

```
{
  "eventType": "Open",
  "mail": {
    "commonHeaders": {
      "from": [
        "sender@example.com"
      ],
      "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
      "subject": "Message sent from Amazon SES",
      "to": [
        "recipient@example.com"
      ]
    },
    "destination": [
      "recipient@example.com"
    ],
    "headers": [
      {
        "name": "X-SES-CONFIGURATION-SET",
        "value": "ConfigSet"
      },
      {

```



```
    "name": "X-SES-MESSAGE-TAGS",
    "value": "myCustomTag1=myCustomValue1, myCustomTag2=myCustomValue2"
  },
  {
    "name": "From",
    "value": "sender@example.com"
  },
  {
    "name": "To",
    "value": "recipient@example.com"
  },
  {
    "name": "Subject",
    "value": "Message sent from Amazon SES"
  },
  {
    "name": "MIME-Version",
    "value": "1.0"
  },
  {
    "name": "Content-Type",
    "value": "multipart/alternative; boundary=\"XBoundary\""
  }
],
"headersTruncated": false,
"messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
"sendingAccountId": "123456789012",
"source": "sender@example.com",
"tags": {
  "myCustomTag1": [
    "myCustomValue1"
  ],
  "myCustomTag2": [
    "myCustomValue2"
  ],
  "ses:caller-identity": [
    "ses-user"
  ],
  "ses:configuration-set": [
    "ConfigSet"
  ],
  "ses:from-domain": [
    "example.com"
  ],
  "ses:source-ip": [
    "192.0.2.0"
  ]
},
"timestamp": "2017-08-09T21:59:49.927Z"
},
"open": {
  "ipAddress": "192.0.2.1",
  "timestamp": "2017-08-09T22:00:19.652Z",
  "userAgent": "Mozilla/5.0 (iPhone; CPU iPhone OS 10_3_3 like Mac OS X)
AppleWebKit/603.3.8 (KHTML, like Gecko) Mobile/14G60"
}
}
```

### Click レコード

Amazon SES が Amazon SNS に発行する Click イベントレコードの例を以下に示します。

```
{
  "eventType": "Click",
  "click": {
```

```
"ipAddress": "192.0.2.1",
"link": "http://docs.aws.amazon.com/ses/latest/DeveloperGuide/send-email-smtp.html",
"linkTags": {
  "samplekey0": [
    "samplevalue0"
  ],
  "samplekey1": [
    "samplevalue1"
  ]
},
"timestamp": "2017-08-09T23:51:25.570Z",
"userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/60.0.3112.90 Safari/537.36"
},
"mail": {
  "commonHeaders": {
    "from": [
      "sender@example.com"
    ],
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "subject": "Message sent from Amazon SES",
    "to": [
      "recipient@example.com"
    ]
  },
  "destination": [
    "recipient@example.com"
  ],
  "headers": [
    {
      "name": "X-SES-CONFIGURATION-SET",
      "value": "ConfigSet"
    },
    {
      "name": "X-SES-MESSAGE-TAGS",
      "value": "myCustomTag1=myCustomValue1, myCustomTag2=myCustomValue2"
    },
    {
      "name": "From",
      "value": "sender@example.com"
    },
    {
      "name": "To",
      "value": "recipient@example.com"
    },
    {
      "name": "Subject",
      "value": "Message sent from Amazon SES"
    },
    {
      "name": "MIME-Version",
      "value": "1.0"
    },
    {
      "name": "Content-Type",
      "value": "multipart/alternative; boundary=\"XBoundary\""
    },
    {
      "name": "Message-ID",
      "value": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000"
    }
  ],
  "headersTruncated": false,
  "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
  "sendingAccountId": "123456789012",
  "source": "sender@example.com",
```

```
"tags": {
  "myCustomTag1": [
    "myCustomValue1"
  ],
  "myCustomTag2": [
    "myCustomValue2"
  ],
  "ses:caller-identity": [
    "ses_user"
  ],
  "ses:configuration-set": [
    "ConfigSet"
  ],
  "ses:from-domain": [
    "example.com"
  ],
  "ses:source-ip": [
    "192.0.2.0"
  ]
},
"timestamp": "2017-08-09T23:50:05.795Z"
}
```

### レンダリング失敗レコード

Amazon SES が Amazon SNS に発行する Rendering Failure イベントレコードの例を以下に示します。

```
{
  "eventType": "Rendering Failure",
  "mail": {
    "timestamp": "2018-01-22T18:43:06.197Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
    "destination": [
      "recipient@example.com"
    ],
    "headersTruncated": false,
    "tags": {
      "ses:configuration-set": [
        "ConfigSet"
      ]
    }
  },
  "failure": {
    "errorMessage": "Attribute 'attributeName' is not present in the rendering data.",
    "templateName": "MyTemplate"
  }
}
```

### DeliveryDelay レコード

Amazon SES が Amazon SNS に発行する DeliveryDelay イベントレコードの例を以下に示します。

```
{
  "eventType": "DeliveryDelay",
  "mail": {
    "timestamp": "2020-06-16T00:15:40.641Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",

```

```
"sendingAccountId": "123456789012",
"messageId": "EXAMPLE7c191be45-e9aedb9a-02f9-4d12-a87d-dd0099a07f8a-000000",
"destination": [
  "recipient@example.com"
],
"headersTruncated": false,
"tags": {
  "ses:configuration-set": [
    "ConfigSet"
  ]
},
"deliveryDelay": {
  "timestamp": "2020-06-16T00:25:40.095Z",
  "delayType": "TransientCommunicationFailure",
  "expirationTime": "2020-06-16T00:25:40.914Z",
  "delayedRecipients": [
    {
      "emailAddress": "recipient@example.com",
      "status": "4.4.1",
      "diagnosticCode": "smtp; 421 4.4.1 Unable to connect to remote host"
    }
  ]
}
}
```

## イベント発行チュートリアル

このセクションでは、データを分析および可視化できる AWS のサービスでの Amazon SES イベント発行方法のチュートリアルを示します。

このセクションのトピック:

- [Amazon Redshift を使用したメール送信イベントの分析 \(p. 348\)](#)
- [Amazon CloudWatch での E メール送信イベントのグラフ化 \(p. 360\)](#)
- [Amazon Kinesis Data Analytics を使用した E メール送信イベントの分析 \(p. 362\)](#)

## Amazon Redshift を使用したメール送信イベントの分析

このチュートリアルでは、Amazon Redshift にデータを発行する Amazon Kinesis Data Firehose 配信ストリームに、イベントを送信する Amazon SES E メールを発行します。その後、Amazon Redshift データベースに接続し、このデータベースで特定の条件を満たす Amazon SES E メール送信イベントを SQL クエリツールでクエリします。

以下のセクションでは、次のプロセスを説明します。

- [前提条件 \(p. 348\)](#)
- [ステップ 1: Amazon Redshift クラスターを作成する \(p. 349\)](#)
- [ステップ 2: Amazon Redshift クラスターに接続する \(p. 349\)](#)
- [ステップ 3: データベーステーブルを作成する \(p. 352\)](#)
- [ステップ 4: Kinesis Data Firehose 配信ストリームを作成する \(p. 354\)](#)
- [ステップ 5: 設定セットをセットアップする \(p. 357\)](#)
- [ステップ 6: メールを送信する \(p. 358\)](#)
- [ステップ 7: E メール送信イベントのクエリを実行する \(p. 359\)](#)

### Prerequisites

このチュートリアルでは、以下が必要になります。

- AWS アカウント - AWS が提供するウェブサービスにアクセスするには、まず <https://aws.amazon.com/> で AWS アカウントを作成する必要があります。
- 検証済み E メールアドレス - Amazon SES 使用して E メールを送信するには、「From」アドレスまたはドメインを検証して、アドレスの所有者本人であることを示す必要があります。サンドボックスの場合、「To」アドレスも確認する必要があります。E メールアドレスまたはドメイン全体を検証できませんが、このチュートリアルでは、Amazon SES コンソール (最も簡単に E メールを送信する方法) から E メールを送信できるように検証済み E メールアドレスが必要です。メールアドレスの確認については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- SQL クエリツール - Amazon Redshift には、SQL クライアントツールまたはライブラリが用意されていないが、インストールされていないため、Amazon SES イベントを含む Amazon Redshift クラスターへのアクセスに使用できるツールまたはライブラリをインストールする必要があります。このチュートリアルでは、DBMS に依存しない無料のクロスプラットフォーム SQL クエリツールである [SQL Workbench/J](#) を使用します。このセクションでは、SQL Workbench/J のインストール手順を示します。

### SQL Workbench/J をインストールする方法

1. [SQL Workbench/J のソフトウェアライセンス](#)を確認します。
2. [SQL Workbench/J ウェブサイト](#) に移動し、使用するオペレーティングシステムに適したパッケージをダウンロードします。
3. 「[Installing and starting SQL Workbench/J](#)」にアクセスして、SQL Workbench/J をインストールします。

#### Important

SQL Workbench/J に必要な Java ランタイムのバージョンを確認して、必ずそのバージョンを使用してください。違うバージョンでは、このクライアントアプリケーションは動作しません。

4. 「[JDBC 接続を設定する](#)」に移動し、Amazon Redshift JDBC ドライバーをダウンロードして、SQL Workbench/J をクラスターに接続できるようにします。

### 次のステップ

[ステップ 1: Amazon Redshift クラスターを作成する \(p. 349\)](#)

## ステップ 1: Amazon Redshift クラスターを作成する

Amazon Redshift クラスターを作成するには、[Amazon Redshift コンソール](#)を選択し、クラスターの起動を選択します。ウィザードでは、クラスターのオプションを選択します。ほとんどのオプションにはデフォルト値があります。

このシンプルなチュートリアルでは、クラスター名とパスワードを入力した後、すべてのデフォルト値を使用できます。Amazon SES イベント発行に固有の値を設定する必要はありません。

#### Important

このチュートリアル用にデプロイしたクラスターは、実働環境で実行されます。実行中は AWS アカウントに課金されます。不要な課金を回避するため、クラスターは実行後に削除してください。料金情報については、[Amazon Redshift の料金ページ](#)を参照してください。

### 次のステップ

[ステップ 2: Amazon Redshift クラスターに接続する \(p. 349\)](#)

## ステップ 2: Amazon Redshift クラスターに接続する

ここでは、SQL クライアントツールを使用してクラスターに接続します。本チュートリアルでは、[前提条件のセクション \(p. 348\)](#)でインストールした SQL Workbench/J クライアントを使用します。

次の手順に従って、このセクションを実行します。

- [接続文字列の取得 \(p. 350\)](#)
- [SQL Workbench/J からクラスターへの接続 \(p. 350\)](#)

### 接続文字列の取得

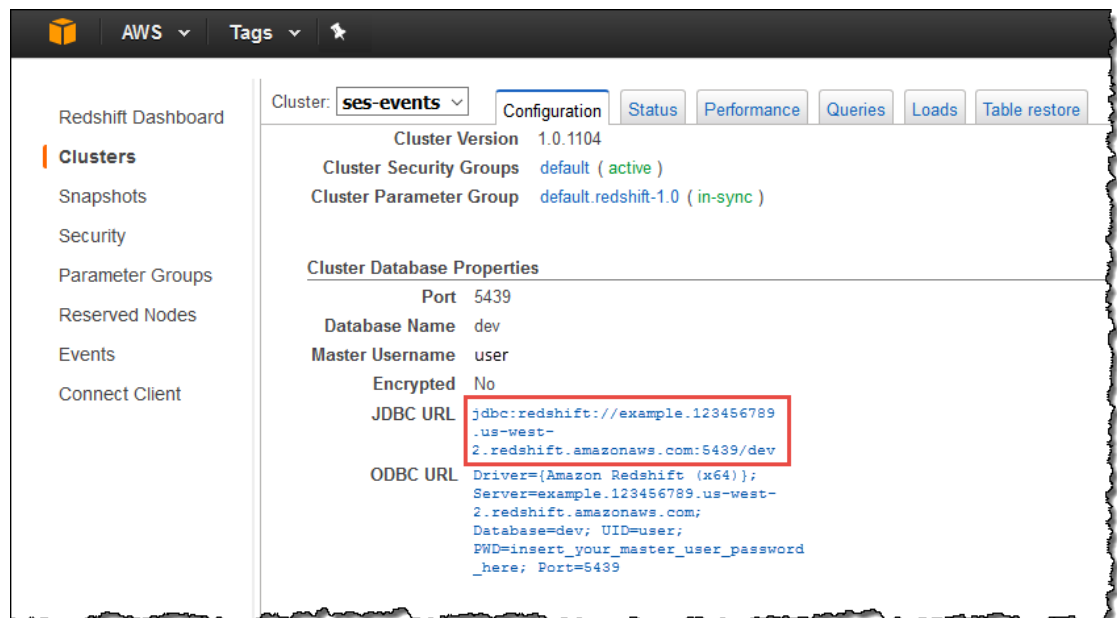
次の手順では、SQL Workbench/J から Amazon Redshift クラスターに接続するときに必要な接続文字列を取得する方法を示します。

接続文字列を取得するには

1. Amazon Redshift コンソールのナビゲーションペインで、クラスターを選択します。
2. クラスターを開くには、クラスター名を選択します。
3. [設定] タブで、[クラスターデータベースのプロパティ] の下にある、クラスターの JDBC URL をコピーします。

#### Note

クラスターのエンドポイントは、クラスターが作成されて使用可能な状態になるまで使用できません。

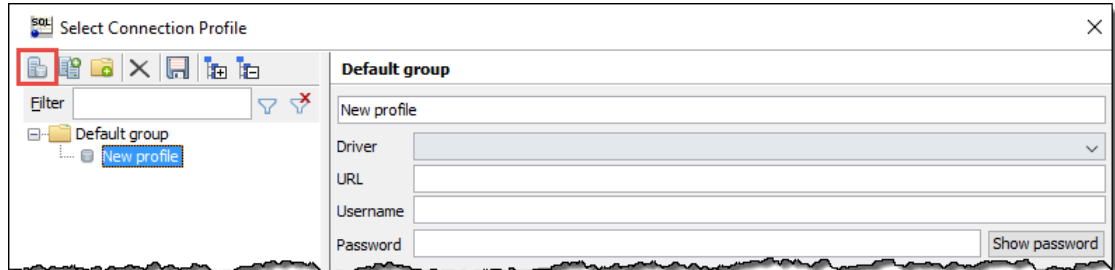


### SQL Workbench/J からクラスターへの接続

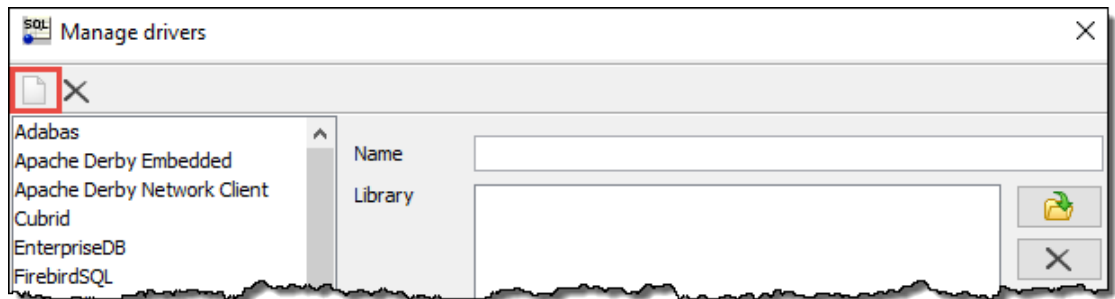
次の手順では、SQL Workbench/J からクラスターに接続する方法を示します。この手順は、「[Prerequisites \(p. 348\)](#)」の説明に従ってコンピュータに SQL Workbench/J をインストールしていることを前提としています。

SQL Workbench/J からクラスターに接続するには

1. SQL Workbench/J を開きます。
2. [File (ファイル)] を選択し、[Connect window (ウィンドウの接続)] を選択します。
3. [Create a new connection profile] ボタンを選択します。



4. [New profile] テキストボックスに、プロファイルの名前を入力します。
5. ウィンドウの下部の左側で、[Manage Drivers] を選択します。
6. [Manage Drivers] ダイアログボックスで、[Create a new entry] ボタンを選択し、次のようにドライバーを追加します。



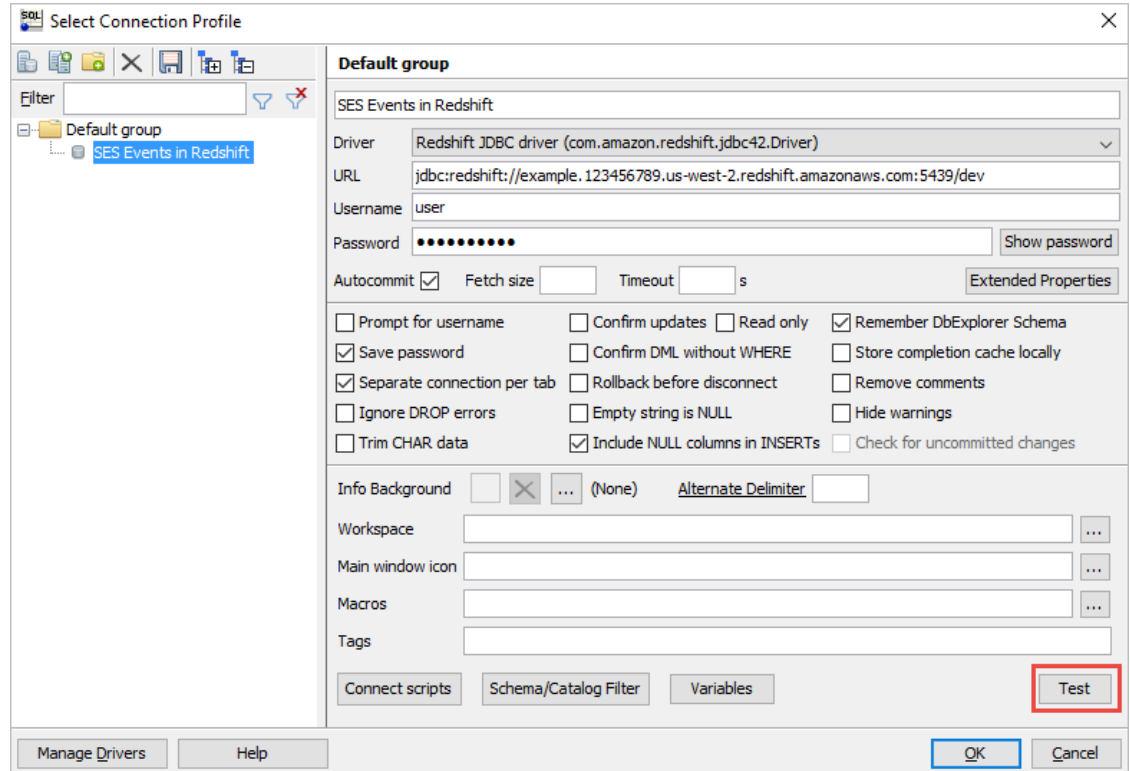
- a. [Name (名前)] ボックスにドライバーの名前を入力します。
- b. [Library] の横で、フォルダアイコンを選択します。
- c. 「[JDBC 接続を設定する](#)」でダウンロードしたドライバーの場所に移動し、ドライバーを選択して [開く] を選択します。
- d. [OK] を選択します。

[Select Connection Profile] ダイアログボックスに自動的に戻ります。

7. [Driver] で、追加したドライバーを選択します。
8. [URL] で、[Amazon Redshift コンソール](#)からコピーした JDBC URL を貼り付けます。
9. [ユーザーネーム] で、[Amazon Redshift クラスターのセットアップ \(p. 349\)](#)時に選択したユーザーネームを入力します。
10. [パスワード]で、Amazon Redshift クラスターのセットアップ時に選択したパスワードを入力します。
11. [Autocommit] を選択します。
12. 接続をテストするには、[Test] を選択します。

#### Note

接続試行がタイムアウトした場合、状況によっては IP アドレスをセキュリティグループに追加して、それらの IP アドレスからの受信トラフィックが許可されるようにする必要があります。詳細については、[Amazon Redshift データベースデベロッパーガイド](#)にある[接続が拒否または失敗する](#)を参照してください。



13. 上部メニューバーで、[Save profile list] ボタンを選択します。

14. [OK] を選択します。

SQL Workbench/J が Amazon Redshift クラスターに接続されます。

## 次のステップ

[ステップ 3: データベーステーブルを作成する \(p. 352\)](#)

## ステップ 3: データベーステーブルを作成する

Amazon Redshift で初期データベースに接続した後、通常はその初期データベースを新しいデータベースを作成するベースとして使用します。ただし、このシンプルなチュートリアルでは、初期データベース内に直接 Amazon SES イベント発行データを保持するテーブルを作成します。

このチュートリアルでは、[メール送信イベントレコード \(p. 309\)](#)内の以下のフィールドに関心があると仮定します。mail.tags.campaign を除くこれらのすべてのフィールドは、Amazon SES により自動的に用意されます。mail.tags.campaign フィールドについては、「campaign」で [ステップ 6: メールを送信する \(p. 358\)](#) をメッセージタグとして使用してメールを送信するときに紹介します。

- mail.messageId
- eventType
- mail.sendingAccountId
- mail.timestamp
- mail.destination
- mail.tags.ses:configuration-set
- mail.tags.campaign



データベースのこの情報にアクセスするには、テーブルを作成する必要があります。次の手順では、データベースにテーブルを作成するときこの情報を指定する方法を説明します。

#### Note

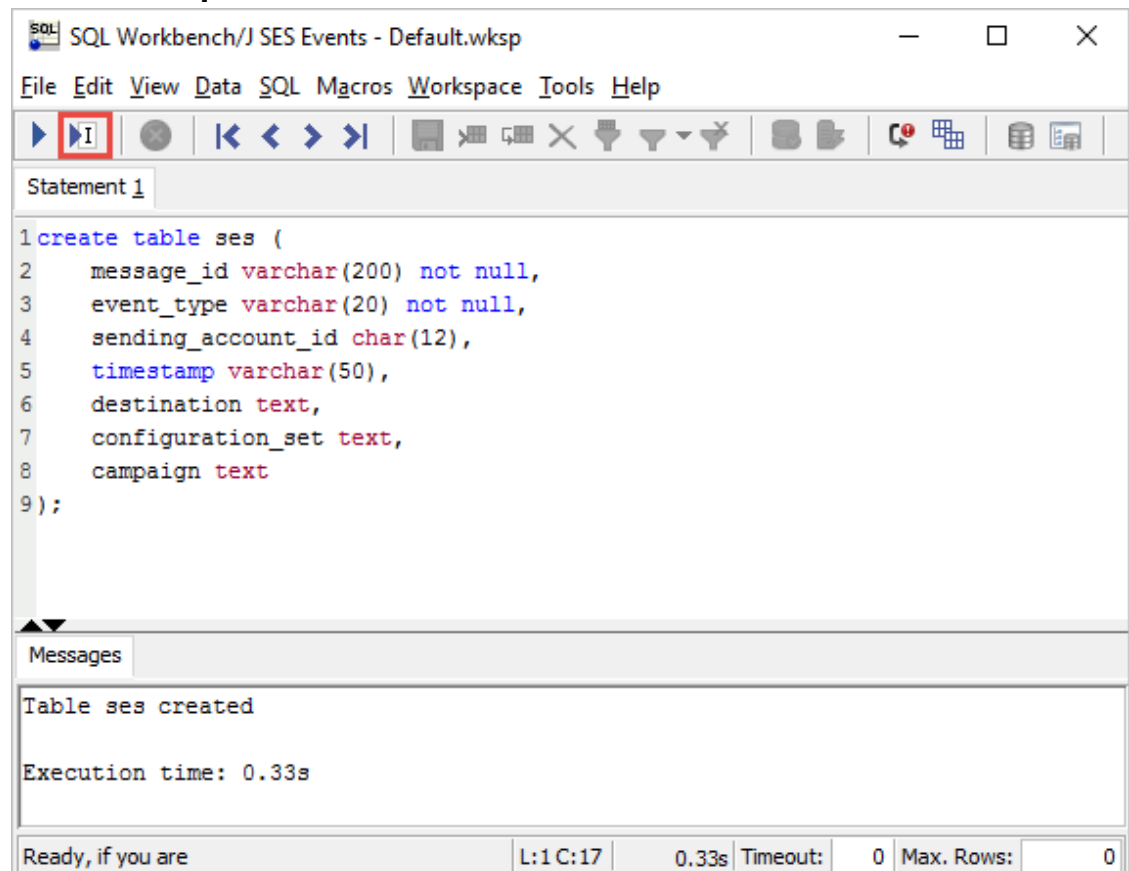
[前のステップ \(p. 349\)](#)で説明したとおり、SQL Workbench/J が現在コンピュータで開いており、Amazon Redshift クラスターに接続されていることを前提としています。

SQL Workbench/J を使用してテーブルを作成するには

1. SQL Workbench/J で、次のコードをコピーし、[Statement 1] ウィンドウに貼り付けます。

```
create table ses (  
  message_id varchar(200) not null,  
  event_type varchar(20) not null,  
  sending_account_id char(12),  
  timestamp varchar(50),  
  destination text,  
  configuration_set text,  
  campaign text  
);
```

2. 次の図に示すように、ステートメント内にカーソルを置き (セミコロンの前の任意の場所)、[Execute current statement] ボタンを選択します。



3. [Messages] ペインで、テーブルが正常に作成されたことを確認します。

#### 次のステップ

[ステップ 4: Kinesis Data Firehose 配信ストリームを作成する \(p. 354\)](#)

## ステップ 4: Kinesis Data Firehose 配信ストリームを作成する

E メール送信イベントを Amazon Kinesis Data Firehose に発行するには、Kinesis Data Firehose 配信ストリームを作成する必要があります。Kinesis Data Firehose 配信ストリームをセットアップしたら、Kinesis Data Firehose がデータを発行する場所を選択します。このチュートリアルでは、データを Amazon Redshift に発行するように Kinesis Data Firehose をセットアップし、Kinesis Data Firehose が中間ステップとしてレコードを Amazon S3 に発行することを選択します。このプロセスでは、Amazon Redshift が Amazon S3 から前のステップ (p. 352) で作成したテーブルにレコードをコピーする方法を指定する必要があります。

このセクションでは、データを Amazon Redshift に送信する Kinesis Data Firehose 配信ストリームを作成する方法と、配信ストリームを編集して Amazon Redshift でどのようにして Amazon SES イベント発行データを Amazon S3 にコピーするかを指定する方法を示します。

### Note

前のステップで説明したとおり、すでに [Amazon Redshift クラスターをセットアップ \(p. 349\)](#) してクラスターに接続 (p. 349) し、[データベーステーブルを作成 \(p. 352\)](#) してある必要があります。

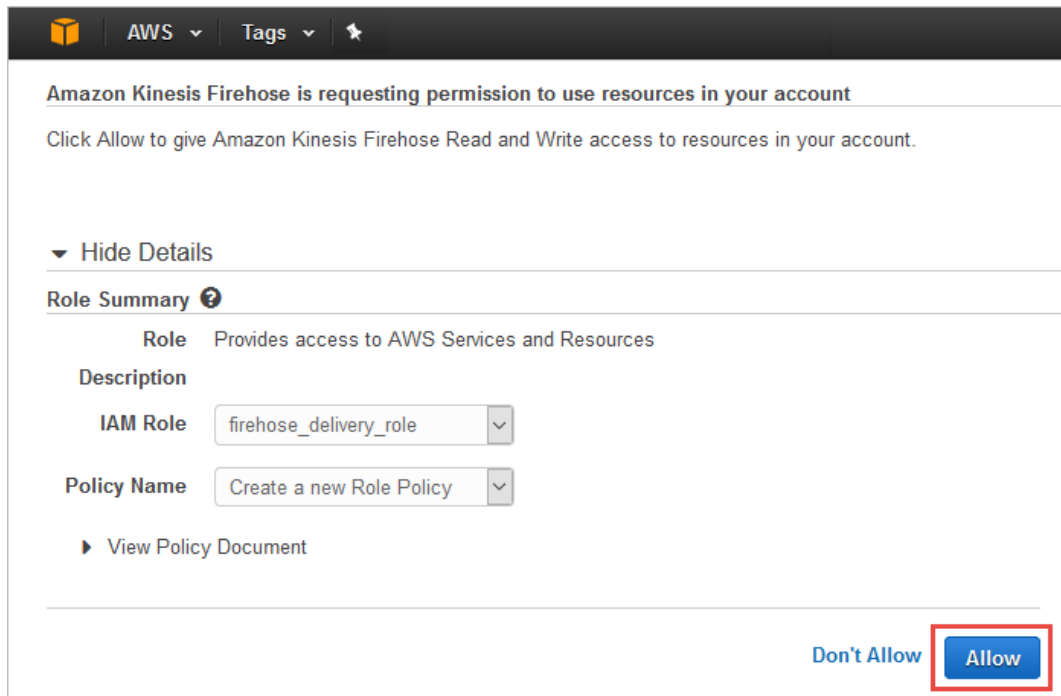
### Kinesis Data Firehose 配信ストリームを作成

次の手順では、Amazon S3 を中間データの場所として使用し、データを Amazon Redshift に発行する Kinesis Data Firehose 配信ストリームを作成する方法を示しています。

Kinesis Data Firehose から Amazon Redshift への配信ストリームを作成するには

1. AWS Management Console にサインインし、Kinesis Data Firehose コンソール (<https://console.aws.amazon.com/firehose/>) を開きます。
2. [配信ストリーム作成] を選択します。
3. [Destination] ページで、次のオプションを選択します。
  - 送信先 – Amazon Redshift を選択します。
  - 配信ストリーム名 – 配信ストリームの名前を入力します。
  - S3 bucket – 新しい S3 バケットを選択してバケット名を入力し、リージョンを選択してからバケットの作成を選択します。
  - Redshift cluster – 前のステップで作成した Amazon Redshift クラスターを選択します。
  - Redshift データベース – デフォルトのデータベース名である dev を入力します。
  - Redshift テーブル – 「[ステップ 3: データベーステーブルを作成する \(p. 352\)](#)」で作成したテーブルである「ses」と入力します。
  - Redshift テーブルの列 – このフィールドは空のままにします。
  - Redshift ユーザーネーム – [Amazon Redshift クラスターのセットアップ \(p. 349\)](#) 時に選択したユーザーネームを入力します。
  - Redshift パスワード – Amazon Redshift クラスターのセットアップ時に選択したパスワードを入力します。
  - Redshift COPY オプション – このフィールドは空のままにします。
  - 再試行期間 – これはデフォルト値のままにします。
  - COPY コマンド – これはデフォルト値のままにします。次の手順で更新します。
4. [次へ] を選択します。
5. [Configuration] ページで、フィールドをこのサンプルチュートリアルのデフォルト設定のままにします。必要なステップは、次の手順で説明されているように、Kinesis Data Firehose がリソースにアクセスできるようにする IAM ロールを選択することだけです。
  - a. [IAM ロール] で、[IAM ロールの選択] を選択します。

- b. ドロップダウンメニューの [Create/Update existing IAM role] で、[Firehose delivery IAM role] を選択します。  
IAM コンソールに移動します。
- c. IAM コンソールで、フィールドをデフォルト設定のままにして、[許可] を選択します。



Kinesis Data Firehose コンソールで、Kinesis データ Firehose 配信ストリームの設定ステップに戻ります。

6. [次へ] を選択します。
7. [Review] ページで設定を確認した後、[配信ストリームを作成] を選択します。

### Amazon Redshift コピーオプションの設定

次に、Amazon Redshift で、Amazon SES イベント発行 JSON レコードを「[ステップ 3: データベーステーブルを作成する \(p. 352\)](#)」で作成したデータベーステーブルにコピーする方法を指定する必要があります。これは、Kinesis Data Firehose 配信ストリームでコピーオプションを編集することにより行います。

この手順では、JSONPaths ファイルを作成する必要があります。JSONPaths ファイルは、Amazon Redshift COPY コマンドに JSON ソースデータを解析する方法を指定するテキストファイルです。この手順では、JSONPaths ファイルが提供されます。JSONPaths ファイルの詳細については、Amazon Redshift データベースデベロッパーガイドにある[JSON 形式からの COPY](#)を参照してください。

Kinesis Data Firehose 配信ストリームの作成時にセットアップした Amazon S3 バケットに JSONPaths ファイルをアップロードし、アップロードした JSONPaths ファイルが使用されるように Kinesis Data Firehose 配信ストリームの COPY オプションを編集します。これらのステップについて、次の手順で説明します。

Amazon Redshift COPY コマンドオプションを設定するには

1. JSONPaths ファイルを作成 - コンピュータで、jsonpaths.json というファイルを作成します。次のテキストをファイルにコピーし、ファイルを保存します。

```
{
  "jsonpaths": [
    "$.mail.messageId",
    "$.eventType",
    "$.mail.sendingAccountId",
    "$.mail.timestamp",
    "$.mail.destination",
    "$.mail.tags.ses:configuration-set",
    "$.mail.tags.campaign"
  ]
}
```

2. JSONPaths ファイルを Amazon S3 バケットにアップロードする - [Amazon S3 コンソール](#)に移動し、「[Kinesis Data Firehose 配信ストリームを作成 \(p. 354\)](#)」 Kinesis Data Firehose で配信ストリームのセットアップ時に作成したバケットにファイルをアップロードします。
3. Kinesis Data Firehose 配信ストリーム設定で COPY コマンドを設定する - ここまでで、作成したテーブルにデータを置くときに Amazon Redshift が使用する COPY コマンドの構文の設定に必要な情報を入手しました。次の手順では、Kinesis Data Firehose 配信ストリーム設定で COPY コマンド情報を更新する方法を示します。

1. [Kinesis Data Firehose コンソール](#)に移動します。

2. [Redshift 配信ストリーム] で、Amazon SES イベント発行用に作成した Kinesis Data Firehose 配信ストリームを選択します。

3. [Details (詳細)] ページで、[Edit (編集)] を選択します。

4. [Redshift COPY options] ボックスに、次のテキストを入力します。以下の値は独自の値に置き換えてください。

- S3-BUCKET-NAME— Amazon Redshift がアクセスできるように Kinesis Data Firehose がデータを置く Amazon S3 バケットの名前。このバケットは、「[ステップ 4: Kinesis Data Firehose 配信ストリームを作成する \(p. 354\)](#)」で Kinesis Data Firehose 配信ストリームをセットアップするときに作成しました。例: 「my-bucket」。
- リージョン— お客様の Amazon SES、Kinesis Data Firehose、Amazon S3、および Amazon Redshift リソースがあるリージョン。例: 「us-west-2」。

```
json 's3://S3-BUCKET-NAME/jsonpaths.json' region 'REGION';
```

5. [Save] を選択します。

Delivery Streams > ses-stream

Use the tabs below to view, edit and monitor your delivery stream.

Details Monitoring S3 Logs Redshift Logs Delete Delivery Stream

Cancel Save

Delivery stream name\* ses-stream

S3 bucket\* example-bucket

S3 prefix S3 Prefix

IAM role\* firehose\_delivery\_role

S3 buffer size (MB)\* 5

S3 buffer interval (sec)\* 300

S3 Compression UNCOMPRESSED

S3 Encryption No Encryption

Status ACTIVE

Error logging  Enable  Disable

Redshift cluster\* ses-events

Redshift database\* dev

Redshift table\* ses

Redshift table columns Redshift table columns

Redshift username\* user

Redshift password\* \*\*\*\*\*

Redshift COPY options json 's3://S3-BUCKET-NAME/jsonpaths.json' region 'REGION';

Retry duration (sec)\* 3600

COPY command COPY ses FROM 's3://example-bucket/<manifest>' CREDENTIALS 'aws\_access\_key\_id=<aws-access-key-id>; aws\_secret\_access\_key=<aws-secret-access-key>' MANIFEST json 's3://S3-BUCKET-NAME/jsonpaths.json' region 'REGION';

## 次のステップ

ステップ 5: 設定セットをセットアップする (p. 357)

## ステップ 5: 設定セットをセットアップする

E メール送信イベントが Amazon Kinesis Data Firehose に発行されるように Amazon SES をセットアップするには、まず設定セットを作成し、次に設定セットに Kinesis Data Firehose イベント送信先を追加します。このセクションでは、これらのタスクを実行する方法を示します。

すでに設定セットがある場合、Kinesis Data Firehose の配信先を既存の設定セットに追加できます。この場合、「[Kinesis Data Firehose イベント送信先の追加 \(p. 357\)](#)」に移動します。

### 設定セットの作成

次の手順では、設定セットを作成する方法を示します。

設定セットを作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. コンテンツペインで、[設定セットを作成] を選択します。
4. 設定セットの名前を入力し、[設定セットを作成] を選択します。
5. [終了] を選択します。

### Kinesis Data Firehose イベント送信先の追加

次の手順では、作成した設定セットに Kinesis Data Firehose イベント送信先を追加する方法を示します。

Kinesis Data Firehose イベント送信先を設定セットに追加するには

1. 設定セットリストから設定セットを選択します。

2. [送信先の追加] で、[送信先タイプの選択] を選択して [Kinesis Data Firehose] を選択します。
3. [名前] に、イベント発行先の名前を入力します。
4. すべての [Event types] を選択します。
5. [Enabled] を選択します。
6. [ストリーミング] で、「[ステップ 4: Kinesis Data Firehose 配信ストリームを作成する \(p. 354\)](#)」で作成した配信ストリームを選択します。
7. [IAM ロール] で、[SES で新規ロールを作成] を選択してロール名を入力します。
8. [Save] を選択します。
9. [Edit Configuration Set] ページを終了するには、ブラウザの戻るボタンを使用します。

## 次のステップ

[ステップ 6: メールを送信する \(p. 358\)](#)

## ステップ 6: メールを送信する

Amazon SES でメールに関連付けられたイベントを発行するには、メールの送信時に設定セットを指定する必要があります。メールを分類するためのメッセージタグも含める必要があります。このセクションでは、Amazon SES コンソールを使用して設定セットとメッセージタグを指定する、シンプルなメールを送信する方法を示します。バウンスや苦情などのメール送信結果をテストできるように、メールを Amazon SES メールボックスシミュレーターに送信します。

Amazon SES コンソールを使用して E メールを送信するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. Amazon SES コンソールのナビゲーションペインで、[ID 管理] の [E メールアドレス] を選択します。
3. ID リストで、[Amazon SES で検証した \(p. 49\)](#) E メールアドレスのチェックボックスを選択します。
4. [テストメール送信] を選択します。
5. [テストメール送信] ダイアログボックスの [メールフォーマット] で、[未フォーマット] を選択します。
6. 「To」アドレスに、[Amazon SES メールボックスシミュレーター \(p. 187\)](#)からのアドレス (complaint@simulator.amazonses.com、bounce@simulator.amazonses.com など) を入力します。
7. 次のメッセージ全体をコピーし、[メッセージ] テキストボックスに貼り付けます。CONFIGURATION-SET-NAME は、「[ステップ 5: 設定セットをセットアップする \(p. 357\)](#)」で作成した設定セットの名前で置き換え、FROM-ADDRESS はこのメールの送信元の確認済みアドレスで置き換えます。

```
X-SES-MESSAGE-TAGS: campaign=book
X-SES-CONFIGURATION-SET: CONFIGURATION-SET-NAME
Subject: Amazon SES Event Publishing Test
From: Amazon SES User <FROM-ADDRESS>
MIME-Version: 1.0
Content-Type: text/plain
```

```
This is a test message.
```

8. [テストメール送信] を選択します。
9. この手順を数回繰り返し、複数のメール送信イベントを生成します。いくつかのメールについては、campaign メッセージタグの値を clothing に変更し、異なるメールキャンペーンの送信をシミュレートします。このようにして、このチュートリアルの最後のステップで Amazon Redshift データベースに対してメール送信イベントレコードのクエリを実行すると、メールキャンペーンに基づくクエリをテストできます。

## 次のステップ

[ステップ 7: E メール送信イベントのクエリを実行する \(p. 359\)](#)

## ステップ 7: E メール送信イベントのクエリを実行する

ここまでで、設定セットとメッセージタグを使用してメールを送信することにより、いくつかのメール送信イベントを生成したので、それらのレコードのクエリを Amazon Redshift で実行できます。

### Note

「[ステップ 2: Amazon Redshift クラスターに接続する \(p. 349\)](#)」で説明したとおり、SQL Workbench/J が現在コンピュータで開かれており、Amazon Redshift クラスターに接続されていることを前提としています。

SQL Workbench/J から Amazon Redshift のメール送信イベントデータをクエリするには

1. すべてのメール送信レコードを表示するには、次のクエリをコピーして [Statement 1] ウィンドウに貼り付けます。

```
select * from ses;
```

2. ステートメント内にカーソルを置き (セミコロンの中の任意の場所)、[Execute current statement] ボタンを選択します。

「[ステップ 6: メールを送信する \(p. 358\)](#)」で送信したすべてのメールに関して、メール送信レコードが表示されます。次の図のレコードは、book キャンペーンに 2 つの苦情があり、clothing キャンペーンに 1 つのバウンスがあったことを示しています。

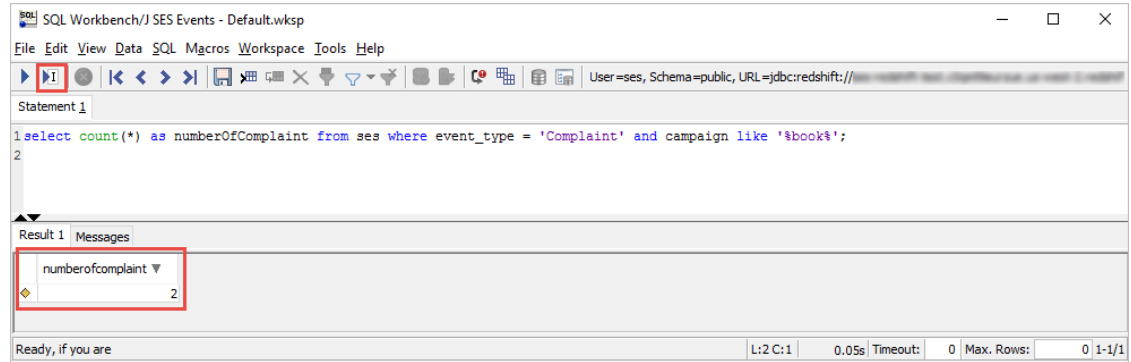
message_id	event_type	sending_account_id	timestamp	destination	configuration_set	campaign
01bf5b3c-c3ae-4015-84c2-18b3dEXAMPLE	Complaint	123456789012	2016-10-17T20:19:26.131Z	[recipient@example.com]	[my-configuration-set]	[book]
05bf5f3c-d3ae-6015-94c2-48b32EXAMPLE	Bounce	123456789012	2016-10-17T20:19:26.131Z	[recipient@example.com]	[my-configuration-set]	[clothing]
06bf5b16-c3ae-7045-84c2-h8b4dEXAMPLE	Complaint	123456789012	2016-10-17T20:19:26.131Z	[recipient@example.com]	[my-configuration-set]	[book]

3. タイプが complaint のキャンペーンの book レコードをカウントするには、次のクエリをコピーして [Statement 1] ウィンドウに貼り付けます。

```
select count(*) as numberOfComplaint from ses where event_type = 'Complaint' and campaign like '%book%';
```

4. ステートメント内にカーソルを置き (セミコロンの中の任意の場所)、[Execute current statement] ボタンを選択します。

結果は次のとおりです。書籍キャンペーンに 2 つの苦情があったことを示しています。



## Amazon CloudWatch での E メール送信イベントのグラフ化

このチュートリアルでは、Amazon SES E メール送信イベントを Amazon CloudWatch に発行し、CloudWatch コンソールを使用してイベントをグラフ化します。

以下のセクションでは、次のプロセスを説明します。

- [前提条件 \(p. 360\)](#)
- [ステップ 1: 設定セットをセットアップする \(p. 360\)](#)
- [ステップ 2: メールを送信する \(p. 361\)](#)
- [ステップ 3: イベントをグラフ化する \(p. 362\)](#)

### Prerequisites

このチュートリアルでは、以下が必要になります。

- **AWS アカウント** - AWS が提供するウェブサービスにアクセスするには、まず <https://aws.amazon.com/> で AWS アカウントを作成する必要があります。
- **検証済み E メールアドレス** - Amazon SES を使用して E メールを送信するには、「From」アドレスまたはドメインを検証して、アドレスの所有者本人であることを示す必要があります。サンドボックスの場合、「To」アドレスも確認する必要があります。E メールアドレスまたはドメイン全体を検証できますが、このチュートリアルでは、Amazon SES コンソール (最も簡単に E メールを送信する方法) から E メールを送信できるように検証済み E メールアドレスが必要です。メールアドレスの確認については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。

### 次のステップ

[ステップ 1: 設定セットをセットアップする \(p. 360\)](#)

### ステップ 1: 設定セットをセットアップする

E メール送信イベントを Amazon CloudWatch に発行するように Amazon SES をセットアップするには、最初に設定セットを作成し、次にこの設定セットに CloudWatch Event 送信先を追加します。このセクションでは、これらのタスクを実行する方法を示します。

すでに設定セットがある場合、CloudWatch 発行先を既存の設定セットに追加できます。この場合、「[CloudWatch Event 送信先の追加 \(p. 361\)](#)」に移動します。

### 設定セットの作成

次の手順では、設定セットを作成する方法を示します。



設定セットを作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで [設定セット] を選択します。
3. [Create Configuration Set] を選択します。
4. 設定セットの名前を入力し、[設定セットを作成] を選択します。

## CloudWatch Event 送信先の追加

次の手順では、作成した設定セットに CloudWatch Event 送信先を追加する方法を示します。

CloudWatch Event 送信先を設定セットに追加するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. ナビゲーションペインで [設定セット] を選択します。
3. 前のセクションで作成した設定セットを選択します。
4. [送信先の追加] で、[送信先タイプの選択] を選択して [CloudWatch] を選択します。
5. [Name (名前)] に、イベント送信先の名前を入力します。
6. [イベントタイプ] で、Amazon CloudWatch でレポートしたいメトリクスを選択します。
7. [Enabled (有効)] を選択します。
8. [値ソース] で、CloudWatch でメトリクスを分類するために使用する値を選択します。たとえば、メッセージタグを選択する場合は、キーバリューのペアを指定する必要があります。E メールにこのキーバリューのペアがメッセージタグとして含まれている場合、Amazon SES は選択したメトリクスを CloudWatch に送信します。CloudWatch でメトリクスを表示すると、メトリクスはメッセージタグのキーによって分類されます。

### Note

値のソースとして [リンクタグ] を選択した場合は、クリックイベントのみを CloudWatch に送信できます。[Link Tag (リンクタグ)] 値ソースを使用すると、Eメールのどのリンクが最も頻繁にクリックされるかを判断できます。

9. [Save] を選択します。
10. [Edit Configuration Set] ページを終了するには、ブラウザの戻るボタンを使用します。

## ステップ 2: メールを送信する

Amazon SES でメールに関連付けられたイベントを発行するには、メールの送信時に設定セットを指定する必要があります。メールを分類するためのメッセージタグも含める必要があります。このセクションでは、Amazon SES コンソールを使用して設定セットとメッセージタグを指定する、シンプルなメールを送信する方法を示します。バウンスや苦情などのメール送信結果をテストできるように、メールを Amazon SES メールボックスシミュレーターに送信します。

Amazon SES コンソールを使用して E メールを送信するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. Amazon SES コンソールのナビゲーションペインで、[ID 管理] の [E メールアドレス] を選択します。
3. ID リストで、[Amazon SES で検証した \(p. 49\)](#) E メールアドレスのチェックボックスを選択します。
4. [テストメール送信] を選択します。
5. [テストメール送信] ダイアログボックスの [メールフォーマット] で、[未フォーマット] を選択します。

6. 「To」アドレスに、[Amazon SES メールボックスシミュレーター \(p. 187\)](#)からのアドレス (complaint@simulator.amazonses.com、bounce@simulator.amazonses.com など) を入力します。
7. 次のメッセージ全体をコピーし、[メッセージ] テキストボックスに貼り付けます。CONFIGURATION-SET-NAME は、「[ステップ 1: 設定セットをセットアップする \(p. 360\)](#)」で作成した設定セットの名前で置き換え、FROM-ADDRESS はこのメールの送信元の確認済みアドレスで置き換えます。

```
X-SES-MESSAGE-TAGS: campaign=book
X-SES-CONFIGURATION-SET: CONFIGURATION-SET-NAME
Subject: Amazon SES Event Publishing Test
From: Amazon SES User <FROM-ADDRESS>
MIME-Version: 1.0
Content-Type: text/plain

This is a test message.
```

8. [テストメール送信] を選択します。
9. この手順を数回繰り返し、複数のメール送信イベントを生成します。いくつかのメールについては、campaign メッセージタグの値を clothing に変更し、異なるメールキャンペーンの送信をシミュレートします。

## 次のステップ

[ステップ 3: メール送信イベントをグラフ化する \(p. 362\)](#)

## ステップ 3: メール送信イベントをグラフ化する

ここまでで、設定セットとメッセージタグを使用して E メールを送信することにより、いくつかの Amazon SES E メール送信イベントを CloudWatch に発行したので、CloudWatch コンソールを使用してそれらのイベントのメトリクスをグラフ化できます。

メール送信イベントメトリクスをグラフ化するには

1. AWS Management Console にサインインして、CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. 左のナビゲーションペインで [Metrics] を選択します。
3. [All metrics] タブで、[SES] を選択します。

### Tip

検索フィールドに「SES」と入力することもできます。

4. 「[CloudWatch Event 送信先の追加 \(p. 361\)](#)」で指定した値ソースを選択します。たとえば、値ソースとしてメッセージタグ「category:books」を指定した場合は、[category] を選択します。
5. 表示するメトリクスを選択します。グラフが詳細ペインに表示されます。

## Amazon Kinesis Data Analytics を使用した E メール送信イベントの分析

Amazon Kinesis Data Analytics では、SQL を使用してストリーミングデータを処理および分析できます。Amazon Kinesis Data Analytics を使用して、Amazon SES E メール送信イベントを分析できます。

このチュートリアルでは、まず Amazon SES 設定セットをセットアップして E メール送信イベントを Amazon Kinesis Data Firehose 配信ストリームに発行し、その設定セットを使用して Amazon SES から E メールを送信します。次に、Amazon Kinesis Data Analytics をセットアップして Kinesis Data Firehose ストリーミングから E メール送信イベントを取得し、SQL を使用して、送信した E メールイベントからキーとなる情報を抽出します。

## Note

このチュートリアルでは、Amazon SES から安定した E メールストリーミングを送信できるアプリケーションを使用します。この要件については、「[前提条件 \(p. 363\)](#)」で説明されています。

以下のセクションでは、チュートリアルについて説明します。

- [前提条件 \(p. 363\)](#)
- [ステップ 1: Kinesis Data Firehose 配信ストリームを作成する \(p. 363\)](#)
- [ステップ 2: 設定セットをセットアップする \(p. 365\)](#)
- [ステップ 3: メールを送信する \(p. 366\)](#)
- [ステップ 4: Amazon Kinesis Data Analytics アプリケーションを作成する。 \(p. 366\)](#)
- [ステップ 5: SQL クエリを実行する \(p. 371\)](#)
- [\(オプション\) ステップ 6: SQL クエリの結果を保存する \(p. 372\)](#)

## Prerequisites

このチュートリアルでは、以下が必要になります。

- AWS アカウント - AWS が提供するウェブサービスにアクセスするには、まず <https://aws.amazon.com/> で AWS アカウントを作成する必要があります。
- 検証済み E メールアドレス - Amazon SES 使用して E メールを送信するには、「From」アドレスまたはドメインを検証して、アドレスの所有者本人であることを示す必要があります。サンドボックスの場合、「To」アドレスも確認する必要があります。E メールアドレスまたはドメイン全体を検証できますが、このチュートリアルでは、Amazon SES コンソール (最も簡単に E メールを送信する方法) から E メールを送信できるように検証済み E メールアドレスが必要です。メールアドレスの確認については、「[Amazon SES E メールアドレスの確認 \(p. 49\)](#)」を参照してください。
- E メールアプリケーション - このチュートリアルの説明に従って Amazon Kinesis Data Analytics を使用するには、安定した E メール送信イベントストリーミングが生成されるように、Amazon SES を通じて安定した E メール ストリーミングを送信する必要があります。これにより、Amazon Kinesis Data Analytics がスキーマを自動的に検出した後、SQL を使用してイベントレコードを処理できるようになります。このチュートリアルでは、5 分間 10 秒おきに 1 通の E メールを送信すれば十分です。

### Important

実際の受取人に送信する既存の E メールキャンペーンがない場合、[Amazon SES メールボックスシミュレーター \(p. 187\)](#)のアドレスに E メールを送信することを強くお勧めします。メールボックスシミュレーターに送信した E メールは、Amazon SES のバウンス率や苦情率、日次送信クォータには加算されません。

## 次のステップ

[ステップ 1: Kinesis Data Firehose 配信ストリームを作成する \(p. 363\)](#)

### ステップ 1: Kinesis Data Firehose 配信ストリームを作成する

Amazon Kinesis Data Analytics を使用して Amazon SES E メール送信イベントを分析するには、Amazon Kinesis Data Firehose 配信ストリームにイベントを発行するように Amazon SES を設定してから、Kinesis Data Firehose からイベントデータを取得するように Amazon Kinesis Data Analytics を設定する必要があります。

Kinesis Data Firehose 配信ストリームをセットアップしたら、データの最終的な送信先を選択します。送信先のオプションは、Amazon Simple Storage Service (Amazon S3)、Amazon Elasticsearch Service、および Amazon Redshift です。Amazon Kinesis Data Analytics を使用して E メール送信イベントを分析するだけの場合、どの送信先を選択しても問題ありません。このチュートリアルでは、データを Amazon S3

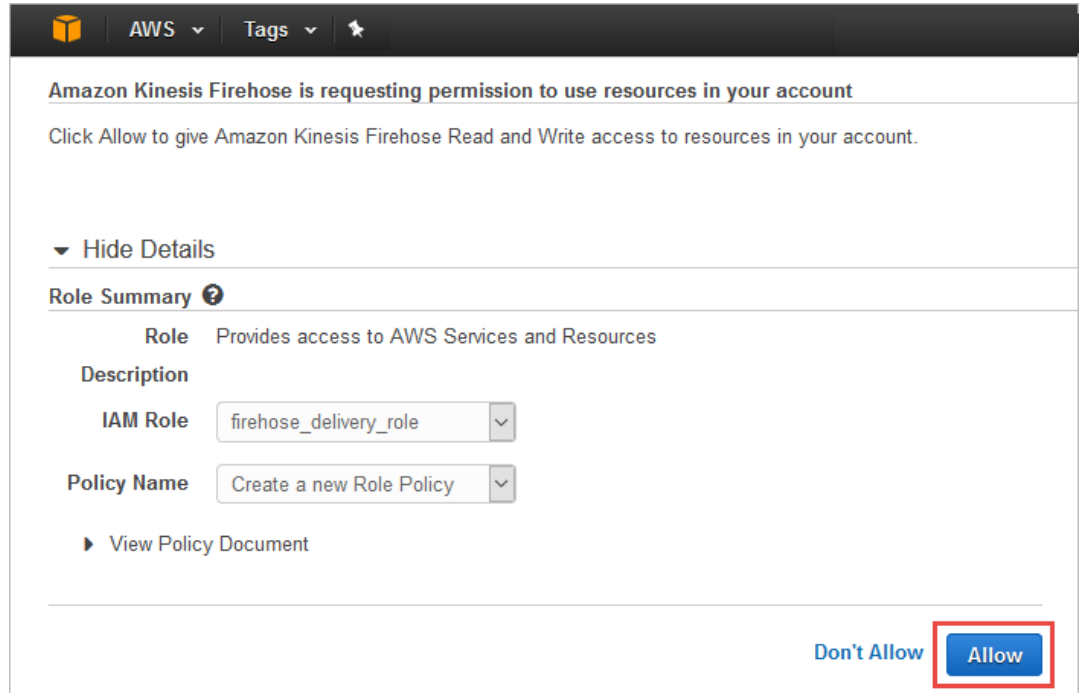
に発行するように Kinesis Data Firehose を設定しますが、Amazon SES 送信および Kinesis Data Firehose 配信ストリームと同じリージョン内であれば、他の送信先オプションを使用できます。

このセクションでは、Kinesis Data Firehose コンソールを使用して Kinesis Data Firehose 配信ストリームを作成する方法について説明します。このチュートリアルではベーシックオプションを選択します。詳細については、Amazon Kinesis Data Firehose デベロッパーガイドの「[Amazon Kinesis Firehose 配信ストリームの作成](#)」を参照してください。

Kinesis Data Firehose から Amazon S3 への配信ストリームを作成するには

1. AWS Management Console にサインインし、Kinesis Data Firehose コンソール (<https://console.aws.amazon.com/firehose/>) を開きます。
2. [配信ストリーム作成] を選択します。
3. [Destination] ページで、次のオプションを選択します。
  - 送信先 – Amazon S3 を選択します。
  - 配信ストリーム名 – 配信ストリームの名前を入力します。
  - S3 バケット - 既存のバケットを選択するか、新しい S3 バケットを選択します。新しいバケットを作成する場合、バケットの名前を入力し、コンソールが現在使用しているリージョンを選択します。
  - S3 プレフィックス – このフィールドは空のままにします。
4. [次へ] を選択します。
5. [Configuration] ページでは、フィールドをデフォルト設定のままにします。必要なステップは、以下にあるように、Kinesis Data Firehose がリソースにアクセスできるようにする IAM ロールを選択することだけです。
  - a. [IAM ロール] で、[IAM ロールの選択] を選択します。
  - b. ドロップダウンメニューの [Create/Update existing IAM role] で、[Firehose delivery IAM role] を選択します。

IAM コンソールに移動します。
  - c. IAM コンソールで、フィールドをデフォルト設定のままにして、[許可] を選択します。



Kinesis Data Firehose コンソールにある Kinesis Data Firehose 配信ストリームのセットアップ手順に戻ります。

6. [次へ] を選択します。
7. [Review] ページで設定を確認した後、[配信ストリームを作成] を選択します。

## 次のステップ

[ステップ 2: 設定セットをセットアップする \(p. 365\)](#)

## ステップ 2: 設定セットをセットアップする

E メール送信イベントを Amazon Kinesis Data Firehose に発行するように Amazon SES をセットアップするには、設定セットを作成し、Kinesis Data Firehose イベント送信先を設定セットに追加します。このセクションでは、これらのタスクを実行する方法について説明します。

すでに設定セットがある場合、Kinesis Data Firehose の配信先を既存の設定セットに追加できます。この場合、「[Kinesis Data Firehose イベント送信先の追加 \(p. 366\)](#)」に移動します。

### 設定セットの作成

次の手順は、設定セットを作成する方法を説明します。

設定セットを作成するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 左のナビゲーションペインで [設定セット] を選択します。
3. コンテンツペインで、[設定セットを作成] を選択します。
4. 設定セットの名前を入力し、[設定セットを作成] を選択します。
5. [終了] を選択します。

## Kinesis Data Firehose イベント送信先の追加

次の手順では、作成した設定セットに Kinesis Data Firehose イベント送信先を追加する方法を示します。

Kinesis Data Firehose イベント送信先を設定セットに追加するには

1. 設定セットリストから設定セットを選択します。
2. [送信先の追加] で、[送信先タイプの選択] を選択して [Kinesis Data Firehose] を選択します。
3. [名前] に、イベント発行先の名前を入力します。
4. すべての [Event types] を選択します。
5. [Enabled] を選択します。
6. [ストリーミング] で、「[ステップ 1: Kinesis Data Firehose 配信ストリームを作成する \(p. 363\)](#)」で作成した配信ストリームを選択します。
7. [IAM ロール] で、[SES で新規ロールを作成] を選択してロール名を入力します。
8. [Save] を選択します。
9. [Edit Configuration Set] ページを終了するには、ブラウザの戻るボタンを使用します。

## 次のステップ

[ステップ 3: メールを送信する \(p. 366\)](#)

## ステップ 3: メールを送信する

このチュートリアルでは Amazon Kinesis Data Analytics コンソールを使用してストリーミングデータを処理および分析するため、Amazon SES を通じて安定した E メールストリーミングをセットアップする必要があります。このチュートリアルは、それらの E メールを送信できるアプリケーションがあることを前提としています。このチュートリアルでは、5 分間 10 秒おきに 1 通の E メールを送信すれば十分です。[Amazon SES メールボックスシミュレーター \(p. 187\)](#)の「To」アドレス (success@simulator.amazonses.com など) を使用することを強くお勧めします。

Eメールのイベント発行を有効にするには、Eメールの送信時に設定セットの名前を Amazon SES に渡します。必要に応じて、メールを分類するためのメッセージタグを含めることができます。この情報を Amazon SES に提供する方法として、Eメール送信 API へのパラメータ、Amazon SES 固有の Eメールヘッダー、または MIME メッセージのカスタムヘッダーがあります。詳細については、「[Amazon SES イベント発行を使用してメールを送信する \(p. 305\)](#)」を参照してください。

たとえば、次の Amazon SES 固有の Eメールヘッダーを Eメールに追加してブックキャンペーンをシミュレーションできます。CONFIGURATION-SET-NAME は、「[ステップ 2: 設定セットをセットアップする \(p. 365\)](#)」で作成した設定セットの名前に置き換えます。

```
X-SES-CONFIGURATION-SET: CONFIGURATION-SET-NAME  
X-SES-MESSAGE-TAGS: campaign=book
```

## 次のステップ

[ステップ 4: Amazon Kinesis Data Analytics アプリケーションを作成する。 \(p. 366\)](#)

## ステップ 4: Amazon Kinesis Data Analytics アプリケーションを作成する。

ここまでで、Amazon SES を使用してイベント発行をセットアップしたので、Amazon Kinesis Data Firehose 配信ストリームから Eメール送信イベントデータをキャプチャするように Amazon Kinesis Data Analytics を設定できます。このために、Amazon Kinesis Data Analytics アプリケーションを作成します。

次の手順では、Amazon Kinesis Data Analytics コンソールを使用して Kinesis Data Firehose 配信ストリームから Amazon SES E メール送信イベントデータをキャプチャするアプリケーションを作成する方法と、データへの SQL クエリだけを実行して「送信」タイプのイベントを返す方法について説明します。

#### Note

E メール送信イベントのイベントタイプが異なると (送信、バウンス、苦情、配信)、[JSON スキーマが異なります \(p. 309\)](#)。運用環境では、このスキーマの複数のフィールドを調べることがありますが、このチュートリアルでは、すべてのイベントタイプに存在する小さいフィールドセットだけを調べます。

Amazon Kinesis Data Analytics アプリケーションを作成するには

1. Amazon SES でイベント発行用に設定した、安定した E メールストリーミングの送信をスタートし、この手順全体で Eメールの送信を続けます。これは、Amazon Kinesis Data Analytics がイベントレコードのスキーマを自動的に検出できるようにするために必要です。このチュートリアルでは、5 分間 10 秒おきに 1 通の E メールを送信すれば十分です。詳細については、「[ステップ 3: メールを送信する \(p. 366\)](#)」を参照してください。

E メールプログラムがいくつかの E メールを送信したら、次のステップに移動します。

2. AWS Management Console にサインインし、Kinesis Data Analytics コンソール (<https://console.aws.amazon.com/kinesisanalytics>) を開きます。
3. [Create new application] を選択します。
4. アプリケーションの名前と説明を入力し、[Save and continue] を選択します。
5. [Connect to a source] を選択します。
6. 「[ステップ 2: 設定セットをセットアップする \(p. 365\)](#)」で作成した Kinesis Data Firehose ストリーミングを選択します。

Amazon Kinesis Data Analytics が、入力レコードに基づいて E メール送信イベントレコードのスキーマの検出を試みます。Amazon Kinesis Data Analytics に「入力スキーマ検出エラー」と表示された場合、Amazon Kinesis Data Analytics がまだ E メール送信レコードを受け取っていないことを意味します。[Rediscover schema] を選択します。場合によっては、このボタンを複数回選択する必要があります。数回試行してもスキーマの検出に成功しなかった場合、E メール送信アプリケーションが E メールを安定して送信していることと、E メールが設定セットを指定していることを確認してください。

Amazon Kinesis Data Analytics がスキーマを検出すると、成功メッセージが表示され、検出されたレコードが一覧表示されます。

#### Important

[Save and continue] を選択しないでください。検出されたスキーマは SQL 命名の制約に従っていないため、選択するとエラーが発生します。次のステップで説明するようにスキーマを編集する必要があります。

7. [Edit schema] を選択します。

**Schema discovery successful**  
Detected JSON format and applied schema

- To define a custom schema, choose "Edit schema" in the stream sample below.
- To capture a new stream sample from the selected source for discovery, [rediscover the schema](#)

(Optional) Send AWS a sample of your data to help improve schema discovery in Amazon Kinesis Analytics

[Help improve schema discovery](#)

Formatted stream sample | Raw stream sample

Filter by column name or column type [Edit schema](#)

messageId	eventType	source	sourceArn
EXAMPLE8d633ffe4-9d79e202-8e68-4d84-8c12-bd80644b270e-000000	Send	sender@example.com	arn:aws:ses:us-east-
EXAMPLE8d633ffe4-9d79e202-8e68-4d84-8c12-bd80644b270e-000000	Send	sender@example.com	arn:aws:ses:us-east-
EXAMPLE8d633ffe4-9d79e202-8e68-4d84-8c12-bd80644b270e-000000	Send	sender@example.com	arn:aws:ses:us-east-
EXAMPLE8d633ffe4-9d79e202-8e68-4d84-8c12-bd80644b270e-000000	Send	sender@example.com	arn:aws:ses:us-east-
EXAMPLE8d633ffe4-9d79e202-8e68-4d84-8c12-bd80644b270e-000000	Send	sender@example.com	arn:aws:ses:us-east-
EXAMPLE8d633ffe4-9d79e202-8e68-4d84-8c12-bd80644b270e-000000	Send	sender@example.com	arn:aws:ses:us-east-

8. このチュートリアルでは、ほとんどの列を削除します。すべての行の横にあるXを選択します。例外は次の列名がある行です。

- eventType
- timestamp
- messageId
- to
- ses:configuration-set

#### Important

[Save schema and update stream samples] は選択しないでください。検出されたスキーマは SQL 命名の制約に従っていないため、選択するとエラーが発生します。次のステップで説明するようにスキーマを編集する必要があります。



Kinesis Analytics dashboard > Example > Source > Edit schema

Format: JSON Record encoding: UTF-8 Row path: \$

Filter by column name

Column order	Column name	Column type	Row path
+	Add column		
×	1	eventType	VARCHAR Length: 8 \$.\$eventType
×	2	timestamp	TIMESTAMP \$.\$mail.timestamp
×	3	source	VARCHAR Length: 32 \$.\$mail.source
×	4	sourceArn	VARCHAR Length: 128 \$.\$mail.sourceArn
×	5	sendingAccountId	BIGINT \$.\$mail.sendingAcco
×	6	messageId	VARCHAR Length: 64 \$.\$mail.messageId
×	7	destination	VARCHAR Length: 64 \$.\$mail.destination[

9. [Column name] の下にある残りのエントリを調べ、次のように SQL 命名要件と比較します。

- 形式 - Amazon Kinesis Data Analytics SQL リファレンスの [識別子](#) で説明しているとおり、引用符で囲まれていない識別子は、文字またはアンダースコアで始まり、その後文字、数字、またはアンダースコアが続きます。Amazon SES 自動タグ名には、コロンとダッシュが含まれているため、これらの要件に準拠していません。これらは次のステップで編集します。
- 予約語 - カラム名は、Amazon Kinesis Data Analytics SQL リファレンスにある [予約語とキーワード](#) の SQL 予約語と食い違ってはなりません。Amazon SES イベントレコードと競合する予約キーワードの例は、timestamp、value、date、from、および to です。

10. 次のように SQL 要件に従って、残りの列名を編集します。

- ses:configuration-set を ses\_configuration\_set に名前変更します。
- timestamp を ses\_timestamp に名前変更します。
- to を ses\_to に名前変更します。

Kinesis Analytics dashboard > Example > Source > Edit schema

Format: JSON Record encoding: UTF-8 Row path: \$

Filter by column name

Column order	Column name	Column type	Length	Row path
1	eventType	VARCHAR	8	\$.eventType
2	ses_timestamp	TIMESTAMP		\$.mail.timestamp
3	messageid	VARCHAR	64	\$.mail.messageid
4	ses_to	VARCHAR	64	\$.mail.commonHeader
5	ses_configuration_set	VARCHAR	16	\$.mail.tags.ses.cc

Cancel Save schema and update stream samples

11. [Save schema and update stream samples] を選択します。検証エラーが発生した場合、ステップ 10 を正しく実行したことを確認してください。「No rows in source stream」エラーが発生した場合、この手順の最初に開始した E メールストリームの送信が続いていることを確認し、[Retrieve rows] を選択します。Amazon Kinesis Data Analytics がレコードを取得する前に、行の取得を数回選択する必要があります。

12. 行の取得に成功したら、[Exit (done)] を選択します。

Filter by column name

Column order	Column name	Column type	Length	Row path
1	eventType	VARCHAR	8	\$.eventType
2	ses_timestamp	TIMESTAMP		\$.mail.timestamp
3	messageid	VARCHAR	64	\$.mail.messageid
4	ses_to	VARCHAR	64	\$.mail.commonHeader
5	ses_configuration_set	VARCHAR	16	\$.mail.tags.ses.cc

Exit (done) Save schema and update stream samples

## 次のステップ

ステップ 5: SQL クエリを実行する (p. 371)

## ステップ 5: SQL クエリを実行する

ここまでで、Amazon Kinesis Data Analytics アプリケーションを作成し、送信元として Amazon Kinesis Data Firehose 配信ストリームを使用するように設定したので、Kinesis Data Firehose 配信ストリームが受信する E メール送信イベントデータをクエリできます。

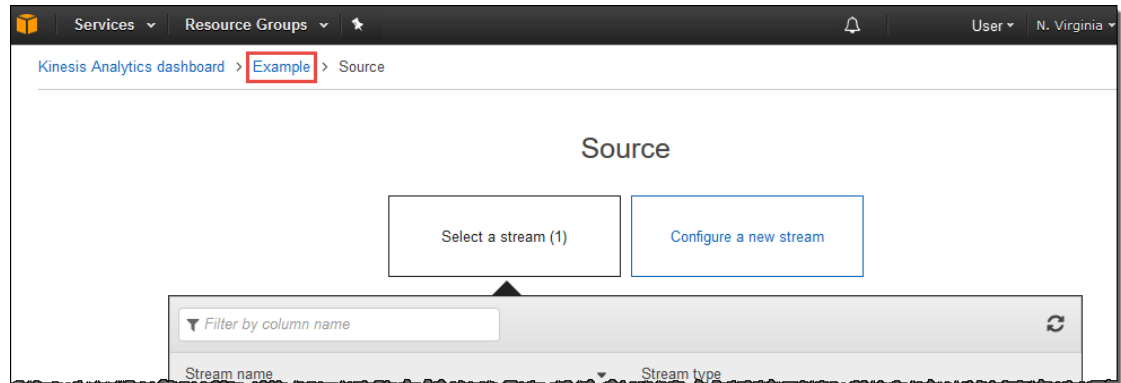
このトピックでは、E メール送信イベントデータで SQL クエリを実行する方法を示します。

### Important

この手順では、「[ステップ 3: メールを送信する \(p. 366\)](#)」で説明しているように、Amazon SES でイベント発行用に設定した、安定した E メールストリーミングの送信を続ける必要があります。

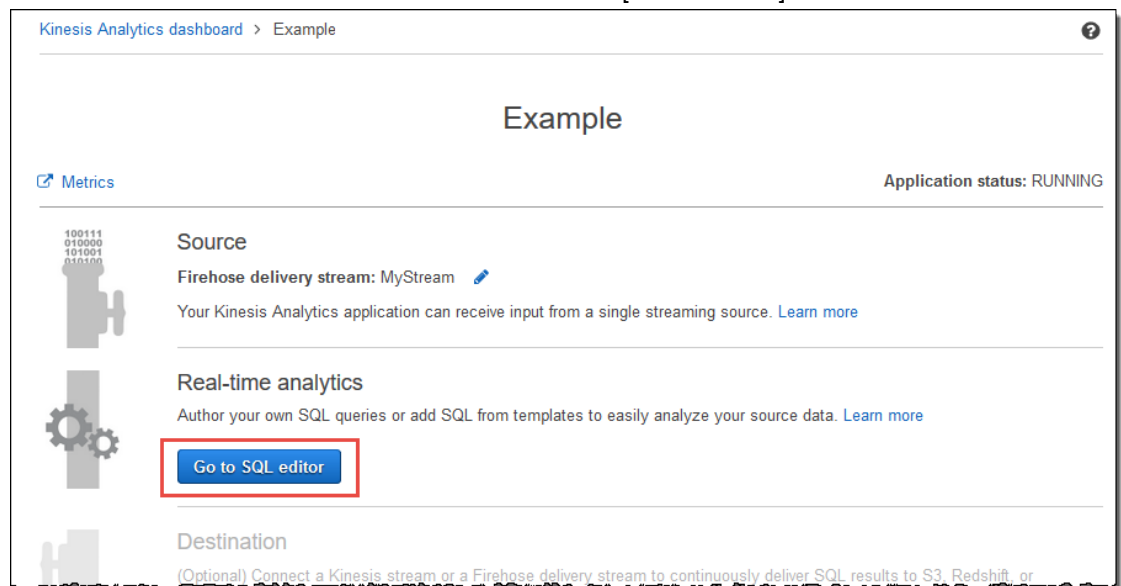
Amazon Kinesis Data Analytics で SQL クエリを実行するには

1. [前のステップ \(p. 366\)](#)を完了した後にこの手順に移動した場合は、Amazon Kinesis Data Analytics コンソールの一番上のメニューに移動してアプリケーションを選択します。



2. [Go to SQL editor] を選択します。

Amazon Kinesis Data Analytics は、Kinesis Data Firehose ストリーミングからイベントデータの読み込みを試行します。「No rows in source stream」エラーが発生した場合、この手順の最初に開始した E メールストリーミングの送信が続いていることを確認し、[Retrieve rows] を選択します。



3. コードエディタダイアログボックスで、次の内容を貼り付けます。

```
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" ("eventType" VARCHAR(16),
"ses_timestamp" timestamp, "messageId" VARCHAR(64), "ses_to" VARCHAR(64),
"ses_configuration_set" VARCHAR(32));

CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"

SELECT STREAM "eventType", "ses_timestamp", "messageId", "ses_to",
"ses_configuration_set"
FROM "SOURCE_SQL_STREAM_001"
WHERE "eventType" = 'Send'
```

4. [Save and run SQL] を選択します。

Amazon Kinesis Data Analytics が入力レコードを取得して処理すると、「送信」タイプのイベントレコードリストが表示されます。

The screenshot displays the Amazon Kinesis Data Analytics SQL editor. The SQL code is as follows:

```
1 CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" ("eventType" VARCHAR(16), "ses_timestamp" timestamp, "messageId" VARCHAR(64), "ses_to" VARCHAR(64), "ses_configuration_set" VARCHAR
2
3 CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
4
5 SELECT STREAM "eventType", "ses_timestamp", "messageId", "ses_to", "ses_configuration_set"
6 FROM "SOURCE_SQL_STREAM_001"
7 WHERE "eventType" = "Send"
```

The interface includes a "Save and run SQL" button, which is highlighted with a red box. Below the code editor, the "Real-time analytics" tab is active, showing a table of results. The table has columns for ROWTIME, eventType, ses\_timestamp, and messageId. The results show five rows of "Send" events with timestamps and message IDs.

ROWTIME	eventType	ses_timestamp	messageId
2016-12-07 00:53:17.389	Send	2016-12-07 00:51:42.039	EXAMPLEd6c3bd57-a4b91dbd-ecb9-4e05-aec6-b75cfee5b86b-00c
2016-12-07 00:53:17.389	Send	2016-12-07 00:52:32.12	EXAMPLEd6c480f8-c0eedbe9-2fd4-4ef0-b8ba-061461f5a169-00000
2016-12-07 00:53:17.389	Send	2016-12-07 00:52:22.032	EXAMPLEd6c45990-e7eab936-3d3c-4fb7-9879-1c55c17b24b3-00c
2016-12-07 00:53:28.303	Send	2016-12-07 00:52:42.036	EXAMPLEd6c4a7b4-2c94da25-2592-4bcf-a784-fc41b255a417-0000i
2016-12-07 00:53:31.307	Send	2016-12-07 00:52:11.983	EXAMPLEd6c4324f-6cde57d2-2f94-4f20-a366-c63b9e5f4cdf-0000c

## 次のステップ

(オプション) ステップ 6: SQL クエリの結果を保存する (p. 372)

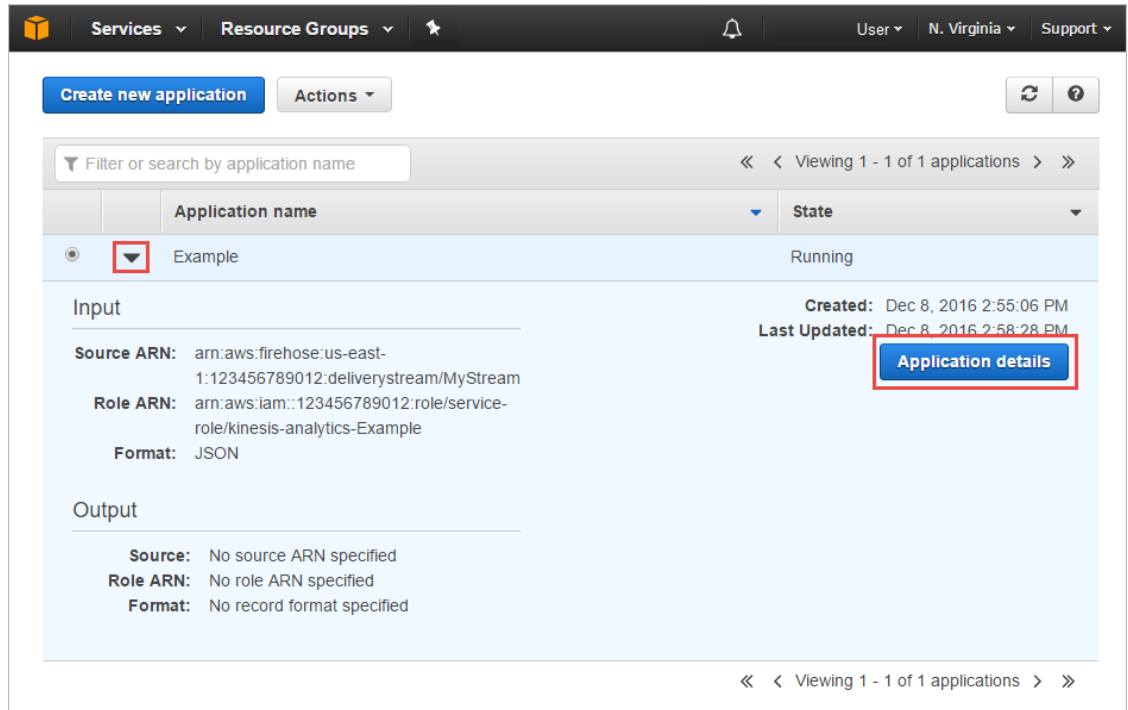
## (オプション) ステップ 6: SQL クエリの結果を保存する

Amazon Kinesis Data Firehose 配信ストリームに SQL クエリの出力を書き込むように、Amazon Kinesis Data Analytics アプリケーションをセットアップすることができます。同じ配信ストリームを Amazon Kinesis Data Analytics アプリケーションの送信元と送信先の両方として使用することはできないため、これを行うには、別の Kinesis Data Firehose 配信ストリームを作成する必要があります。他の Kinesis Data Firehose 配信ストリームと同様に、送信先は、Amazon Simple Storage Service (Amazon S3)、Amazon Elasticsearch Service、または Amazon Redshift のいずれかを選択できます。

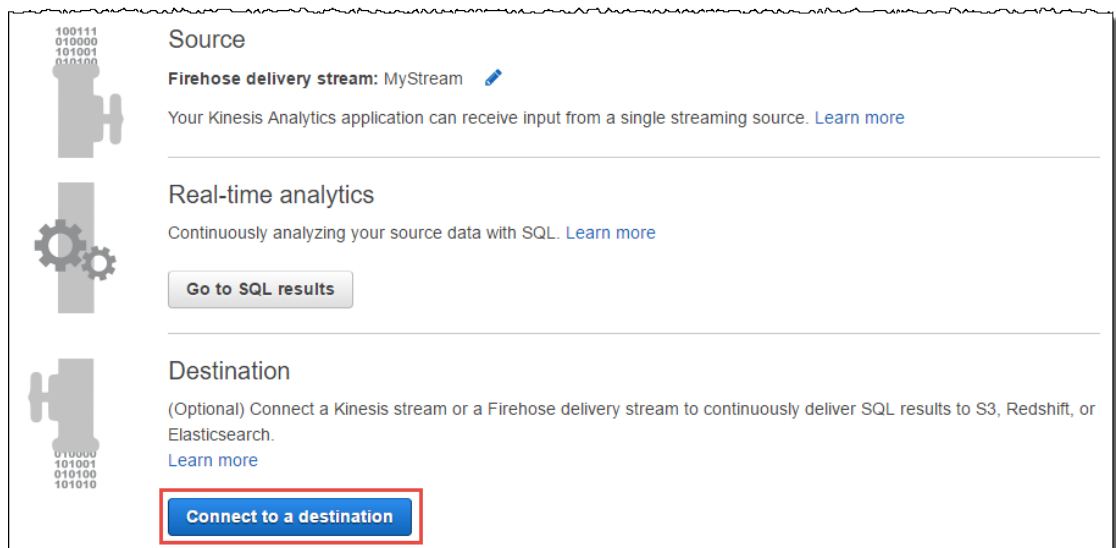
次の手順は、SQL クエリの結果が Amazon S3 にデータを書き込む Kinesis Data Firehose 配信ストリームに JSON 形式で保存されるように Amazon Kinesis Data Analytics を構成する方法を示しています。次に、SQL クエリを実行して保存されたデータにアクセスします。

## SQL クエリの結果を Amazon S3 に保存するには

1. 送信先として Amazon S3 を使用する新しい Kinesis Data Firehose ストリーミングをセットアップします。これは、「[ステップ 1: Kinesis Data Firehose 配信ストリームを作成する \(p. 363\)](#)」と同じ手順です。
2. [Amazon Kinesis Data Analytics コンソール](#)に移動し、アプリケーションの横の矢印を選択して [アプリケーションの詳細] を選択します。



3. [Connect to a destination] を選択します。



4. ステップ 1 で作成した Kinesis Data Firehose ストリーミングを選択し、残りのオプションをデフォルト設定のままにして [保存して続行] を選択します。

数秒後、アプリケーションのメインページに戻ります。

Destination

Select a stream (1)    Configure a new stream

Filter by column name

Stream name	Stream type
StreamForSQL	Firehose delivery stream

In your SQL, refer to this stream as DESTINATION\_SQL\_STREAM

Output format: JSON

Permission to access the stream:  Create/update Example IAM role  Choose an IAM role

Cancel    Save and continue

5. [Go to SQL results] を選択します。

Source

Firehose delivery stream: MyStream

Your Kinesis Analytics application can receive input from a single streaming source. [Learn more](#)

Real-time analytics

Continuously analyzing your source data with SQL. [Learn more](#)

Go to SQL results

Destination

Firehose delivery stream: StreamForSQL

Connect a Kinesis Stream, or a Firehose delivery stream to continuously deliver SQL results to S3, Redshift or Elasticsearch. [Learn more](#)

6. [Save and run SQL] を選択し、「ステップ 5: SQL クエリを実行する (p. 371)」で実行したクエリを再実行します。

Amazon Kinesis Data Analytics は、Kinesis Data Firehose 配信ストリームから受信したイベントデータの処理を試みます。「No rows have arrived yet (行がまだ到着していません)」エラーが発生した場合、Amazon Kinesis Data Analytics に処理する E メール送信イベントが到達するように Eメールの送信が続いていることを確認してください。

Amazon Kinesis Data Analytics がレコードを処理すると、結果がリアルタイム分析タブに表示されま  
す。Kinesis Data Firehose 配信ストリームをステップ 1 でセットアップしたときに指定した Amazon  
S3 バケットに、Amazon Kinesis Data Analytics の結果が自動的に保存されます。

The screenshot shows the Amazon Kinesis Data Analytics console. At the top, there's a navigation bar with "Kinesis Analytics dashboard", "Example", and "SQL editor". Below that, there's a section for "Add SQL from templates" with an "Export SQL" link. The main area is a SQL editor with the following code:

```
1 CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" ("eventType" VARCHAR(16), "ses_timestamp" timestamp, "messageId" VARCHAR(64), "ses_to" VARCHAR(64), "ses_configuration_set" VARCHAR(64))
2
3 CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
4
5 SELECT STREAM "eventType", "ses_timestamp", "messageId", "ses_to", "ses_configuration_set"
6 FROM "SOURCE_SQL_STREAM_001"
7 WHERE "eventType" = 'Send'
```

At the bottom right of the editor, there are "Cancel" and "Save and run SQL" buttons. Below the editor, there's a section for "Source data" with tabs for "Real-time analytics" and "Destination". The "Real-time analytics" tab is active, showing "In-application streams" with "DESTINATION\_SQL\_STREAM" selected. There's a "Pause results" button and a checkbox for "Scroll to bottom when new results arrive". Below that, there's a table with the following data:

ROWTIME	eventType	ses_timestamp	messageId
2016-12-07 00:53:17.389	Send	2016-12-07 00:51:42.039	EXAMPLEd6c3bd57-a4b91dbd-ecb9-4e05-aec6-b75cee5b86b-00c
2016-12-07 00:53:17.389	Send	2016-12-07 00:52:32.12	EXAMPLEd6c480f8-c0eedbe9-2fdf-4ef0-b8ba-06f461f5a169-00000
2016-12-07 00:53:17.389	Send	2016-12-07 00:52:22.032	EXAMPLEd6c45990-e7eab936-3d3c-4fb7-9879-1c55c17b24b3-00c
2016-12-07 00:53:28.303	Send	2016-12-07 00:52:42.036	EXAMPLEd6c4a7b4-2c94da25-2592-4bcf-a784-fc41b255a417-0000
2016-12-07 00:53:31.307	Send	2016-12-07 00:52:11.983	EXAMPLEd6c4324f-6c0e57d2-2194-4f20-a366-c63b9e5f4cdf-0000c

7. 結果を取得するには、[Amazon S3 コンソール](#)に移動します。
8. Amazon Kinesis Data Analytics アプリケーションが宛先として使用する Kinesis Data Firehose 配信ストリームに関連付けられている Amazon S3 バケットを選択します。
9. データに移動します。データは、デフォルトでは結果がバケットに保存された日付に基づいてフォルダ階層に整理されています。

バケットが空の場合、数分待つてから再度試してください。Amazon Kinesis Data Analytics から Amazon S3 バケットにデータが移動するのに数分かかることがあります。

10. ファイルを選択し、[Actions] メニューで [Download] を選択します。

The screenshot shows the Amazon S3 console. At the top, there's a navigation bar with "Services", "Resource Groups", "User", "Global", and "Support". Below that, there's a section for "All Buckets / example" with a search bar and a "Search by prefix" input. There's a table with the following data:

Name	Storage Class	Size	Last Modified
StreamForSQL-1-2016-12-08-4f22-b361-f21c7694e9bc	Standard	6.2 KB	Thu Dec 08 15:51:50 GMT-800 2016
StreamForSQL-1-2016-12-08-3-47f8-b230-4c60284d403f	Standard	6.9 KB	Thu Dec 08 15:56:55 GMT-800 2016

A context menu is open over the first file, with the "Download" option highlighted. The menu options are: Open, Download, Create Folder..., Upload, Make Public, Rename, Delete, Initiate Restore, Cut, Copy, Paste, and Properties.

11. 画面の指示に従ってファイルをコンピュータにダウンロードします。

12. コンピュータで、テキストエディタを使用してファイルを開きます。レコードは JSON 形式であり、各レコードは中括弧で囲まれています。以下に、2 つのレコードを含むファイルの例を示します。

```
{ "eventType": "Send", "ses_timestamp": "2016-12-08
18:51:12.092", "messageId": "EXAMPLE8dfc6695c-5f048b74-
ca83-4052-8348-4e7da9669fc3-000000", "ses_to": "[\"success@simulator.amazonses.com
\" ]", "ses_configuration_set": "[\"MyConfigSet\" ]"}
{"eventType": "Send", "ses_timestamp": "2016-12-08
18:50:42.181", "messageId": "EXAMPLEdfc5f485-
d40a2543-2cac-4b84-8a8f-30bebdf3820c-000000", "ses_to": "[\"success@simulator.amazonses.com
\" ]", "ses_configuration_set": "[\"MyConfigSet\" ]" }
```



# Amazon SES 送信者評価のモニタリング

Amazon SES では、送信者としての評価にダメージを与える可能性のあるメトリクスや、メール配信率の低下を招く可能性のあるメトリクスなど、複数のメトリクスがアクティブに追跡されます。このプロセスで監視する 2 つの重要なメトリクスは、アカウントのバウンス率と苦情率です。アカウントのバウンス率や苦情率が高すぎる場合、アカウントを確認し、アカウントの E メール送信機能を一時停止することがあります。

アカウントの状態にとってバウンス率および苦情率が非常に重要であるため、Amazon SES にはこれらのメトリクスを追跡するための評価ダッシュボードが用意されています。評価ダッシュボードでは、バウンスや苦情に関係なく送信者の評価にダメージを与える可能性のある要因に関する情報も確認できます。たとえば、既知の**スパムトラップ**にメールを送信すると、このダッシュボードにメッセージが表示されます。

このセクションには、評価ダッシュボードへのアクセスや評価ダッシュボード内の情報の解釈に関する情報のほか、送信者評価に影響する可能性のある要因をアクティブに通知するためのシステム設定に関する情報が含まれています。

このセクションでは、以下のトピックについて説明します。

- [評価ダッシュボードを使用してバウンス率と苦情率を追跡する \(p. 377\)](#)
- [評価ダッシュボードのメッセージ \(p. 378\)](#)
- [CloudWatch を使用して評価モニタリングアラームを作成する \(p. 390\)](#)
- [専用 IP の SNDS メトリクス \(p. 394\)](#)
- [E メール送信の自動的な一時停止 \(p. 395\)](#)

## 評価ダッシュボードを使用してバウンス率と苦情率を追跡する

評価ダッシュボードには、Amazon SES チームが個人アカウントの状態を判断する際に確認するものと同じ情報が含まれています。

評価ダッシュボードを表示するには

1. AWS Management Console にサインインして Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. 画面左側のナビゲーションペインで、[Reputation Dashboard] を選択します。

ダッシュボードには、以下の内容が表示されます。

- [Account status (アカウントのステータス)] - アカウントの状態を示す簡単な説明。以下に示しているのは、可能な値です。
  - [正常] - 現在、アカウントに影響する問題はありません。
  - [Under review (レビュー対象)] - お客様のアカウントはレビュー対象です。アカウントのレビュー対象となった原因の問題が、レビュー期間の終了までに修正されなかった場合、アカウントの E メール送信が一時停止される可能性があります。
  - [Pending end of review decision (レビューの決定終了保留中)] - お客様のアカウントはレビュー対象です。お客様のアカウントをレビュー対象とした問題の内容により、何らかの処置を行う前に、手動でレビューを実行する必要があります。

- [Sending paused (送信一時停止)] - アカウントの E メール送信機能を一時停止しました。アカウントが一時停止されている間は、Amazon SES を使用して E メールを送信することはできません。この決定の見直しをリクエストできます。レビューをリクエストする方法の詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。
- [Pending sending pause (送信一時停止保留中)] - お客様のアカウントはレビュー対象です。アカウントをレビュー対象とした問題は解決されていません。このような状況では、通常アカウントの E メール送信機能を一時停止します。ただし、アカウントの性質により、何らかの処置を行う前に、アカウントの確認が行われます。
- [Bounce Rate (バウンス率)] - アカウントから送信され、ハードバウンスとなった Eメールのパーセンテージ。
- [Complaint Rate] - アカウントから送信され、受信者によってスパムと報告されたメールのパーセンテージ。

#### Note

[Bounce Rate] セクションおよび [Complaint Rate] セクションには、それぞれのメトリクスに関するステータスメッセージも含まれています。以下は、これらに対してメトリクスについて表示される可能性のあるステータスメッセージのリストです。

- Healthy - メトリクスは通常のレベルです。
- Almost healed - メトリクスにより、アカウントがレビュー対象になっています。レビュー期間が開始してから、メトリクスは最大レート未満を維持しています。メトリクスが最大レート未満のままであれば、レビュー期間が終了する前にこのメトリクスのステータスが Healthy に変化する可能性があります。
- Under review - メトリクスによってアカウントがレビュー対象になり、メトリクスはまだ最大レートを超過しています。メトリクスが最大レートを超過した問題が、レビュー期間の終了までに修正されなかった場合、アカウントの E メール送信が一時停止される可能性があります。
- Sending pause - メトリクスにより、アカウントの E メール送信機能が一時停止されます。アカウントが一時停止されている間は、Amazon SES を使用して E メールを送信することはできません。この決定の見直しをリクエストできます。レビューのリクエストを送信する方法の詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。
- [Pending sending pause (送信一時停止保留中)] - メトリクスによりアカウントがレビューされています。このレビューの原因となった問題が、解決していません。この問題により、アカウントの E メール送信機能が一時停止される可能性があります。その先のアクションが発生する前に、Amazon SES チームのメンバーによってアカウントの確認が行われます。
- [Other Notifications (その他の通知)] - バウンスや苦情に関係のない評価関連の問題がアカウントで発生している場合、ここに短いメッセージが表示されます。ここに表示される可能性のある通知の詳細については、「[評価ダッシュボードのメッセージ \(p. 378\)](#)」を参照してください。

#### Note

評価ダッシュボードは、AWS コンソールへのアクセス権を持つすべてのユーザーが利用できません。IAM ポリシーを使用して評価ダッシュボードへのアクセスを制限することはできません。

## 評価ダッシュボードのメッセージ

Amazon SES 評価ダッシュボードには、アカウントに関する重要なメトリクスが表示されます。以下のセクションでは、このダッシュボードに表示される可能性のあるメッセージについて説明し、送信者評価に関連する問題の解決に役立つ可能性があるヒントと使用できる情報を示します。

このセクションでは、以下のタイプの通知についての情報を提供します。

- [ステータスメッセージ \(p. 379\)](#)
- [バウンス率の通知 \(p. 380\)](#)
- [苦情率の通知 \(p. 381\)](#)
- [アンチスパム組織の通知 \(p. 382\)](#)
- [直接フィードバック通知 \(p. 383\)](#)
- [ドメインブロックリスト通知 \(p. 384\)](#)
- [内部レビュー通知 \(p. 385\)](#)
- [メールボックスプロバイダー通知 \(p. 386\)](#)
- [受信者フィードバック通知 \(p. 387\)](#)
- [関連アカウント通知 \(p. 388\)](#)
- [スパムトラップ通知 \(p. 388\)](#)
- [脆弱サイト通知 \(p. 389\)](#)
- [その他の通知 \(p. 390\)](#)

## ステータスメッセージ

評価ダッシュボードを使用する際に、Amazon SES アカウントのステータスを表すメッセージが表示されます。表示される可能性のあるアカウントステータス値の一覧を以下に示します。

- [正常] - 現在、アカウントに影響する問題はありません。
- [Under review (レビュー対象)] - お客様のアカウントはレビュー対象です。アカウントのレビュー対象となった原因の問題が、レビュー期間の終了までに修正されなかった場合、アカウントの E メール送信が一時停止される可能性があります。
- [Pending end of review decision (レビューの決定終了保留中)] - お客様のアカウントはレビュー対象です。お客様のアカウントをレビュー対象とした問題の内容により、何らかの処置を行う前に、手動でレビューを実行する必要があります。
- [Sending paused (送信一時停止)] - アカウントの E メール送信機能を一時停止しました。アカウントが一時停止されている間は、Amazon SES を使用して E メールを送信することはできません。この決定の見直しをリクエストできます。レビューをリクエストする方法の詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。
- [Pending sending pause (送信一時停止保留中)] - お客様のアカウントはレビュー対象です。アカウントをレビュー対象とした問題は解決されていません。このような状況では、通常アカウントの E メール送信機能を一時停止します。ただし、アカウントの性質により、何らかの処置を行う前に、アカウントの確認が行われます。

さらに、評価ダッシュボードの [Bounce Rate] セクションと [Complaint Rate] セクションには、それぞれのメトリクスのステータス要約が表示されます。表示される可能性のあるメトリクスのステータス値の一覧を以下に示します。

- Healthy - メトリクスは通常のレベルです。
- Almost healed - メトリクスにより、アカウントがレビュー対象になっています。レビュー期間が開始してから、メトリクスは最大レート未満を維持しています。メトリクスが最大レート未満のままであれば、レビュー期間が終了する前にこのメトリクスのステータスが Healthy に変化する可能性があります。
- Under review - メトリクスによってアカウントがレビュー対象になり、メトリクスはまだ最大レートを超過しています。メトリクスが最大レートを超過した問題が、レビュー期間の終了までに修正されなかった場合、アカウントの E メール送信が一時停止される可能性があります。

- Sending pause - メトリクスにより、アカウントの E メール送信機能が一時停止されます。アカウントが一時停止されている間は、Amazon SES を使用して E メールを送信することはできません。この決定の見直しをリクエストできます。レビューのリクエストを送信する方法の詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。
- [Pending sending pause (送信一時停止保留中)] - メトリクスによりアカウントがレビューされています。このレビューの原因となった問題が、解決していません。この問題により、アカウントの E メール送信機能が一時停止される可能性があります。その先のアクションが発生する前に、Amazon SES チームのメンバーによってアカウントの確認が行われます。

## バウンス率の通知

このセクションでは、Amazon SES の評価ダッシュボードに表示されるバウンス率通知についての追加情報を提供します。

### この通知を受け取った理由

この通知は、アカウントのバウンス率が高すぎたために受け取りました。バウンス率は、Amazon SES アカウントによって生成されたハードバウンスの数に基づいています。E メールプロバイダーでは、バウンス率の高さは、送信者が受取人リストを適切に管理しておらず、未承諾 E メールを送信している可能性を示すサインとみなします。

ハードバウンスは、存在しないアドレスに E メールが送信されると発生します。Amazon SES はこの計算においてソフトバウンス (受信者のアドレスで一時的にメッセージを受信できない場合に発生するもの) を考慮しません。確認済みのアドレスおよびドメインに送信するバウンスメールや、[Amazon SES インボックスシミュレーター \(p. 187\)](#)に送信するメールも、この計算に反映されません。

バウンス率は、Eメールの代表ボリュームに基づいて計算されます。代表ボリュームは、通常の送信行為を表す Eメールの量です。大量のメールの送信者と少量のメールの送信者を公平に扱うために、代表ボリュームはアカウントごとに異なっており、アカウントの送信パターンの変化にともなって代表ボリュームも変わります。

最良の結果を得るには、バウンス率を 5% 未満に保ちます。これより高いバウンス率は、Eメールの配信に影響する可能性があります。バウンス率が 5% 以上になると、アカウントは自動的にレビュー対象になります。バウンス率が 10% を超える場合は、高いバウンス率の原因となった問題が解決するまで、お客様のアカウントの今後の Eメール送信を一時停止することがあります。

### 問題を解決する方法

バウンスおよび苦情をキャプチャおよび管理するためのプロセスを正しく配備してください (まだの場合)。Amazon SES アカウントはすべて、これらのプロセスを正しく配備する必要があります。詳細については、「[Eメールプログラムの成功のメトリクス \(p. 475\)](#)」を参照してください。

次に、どのメールアドレスでバウンスが発生しているか特定し、バウンスの発生を抑えて排除するためのプランを作成および実施します。アカウントの Eメール送信機能がすでに一時停止している場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。

### お客様のアカウントがレビュー対象になっている場合

レビュー期間の終了時にアカウントのバウンス率が 10% を超えている場合、問題が解決されるまで、お客様のアカウントの Eメール送信機能を一時停止することがあります。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。ケースへのレスポンスで、実装した変更について説明します。この変更によってバウンス率が低下すると見なされた場合は、変更の実施後に受信するバウンスのみが含まれるように計算が調整されます。

## アカウントの E メール送信機能が一時停止された場合

この決定の再検討をリクエストできます。詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

問題解決のための変更を実施する場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。

## 苦情率の通知

このセクションでは、Amazon SES の評価ダッシュボードに表示される苦情率通知についての追加情報を提供します。

### この通知を受け取った理由

この通知は、アカウントの苦情率が高すぎたために受け取りました。苦情率は、Amazon SES アカウントによって生成された苦情の数に基づいています。E メールプロバイダーでは、苦情率の高さは、送信者が受取人リストを適切に管理しておらず、未承諾 E メールを送信している可能性を示すサインとみなします。

苦情は、送信した E メールを受信者がスパムと認識した場合に発生します。これは通常、受信者が E メールクライアントの [スパムを報告] ボタンを使用する場合に発生します。[Amazon SES メールボックスシミュレーター \(p. 187\)](#) に送信する E メールによって生成された苦情は、この計算では考慮されません。

苦情率は、Eメールの代表ボリュームに基づいて計算されます。代表ボリュームは、通常の送信行為を表す Eメールの量です。大量のメールの送信者と少量のメールの送信者を公平に扱うために、代表ボリュームはアカウントごとに異なっており、アカウントの送信パターンの変化にともなって代表ボリュームも変わります。

最良の結果を得るには、苦情率を 0.1% 未満に維持してください。これより高い苦情率は、Eメールの配信に影響する可能性があります。苦情率が 0.1% 以上になると、アカウントは自動的にレビュー対象になります。苦情率が 0.5% を超える場合は、高い苦情率の原因となった問題が解決するまで、お客様のアカウントの今後の Eメール送信を一時停止することがあります。

### 問題を解決する方法

バウンスおよび苦情をキャプチャおよび管理するためのプロセスを正しく配備してください (まだの場合)。Amazon SES アカウントはすべて、これらのプロセスを正しく配備する必要があります。詳細については、「[Eメールプログラムの成功のメトリクス \(p. 475\)](#)」を参照してください。

次に、送信したどのメッセージが苦情を招いたか特定し、このような苦情を減少させるためのプランを実施します。アカウントの Eメール送信機能がすでに一時停止している場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信します。

苦情があったアドレスへの送信は直ちに停止する必要があると同時に、受信側で苦情が発生する要因を特定することも重要です。これらの要因を特定した後、対処できるようにメール送信の動作を調整します。

### お客様のアカウントがレビュー対象になっている場合

レビュー期間の終了時にアカウントの苦情率が 0.5% を超えている場合、問題が解決されるまで、お客様のアカウントの Eメール送信機能を一時停止することがあります。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。ケースへのレスポンスで、実装した変更について説明します。この変更によって苦情率が低下すると見なされた場合は、変更の実施後に受信した苦情のみが考慮されるように計算が調整されます。

## アカウントの E メール送信機能が一時停止された場合

この決定の再検討をリクエストできます。詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。

## アンチスパム組織の通知

このセクションでは、Amazon SES の評価ダッシュボードに表示されるアンチスパム組織通知についての追加情報を提供します。

### この通知を受け取った理由

信頼できるアンチスパム組織により、お客様の Amazon SES アカウントから送信されているコンテンツの一部について、未承諾または問題ありというシステムによるフラグ付けがされたと報告されています。

アンチスパム組織がコンテンツに問題があると判断した原因となった特定のメッセージに関する情報を提供することはできません。また、そのレポートを発行した組織の名前を開示することもできません。一般的にアンチスパム組織では、受信者のフィードバック、メッセージのエンゲージメントに関するメトリクス、無効なアドレスへの配信の試行数、スパムフィルタによってフラグ付けされたコンテンツ、スパムトラップのヒット数などの要因の組み合わせが考慮されます。これはすべてを網羅したリストではありません。これ以外の要因によって、お客様のコンテンツへのフラグ付けが行われることもあります。

### 問題を解決する方法

この問題を解決するには、E メール送信プログラムのどのような点が原因で、アンチスパム組織によって問題のあるメールとしてのフラグ付けが行われているのか調べる必要があります。そのうえで、問題に対処できるようにメール送信プログラムを変更する必要があります。

### お客様のアカウントがレビュー対象になっている場合

レビュー期間の終了時に、アンチスパム組織が引き続きお客様のアカウントから送信された E メールを問題のあるものとして特定した場合は、問題が解決されるまで、お客様のアカウントの E メール送信機能を一時停止する可能性があります。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージには、加えた変更の詳細を入力します。この情報の到着後、レビュー期間が延長されます。これは、お客様による変更の実施後にアンチスパム組織から送信される通知のみを分析するためです。延長されたレビュー期間の終了時に、お客様のアカウントはアンチスパム組織のリストに含まれなくなり、お客様のアカウントのレビュー期間は解除されます。

### アカウントの E メール送信機能が一時停止された場合

この決定の再検討をリクエストできます。詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。

## 直接フィードバック通知

このセクションでは、Amazon SES の評価ダッシュボードに表示される直接フィードバック通知についての追加情報を提供します。

### この通知を受け取った理由

お客様の Amazon SES アカウントに関連するアドレスまたはドメインから受信したメッセージについてのレポートが、相当数のユーザーから Amazon SES に直接寄せられました。このタイプのフィードバックは、メールボックスプロバイダーから直接報告される苦情には反映されておらず、評価ダッシュボードに表示されるバウンスと苦情のメトリクスにも含まれていません。

これらの問題を報告したユーザーのプライバシーを保護するために、そのユーザーの E メールアドレスを開示することはできません。

受信者から Amazon SES への苦情は、受信者が受信登録していないメッセージを受信した場合や、受信する予定のタイプのメールを受信しなかった場合、受信したメールが役立つメールまたは興味を引くメールではなかった場合、受信したメッセージが受信登録したものではないと思われる場合、受信するメッセージが多すぎる場合などに生じます。これはすべてを網羅したリストではありません。お客様のケースに該当する要因は、お使いの E メール送信プログラムによって異なります。

### 問題を解決する方法

新しいアドレスを取得するには「[リストの構築とメンテナンス \(p. 478\)](#)」に記載されているダブルオプトイン戦略を実装し、ダブルオプトインプロセスを完了したアドレスのみにメールを送信することをお勧めします。

さらに、送信したメールに対してしばらく応答がなかった一連のアドレスは、消去する必要があります。送信したコンテンツに対して閲覧および応答を行っているユーザーを特定するには、「[Amazon SES 送信アクティビティのモニタリング \(p. 267\)](#)」に記載されているオープンとクリックの追跡を使用できます。

### お客様のアカウントがレビュー対象になっている場合

レビュー期間の終了時に、Amazon SES が引き続きお客様のアカウントから送信されたメッセージに関する直接的な苦情を多数受け取る場合、問題が解決されるまで、お客様のアカウントの E メール送信機能を一時停止する可能性があります。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。問題を解決するために行った手順の詳細情報を提供し、これらの手順を実行して今後問題が再発するのを防ぐ方法を説明してください。お客様が加えた変更により問題が適切に対処されたと思われた場合は、アカウントのレビュー期間をキャンセルします。

### アカウントの E メール送信機能が一時停止された場合

この決定の再検討をリクエストできます。詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解

決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。

## ドメインブロックリスト通知

このセクションでは、Amazon SES の評価ダッシュボードに表示されるドメインブロックリスト通知についての追加情報を提供します。

### この通知を受け取った理由

信頼できるメインブロックリストに記載のあるドメインへの参照が、お客様の Amazon SES アカウントから送信されたメールに含まれています。これらのリストに記載のあるドメインは一般的に、不正または悪意のある動作と関連しています。そのドメインがメールの送信元ドメインかどうかは問題ではありません。ブロックリストに記載のあるドメインへの参照またはリンクが含まれるメッセージのほか、このようなドメインでホストされている画像が含まれたメッセージにもフラグが付けられる可能性があります。

お客様のメッセージにフラグが付く原因となったドメイン名を提供することも、どのメッセージにフラグが付いたかを特定することもできません。

### 問題を解決する方法

まず、Amazon SES 経由で送信する E メールで参照されているすべてのドメインのリストを作成します。次に、[Spamhaus ドメインルックアップツール](#)を使用して、Eメールのどのドメインがドメインブロックリストに含まれているかを判断します。送信した E メール内で参照されている複数のドメインが、このブロックリストに含まれていることもあります。

Spamhaus ドメインブロックリストは、Amazon SES や AWS とは関係がありません。当社では、このリストのドメインの正確性を保証していません。Spamhaus Domain Blocklist および Domain Lookup Tool は、[Spamhaus Project](#) によって所有、運用、保守されています。

### お客様のアカウントがレビュー対象になっている場合

レビュー期間中に送信する E メールで、悪意のある目的で使用されたドメインへの参照がないか確認されます。これらのドメインへの参照が E メールに引き続き多数含まれている場合、問題が解決されるまで、アカウントの E メール送信機能が一時停止されることがあります。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージには、加えた変更の詳細を入力します。この情報の到着後、レビュー期間が延長されます。これは、お客様による変更の実施後にお客様の E メールに含まれるブロックリスト記載ドメイン数のみを分析するためです。この延長されたレビュー期間の終了時に、ドメインブロックリスト通知の数が減ったり削除されたりした場合、およびお客様がこの問題の今後の再発を防ぐための対策を講じたと考えられる場合、アカウントのレビュー期間はキャンセルされます。

### アカウントの E メール送信機能が一時停止された場合

この決定の再検討をリクエストできます。詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。



## 内部レビュー通知

このセクションでは、Amazon SES の評価ダッシュボードに表示される内部レビュー通知についての追加情報を提供します。

### この通知を受け取った理由

お客様のアカウントに関する総合的なレビューにより、メールボックスプロバイダーまたは受信者によってお客様のメッセージがスパムであると見なされる原因となった特性がいくつか特定されました。

不正検出プロセスを保護するために、お客様のアカウントがこのようにフラグ付けされる原因となった特定の要因を開示することはできません。

この判断につながる可能性のある一般的な要因としては、次のようなものがあります。

- 商用のアンチスパムシステムによってフラグ付けされるメッセージ。
- 受取人がメールを明示的にリクエストしていないことを示すメッセージコンテンツ。
- メッセージの送信者とメール本文内に記載のあるブランドとの不一致。
- 送信者を明らかにしていないコンテンツ。
- 未承諾メールに関連したコンテンツを扱うメッセージの送信。
- 未承諾メールに関連する書式パターン。
- 評価の低いドメインからの送信または評価の低いドメインの参照。

これは包括的なリストではありません。この通知が発生した理由は、これらの要因の組み合わせである場合も、ここに記載されていない場合もあります。

### 問題を解決する方法

問題の深刻度の軽減には、以下の推奨事項が役立つ可能性があります。

- お客様からの E メールを明示的に要求した人のみがメールの受信者になっていることを確認します。
- メールを受取人のリストは、絶対に購入、賃借、借用しないでください。
- 送信するメッセージでは、送信者名やメールの目的を隠そうとしないでください。
- Amazon SES を介して送信する E メール内で参照されているすべてのドメインのリストを作成し、Spamhaus Domain Lookup ツール (<https://www.spamhaus.org/lookup/>) を使用して、これらのドメインが Spamhaus Domain Blocklist に記載されていないか調べます。
- メールを設計するときは、業界のベストプラクティスに従っていることを確認します。

これはすべてを網羅したリストではありませんが、お客様の E メールにフラグが付けられる原因となった一般的な要因の一部を特定するために役立つ可能性があります。

Spamhaus ドメインブロックリストは、Amazon SES や AWS とは関係がありません。当社では、このリストのドメインの正確性を保証していません。Spamhaus Domain Blocklist および Domain Lookup Tool は、[Spamhaus Project](#) によって所有、運用、保守されています。

### アカウントがレビュー対象になっている場合、またはアカウントの E メール送信機能が一時停止している場合

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。問題を解決するために行っ

た手順の詳細情報を提供し、これらの手順を実行して今後問題が再発するのを防ぐ方法を説明してください。お客様が加えた変更により問題が適切に対処されたと見なされた場合は、アカウントのレビュー期間をキャンセル、または一時停止を解除します。

お客様のアカウントからレビュー期間または送信一時停止を解除したものの、後で同じ問題が発生した場合、アカウントを再度レビュー対象としたり、お客様の E メール送信機能を一時停止する可能性があります。極端なケースや、同じ問題が繰り返し発生した場合は、お客様のアカウントの E メール送信機能を完全に停止する可能性があります。

アカウントがレビュー対象になった場合、またはアカウントの E メール送信機能が一時停止された場合の対処方法については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

## メールボックスプロバイダー通知

このセクションでは、Amazon SES の評価ダッシュボードに表示されるメールボックスプロバイダー通知についての追加情報を提供します。

### この通知を受け取った理由

お客様の Amazon SES アカウントに関連するアドレスまたはドメインから、未承諾メールまたは悪意のあるメールが送信されたと、主要なメールボックスプロバイダーから報告されました。

このレポートを発行した組織の名前を開示することはできません。また、メールボックスプロバイダーからレポートが発行される原因となった特定の要因に関する情報も提供できません。一般的に、メールボックスプロバイダーでは、顧客フィードバック、顧客エンゲージメントに関するメトリクス、無効アドレスに対する配信の試行数、スパムフィルタによってフラグ付けされたコンテンツなどの情報に基づいて、このような判断が行われます。これはすべてを網羅したリストではありません。メールボックスプロバイダーでお客様のコンテンツにフラグが付けられる原因となった要因は、他にも存在する可能性があります。

### 問題を解決する方法

この問題を解決するには、E メール送信プログラムのどのような点が原因で、メールボックスプロバイダーでお客様のメールに問題があるとしてフラグが付けられたのか調べる必要があります。そのうえで、問題に対処できるようにメール送信プログラムを変更する必要があります。

### お客様のアカウントがレビュー対象になっている場合

レビュー期間の終了時に、メールボックスプロバイダーが引き続きお客様のアカウントから送信された E メールを問題があるとして特定する場合は、問題が解決されるまで、お客様のアカウントの E メール送信機能を一時停止する可能性があります。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージには、加えた変更の詳細を入力します。この情報の到着後、レビュー期間が延長されます。これは、お客様による変更の実施後に受信したメールボックスプロバイダー通知の数のみを分析するためです。延長されたレビュー期間の終了時に、お客様のコンテンツに問題があるとメールボックスプロバイダーから報告されなければ、お客様のアカウントからレビューが解除される場合があります。

### アカウントの E メール送信機能が一時停止された場合

この決定の再検討をリクエストできます。詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。

## 受信者フィードバック通知

このセクションでは、Amazon SES の評価ダッシュボードに表示される受信者フィードバック通知についての追加情報を提供します。

### この通知を受け取った理由

主要なメールボックスプロバイダーから受けた報告により、そのプロバイダーの多くのユーザーによって、お客様の Amazon SES アカウントから送信されたメールが未承諾であるとされていることが判明しました。このタイプのフィードバックは、メールボックスプロバイダーから直接報告される苦情には反映されておらず、Amazon SES のバウンス通知および苦情通知にも含まれていません。

多数の苦情が発生すると、すべての Amazon SES ユーザーに悪影響が及ぶ可能性があります。アカウントに対して一定数の苦情が寄せられた場合は、お客様およびその他の Amazon SES ユーザーの評価を保護するために、迅速なアクションが実行されます。

お客様のメールを未承諾であるとしている特定のメールアドレスのリストは、提供できません。さらに、この問題に関する報告元であるメールボックスプロバイダー名も開示できません。

### 問題を解決する方法

この問題を解決するには、E メール送信プログラムのどのような点が原因で、受取人がお客様から受け取った E メールメッセージに関する苦情が生じたのか調べる必要があります。これらの要因を特定した後、対処できるようにメール送信の処理を変更します。

新しいアドレスを取得するには「[リストの構築とメンテナンス \(p. 478\)](#)」に記載されているダブルオプトイン戦略を実装することをお勧めします。また、ダブルオプトインプロセスを完了したアドレスのみにメールを送信することをお勧めします。

さらに、送信したメールに対してしばらく応答がなかった一連のアドレスは、消去する必要があります。送信したコンテンツに対して閲覧および応答を行っているユーザーを特定するには、「[Amazon SES 送信アクティビティのモニタリング \(p. 267\)](#)」に記載されているオープンとクリックの追跡を使用できます。

### お客様のアカウントがレビュー対象になっている場合

レビュー期間の終了時に、メールボックスプロバイダーが直接的な苦情を引き続き多数報告する場合、問題が解決されるまで、お客様のアカウントの E メール送信機能を一時停止する可能性があります。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージには、加えた変更の詳細を入力します。この情報の到着後、レビュー期間が延長されます。これは、お客様による変更の実施後に受信したメールボックスプロバイダーからの苦情数のみを分析するためです。延長されたレビュー期間の終了時に、メールボックスプロバイダーからの苦情数が減少するかゼロになっていれば、お客様のアカウントからレビューが解除される場合があります。

### アカウントの E メール送信機能が一時停止された場合

この決定の再検討をリクエストできます。詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。

## 関連アカウント通知

このセクションでは、Amazon SES の評価ダッシュボードに表示される関連アカウント通知についての追加情報を提供します。

### この通知を受け取った理由

別の Amazon SES アカウントから送信されたメールに関連する重要な問題が検出されました。問題のあるアカウントはお客様の AWS アカウントと関連していると思われる、同様の問題を回避するためにアクションが実行されました。

### 問題を解決する方法

あるアカウントの E メール送信機能を一時停止する場合、送信一時停止の理由に関する情報が常にそのアカウントの所有者に送信されます。詳細については、関連するアカウントの所有者に送信した E メールを参照してください。

まず、この関連アカウントの問題に対処してください。問題解決のための変更を実施した後は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。問題を解決するために行った手順の詳細情報を提供し、これらの手順を実行して今後問題が再発するのを防ぐ方法を説明してください。お客様が加えた変更により問題が適切に対処されたと思われた場合は、アカウントのレビュー期間をキャンセル、または一時停止を解除します。

## スパムトラップ通知

このセクションでは、Amazon SES の評価ダッシュボードに表示されるスパムトラップ通知についての追加情報を提供します。

### この通知を受け取った理由

サードパーティのアンチスパム組織から、その組織のスパムトラップアドレスで、お客様の Amazon SES アカウントに関連付けられている確認済みアドレスまたはドメインからメールを受信したという報告を受けました。

スパムトラップは、未承諾メール (スパム) だけを引き付けるために使用される休眠メールアドレスです。多数のスパムトラップレポートが発生すると、すべての Amazon SES ユーザーに悪影響が及ぶ可能性があります。アカウントから一定量のメールがスパムトラップアドレスに送信された場合は、お客様およびその他の Amazon SES ユーザーの評価を保護するために、迅速なアクションが実行されます。

### 問題を解決する方法

使用されたスパムトラップに関連する E メールアドレスを開示することはできません。これらのアドレスは、公開されると意味がなくなるため、所有元の組織で厳重に保護されています。

スパムトラップアドレスにメールが送信される場合は、一般的に、顧客のメールアドレス入手方法に問題があることを示します。たとえば、購入したメールアドレスリストには、スパムトラップアドレスが含まれている可能性があります。Amazon SES のサービスの利用条件で、購入または借用したリストへの送信が禁止されているのは、このためです。新しいアドレスを取得するには「[リストの構築とメンテナンス](#)」

ス (p. 478)」に記載されているダブルオプトイン戦略を実装することをお勧めします。また、ダブルオプトインプロセスを完了したアドレスのみにメールを送信することをお勧めします。

さらに、送信したメールに対してしばらく応答がなかった一連のアドレスは、消去する必要があります。送信したコンテンツに対して閲覧および応答を行っているユーザーを特定するには、「[Amazon SES 送信アクティビティのモニタリング \(p. 267\)](#)」に記載されているオープンとクリックの追跡を使用できます。

## お客様のアカウントがレビュー対象になっている場合

レビュー期間の終了時に、お客様のアカウントから引き続きスパムトラップアドレスにメッセージが送信されている場合は、問題が解決されるまで、お客様のアカウントの E メール送信機能を一時停止する可能性があります。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージには、加えた変更の詳細を入力します。この情報の到着後、レビュー期間が延長されます。これは、お客様による変更の実施後に受信したスパムトラップレポートの数のみを分析するためです。延長されたレビュー期間の終了時に、スパムトラップレポートの数が減少するかゼロになっていれば、お客様のアカウントからレビューが解除される場合があります。

## アカウントの E メール送信機能が一時停止された場合

この決定の再検討をリクエストできます。詳細については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。

## 脆弱サイト通知

このセクションでは、Amazon SES の評価ダッシュボードに表示される脆弱性があるサイト通知についての追加情報を提供します。

### この通知を受け取った理由

お客様のアカウントに関する総合的なレビューにより、お客様が意図していないと思われるメッセージがお客様のアカウントから送信されたことがわかりました。これらのメッセージは、高い確率でメールボックスプロバイダーおよび受取人からスパムとしてフラグ付けされます。

このような場合は、お客様のウェブサイトの機能を悪用して第三者が迷惑メールを送信していることが考えられます。たとえば、お客様のウェブサイトに、「友だちにメールを送る」、「お問い合わせ」、「友だちを招待する」などの機能がある場合、第三者はその機能を使用して未承諾メールを送信できます。

### 問題を解決する方法

まず、お客様が知らない間に Amazon SES を使用して第三者が E メールを送信する可能性のある、お客様のウェブサイトまたはアプリケーションの機能を特定します。サポートセンターのケースでは、この方法で送信されたメッセージのサンプルをリクエストできます。

次に、未承諾送信を回避できるように、アプリケーションまたはウェブサイトを修正します。たとえば、CAPTCHA を追加する、メールを送信できる速度を制限する、ユーザーによるカスタムコンテンツの

送信を禁止する、メール送信前にユーザーにログインを求める、アプリケーションによる複数の同時通知の生成を禁止するなどの方法があります。

## アカウントがレビュー対象になっている場合、またはアカウントの E メール送信機能が一時停止している場合

問題解決のための変更を実施した場合は、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メールには、この問題の解決のために実行したアクションの詳細と、この問題を再度発生させないための計画の詳細を記入してください。リクエストを受け取った後、ご提供いただいた情報を確認し、必要に応じてアカウントのステータスを変更します。

お客様のアカウントからレビュー期間または送信一時停止を解除したものの、後で同じ問題が発生した場合、アカウントを再度レビュー対象としたり、お客様の E メール送信機能を一時停止する可能性があります。極端な問題や、同じ問題が繰り返し確認された場合は、お客様のアカウントの E メール送信機能を完全に停止する可能性があります。

アカウントがレビュー対象になった場合、またはアカウントの E メール送信機能が一時停止された場合の対処方法については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。

## その他の通知

このセクションでは、Amazon SES の評価ダッシュボードに表示されるその他の通知についての追加情報を提供します。

### この通知を受け取った理由

自動レビューまたは人によるレビューで、このドキュメントのこれまでのセクションに記載されていない問題が見つかりました。

### 問題を解決する方法

特定の問題の詳細について、お客様に代わって AWS がオープンしたサポートセンターのケースを参照します。サポートセンターにアクセスするには、AWS コンソールにサインインし、[Support Center (サポートセンター)] を選択します。ケースへのレスポンスで、実装した変更について説明します。検出された問題の状況と内容に応じて、レビュー期間が終了したり、お客様のアカウントの E メール送信機能が復旧する可能性があります。

## CloudWatch を使用して評価モニタリングアラームを作成する

Amazon SES は、一連の評価関連のメトリクスを自動的に Amazon CloudWatch に発行します。これらのメトリクスを使用して、バウンス率または苦情率がアカウントのメール送信機能に影響するレベルに達すると通知するアラームを作成できます。

### Note

このセクションの手順では、CloudWatch アラームのオプション設定に関する情報が一部省略されています。詳細な手順については、Amazon CloudWatch ユーザーガイドの「[Amazon CloudWatch アラームの作成](#)」を参照してください。

## CloudWatch アラームの作成

1. 新しい Amazon SNS トピックを作成し、任意のエンドポイント (メールや SMS など) を使用してそのトピックをサブスクライブします。詳細については、Amazon Simple Notification Service デベロッパーガイドの「[トピックの作成](#)」および「[トピックにエンドポイントをサブスクライブする](#)」を参照してください。
2. CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
3. ナビゲーションペインで、[Alarms] を選択します。
4. [Specify metric and conditions (メトリクスと条件を指定)] ページで、以下の操作を行います。
  - a. [メトリクス] で、[メトリクスの選択] を選択します。
  - b. メトリクスのリストで、[SES] を選択します。

### Note

現在の AWS リージョンで E メールを送信したことがない場合は、利用可能なメトリクスのリストに [SES] が表示されない場合があります。SES メトリクスを表示するには、[\[Amazon SES メールボックスシミュレーター \(p. 187\)\]](#) にテストメールを送信します。メトリクスが数分以内に CloudWatch に表示されます。

- c. [アカウントメトリクス] を選択します。
- d. アラームを作成する対象のメトリクスを選択します。

たとえば、バウンス率が特定のレベルに達したときにアラームを作成するには、Reputation.BounceRate を選択します。苦情率が特定のレベルに達したときにアラームを作成するには、Reputation.ComplaintRate を選択します。

### Note

アカウントでバウンスまたは苦情が発生したことがない場合、Reputation.BounceRate および Reputation.ComplaintRate メトリクスはこのページに表示されません。

メトリクスを選択したら、[メトリクスの選択] を選択します。

- e. [条件] セクションの [しきい値のタイプ] で、[静的] を選択します。
- f. [Whenever Reputation.**MetricName** is] で、[Greater/Equal] を選択します。
- g. [than] で、CloudWatch でアラームを発生させる値を指定します。

アラームを作成してバウンス率をモニタリングする場合、Amazon SES ではバウンス率を 5% よりも低く維持することをお勧めします。アカウントのバウンス率が 10% を超える場合、当社はお客様のアカウントによる E メール送信機能を一時停止することがあります。このため、次のイメージで示すように、アカウントのバウンス率が 0.05 (5%) 以上になると通知が送信されるように CloudWatch を設定する必要があります。

## Conditions

### Threshold type

**Static**  
Use a value as a threshold

**An**  
Us

### Whenever Reputation.BounceRate is...

Define the alarm condition

**Greater**  
> threshold

**Greater/Equal**  
>= threshold

**Lo**  
<=

### than...

Define the threshold value

0.05



Must be a number

アラームを作成して苦情率をモニタリングする場合、Amazon SES では苦情率を 0.1% よりも低く維持することをお勧めします。アカウントの苦情率が 0.5% を超える場合、当社はお客様のアカウントによる Eメール送信機能を一時停止することがあります。このため、次のイメージで示すように、アカウントの苦情率が 0.001 (0.1%) 以上になると通知が送信されるように CloudWatch を設定する必要があります。





## 専用 IP の SNDS メトリクス

Amazon SES を使用する各 AWS リージョンで、リース専用 IP アドレスのスマートネットワークデータサービス (SNDS) データを表示できます。この SNDS データは、Amazon CloudWatch コンソールで使用できます。

SNDS は、IP 所有者が IP 空間内のスパムを防止できるようにする Outlook プログラムです。Amazon SES は、専用 IP をリースするユーザーのために、この重要なデータを提供します。SNDS データは、IP のメール送信動作に関するインサイトを提供し、送信者の評価が懸念される領域を呼び出します。

### Note

Outlookを参照する際、これはトラッキングするすべてのドメインをカバーします。例えば、これは Hotmail.com、Outlook.com、および Live.com をカバーできます。

専用 IP アドレスの SNDS データを表示するには

1. Amazon CloudWatch コンソールにサインインします。<https://console.aws.amazon.com/cloudwatch/>
2. ナビゲーションペインで メトリクスを選択します。
3. 選択した AWS リージョンの [すべてのメトリクス] タブで、AWS の名前空間下の [SES] を選びます。
4. 選択 IP メトリクスを選ぶと、SNDS によって追跡される専用の IP がすべて表示されます。
5. 一覧の SNDS によって追跡されるすべての専用 IP を表示するか、メトリクスのみを表示するために個別の IP アドレスを選択します。

次のメトリクスは、専用 IP アドレスごとに提供され、Outlook によって定義されます。詳細については、Outlook SNDS の [よくある質問](#) を参照してください。

### Note

これらのメトリクスは、1 日に 1 回更新データを提供する活動期間を表します。メトリクスには、24 時間に対応するタイムスタンプもあります。

- SNDS.RCPTCommands - これは、活動期間中に特定の IP アドレスに対して SNDS によって認識される RCPT コマンドの数。RCPT コマンドは、メールの送信に使用される SMTP プロトコルの部分です。SMTP プロトコルでは、電子メールの配信先となる受信者アドレスを指定します。
- SNDS.DATACommands - 活動期間中に特定の IP アドレスに対して SNDS によって認識される DATA コマンドの数。DATA コマンドは、メールを送信するために使用される SMTP プロトコルの部分です。具体的には、以前に確立された意図された受信者に実際にメッセージを送信する部分です。
- SNDS.MessageRecipients - 活動期間中に特定の IP アドレスについて、SNDS によって認識されたメッセージの受信者数。
- SNDS.SpamRate - 指定した活動期間中に IP アドレスによって送信されたすべてのメッセージに適用されたスパムフィルタリングの集計結果を表示。
  - スпамレート 0 は、IP アドレスのスパムが 10% 未満であることを意味します。
  - スпамレート 0.5 は、IP アドレスから 10% ~ 90% のスパムが生成されたことを意味します。
  - スпамレート 1 は、IP アドレスから 90% 以上のスパムが生成されたことを意味します。
- SNDS.ComplaintRate - 活動期間中に、IP から受信したメッセージが Outlook ユーザーによって不平を言われた時間の割合です。
  - クレームレート 1 は、苦情率が 100% を意味します。
  - たとえば、クレームレートが 0.05 の場合、苦情率が 5% を意味します。
  - クレームレート 0 は、レートが 0.1% 未満であることを意味します。
- SNDS.TrapHits - 「トラップアカウント」に送信されたメッセージの数。トラップアカウントは、Outlook によって管理され、メールを要求しないアカウントです。したがって、トラップアカウントに送信されたメッセージは、スパムである可能性が非常に高いです。

## トラブルシューティングに関する質問

Q1：データが毎日移入されないのはなぜですか？ 次のシナリオのいずれかが適用できます。

- SNDS データは、Outlook SNDS プログラムに依存します。
- 値を計算するためには SNDS が受信する必要がある E メール の最小値があります。IP 上の E メール のボリュームが少ない場合、データは利用できない場合があります。

Q2：SNDS.SpamRate および SNDS.ComplaintRate メトリクスが変更されるのはなぜですか。また、レートが 1 の値に変更された場合は何をすればよいですか？

これは、送信動作の何かが Outlook SNDS プログラムから否定的な応答をトリガーしたことを示すインジケータです。この場合、他のインターネットサービスプロバイダー (ISP) と契約番号を確認して、それがグローバルな問題ではないことをチェックします。これがグローバルな問題である場合は、複数の ISP での問題が発生する可能性があり、リスト、コンテンツ、ディストリビューション、またはアクセス許可の問題が考えられます。Outlook に固有のものである場合は、[Outlook に最適に配信する方法を参照ください](#)。ベストプラクティスの詳細については、「[Amazon SES を使用した E メール送信のベストプラクティス \(p. 475\)](#)」を参照してください。

Q3：SNDS.SpamRate の値が 0 (または 0.5) から 1 に変更された場合、AWS Support はどのようなアクションを実行しますか？

AWS には SNDS に対する制御がないため、SNDS に対する影響はありません。すべての緩和リクエストは [新しいサポートリクエストフォーム](#) を通じて Outlook に直接ファイルする必要があります。

## E メール送信の自動的な一時停止

送信者の評価の低下を防ぐために、特定の設定セットを使用して送信されているメッセージ、または特定の AWS リージョンの Amazon SES アカウントから送信されているメッセージすべてについて、Eメールの送信を一時的に停止できます。

Amazon CloudWatch と Lambda を使用して、自動化することで、評価メトリクス (バウンス率や苦情率など) が一定の特定のしきい値を超えた場合に、Eメール送信を自動的に停止させるソリューションを作成できます。このトピックでは、このソリューションをセットアップする手順について説明します。

このセクションのトピック:

- [Amazon SES アカウントの E メール送信を自動的に一時停止する \(p. 395\)](#)
- [設定セットの E メール送信を自動的に一時停止する \(p. 400\)](#)

## Amazon SES アカウントの E メール送信を自動的に一時停止する

このセクションの手順では、単一の AWS リージョンで Amazon SES アカウントの Eメール送信を自動的に一時停止するように Amazon SES、Amazon SNS、Amazon CloudWatch、AWS Lambda をセットアップするためのステップを説明します。複数のリージョンから Eメールを送信する場合は、このソリューションを実装するリージョンごとにこのセクションの手順を繰り返します。

このセクションのトピック:

- [パート 1: IAM ロールの作成 \(p. 396\)](#)
- [パート 2: Lambda 関数を作成する \(p. 396\)](#)
- [パート 3: アカウントの E メール送信の再有効化 \(p. 397\)](#)

- [パート 4: Amazon SNS トピックを作成する \(p. 398\)](#)
- [パート 5: CloudWatch アラームの作成 \(p. 398\)](#)
- [パート 6: ソリューションをテストする \(p. 399\)](#)

## パート 1: IAM ロールの作成

E メール送信の自動一時停止を設定するための最初のステップは、UpdateAccountSendingEnabled API オペレーションを実行できる IAM ロールを作成することです。

IAM ロールを作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [Roles (ロール)] を選択します。
3. [ロールの作成] を選択します。
4. [信頼されたエンティティの種類を選択] で、[AWS のサービス] を選択します。
5. [このロールを使用するサービスを選択] の下で、[Lambda] を選択します。[Next: Permissions (次へ: アクセス許可)] を選択します。
6. [Attach permissions policies] ページで、次のいずれかのポリシーを選択します。
  - AWSLambdaBasicExecutionRole
  - AmazonSESEFullAccess

### Tip

ポリシーのリストの上部にある検索ボックスを使用すると、これらのポリシーをすばやく特定できます。

[Next: Review] を選択します。

7. [Review] ページで、[Name] にロールの名前を入力します。[ロールの作成] を選択します。

## パート 2: Lambda 関数を作成する

IAM ロールを作成した後で、Lambda 関数を作成して、アカウントの E メール送信を一時停止することができます。

Lambda 関数を作成するには

1. AWS Lambda コンソール (<https://console.aws.amazon.com/lambda/>) を開きます。
2. リージョンセレクターを使用して、この Lambda 関数をデプロイするリージョンを選択します。

### Note

この関数では、このステップで選択した AWS リージョン内の E メール送信を一時停止します。複数のリージョンから E メールを送信する場合は、E メール送信を自動的に一時停止する各リージョンごとにこのセクションの手順を繰り返します。

3. [Create function] を選択します。
4. [Create function] で [Author from scratch] を選択します。
5. [Author from scratch] で、次のステップを実行します。
  - [名前] に Lambda 関数の名前を入力します。
  - [Runtime (ランタイム)] で、[Node.js 6.10] を選択します。

- [Role] で、[Choose an existing role] を選択します。
- [Existing role] には、[the section called “パート 1: IAM ロールの作成” \(p. 396\)](#) で作成した IAM ロールを選択します。

[Create function] を選択します。

6. コードエディタの [Function code] に、以下のコードを貼り付けます。

```
'use strict';

var aws = require('aws-sdk');

// Create a new SES object.
var ses = new aws.SES();

// Specify the parameters for this operation. In this case, there is only one
// parameter to pass: the Enabled parameter, with a value of false
// (Enabled = false disables email sending, Enabled = true enables it).
var params = {
  Enabled: false
};

exports.handler = (event, context, callback) => {
  // Pause sending for your entire SES account
  ses.updateAccountSendingEnabled(params, function(err, data) {
    if(err) {
      console.log(err.message);
    } else {
      console.log(data);
    }
  });
};
```

[Save] を選択します。

7. [Test] を選択します。[Configure test event] ウィンドウが表示されたら、[Event name] フィールドに名前を入力してから、[Create] を選択します。
8. ページ上部の通知バーに `Execution result: succeeded` と表示されていることを確認します。関数の実行に失敗した場合は、次のことを実行します。
  - [the section called “パート 1: IAM ロールの作成” \(p. 396\)](#) で作成した IAM ロールに正しいポリシーが含まれていることを確認します。
  - Lambda 関数のコードにエラーがないことを確認します。Lambda コードエディタでは、構文エラーやその他の潜在的な問題が自動的にハイライトされます。

## パート 3: アカウントの E メール送信の再有効化

[the section called “パート 2: Lambda 関数を作成する” \(p. 396\)](#) で Lambda 関数のテストをする副作用は、Amazon SES アカウントの E メール送信が一時停止していることです。ほとんどの場合は、CloudWatch アラームがトリガーされるまで、アカウントの送信を一時停止しません。

このセクションの手順では、Amazon SES アカウントの E メール送信を再度有効にします。これらの手順を完了するには、AWS Command Line Interface をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

E メール送信を再有効化するには

1. コマンドラインで次のコマンドを入力して、アカウントの E メール送信を再度有効にします。aws ses update-account-sending-enabled --enabled --region **us-west-2**

## Note

前述のコマンドで、`us-west-2` を、E メール送信を再び有効にするリージョンの名前に置き換えます。

2. コマンドラインで次のコマンドを入力して、アカウントの Eメールの送信ステータスを確認します。`aws ses get-account-sending-enabled --region us-west-2`

次の出力が表示された場合は、アカウントの Eメール送信が正常に再有効化されています。

```
{
  "Enabled": true
}
```

## パート 4: Amazon SNS トピックを作成する

アラームが発生したときに CloudWatch が Lambda 関数を実行するには、まず Amazon SNS トピックを作成し、Lambda 関数をサブスクライブする必要があります。

Amazon SNS トピックを作成するには

1. Amazon SNS コンソール (<https://console.aws.amazon.com/sns/v3/home>) を開きます。
2. リージョンセレクターを使用して、Eメール送信を自動的に一時停止するリージョンを選択します。
3. ナビゲーションペインで、[トピック] を選択します。
4. [Create new topic] を選択します。
5. [Create new topic] ウィンドウで、[Topic name] にトピックの名前を入力します。必要に応じて、[Display name] フィールドにわかりやすい名前を入力することもできます。

[トピックの作成] を選択します。

6. トピックのリストで、前のステップで作成したトピックの横にあるチェックボックスをオンにします。[Actions] メニューで、[Subscribe to topic] を選択します。
7. [Create subscription] ウィンドウで、以下の選択を行います。
  - [プロトコル] で [AWS Lambda] を選択します。
  - [Endpoint] に、[the section called “パート 2: Lambda 関数を作成する” \(p. 396\)](#) で作成した Lambda 関数を選択します。
  - [Version or alias] に [default] を選択します。
8. [Create subscription] を選択します。

## パート 5: CloudWatch アラームの作成

このセクションでは、メトリクスが特定のしきい値に到達するとトリガーされるアラームを CloudWatch に作成する手順を紹介します。アラームが発生すると、[the section called “パート 4: Amazon SNS トピックを作成する” \(p. 398\)](#) で作成した Amazon SNS トピックに通知が送信され、[the section called “パート 2: Lambda 関数を作成する” \(p. 396\)](#) で作成した Lambda 関数が実行されます。

CloudWatch アラームの作成

1. CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. リージョンセレクターを使用して、Eメール送信を自動的に一時停止するリージョンを選択します。
3. ナビゲーションペインで、[Alarms] を選択します。
4. [Create Alarm (アラーム作成)] を選択します。

5. [Create Alarm] ウィンドウの、[SES Metrics] の下で、[Account Metrics] を選択します。
  6. [Metric Name] で、次のいずれかのオプションを選択します。
    - Reputation.BounceRate – アカウントの全体のハードバウンス率が定義したしきい値を超えた場合に、アカウントの E メール送信を一時停止する場合は、このメトリクスを選択します。
    - Reputation.ComplaintRate – アカウントの全体の苦情率が定義したしきい値を超えた場合に、アカウントの E メール送信を一時停止する場合は、このメトリクスを選択します。
- [Next] を選択します。
7. 以下の手順を実行します。
    - [Alarm Threshold] の [Name] に、アラームの名前を入力します。
    - [Whenever: Reputation.BounceRate] または [Whenever: Reputation.ComplaintRate] で、アラームをトリガーするしきい値を指定します。

#### Note

バウンス率が 10% を超えるか苦情率が 0.5% を超えると、アカウントは自動的に確認中になります。CloudWatch アラームを発生させるバウンス率または苦情率を指定する場合は、これらの割合より低い値を使用して、お客様のアカウントが確認中とならないようにすることをお勧めします。

- [アクション] の [アラームが次の時:] で、[状態: 警告] を選択します。[Send notification to] に [the section called “パート 4: Amazon SNS トピックを作成する” \(p. 398\)](#) で作成した Amazon SNS トピックを選択します。

[Create Alarm] を選択します。

## パート 6: ソリューションをテストする

ALARM 状態に入ったときに Lambda 関数を実行するようにアラームをテストできるようになりました。SetAlarmState API オペレーションを使用して、アラームの状態を一時的に変更することができません。

このセクションの手順はオプションですが、ソリューション全体が正しく設定されていることを確認するために、これらの手順を完了することをお勧めします。

1. コマンドラインで次のコマンドを入力して、アカウントの E メール送信ステータスを確認します。

```
aws ses get-account-sending-enabled --region us-west-2
```

#### Note

前述のコマンドの **us-west-2** を前のステップで指定したリージョンの名前に置き換えます。

送信がアカウントに対して有効になっている場合、次の出力が表示されます。

```
{
  "Enabled": true
}
```

2. コマンドラインで次のコマンドを入力して、アラーム状態を一時的に ALARM に変更します。

```
aws cloudwatch set-alarm-state --alarm-name MyAlarm --state-value ALARM --state-reason "Testing execution of Lambda function" --region us-west-2
```

前述のコマンドで、**MyAlarm** を、「[the section called “パート 5: CloudWatch アラームの作成” \(p. 398\)](#)」で作成したアラームの名前に置き換え、**us-west-2** を、E メール送信を自動的に一時停止するリージョンに置き換えます。

#### Note

このコマンドを実行すると、アラームの状態は OK から ALARM に切り替わり、数秒で OK に戻ります。これらのステータスの変更は、CloudWatch コンソールのアラームの [History] タブで [DescribeAlarmHistory](#) オペレーションを使用して表示できます。

3. コマンドラインで次のコマンドを入力して、アカウントの E メール送信ステータスを確認します。`aws ses get-account-sending-enabled --region us-west-2`

Lambda 関数が正常に実行された場合、次の出力が表示されます。

```
{
  "Enabled": false
}
```

4. [the section called “パート 3: アカウントの E メール送信の再有効化” \(p. 397\)](#) の手順を実行して、アカウントの E メール送信を再度有効にします。

## 設定セットの E メール送信を自動的に一時停止する

Amazon SES に設定された特定の設定を使用して送信された E メールに固有の評価メトリクスをエクスポートするように Amazon CloudWatch を設定できます。これらのメトリクスを使用して、これらの設定セットに固有の CloudWatch アラームを作成できます。これらのアラームが特定のしきい値を超えると、Amazon SES アカウントの E メール送信機能全体に影響を与えずに、指定した設定セットを使用する Eメールの送信を自動的に一時停止することができます。

#### Note

このセクションで説明するソリューションは、単一の AWS リージョン内の特定の設定セットに対する E メール送信を一時停止します。複数のリージョンから E メールを送信する場合は、このソリューションを実装するリージョンごとにこのセクションの手順を繰り返します。

このセクションのトピック:

- [パート 1: 設定セットの評価メトリクスレポートを有効にする \(p. 400\)](#)
- [パート 2: IAM ロールの作成 \(p. 400\)](#)
- [パート 3: Lambda 関数を作成する \(p. 401\)](#)
- [パート 4: 設定セットの E メール送信の再有効化 \(p. 402\)](#)
- [パート 5: Amazon SNS トピックを作成する \(p. 403\)](#)
- [パート 6: CloudWatch アラームの作成 \(p. 403\)](#)
- [パート 7: ソリューションをテストする \(p. 404\)](#)

### パート 1: 設定セットの評価メトリクスレポートを有効にする

設定セットの E メール送信を自動的に一時停止するように Amazon SES を設定する前に、まず設定セットの評価メトリクスのエクスポートを有効にする必要があります。

設定セットのバウンスと苦情メトリクスのエクスポートを有効にするには、[the section called “評価メトリクスのエクスポート” \(p. 265\)](#) の手順を完了します。

### パート 2: IAM ロールの作成

E メール送信の自動一時停止を設定するための最初のステップは、UpdateConfigurationSetSendingEnabled API オペレーションを実行できる IAM ロールを作成することです。



IAM ロールを作成するには

1. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
2. ナビゲーションペインで [Roles (ロール)] を選択します。
3. [ロールの作成] を選択します。
4. [信頼されたエンティティの種類を選択] で、[AWS のサービス] を選択します。
5. [このロールを使用するサービスを選択] の下で、[Lambda] を選択します。[Next: Permissions (次へ: アクセス許可)] を選択します。
6. [Attach permissions policies] ページで、次のいずれかのポリシーを選択します。
  - AWS Lambda[BasicExecutionRole]
  - AmazonSESFullAccess

#### Tip

ポリシーのリストの上部にある検索ボックスを使用すると、これらのポリシーをすばやく特定できます。

[Next: Review] を選択します。

7. [Review] ページで、[Name] にロールの名前を入力します。[ロールの作成] を選択します。

## パート 3: Lambda 関数を作成する

IAM ロールを作成した後で、Lambda 関数を作成して、設定セットの E メール送信を一時停止することができます。

Lambda 関数を作成するには

1. AWS Lambda コンソール (<https://console.aws.amazon.com/lambda/>) を開きます。
2. リージョンセレクターを使用して、この Lambda 関数をデプロイするリージョンを選択します。

#### Note

この関数では、このステップで選択した AWS リージョン内の設定セットの E メール送信を一時停止します。複数のリージョンから E メールを送信する場合は、E メール送信を自動的に一時停止する各リージョンごとにこのセクションの手順を繰り返します。

3. [Create function] を選択します。
4. [Create function] で [Author from scratch] を選択します。
5. [Author from scratch] で、次のステップを実行します。
  - [名前] に Lambda 関数の名前を入力します。
  - [Runtime (ランタイム)] で、[Node.js 6.10] を選択します。
  - [Role] で、[Choose an existing role] を選択します。
  - [Existing role] には、[the section called “パート 2: IAM ロールの作成” \(p. 400\)](#) で作成した IAM ロールを選択します。

[Create function] を選択します。

6. コードエディタの [Function code] に、以下のコードを貼り付けます。

```
'use strict';
```

```
var aws = require('aws-sdk');

// Create a new SES object.
var ses = new aws.SES();

// Specify the parameters for this operation. In this example, you pass the
// Enabled parameter, with a value of false (Enabled = false disables email
// sending, Enabled = true enables it). You also pass the ConfigurationSetName
// parameter, with a value equal to the name of the configuration set for
// which you want to pause email sending.
var params = {
  ConfigurationSetName: ConfigSet,
  Enabled: false
};

exports.handler = (event, context, callback) => {
  // Pause sending for a configuration set
  ses.updateConfigurationSetSendingEnabled(params, function(err, data) {
    if(err) {
      console.log(err.message);
    } else {
      console.log(data);
    }
  });
};
```

前述のコードの *ConfigSet* を設定セットの名前と置き換えます。[Save] を選択します。

- [Test] を選択します。[Configure test event] ウィンドウが表示されたら、[Event name] フィールドに名前を入力してから、[Create] を選択します。
- ページ上部の通知バーに Execution result: succeeded と表示されていることを確認します。関数の実行に失敗した場合は、次のことを実行します。
  - the section called “[パート 2: IAM ロールの作成](#)” (p. 400) で作成した IAM ロールに正しいポリシーが含まれていることを確認します。
  - Lambda 関数のコードにエラーがないことを確認します。Lambda コードエディタでは、構文エラーやその他の潜在的な問題が自動的にハイライトされます。

## パート 4: 設定セットの E メール送信の再有効化

the section called “[パート 3: Lambda 関数を作成する](#)” (p. 401) で Lambda 関数のアップロードとテストをする副作用は、設定セットの E メール送信が一時停止していることです。ほとんどの場合は、CloudWatch アラームがトリガーされるまで、設定セットの送信を一時停止しません。

このセクションの手順では、E メール送信設定を再度有効にします。これらの手順を完了するには、AWS Command Line Interface をインストールして設定する必要があります。詳細については、[AWS Command Line Interface ユーザーガイド](#)を参照してください。

E メール送信を再有効化するには

- コマンドラインで次のコマンドを入力して、設定セットの E メール送信を再度有効にします。

```
aws ses update-configuration-set-sending-enabled \
--configuration-set-name ConfigSet \
--enabled
```

前述のコマンドでは、*ConfigSet* を、E メール送信を一時停止したい設定セットの名前に置き換え、*us-west-2* を、E メール送信を自動的に一時停止するリージョンに置き換えます。

2. コマンドラインで次のコマンドを入力して、Eメールの送信が有効であることを確認します。

```
aws ses describe-configuration-set \  
--configuration-set-name ConfigSet \  
--configuration-set-attribute-names reputationOptions
```

このコマンドは、次の例と同様の出力を生成します。

```
{  
  "ConfigurationSet": {  
    "Name": "ConfigSet"  
  },  
  "ReputationOptions": {  
    "ReputationMetricsEnabled": true,  
    "SendingEnabled": true  
  }  
}
```

SendingEnabled の値が true の場合、設定セットの Eメール送信は正常に再有効化されています。

## パート 5: Amazon SNS トピックを作成する

アラームが発生したときに CloudWatch が Lambda 関数を実行するには、まず Amazon SNS トピックを作成し、Lambda 関数をサブスクライブする必要があります。

Amazon SNS トピックを作成するには

1. Amazon SNS コンソール (<https://console.aws.amazon.com/sns/v3/home>) を開きます。
2. リージョンセレクターを使用して、Eメール送信を自動的に一時停止するリージョンを選択します。
3. ナビゲーションペインで、[トピック] を選択します。
4. [Create new topic] を選択します。
5. [Create new topic] ウィンドウで、[Topic name] にトピックの名前を入力します。必要に応じて、[Display name] フィールドにわかりやすい名前を入力することもできます。

[トピックの作成] を選択します。

6. トピックのリストで、前のステップで作成したトピックの横にあるチェックボックスをオンにします。[Actions] メニューで、[Subscribe to topic] を選択します。
7. [Create subscription] ウィンドウで、以下の選択を行います。
  - [プロトコル] で [AWS Lambda] を選択します。
  - [Endpoint] に、[the section called “パート 3: Lambda 関数を作成する” \(p. 401\)](#) で作成した Lambda 関数を選択します。
  - [Version or alias] に [default] を選択します。
8. [Create subscription] を選択します。

## パート 6: CloudWatch アラームの作成

このセクションでは、メトリクスが特定のしきい値に到達するとトリガーされるアラームを CloudWatch に作成する手順を紹介します。アラームが発生すると、[the section called “パート 5: Amazon SNS トピックを作成する” \(p. 403\)](#) で作成した Amazon SNS トピックに通知が送信され、[the section called “パート 3: Lambda 関数を作成する” \(p. 401\)](#) で作成した Lambda 関数が実行されます。

## CloudWatch アラームの作成

1. CloudWatch コンソール (<https://console.aws.amazon.com/cloudwatch/>) を開きます。
2. リージョンセレクターを使用して、E メール送信を自動的に一時停止するリージョンを選択します。
3. 左側のナビゲーションペインで、[Alarms] を選択します。
4. [Create Alarm] を選択します。
5. [Create Alarm] ウィンドウの、[SES Metrics] の下で、[Configuration Set Metrics] を選択します。
6. [ses:configuration-set] 列で、アラームを作成する設定セットを検索します。[Metric Name] で、次のいずれかのオプションを選択します。

- Reputation.BounceRate – 設定セットの全体のハードバウンス率が定義したしきい値を超えた場合に、設定セットの E メール送信を一時停止する場合は、このメトリクスを選択します。
- Reputation.ComplaintRate – 設定セットの全体の苦情率が定義したしきい値を超えた場合に、設定セットの E メール送信を一時停止する場合は、このメトリクスを選択します。

[Next] を選択します。

7. 以下の手順を実行します。
  - [Alarm Threshold] の [Name] に、アラームの名前を入力します。
  - [Whenever: Reputation.BounceRate] または [Whenever: Reputation.ComplaintRate] で、アラームをトリガーするしきい値を指定します。

### Note

Amazon SES アカウントの全体的なバウンス率が 10% を超える場合、または Amazon SES アカウントの全体的な苦情率が 0.5% を超える場合、Amazon SES アカウントは自動的に確認中となります。CloudWatch アラームを発生させるバウンス率または苦情率を指定する場合は、これらの割合よりはるかに低い値を使用して、お客様のアカウントが確認中にならないようにすることをお勧めします。

- [アクション] の [アラームが次の時:] で、[状態: 警告] を選択します。[Send notification to] に [the section called “パート 5: Amazon SNS トピックを作成する” \(p. 403\)](#) で作成した Amazon SNS トピックを選択します。

[Create Alarm] を選択します。

## パート 7: ソリューションをテストする

ALARM 状態に入ったときに Lambda 関数を実行するようにアラームをテストできるようになりました。CloudWatch API で `SetAlarmState` オペレーションを使用して、アラームの状態を一時的に変更することができます。

このセクションの手順はオプションですが、ソリューション全体が正しく設定されていることを確認するために、これらの手順を完了することをお勧めします。

ソリューションをテストするには

1. コマンドラインで次のコマンドを入力して、設定セットの E メール送信ステータスを確認します。`aws ses describe-configuration-set --configuration-set-name ConfigSet --region us-west-2`  
送信が設定セットに対して有効になっている場合、次の出力が表示されます。

```
{
  "ConfigurationSet": {
    "Name": "ConfigSet"
  }
}
```

```
    },  
    "ReputationOptions": {  
      "ReputationMetricsEnabled": true,  
      "SendingEnabled": true  
    }  
  }  
}
```

`SendingEnabled` の値が `true` の場合、現在設定セットの E メール送信は有効になっています。

2. コマンドラインで次のコマンドを入力して、アラーム状態を一時的に `ALARM` に変更します。

```
aws cloudwatch set-alarm-state \  
--alarm-name MyAlarm \  
--state-value ALARM \  
--state-reason "Testing execution of Lambda function"
```

前述のコマンドの *MyAlarm* を [the section called “パート 6: CloudWatch アラームの作成” \(p. 403\)](#) で作成したアラームの名前に置き換えます。

#### Note

このコマンドを実行すると、アラームの状態は `OK` から `ALARM` に切り替わり、数秒で `OK` に戻ります。これらのステータスの変更は、CloudWatch コンソールのアラームの [History] タブで [DescribeAlarmHistory](#) オペレーションを使用して表示できます。

3. コマンドラインで次のコマンドを入力して、設定セットの Eメールの送信ステータスを確認します。

```
aws ses describe-configuration-set \  
--configuration-set-name ConfigSet
```

Lambda 関数が正常に実行された場合、次の例のような出力が表示されます。

```
{  
  "ConfigurationSet": {  
    "Name": "ConfigSet"  
  },  
  "ReputationOptions": {  
    "ReputationMetricsEnabled": true,  
    "SendingEnabled": false  
  }  
}
```

`SendingEnabled` の値が `false` の場合、設定セットの E メール送信は無効になり、Lambda 関数が正常に実行されたことを示します。

4. [the section called “パート 4: 設定セットの Eメール送信の再有効化” \(p. 402\)](#) の手順を実行して、設定セットの Eメール送信を再度有効にします。

# Amazon SES へのアクセスのコントロール

AWS Identity and Access Management (IAM) と Amazon Simple Email Service (Amazon SES) を使用して、IAM ユーザー、グループ、またはロールで実行できる Amazon SES API アクションを指定できます。(このトピックでは、これらのエンティティをまとめて「ユーザー」と呼びます)。ユーザーが「From」、受取人、「Return-Path」の E メールアドレスに使用できる E メールアドレスをコントロールすることもできます。

例えば、IAM ポリシーを作成して、組織内のユーザーはメールを送信できるが送信統計の確認アクションなどのチェック作業を実行できないようにすることができます。別の例として、ユーザーが特定の「From」アドレスを使用している場合のみ、アカウントから Amazon SES を介して E メールを送信することを許可するポリシーを作成できます。

IAM を使用するには、IAM ポリシー (明示的にアクセス許可を定義するドキュメント) を定義し、ユーザーにポリシーを添付します。IAM ポリシーを作成する方法については、「IAM ユーザーガイド」を参照してください。ポリシーで設定した制限が適用される以外に、ユーザーが Amazon SES を操作する方法や Amazon SES がリクエストを実行する方法に変更はありません。

## Note

送信認可ポリシーを使用することで、Amazon SES へのアクセスを制御することもできます。IAM ポリシーは、個々の IAM ユーザーが実行できる操作を制限しますが、送信認可ポリシーは個々の検証済みアイデンティティの使用方法を制限します。さらに、クロスアカウントアクセスを許可できるのは送信承認ポリシーのみです。送信承認の詳細については、「[Amazon SES での送信承認の使用 \(p. 154\)](#)」を参照してください。

既存の IAM ユーザーが使用する Amazon SES SMTP の認証情報を生成する方法については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

## Amazon SES にアクセスするための IAM ポリシーの作成

このセクションでは、特に Amazon SES で IAM ポリシーを使用する方法を説明します。IAM ポリシーを作成する一般的な方法については、「IAM ユーザーガイド」を参照してください。

Amazon SES で IAM を使用する理由は次の 3 つです。

- E メール送信アクションを制限するため。
- ユーザーが送信する Eメールの「From」、受取人、「Return-Path」のアドレスを制限するため。
- ユーザーが使用を承認された API の呼び出しを許可される期間など、API の使用の一般的な側面をコントロールするため。

## アクションの制限

ユーザーが実行できる Amazon SES アクションをコントロールするには、IAM ポリシーの Action 要素を使用します。API 名の先頭に小文字の文字列 Action を付けることで、ses:要素をすべての Amazon SES API アクションに設定できます。たとえば、Action を ses:SendEmail、ses:GetSendStatistics、または ses:\* (すべてのアクションの場合) に設定できます。

次に、Action に応じて、次のように Resource 要素を指定します。

**Action** 要素により E メール送信 API (つまり、`ses:SendEmail` や `ses:SendRawEmail`) へのアクセスのみ許可されている場合、次のようにします。

- ユーザーが AWS アカウントの任意の ID から送信できるようにするには、Resource を \* に設定します。
- ユーザーの送信元のアイデンティティを制限するには、Resource をユーザーに使用を許可するアイデンティティの ARN に設定します。

**Action** 要素によりすべての API へのアクセスが許可されている場合、次のようにします。

- ユーザーの送信元のアイデンティティを制限しない場合、Resource を \* に設定します。
- ユーザーの送信元のアイデンティティを制限する場合、2 つのポリシー (または、1 つのポリシー内に 2 つのステートメント) を作成する必要があります。
  - Action が、許可される E メール送信以外の API の明示的なリストに設定され、Resource が \* に設定されているポリシー
  - Action が E メール送信 API (`ses:SendEmail` や `ses:SendRawEmail`) のいずれかに設定され、Resource がユーザーに使用を許可するアイデンティティの ARN に設定されたポリシー

使用可能な Amazon SES アクションのリストについては、[Amazon Simple Email Service API リファレンス](#)および[Amazon Simple Email Service API v2 リファレンス](#)を参照してください。IAM ユーザーが SMTP インターフェイスを使用する場合、許可する `ses:SendRawEmail` へのアクセスを最小限にする必要があります。

## E メールアドレスの制限

ユーザーを特定の E メールアドレスに制限する場合、Condition ブロックを使用できます。Condition ブロックでは、[IAM ユーザーガイド](#) で説明されている条件キーを使用して条件を指定します。条件キーを使用して、次の E メールアドレスをコントロールできます。

### Note

これらの E メールアドレス条件キーは、以下の表に記載された API にも適用されます。

条件キー	説明	API
<code>ses:Recipients</code>	受取人アドレスを制限します。To:、"CC"、"BCC" アドレスが含まれます。	<code>SendEmail</code> 、 <code>SendRawEmail</code>
<code>ses:FromAddress</code>	「From」アドレスを制限します。	<code>SendEmail</code> 、 <code>SendRawEmail</code> 、 <code>SendBounce</code>
<code>ses:FromDisplayName</code>	表示名として使用される「From」アドレスを制限します。	<code>SendEmail</code> 、 <code>SendRawEmail</code>
<code>ses:FeedbackAddress</code>	"Return-Path" アドレス (Eメールのフィードバック転送によりバウンスや苦情を送信できるアドレス) を制限します。Eメールのフィードバック転送の詳細については、「 <a href="#">Eメールで送信された Amazon SES 通知 (p. 274)</a> 」を参照してください。	<code>SendEmail</code> 、 <code>SendRawEmail</code>

## 一般的な API の使用の制限

条件で AWS 全体のキーを使用することで、ユーザーが API にアクセスを許可される日時のような側面に基づいて Amazon SES へのアクセスを制限できます。Amazon SES は、以下の AWS 全体のポリシーキーのみを実装しています。

- `aws:CurrentTime`
- `aws:EpochTime`
- `aws:SecureTransport`
- `aws:SourceIp`
- `aws:UserAgent`

これらのキーについては、「[IAM ユーザーガイド](#)」を参照してください。

## Amazon SES の IAM ポリシーの例

このトピックでは、特定の条件の下でのみ Amazon SES へのユーザーアクセスを許可するポリシーの例を示します。

このセクションのポリシーの例。

- [すべての Amazon SES アクションへのフルアクセスを許可 \(p. 408\)](#)
- [E メール送信アクションへのアクセスのみを許可 \(p. 408\)](#)
- [送信期間の制限 \(p. 409\)](#)
- [受取人アドレスの制限 \(p. 409\)](#)
- ["From" アドレスの制限 \(p. 410\)](#)
- [E メール送信者の表示名の制限 \(p. 410\)](#)
- [バウンスや苦情のフィードバックの制限と送信先 \(p. 411\)](#)

## すべての Amazon SES アクションへのフルアクセスを許可

次のポリシーでは、任意の Amazon SES アクションの呼び出しをユーザーに許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ses:*"
      ],
      "Resource": "*"
    }
  ]
}
```

## E メール送信アクションへのアクセスのみを許可

次のポリシーでは、Amazon SES を使用した E メールの送信をユーザーに許可しますが、Amazon SES 送信統計へのアクセスなどの管理作業を実行することは許可しません。



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource": "*"
    }
  ]
}
```

## 送信期間の制限

次のポリシーでは、2018 年の 9 月中に限り、E メール送信 Amazon SES API の呼び出しをユーザーに許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource": "*",
      "Condition": {
        "DateGreaterThan": {
          "aws:CurrentTime": "2018-08-31T12:00Z"
        },
        "DateLessThan": {
          "aws:CurrentTime": "2018-10-01T12:00Z"
        }
      }
    }
  ]
}
```

## 受取人アドレスの制限

次のポリシーでは、example.com ドメインの受信者アドレスに対してのみ、Amazon SES E メール送信 API の呼び出しをユーザーに許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringLike": {
          "ses:Recipients": [
            "*@example.com"
          ]
        }
      }
    }
  ]
}
```

```
    ]  
  }  
}  
]  
}
```

## "From" アドレスの制限

次のポリシーでは、marketing@example.com の受信者アドレスに対してのみ、Amazon SES E メール送信 API の呼び出しをユーザーに許可します。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ses:SendEmail",  
        "ses:SendRawEmail"  
      ],  
      "Resource": "*",  
      "Condition": {  
        "StringEquals": {  
          "ses:FromAddress": "marketing@example.com"  
        }  
      }  
    }  
  ]  
}
```

次のポリシーでは、bounce@example.com の受信者アドレスに対してのみ、SendBounce API の呼び出しをユーザーに許可します。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ses:SendBounce"  
      ],  
      "Resource": "*",  
      "Condition": {  
        "StringEquals": {  
          "ses:FromAddress": "bounce@example.com"  
        }  
      }  
    }  
  ]  
}
```

## E メール送信者の表示名の制限

次のポリシーでは、受信者アドレスの表示名が「Marketing」を含む場合に対してのみ、Amazon SES E メール送信 API の呼び出しをユーザーに許可します。

```
{  
  "Version": "2012-10-17",
```

```
"Statement":[
  {
    "Effect":"Allow",
    "Action":[
      "ses:SendEmail",
      "ses:SendRawEmail"
    ],
    "Resource": "*",
    "Condition":{
      "StringLike":{
        "ses:FromDisplayName":"Marketing"
      }
    }
  }
]
```

## バウンスや苦情のフィードバックの制限と送信先

次のポリシーでは、「Return-Path」が feedback@example.com の受信者アドレスに対してのみ、Amazon SES E メール送信 API の呼び出しをユーザーに許可します。

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource": "*",
      "Condition":{
        "StringEquals":{
          "ses:FeedbackAddress":"feedback@example.com"
        }
      }
    }
  ]
}
```

# AWS CloudTrail を使用した Amazon SES API コールのログ作成

Amazon SES は、Amazon SES のユーザー、ロール、または AWS のサービスによって実行されたアクションを記録するサービスである AWS CloudTrail と統合されています。CloudTrail は Amazon SES の API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、Amazon SES コンソールからの呼び出しと、Amazon SES API オペレーションへのコード呼び出しが含まれます。追跡を作成する場合は、Amazon SES のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [Event history (イベント履歴)] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、Amazon SES に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

設定や有効化の方法など、CloudTrail の詳細については、「[AWS CloudTrail ユーザーガイド](#)」を参照してください。

## CloudTrail 内の Amazon SES 情報

CloudTrail は、アカウント作成時に AWS アカウントで有効になります。Amazon SES でサポートされているイベントアクティビティが発生すると、そのアクティビティは [イベント履歴] の他の AWS のサービスのイベントとともに CloudTrail イベントに記録されます。最近のイベントは、AWS アカウントで表示、検索、ダウンロードできます。詳細については、[CloudTrail イベント履歴でのイベントの表示](#)を参照してください。

Amazon SES のイベントなど、AWS アカウントのイベントの継続的な記録については、追跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで作成した証跡がすべての AWS リージョンに適用されます。証跡では、AWS パーティションのすべてのリージョンからのイベントがログに記録され、指定した Amazon S3 バケットにログファイルが配信されます。さらに、その他の AWS サービスを設定して、CloudTrail ログで収集されたデータをより詳細に分析し、それに基づく対応を行うことができます。詳細については、以下を参照してください。

- [証跡を作成するための概要](#)
- [CloudTrail のサポート対象サービスと統合](#)
- [Amazon SNS の CloudTrail 通知の設定](#)
- [複数のリージョンから CloudTrail ログファイルを受け取る、および複数のアカウントから CloudTrail ログファイルを受け取る](#)

Amazon SES は、CloudTrail ログファイルのイベントとして以下のアクションのログ付けをサポートしません。

- [CloneReceiptRuleSet](#)
- [CreateReceiptFilter](#)
- [CreateReceiptRule](#)
- [CreateReceiptRuleSet](#)
- [DeleteIdentity](#)
- [DeleteIdentityPolicy](#)
- [DeleteReceiptFilter](#)

- [DeleteReceiptRule](#)
- [DeleteReceiptRuleSet](#)
- [DeleteVerifiedEmailAddress](#)
- [DescribeActiveReceiptRuleSet](#)
- [DescribeReceiptRule](#)
- [DescribeReceiptRuleSet](#)
- [GetIdentityDkimAttributes](#)
- [GetIdentityNotificationAttributes](#)
- [GetIdentityPolicies](#)
- [GetIdentityVerificationAttributes](#)
- [GetSendQuota](#)
- [GetSendStatistics](#)
- [ListIdentities](#)
- [ListIdentityPolicies](#)
- [ListReceiptFilters](#)
- [ListReceiptRuleSets](#)
- [ListVerifiedEmailAddresses](#)
- [PutIdentityPolicy](#)
- [ReorderReceiptRuleSet](#)
- [SetActiveReceiptRuleSet](#)
- [SetReceiptRulePosition](#)
- [SetIdentityDkimEnabled](#)
- [SetIdentityFeedbackForwardingEnabled](#)
- [SetIdentityHeadersInNotificationsEnabled](#)
- [SetIdentityNotificationTopic](#)
- [UpdateReceiptRule](#)
- [VerifyDomainDkim](#)
- [VerifyDomainIdentity](#)
- [VerifyEmailAddress](#)
- [VerifyEmailIdentity](#)

#### Note

Amazon SES は、管理イベントを CloudTrail に提供します。管理イベントには、AWS アカウント内のリソースの作成と管理に関連するアクションが含まれます。Amazon SES で、管理イベントには、ID や受信ルールの作成および削除などのアクションが含まれます。管理イベントは、データイベントとは異なります。データイベントは、AWS アカウント内のデータにアクセスしてやり取りすることに関連するイベントです。Amazon SES で、データイベントには、Eメールの送信などのアクションが含まれます。Amazon SES は管理イベントを CloudTrail にのみ配信するため、次のイベントは CloudTrail に記録されません。

- [SendEmail](#)
- [SendRawEmail](#)
- [SendTemplatedEmail](#)
- [SendBulkTemplatedEmail](#)
- [SendCustomVerificationEmail](#)

E メール送信に関連するイベントを記録するイベント発行を使用できます。詳細については、「[Amazon SES イベント発行を使用して E メール送信をモニタリングする \(p. 296\)](#)」を参照してください。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。この ID 情報は以下のことを確認するのに役立ちます。

- リクエストが、root 認証情報と AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか。
- リクエストが、ロールとフェデレーテッドユーザーのどちらの一時的なセキュリティ認証情報を使用して送信されたか。
- リクエストが別の AWS のサービスによって生成されたかどうか。

詳細については、「[CloudTrail userIdentity 要素](#)」を参照してください。

## 例: Amazon SES ログファイルのエントリ

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail ログファイルには、1 つ以上のログエントリがあります。イベントは任意の発生元からの 1 つのリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次は、DeleteIdentity と VerifyEmailIdentity のアクションを示す CloudTrail ログエントリの例です。

```
{
  "Records": [
    {
      "awsRegion": "us-west-2",
      "eventID": "0ffa308d-1467-4259-8be3-c749753be325",
      "eventName": "DeleteIdentity",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2018-02-02T21:34:50Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "50b87bfe-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": {
        "identity": "amazon.com"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-sdk-java/unknown-version",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "111122223333",
        "arn": "arn:aws:iam::111122223333:root",
        "principalId": "111122223333",
        "type": "Root"
      }
    },
    {
      "awsRegion": "us-west-2",
      "eventID": "5613b0ff-d6c6-4526-9b53-a603a9231725",
      "eventName": "VerifyEmailIdentity",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2018-02-04T01:05:33Z",
      "eventType": "AwsApiCall",
```

```
"eventVersion":"1.02",
"recipientAccountId":"111122223333",
"requestID":"eb2ff803-ac09-11e4-8ff5-a56a3119e253",
"requestParameters":{"
  "emailAddress":"sender@example.com"
},
"responseElements":null,
"sourceIPAddress":"192.0.2.0",
"userAgent":"aws-sdk-java/unknown-version",
"userIdentity":{"
  "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
  "accountId":"111122223333",
  "arn":"arn:aws:iam::111122223333:root",
  "principalId":"111122223333",
  "type":"Root"
}
}
]
}
```

# Amazon SES API の使用

Amazon SES API の低レベル機能を、詳細を自動的に処理する高レベルのデータ型および関数の呼び出しでラップする [AWS SDK](#) を使用して、Amazon SES API にアクセスできます。Amazon SES API に対して HTTPS リクエストを行うこともできます。個々の API オペレーションに関する詳細については、[Amazon Simple Email Service API リファレンス](#) を参照してください。

このセクションは、以下のトピックで構成されます。

- [Amazon SES API リクエスト \(p. 416\)](#)
- [Amazon SES API へのリクエストの認証 \(p. 418\)](#)
- [Amazon SES API の GET と POST の例 \(p. 420\)](#)
- [Amazon SES API レスポンス \(p. 421\)](#)
- [Amazon SES API から返されるエラーコード \(p. 422\)](#)

## Amazon SES API リクエスト

Amazon SES API エンドポイントにリクエストを送信することで、Amazon SES API を直接呼び出すことができます。Amazon SES API へのリクエストは、GET や POST メソッドを使用するシンプルな HTTPS リクエストです。API リクエストには実行されるアクションを表す `Action` パラメータが必ず含まれます。

### Important

Amazon SES は、HTTP リクエストをサポートしていません。代わりに HTTPS を使用する必要があります。

## GET リクエストの構造

このガイドでは、Amazon SES GET リクエストを URL として示しています。各 URL には次が含まれます。

- エンドポイント リクエストが影響を与えるリソース。Amazon SES を使用できる AWS リージョンのエンドポイント URL のリストについては、AWS 全般のリファレンスにある [Amazon Simple Email Service \(Amazon SES\)](#) を参照してください。
- アクション メッセージの送信など、エンドポイントで実行したいアクション。
- パラメータ 任意のリクエストパラメータ。

以下に、(米国西部 (オレゴン) リージョン) の Amazon SES エンドポイントを使用してメッセージを送信する GET リクエストの例を示します。

```
https://email.us-west-2.amazonaws.com?Action=SendEmail&Source=user%40example.com&Destination.ToAddresses.member.1=allan%40example.com&Message.Subject.Data=This%20is%20the%20subject%20line.&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20having%20a%20good%20day.
```



## Important

GET リクエストは URL であるため、パラメータ値を URL エンコードする必要があります。たとえば、前の例のリクエストで、`Source` パラメータの値は実際には `user@example.com` です。しかし、URL では "@" 文字が許可されていないので、各 "@" を URL エンコードして "%40" とします。

このガイドでは、次の構文解析形式で GET の例を表すことで読みやすくしています。

```
https://email.us-west-2.amazonaws.com
?Action=SendEmail
&Source=user%40example.com
&Destination.ToAddresses.member.1=allan%40example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20having%20a%20good%20day.
```

最初の行は、リクエストのエンドポイントを表しています。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。各パラメータは、アンパーサンド (&) で区切られています。

Action パラメーターは実行するアクションを示します。アクションと各アクションで使用されるパラメーターの完全なリストについては、[Amazon Simple Email Service API リファレンス](#)をご覧ください。

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。たとえば、複数の受信者にメールを送信する場合は、E メールアドレスのリストを指定できます。このようなリストは、`param.n` 表記法で指定します。ここで、`n` の値は 1 で始まる整数です。たとえば、最初の「To:」アドレスは `Destination.ToAddresses.1` と指定し、2 つ目は `Destination.ToAddresses.2` というように指定します。

Amazon SES では、いずれのパラメータ値でもスペースを使用できません。このガイドでは、スペースが含まれているサンプルのリクエストパラメータ値を、次の 2 つの方法のいずれかで表記しています。

- URL エンコード (たとえば %20)。
- プラス記号 (「+」) で表記。リクエスト内では、プラス記号はスペースの簡易表記として予約されています。パラメータに、変換されないプラス記号自体を含めるには、%2B のように URL エンコードする必要があります。

## Note

リクエストは、`x-Amzn-Authorization` HTTP ヘッダーとともに使用しなければなりません。詳細については、[Amazon SES API へのリクエストの認証 \(p. 418\)](#) を参照してください。

## POST リクエストの構造

Amazon SES は、POST リクエストも受け入れます。POST リクエストでは、以下の手順で説明しているように、HTTP リクエストボディ内のフォームとしてパラメータを送信します。

POST リクエストを作成するには

1. パラメータ名と値をフォームとして組み立てます。

GET リクエストの場合と同様に、パラメータと値を組み合わせます (名前と値の各ペアは、アンパーサンドで区切ります)。次の例は、`SendEmail` リクエストを示しています。このガイドでは、情報を読みやすくするために改行を使用しています。

```
Action=SendEmail
```

```
&Source=user@example.com
&Destination.ToAddresses.member.1=allan@example.com
&Message.Subject.Data=This is the subject line.
&Message.Body.Text.Data=Hello. I hope you are having a good day.
```

- HTML 仕様の Form Submission セクションに従って、フォームを Form-URL エンコードします。

詳細については、[http://www.w3.org/MarkUp/html-spec/html-spec\\_toc.html#SEC8.2.1](http://www.w3.org/MarkUp/html-spec/html-spec_toc.html#SEC8.2.1) を参照してください。

```
Action=SendEmail
&Source=user%40example.com
&Destination.ToAddresses.member.1=allan%40example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20having%20a%20good%20day.
```

- POST リクエストの本文として、結果フォームを用意します。
- リクエストに以下の HTTP ヘッダーを含めます。
  - Content-Type、(値は application/x-www-form-urlencoded に設定)
  - Content-Length
  - Date
  - Authorization (詳しくは、[Amazon SES API へのリクエストの認証 \(p. 418\)](#) を参照してください)。
- 完成したリクエストを送信します。

```
POST / HTTP/1.1
Date: Thu, 26 May 2011 06:49:50 GMT
Host: email.us-west-2.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE,SignedHeaders=Date;x-amz-date,Signature=9d63c3b5b7623d1fa3dc7fd1547313b9546c6d0fbbb6773a420613b7EXAMPLE
Content-Length: 230

Action=SendEmail
&Source=user%40example.com
&Destination.ToAddresses.member.1=allan%40example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20having%20a%20good%20day.
```

#### Note

HTTP クライアントは、通常、クライアントが使用している HTTP のバージョンでの要件に従って、HTTP リクエストに他の項目を追加します。このガイドの例には、これらの追加の項目は含まれていません。

## Amazon SES API へのリクエストの認証

Amazon SES API にアクセスするときは、AWS 署名を使用してリクエストを認証します。リクエストに有効な署名が含まれていない場合、Amazon SES はエラーを返し、リクエストを処理しません。

AWS リクエストの署名の詳細については、AWS 全般のリファレンスの「[AWS リクエストの署名](#)」を参照してください。

## Important

2020 年 10 月 1 日以降、Amazon SES は署名バージョン 4 を使用して署名されたリクエストのみをサポートします。古いバージョンの AWS 署名を使用する場合は、その日より前に AWS 署名バージョン 4 を導入する必要があります。

## 認証エラー

E メールを送信しようとしたときに、リクエストで以前のバージョンの AWS 署名が使用されている場合、次の例のようなエラーメッセージが表示されます。

```
<ErrorResponse xmlns="http://ses.amazonaws.com/doc/2010-12-01/">
  <Error>
    <Type>Sender</Type>
    <Code>InvalidClientTokenId</Code>
    <Message>The security token included in the request is invalid</Message>
  </Error>
</ErrorResponse>
```

このエラーは、古いバージョンの AWS SDK または AWS CLI を使用するときに表示される可能性があります。このエラーは、古いバージョンの AWS 署名を使用する Amazon SES API に対して直接 HTTPS リクエストを行う場合にも表示される可能性があります。

このエラーメッセージが表示された場合は、セットアップに関連する適切な手順に従って既存の SMTP 認証情報を置き換えて、AWS 署名バージョン 4 を使用するようにリクエストを更新する必要があります。

- SES コンソールを使用して SMTP 認証情報を生成した場合は、単に新しい認証情報を作成し、既存の認証情報を新しい認証情報に置き換えます。
- SMTP 認証情報を AWS 認証情報から取得した場合は、AWS 署名バージョン 4 アルゴリズムを使用していることを確認します。この変換がライブラリに依存している場合は、ライブラリに AWS 署名バージョン 4 アルゴリズムを使用する新しいリリースがあるかどうかをチェックし、ライブラリに移行します。それ以外の場合は、AWS 署名バージョン 4 アルゴリズムを使用する別のライブラリから認証情報を派生させるか、SES コンソールを使用して認証情報を生成する必要があります。

## AWS 署名バージョン 4 への移行

AWS SDK または AWS CLI を使用する場合は、SDK の最新バージョンまたは AWS CLI にアップデートする必要があります。Amazon SES API に直接の HTTPS リクエストを行う場合は、リクエストのヘッダーを更新して AWS 署名バージョン 4 を使用します。以前のバージョンの AWS 署名を使用する API リクエストは、リクエストヘッダーを調べることで簡単に識別できます。たとえば、AWS 署名バージョン 3 を使用するリクエストは、次の例のようになります。

```
X-Amzn-Authorization: AWS3-HTTPS
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Algorithm=HMACSHA256,Signature=lBP67vCvG1 ...
```

AWS 署名バージョン 4 を使用するリクエストには、次の情報を含む Authorization ヘッダーが含まれます。

- 署名に使用したアルゴリズム (AWS4-HMAC-SHA256)
- 認証情報スコープ (アクセスキー ID を含む)
- 署名付きヘッダーの一覧
- 計算された署名。この署名はリクエスト情報に基づいており、AWS シークレットアクセスキーを使用して署名を生成します。署名により、AWS の ID が確認されます。

AWS 署名バージョン 4 を使用する Amazon SES API への呼び出しの例は、「[Amazon SES API リクエスト \(p. 416\)](#)」を参照してください。AWS 署名バージョン 4 の使用については、『AWS 全般のリファレンス』の「[署名バージョン 4 の署名プロセス](#)」を参照してください。

## Amazon SES API の GET と POST の例

このセクションでは、Amazon SES API に対して発行できるリクエストの例を示します。

### GET リクエストの例

次のコードは、GET リクエストの例です。これには、計算された署名が含まれています。リクエスト内のすべてのパラメータが URL エンコードされていることに注意してください。

```
https://email.us-west-2.amazonaws.com/  
?Action=SendEmail  
&Source=user%40example.com  
&Destination.ToAddresses.member.1=allan%40example.com  
&Message.Subject.Data=This%20is%20the%20subject%20line.  
&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20having%20a%20good%20day.  
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE  
&Signature=RhU864jFu893mg7g9N9j9nr6h7EXAMPLE  
&Algorithm=HMACSHA256
```

### POST リクエストの例

次のコードは、POST リクエストの例です。これには、計算された署名が含まれています。リクエスト内のすべてのパラメータが URL エンコードされていることに注意してください。

```
POST / HTTP/1.1  
Host: email.us-west-2.amazonaws.com  
Content-Type: application/x-www-form-urlencoded  
Date: Tue, 25 May 2010 21:20:27 +0000  
Content-Length: 174  
  
Action=SendRawEmail  
&Destinations.member.1=allan%40example.com  
&RawMessage.Data=RnJvbTp1c2VyQG4YW1wbGUuY29tDQpTdWJqZWNOOiBUZXN0QoNCK1lc3 ...
```

RawMessage.Data の値は、次のテキストを base64 エンコードした表現です。

```
From:user@example.com  
Subject: Test  
  
Message sent using SendRawEmail.
```

以下は、SendRawEmail 演算子に対する完全な POST リクエストです。リクエストには X-Amzn-Authorization ヘッダーが含まれます。どのヘッダーも URL エンコードされていません。

```
POST / HTTP/1.1  
Host: email.us-west-2.amazonaws.com  
Content-Type: application/x-www-form-urlencoded  
Date: Tue, 25 May 2010 21:20:27 +0000  
Content-Length: 174  
Authorization: AWS4-HMAC-SHA256 Credential=AKIDEXAMPLE/20150830/us-east-1/iam/aws4_request,  
SignedHeaders=content-type;host;x-amz-date,Signature=5d672d79c1EXAMPLE
```

```
Action=SendRawEmail
&Destinations.member.1=allan%40example.com
&RawMessage.Data=RnJvbTp1c2VyQG4YV1wbGUuY29tDQpTdWJqZWN0OiBUZXN0DQoNCK1lc3 ...
```

## Amazon SES API レスポンス

API リクエストに対するレスポンスで、Amazon SES はリクエスト結果を含む XML データ構造を返します。

すべての Amazon SES レスポンスには、RequestId 要素のリクエスト ID が含まれています。値は AWS によって割り当てられる一意の文字列です。特定のリクエストに問題が発生すると、問題のトラブルシューティングを行うために AWS によってリクエスト ID が求められます。

また正常な Amazon SES レスポンスには、1 つ以上のメッセージ ID が含まれます。メッセージ ID は、Amazon SES が送信する E メールメッセージの受信報告と見なすことができます。メッセージが拒否されるか、バウンスメールになった場合、受け取るすべての苦情またはバウンス通知にメッセージ ID が表示されます。そのため、メッセージ ID を利用して、問題のある送信 E メールメッセージを特定し、修正作業を行えます。

### 正常なレスポンスの構造

リクエストが正常に行われた場合、主レスポンスの要素には、「Response」で終わる、アクションに関連する名前が付きます。たとえば、SendEmailResponse は、正常な SendEmail リクエストに対するレスポンス要素です。この要素には、以下の子要素が含まれています。

- 子要素 RequestId を含む ResponseMetadata。
- アクション固有の結果を含む任意の要素。たとえば、SendEmailResult という要素を含む SendEmailResponse 要素。

XML スキーマでは、各 Amazon SES アクション用の XML レスポンスメッセージについて記述します。

以下に、正常なレスポンスの例を示します。

```
<SendEmailResponse xmlns="https://email.amazonaws.com/doc/2010-03-31/">
  <SendEmailResult>
    <MessageId>000001271b15238a-fd3ae762-2563-11df-8cd4-6d4e828a9ae8-000000</MessageId>
  </SendEmailResult>
  <ResponseMetadata>
    <RequestId>fd3ae762-2563-11df-8cd4-6d4e828a9ae8</RequestId>
  </ResponseMetadata>
</SendEmailResponse>
```

### Amazon SES API エラーレスポンスの構造

リクエストが正常に行われなかった場合、主レスポンス要素は、呼び出されたアクションに関係なく、ErrorResponse と呼ばれます。この要素には Error 要素と RequestId 要素が含まれます。各 Error には、以下のものが含まれます：

- エラーが受取人エラーが送信者エラーのどちらかを特定する Type 要素
- 発生したエラータイプを識別する Code 要素
- 人間が読みやすい形式でエラー条件について記述する Message 要素
- 補足でエラーの詳細を提供する、または空である Detail 要素

以下に、エラーレスポンスの例を示します。

```
<ErrorResponse>
  <Error>
    <Type>
      Sender
    </Type>
    <Code>
      ValidationError
    </Code>
    <Message>
      Value null at 'message.subject' failed to satisfy constraint: Member must not be
      null
    </Message>
  </Error>
  <RequestId>
    42d59b56-7407-4c4a-be0f-4c88daeea257
  </RequestId>
</ErrorResponse>
```

## Amazon SES API から返されるエラーコード

このトピックには、Amazon SES API から返されるエラーコードのリストが含まれています。Amazon SES API の使用については、[Amazon Simple Email Service API リファレンス](#)を参照してください。

5xx エラーを受け取った HTTPS リクエストは、再試行する必要があります。この場合、重複が生成される可能性を下げるため、タイムアウトごとに待機時間が徐々に長くなる (5 秒、10 秒、30 秒) 段階的な再試行方法を実装することをお勧めします。3 回目の再試行呼び出しが成功しない場合、20 分後に一連の再試行をもう一度実行します。Amazon SES で段階的な再試行ポリシーを使用する実装例については、Amazon SES ブログで「[Throttling – Maximum sending rate exceeded \(スロットリング – 最大送信レートの超過\)](#)」エラーの対処法」を参照してください。

### Note

AWS SDK は、再試行ロジックを自動的に実装します。

HTTPS クライアントエラー (4xx) は、再試行する前にリクエストを修正して問題を解決する必要があることを示しています。たとえば、AWS 認証情報が無効な場合、E メールを再度送信する前に適切な認証情報が使用されるように設定を更新する必要があります。

エラー	説明	HTTPS ステータスコード	このコードを返すアクション
ConfigurationSetDoesNotExist	指定した設定セットが存在しません。設定セットは、メール送信イベントの発行に使用するオプションのパラメータです。詳細については、「 <a href="#">Amazon SES イベント発行を使用して E メール送信をモニタリングする (p. 296)</a> 」を参照してください。	400	SendEmail, SendRawEmail
IncompleteSignature	リクエストの署名が AWS 基準に適合しません。	400	すべて

エラー	説明	HTTPS ステータスコード	このコードを返すアクション
InternalFailure	不明なエラー、例外、または障害により、リクエスト処理が失敗しました。	500	すべて
InvalidAction	要求されたアクションまたはオペレーションは無効です。アクションが正しく入力されていることを確認します。	400	すべて
InvalidClientTokenId	指定された x.509 証明書、または AWS アクセスキー ID が見つかりません。	403	すべて
InvalidParameterCombination	同時に使用できないパラメータが、同時使用されています。	400	すべて
InvalidParameterValue	無効な値または範囲外の値が入力パラメータとして指定されました。	400	すべて
InvalidQueryParameter	AWS クエリ文字列が正しい形式でないか、AWS 標準を順守していません。	400	すべて
MailFromDomainNotVerified	Amazon SES では、指定された MAIL FROM ドメインを使用するために必要な MX レコードを読み取ることができず、メッセージを送信できませんでした。	400	SendEmail, SendRawEmail
MalformedQueryString	クエリ文字列に構文エラーがあります。	404	すべて
MessageRejected	アクションが失敗し、メッセージを送信できなかったことを示します。エラーの原因については、エラースタックを確認してください。このエラーの原因となる可能性がある問題については、「 <a href="#">Amazon SES の E メール送信エラー (p. 486)</a> 」を参照してください。	400	SendEmail, SendRawEmail

エラー	説明	HTTPS ステータスコード	このコードを返すアクション
MissingAction	リクエストに、アクションまたは必須パラメータが含まれていません。	400	すべて
MissingAuthenticationToken	リクエストには、有効な (登録された) AWS アクセスキー ID、または X.509 証明書のどちらか一方が含まれている必要があります。	403	すべて
MissingParameter	指定したアクションの必須パラメータが指定されていません。	400	すべて
OptInRequired	サービスを利用するためには、AWS アクセスキー ID を取得する必要があります。	403	すべて
RequestExpired	リクエストの日付スタンプの 15 分以上後またはリクエストの有効期限 (署名付き URL の場合など) の 15 分以上後に、リクエストが到着しました。または、リクエストの日付スタンプが現在より 15 分以上先です。	400	すべて
ServiceUnavailable	サーバーの一時的障害のため、リクエストは失敗しました。	503	すべて
Throttling	リクエストは、制限が必要なために実行が拒否されました。	400	すべて



# Amazon SES コード例

このセクションでは、Amazon SES を使用した Eメールの送信と Amazon SES アカウントの管理に役立つコード例を示します。コード例は、[C#](#)、[Go](#)、[Java](#)、[PHP](#)、[Python](#)、および [Ruby](#) の各言語で利用できます。

以下のいずれかのリンクを選択すると、そのタスクのコード例が表示されます。

- [AWS SDK を使用して Eメールを送信する \(p. 425\)](#)
- [Amazon SES SMTP インターフェイスを使用した Eメールの送信 \(p. 439\)](#)
- [AWS SDK を使用して raw Eメールを送信する \(p. 451\)](#)
- [複数の Eメールアドレスの検証 \(p. 459\)](#)
- [AWS リージョンで Eメールアイデンティティをレプリケートする \(p. 460\)](#)

## AWS SDK を使用して Eメールを送信する

AWS SDK には、Amazon SES および 他の AWS のサービスを操作するためのメソッドが組み込まれています。Amazon SES を他のAWSのサービスと併用する場合は、SDKの使用をお勧めします。AWS SDKの詳細については、「[Amazon Web Services のツール](#)」を参照してください。

このセクションでは、AWS SDK を使用した Amazon SES での Eメール送信プロセスを各種プログラミング言語のコード例で示します。

### C#

次のコード例は、[AWS SDK for .NET](#) を使用した Amazon SES でのメール送信の完全なソリューションを示しています。このコード例では、AWS SDK for .NET をインストール済みで、共有認証情報ファイルを作成済みであるものとします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

#### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、SDK ストアを使用して AWS アクセスキー ID とシークレットアクセスキーを指定できます。詳細については、AWS SDK for .NET デベロッパーガイドの「[AWS 認証情報の設定](#)」を参照してください。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
using Amazon;
using System;
using System.Collections.Generic;
using Amazon.SimpleEmail;
using Amazon.SimpleEmail.Model;

namespace AmazonSESSample
{
    class Program
    {
        // Replace sender@example.com with your "From" address.
```

```
// This address must be verified with Amazon SES.
static readonly string senderAddress = "sender@example.com";

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
static readonly string receiverAddress = "recipient@example.com";

// The configuration set to use for this email. If you do not want to use a
// configuration set, comment out the following property and the
// ConfigurationSetName = configSet argument below.
static readonly string configSet = "ConfigSet";

// The subject line for the email.
static readonly string subject = "Amazon SES test (AWS SDK for .NET)";

// The email body for recipients with non-HTML email clients.
static readonly string textBody = "Amazon SES Test (.NET)\r\n"
    + "This email was sent through Amazon SES "
    + "using the AWS SDK for .NET.";

// The HTML body of the email.
static readonly string htmlBody = @"<html>
<head></head>
<body>
  <h1>Amazon SES Test (AWS SDK for .NET)</h1>
  <p>This email was sent with
  <a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the
  <a href='https://aws.amazon.com/sdk-for-net/'>
  AWS SDK for .NET</a>.</p>
</body>
</html>";

static void Main(string[] args)
{
    // Replace USWest2 with the AWS Region you're using for Amazon SES.
    // Acceptable values are EUWest1, USEast1, and USWest2.
    using (var client = new
AmazonSimpleEmailServiceClient(RegionEndpoint.USWest2))
    {
        var sendRequest = new SendEmailRequest
        {
            Source = senderAddress,
            Destination = new Destination
            {
                ToAddresses =
                new List<string> { receiverAddress }
            },
            Message = new Message
            {
                Subject = new Content(subject),
                Body = new Body
                {
                    Html = new Content
                    {
                        {
                            Charset = "UTF-8",
                            Data = htmlBody
                        },
                    },
                    Text = new Content
                    {
                        {
                            Charset = "UTF-8",
                            Data = textBody
                        }
                    }
                }
            },
        },
        // If you are not using a configuration set, comment
        // or remove the following line
    }
}
```

```
        ConfigurationSetName = configSet
    };
    try
    {
        Console.WriteLine("Sending email using Amazon SES...");
        var response = client.SendEmail(sendRequest);
        Console.WriteLine("The email was sent successfully.");
    }
    catch (Exception ex)
    {
        Console.WriteLine("The email was not sent.");
        Console.WriteLine("Error message: " + ex.Message);
    }
}

Console.Write("Press any key to continue...");
Console.ReadKey();
}
}
```

## Go

次のコード例は、[AWS SDK for Go](#) を使用した Amazon SES でのメール送信の完全なソリューションを示しています。このコード例では、AWS SDK for Go をインストールしていること、および共有認証情報ファイルを作成しているものとします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2 つの環境変数 (AWS\_ACCESS\_KEY\_ID と AWS\_SECRET\_ACCESS\_KEY) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
package main

import (
    "fmt"

    //go get -u github.com/aws/aws-sdk-go
    "github.com/aws/aws-sdk-go/aws"
    "github.com/aws/aws-sdk-go/aws/session"
    "github.com/aws/aws-sdk-go/service/ses"
    "github.com/aws/aws-sdk-go/aws/awsserr"
)

const (
    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    Sender = "sender@example.com"

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    Recipient = "recipient@example.com"

    // Specify a configuration set. If you do not want to use a configuration
    // set, comment out the following constant and the
    // ConfigurationSetName: aws.String(ConfigurationSet) argument below
    ConfigurationSet = "ConfigSet"
)
```

```
// Replace us-west-2 with the AWS Region you're using for Amazon SES.
AwsRegion = "us-west-2"

// The subject line for the email.
Subject = "Amazon SES Test (AWS SDK for Go)"

// The HTML body for the email.
HtmlBody = "<h1>Amazon SES Test Email (AWS SDK for Go)</h1><p>This email was sent
with " +
    "<a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the " +
    "<a href='https://aws.amazon.com/sdk-for-go/'>AWS SDK for Go</a>.</p>"

//The email body for recipients with non-HTML email clients.
TextBody = "This email was sent with Amazon SES using the AWS SDK for Go."

// The character encoding for the email.
CharSet = "UTF-8"
)

func main() {

    // Create a new session and specify an AWS Region.
    sess, err := session.NewSession(&aws.Config{
        Region:aws.String(AwsRegion)},
    )

    // Create an SES client in the session.
    svc := ses.New(sess)

    // Assemble the email.
    input := &ses.SendEmailInput{
        Destination: &ses.Destination{
            CcAddresses: []*string{
            },
            ToAddresses: []*string{
                aws.String(Recipient),
            },
        },
        Message: &ses.Message{
            Body: &ses.Body{
                Html: &ses.Content{
                    CharSet: aws.String(CharSet),
                    Data:    aws.String(HtmlBody),
                },
                Text: &ses.Content{
                    CharSet: aws.String(CharSet),
                    Data:    aws.String(TextBody),
                },
            },
            Subject: &ses.Content{
                CharSet: aws.String(CharSet),
                Data:    aws.String(Subject),
            },
        },
        Source: aws.String(Sender),
        // Comment or remove the following line if you are not using a
        configuration set
        ConfigurationSetName: aws.String(ConfigurationSet),
    }

    // Attempt to send the email.
    result, err := svc.SendEmail(input)

    // Display error messages if they occur.
    if err != nil {
        if aerr, ok := err.(awserr.Error); ok {
```

```
        switch aerr.Code() {
        case ses.ErrCodeMessageRejected:
            fmt.Println(ses.ErrCodeMessageRejected, aerr.Error())
        case ses.ErrCodeMailFromDomainNotVerifiedException:
            fmt.Println(ses.ErrCodeMailFromDomainNotVerifiedException,
aerr.Error())
        case ses.ErrCodeConfigurationSetDoesNotExistException:
            fmt.Println(ses.ErrCodeConfigurationSetDoesNotExistException,
aerr.Error())
        default:
            fmt.Println(aerr.Error())
        }
    } else {
        // Print the error, cast err to awserr.Error to get the Code and
        // Message from an error.
        fmt.Println(err.Error())
    }
    return
}

fmt.Println("Email Sent!")
fmt.Println(result)
}
```

#### Java SDK v1

次のコード例は、[AWS SDK for Java](#) を使用した Amazon SES でのメール送信の完全なソリューションを示しています。このコード例では、AWS SDK for Java をインストールしていること、および共有認証情報ファイルを作成しているものとし、共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

```
package com.amazonaws.samples;

import java.io.IOException;

import com.amazonaws.regions.Regions;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailService;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailServiceClientBuilder;
import com.amazonaws.services.simpleemail.model.Body;
import com.amazonaws.services.simpleemail.model.Content;
import com.amazonaws.services.simpleemail.model.Destination;
import com.amazonaws.services.simpleemail.model.Message;
import com.amazonaws.services.simpleemail.model.SendEmailRequest;

public class AmazonSESSample {

    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    static final String FROM = "sender@example.com";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    static final String TO = "recipient@example.com";

    // The configuration set to use for this email. If you do not want to use a
    // configuration set, comment the following variable and the
    // .withConfigurationSetName(CONFIGSET); argument below.
    static final String CONFIGSET = "ConfigSet";

    // The subject line for the email.
    static final String SUBJECT = "Amazon SES test (AWS SDK for Java)";

    // The HTML body for the email.
    static final String HTMLBODY = "<h1>Amazon SES test (AWS SDK for Java)</h1>"
```

```
+ "<p>This email was sent with <a href='https://aws.amazon.com/ses/'>"
+ "Amazon SES</a> using the <a href='https://aws.amazon.com/sdk-for-java/'>"
+ "AWS SDK for Java</a>";

// The email body for recipients with non-HTML email clients.
static final String TEXTBODY = "This email was sent through Amazon SES "
    + "using the AWS SDK for Java.";

public static void main(String[] args) throws IOException {

    try {
        AmazonSimpleEmailService client =
            AmazonSimpleEmailServiceClientBuilder.standard()
                // Replace US_WEST_2 with the AWS Region you're using for
                // Amazon SES.
                .withRegion(Regions.US_WEST_2).build();
        SendEmailRequest request = new SendEmailRequest()
            .withDestination(
                new Destination().withToAddresses(TO))
            .withMessage(new Message()
                .withBody(new Body()
                    .withHtml(new Content()
                        .withCharset("UTF-8").withData(HTMLBODY))
                    .withText(new Content()
                        .withCharset("UTF-8").withData(TEXTBODY)))
                .withSubject(new Content()
                    .withCharset("UTF-8").withData(SUBJECT)))
            .withSource(FROM)
            // Comment or remove the next line if you are not using a
            // configuration set
            .withConfigurationSetName(CONFIGSET);
        client.sendEmail(request);
        System.out.println("Email sent!");
    } catch (Exception ex) {
        System.out.println("The email was not sent. Error message: "
            + ex.getMessage());
    }
}
}
```

## Java SDK v2

次のコード例は、[AWS SDK for Java 2.x](#) と [JavaMail API](#) を使用した Amazon SES でのメール送信の完全なソリューションを示しています。このコード例では、SDK for Java 2.x をインストールしていること、および共有認証情報ファイルを作成しているものとします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

```
package com.example.ses;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ses.SesClient;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;
import javax.mail.internet.MimeBodyPart;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.util.Properties;
import software.amazon.awssdk.core.SdkBytes;
```

```
import software.amazon.awssdk.services.ses.model.SendRawEmailRequest;
import software.amazon.awssdk.services.ses.model.RawMessage;
import software.amazon.awssdk.services.ses.model.SesException;

/**
 * To run this Java V2 code example, ensure that you have setup your development
 * environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendMessage {

    public static void main(String[] args) throws IOException {

        final String USAGE = "\n" +
            "Usage:\n" +
            "  SendMessage <sender> <recipient> <subject> \n\n" +
            "Where:\n" +
            "  sender - an email address that represents the sender. \n"+
            "  recipient - an email address that represents the recipient. \n"+
            "  subject - the subject line. \n" ;

        if (args.length != 3) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String sender = args[0];
        String recipient = args[1];
        String subject = args[2];

        Region region = Region.US_EAST_1;
        SesClient client = SesClient.builder()
            .region(region)
            .build();

        // The email body for non-HTML email clients
        String bodyText = "Hello,\r\n" + "See the list of customers. ";

        // The HTML body of the email
        String bodyHTML = "<html>" + "<head></head>" + "<body>" + "<h1>Hello!</h1>"
            + "<p> See the list of customers.</p>" + "</body>" + "</html>";

        try {
            send(client, sender, recipient, subject, bodyText, bodyHTML);
            client.close();
            System.out.println("Done");
        } catch (IOException | MessagingException e) {
            e.printStackTrace();
        }
    }

    public static void send(SesClient client,
        String sender,
        String recipient,
        String subject,
        String bodyText,
        String bodyHTML
    ) throws AddressException, MessagingException, IOException
    {

        Session session = Session.getDefaultInstance(new Properties());
        MimeMessage message = new MimeMessage(session);
```

```
// Add subject, from and to lines
message.setSubject(subject, "UTF-8");
message.setFrom(new InternetAddress(sender));
message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(recipient));

// Create a multipart/alternative child container
MimeMultipart msgBody = new MimeMultipart("alternative");

// Create a wrapper for the HTML and text parts
MimeBodyPart wrap = new MimeBodyPart();

// Define the text part
MimeBodyPart textPart = new MimeBodyPart();
textPart.setContent(bodyText, "text/plain; charset=UTF-8");

// Define the HTML part
MimeBodyPart htmlPart = new MimeBodyPart();
htmlPart.setContent(bodyHTML, "text/html; charset=UTF-8");

// Add the text and HTML parts to the child container
msgBody.addBodyPart(textPart);
msgBody.addBodyPart(htmlPart);

// Add the child container to the wrapper object
wrap.setContent(msgBody);

// Create a multipart/mixed parent container
MimeMultipart msg = new MimeMultipart("mixed");

// Add the parent container to the message
message.setContent(msg);

// Add the multipart/alternative part to the message
msg.addBodyPart(wrap);

try {
    System.out.println("Attempting to send an email through Amazon SES " +
"using the AWS SDK for Java...");

    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    message.writeTo(outputStream);
    ByteBuffer buf = ByteBuffer.wrap(outputStream.toByteArray());

    byte[] arr = new byte[buf.remaining()];
    buf.get(arr);

    SdkBytes data = SdkBytes.fromByteArray(arr);
    RawMessage rawMessage = RawMessage.builder()
        .data(data)
        .build();

    SendRawEmailRequest rawEmailRequest = SendRawEmailRequest.builder()
        .rawMessage(rawMessage)
        .build();

    client.sendRawEmail(rawEmailRequest);

} catch (SesException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}
```



## JavaScript

次のコード例は、Amazon SES でのメール送信の完全なソリューションを [AWS SDK for JavaScript in Node.js](#) で示しています。このコード例では、Node.js に SDK for JavaScript がインストールされていることが前提になっています。AWS アクセスキー ID、シークレットアクセスキー、および使用する AWS リージョンを含む設定ファイルも作成する必要があります。このファイルの作成の詳細については、「[Loading Credentials in Node.js from a JSON File](#)」を参照してください。

### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2つの環境変数 (AWS\_ACCESS\_KEY\_ID と AWS\_SECRET\_ACCESS\_KEY) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
'use strict';

var aws = require('aws-sdk');

// Provide the full path to your config.json file.
aws.config.loadFromPath('./config.json');

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
const sender = "Sender Name <sender@example.com>";

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
const recipient = "recipient@example.com";

// Specify a configuration set. If you do not want to use a configuration
// set, comment the following variable, and the
// ConfigurationSetName : configuration_set argument below.
const configuration_set = "ConfigSet";

// The subject line for the email.
const subject = "Amazon SES Test (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
const body_text = "Amazon SES Test (SDK for JavaScript in Node.js)\r\n"
  + "This email was sent with Amazon SES using the "
  + "AWS SDK for JavaScript in Node.js.";

// The HTML body of the email.
const body_html = `
<head></head>
<body>
  <h1>Amazon SES Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
  <a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the
  <a href='https://aws.amazon.com/sdk-for-node-js/'>
    AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding for the email.
const charset = "UTF-8";

// Create a new SES object.
```

```
var ses = new aws.SES();

// Specify the parameters to pass to the API.
var params = {
  Source: sender,
  Destination: {
    ToAddresses: [
      recipient
    ],
  },
  Message: {
    Subject: {
      Data: subject,
      Charset: charset
    },
    Body: {
      Text: {
        Data: body_text,
        Charset: charset
      },
      Html: {
        Data: body_html,
        Charset: charset
      }
    }
  },
  ConfigurationSetName: configuration_set
};

//Try to send the email.
ses.sendEmail(params, function(err, data) {
  // If something goes wrong, print an error message.
  if(err) {
    console.log(err.message);
  } else {
    console.log("Email sent! Message ID: ", data.MessageId);
  }
});
```

## PHP

次のコード例は、[AWS SDK for PHP](#) を使用した Amazon SES でのメール送信の完全なソリューションを示しています。このコード例では、AWS SDK for PHP をインストールしていること、および共有認証情報ファイルを作成しているものとします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2つの環境変数 (`AWS_ACCESS_KEY_ID` と `AWS_SECRET_ACCESS_KEY`) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
<?php

// If necessary, modify the path in the require statement below to refer to the
// location of your Composer autoload.php file.
require 'vendor/autoload.php';

use Aws\Ses\SesClient;
use Aws\Exception\AwsException;
```

```
// Create an SesClient. Change the value of the region parameter if you're
// using an AWS Region other than US West (Oregon). Change the value of the
// profile parameter if you want to use a profile in your credentials file
// other than the default.
$SesClient = new SesClient([
    'profile' => 'default',
    'version' => '2010-12-01',
    'region' => 'us-west-2'
]);

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
$sender_email = 'sender@example.com';

// Replace these sample addresses with the addresses of your recipients. If
// your account is still in the sandbox, these addresses must be verified.
$recipient_emails = ['recipient1@example.com', 'recipient2@example.com'];

// Specify a configuration set. If you do not want to use a configuration
// set, comment the following variable, and the
// 'ConfigurationSetName' => $configuration_set argument below.
$configuration_set = 'ConfigSet';

$subject = 'Amazon SES test (AWS SDK for PHP)';
$plaintext_body = 'This email was sent with Amazon SES using the AWS SDK for PHP.';
$html_body = '<h1>AWS Amazon Simple Email Service Test Email</h1>'.
    '<p>This email was sent with <a href="https://aws.amazon.com/ses/">.
    'Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-
php/">'.
    'AWS SDK for PHP</a>.</p>';
$char_set = 'UTF-8';

try {
    $result = $SesClient->sendEmail([
        'Destination' => [
            'ToAddresses' => $recipient_emails,
        ],
        'ReplyToAddresses' => [$sender_email],
        'Source' => $sender_email,
        'Message' => [
            'Body' => [
                'Html' => [
                    'Charset' => $char_set,
                    'Data' => $html_body,
                ],
                'Text' => [
                    'Charset' => $char_set,
                    'Data' => $plaintext_body,
                ],
            ],
            'Subject' => [
                'Charset' => $char_set,
                'Data' => $subject,
            ],
        ],
    ],
    // If you aren't using a configuration set, comment or delete the
    // following line
    'ConfigurationSetName' => $configuration_set,
]);
$messageId = $result['MessageId'];
echo("Email sent! Message ID: $messageId.\n");
} catch (AwsException $e) {
    // output error message if fails
    echo $e->getMessage();
    echo("The email was not sent. Error message: ".$e->getAwsErrorMessage()."\n");
    echo "\n";
}
```

```
}
```

## Python

次のコード例は、[AWS SDK for Python \(Boto\)](#) を使用した Amazon SES でのメール送信の完全なソリューションを示しています。このコード例では、AWS SDK for Python (Boto) をインストールしていること、および共有認証情報ファイルを作成しているものとします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2つの環境変数 (AWS\_ACCESS\_KEY\_ID と AWS\_SECRET\_ACCESS\_KEY) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
import boto3
from botocore.exceptions import ClientError

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
SENDER = "Sender Name <sender@example.com>"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
RECIPIENT = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable, and the
# ConfigurationSetName=CONFIGURATION_SET argument below.
CONFIGURATION_SET = "ConfigSet"

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# The subject line for the email.
SUBJECT = "Amazon SES Test (SDK for Python)"

# The email body for recipients with non-HTML email clients.
BODY_TEXT = ("Amazon SES Test (Python)\r\n"
             "This email was sent with Amazon SES using the "
             "AWS SDK for Python (Boto).")

# The HTML body of the email.
BODY_HTML = """<html>
<head></head>
<body>
  <h1>Amazon SES Test (SDK for Python)</h1>
  <p>This email was sent with
  <a href='https://aws.amazon.com/ses/'>Amazon SES</a> using the
  <a href='https://aws.amazon.com/sdk-for-python/'>
    AWS SDK for Python (Boto)</a>.</p>
</body>
</html>
"""

# The character encoding for the email.
CHARSET = "UTF-8"

# Create a new SES resource and specify a region.
client = boto3.client('ses', region_name=AWS_REGION)
```

```
# Try to send the email.
try:
    #Provide the contents of the email.
    response = client.send_email(
        Destination={
            'ToAddresses': [
                RECIPIENT,
            ],
        },
        Message={
            'Body': {
                'Html': {
                    'Charset': CHARSET,
                    'Data': BODY_HTML,
                },
                'Text': {
                    'Charset': CHARSET,
                    'Data': BODY_TEXT,
                },
            },
            'Subject': {
                'Charset': CHARSET,
                'Data': SUBJECT,
            },
        },
        Source=SENDER,
        # If you are not using a configuration set, comment or delete the
        # following line
        ConfigurationSetName=CONFIGURATION_SET,
    )
# Display an error if something goes wrong.
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print("Email sent! Message ID:"),
    print(response['MessageId'])
```

## Ruby

次のコード例は、[AWS SDK for Ruby](#) を使用した Amazon SES でのメール送信の完全なソリューションを示しています。このコード例では、AWS SDK for Ruby をインストール済みで、共有認証情報ファイルを作成済みであるものとします。SDK for Ruby のインストールの詳細については、[AWS SDK for Ruby デベロッパーガイド](#)の「AWS SDK for Ruby のインストール」を参照してください。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2つの環境変数 (AWS\_ACCESS\_KEY\_ID と AWS\_SECRET\_ACCESS\_KEY) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
require 'aws-sdk'

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = "sender@example.com"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
```

```
recipient = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable and the
# configuration_set_name: configsetname argument below.
configsetname = "ConfigSet"

# Replace us-west-2 with the AWS Region you're using for Amazon SES.
awsregion = "us-west-2"

# The subject line for the email.
subject = "Amazon SES test (AWS SDK for Ruby)"

# The HTML body of the email.
htmlbody =
  '<h1>Amazon SES test (AWS SDK for Ruby)</h1>'\
  '<p>This email was sent with <a href="https://aws.amazon.com/ses/">'\
  'Amazon SES</a> using the <a href="https://aws.amazon.com/sdk-for-ruby/">'\
  'AWS SDK for Ruby</a>.'
```

```
end
```

## Amazon SES SMTP インターフェイスを使用した Eメールの送信

いくつかのプログラミング言語には、SMTP を使用したメール送信用の標準ライブラリが含まれています。これらのライブラリを使用して、軽量で高度に設定可能なメール送信アプリケーションを作成できます。

このセクションでは、SMTP インターフェイスを使用した Amazon SES でのメール送信プロセスを各種プログラミング言語のコード例で示します。以下のコード例では、可能な限り、標準ライブラリを使用しています。

### C#

次のコード例は、Amazon SES SMTP インターフェイスでのメール送信の完全なソリューションを C# で示しています。このコード例を実行するには、SMTP 認証情報を取得する必要があります。詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

```
using System;
using System.Net;
using System.Net.Mail;

namespace AmazonSESSample
{
    class Program
    {
        static void Main(string[] args)
        {
            // Replace sender@example.com with your "From" address.
            // This address must be verified with Amazon SES.
            String FROM = "sender@example.com";
            String FROMNAME = "Sender Name";

            // Replace recipient@example.com with a "To" address. If your account
            // is still in the sandbox, this address must be verified.
            String TO = "recipient@amazon.com";

            // Replace smtp_username with your Amazon SES SMTP user name.
            String SMTP_USERNAME = "smtp_username";

            // Replace smtp_password with your Amazon SES SMTP password.
            String SMTP_PASSWORD = "smtp_password";

            // (Optional) the name of a configuration set to use for this message.
            // If you comment out this line, you also need to remove or comment out
            // the "X-SES-CONFIGURATION-SET" header below.
            String CONFIGSET = "ConfigSet";

            // If you're using Amazon SES in a region other than US West (Oregon),
            // replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
            // endpoint in the appropriate AWS Region.
            String HOST = "email-smtp.us-west-2.amazonaws.com";

            // The port you will connect to on the Amazon SES SMTP endpoint. We
            // are choosing port 587 because we will use STARTTLS to encrypt
            // the connection.
            int PORT = 587;
```

```
// The subject line of the email
String SUBJECT =
    "Amazon SES test (SMTP interface accessed using C#)";

// The body of the email
String BODY =
    "<h1>Amazon SES Test</h1>" +
    "<p>This email was sent through the " +
    "<a href='https://aws.amazon.com/ses'>Amazon SES</a> SMTP interface " +
    "using the .NET System.Net.Mail library.</p>";

// Create and build a new MailMessage object
MailMessage message = new MailMessage();
message.IsBodyHtml = true;
message.From = new MailAddress(FROM, FROMNAME);
message.To.Add(new MailAddress(TO));
message.Subject = SUBJECT;
message.Body = BODY;
// Comment or delete the next line if you are not using a configuration set
message.Headers.Add("X-SES-CONFIGURATION-SET", CONFIGSET);

using (var client = new System.Net.Mail.SmtpClient(HOST, PORT))
{
    // Pass SMTP credentials
    client.Credentials =
        new NetworkCredential(SMTP_USERNAME, SMTP_PASSWORD);

    // Enable SSL encryption
    client.EnableSsl = true;

    // Try to send the message. Show status in console.
    try
    {
        Console.WriteLine("Attempting to send email...");
        client.Send(message);
        Console.WriteLine("Email sent!");
    }
    catch (Exception ex)
    {
        Console.WriteLine("The email was not sent.");
        Console.WriteLine("Error message: " + ex.Message);
    }
}
}
```

## Go

次のコード例は、Amazon SES SMTP インターフェイスでの E メール送信の完全なソリューションを Go プログラミング言語で示しています。このコード例を実行するには、SMTP 認証情報を取得する必要があります。詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。さらに、[Gomail パッケージ](#)もインストールする必要があります。

```
package main

import (
    "fmt"
    "gopkg.in/gomail.v2" //go get gopkg.in/gomail.v2
)

const (
    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.

```



Amazon Simple Email Service Classic デベロッパーガイド  
SMTP インターフェイスを使用した Eメールの送信

```
Sender = "sender@example.com"
SenderName = "Sender Name"

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
Recipient = "recipient@example.com"

// Replace SmtplibUser with your Amazon SES SMTP user name.
SmtplibUser = "SmtplibUser"

// Replace SmtplibPass with your Amazon SES SMTP password.
SmtplibPass = "SmtplibPass"

// The name of the configuration set to use for this message.
// If you comment out or remove this variable, you will also need to
// comment out or remove the header below.
ConfigSet = "ConfigSet"

// If you're using Amazon SES in an AWS Region other than US West (Oregon),
// replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
// endpoint in the appropriate region.
Host = "email-smtp.us-west-2.amazonaws.com"
Port = 587

// The subject line for the email.
Subject = "Amazon SES Test (Gomail)"

// The HTML body for the email.
HtmlBody = "<html><head><title>SES Sample Email</title></head><body>" +
  "<h1>Amazon SES Test Email (Gomail)</h1>" +
  "<p>This email was sent with " +
  "<a href='https://aws.amazon.com/ses/'>Amazon SES</a> using " +
  "the <a href='https://github.com/go-gomail/gomail/'>Gomail " +
  "package</a> for <a href='https://golang.org/'>Go</a>.</p>" +
  "</body></html>"

//The email body for recipients with non-HTML email clients.
TextBody = "This email was sent with Amazon SES using the Gomail package."

// The tags to apply to this message. Separate multiple key-value pairs
// with commas.
// If you comment out or remove this variable, you will also need to
// comment out or remove the header on line 80.
Tags = "genre=test,genre2=test2"

// The character encoding for the email.
Charset = "UTF-8"
)

func main() {
  // Create a new message.
  m := gomail.NewMessage()

  // Set the main email part to use HTML.
  m.SetBody("text/html", HtmlBody)

  // Set the alternative part to plain text.
  m.AddAlternative("text/plain", TextBody)

  // Construct the message headers, including a Configuration Set and a Tag.
  m.SetHeaders(map[string][]string{
    "From": {m.FormatAddress(Sender, SenderName)},
    "To": {Recipient},
    "Subject": {Subject},
  })
}
```

```
// Comment or remove the next line if you are not using a configuration set
"X-SES-CONFIGURATION-SET": {ConfigSet},
// Comment or remove the next line if you are not using custom tags
"X-SES-MESSAGE-TAGS": {Tags},
})

// Send the email.
d := gomail.NewPlainDialer(Host, Port, SmtplibUser, SmtplibPass)

// Display an error message if something goes wrong; otherwise,
// display a message confirming that the message was sent.
if err := d.DialAndSend(m); err != nil {
    fmt.Println(err)
} else {
    fmt.Println("Email sent!")
}
}
```

## Java

次のコード例は、Amazon SES SMTP インターフェイスでのメール送信の完全なソリューションを Java で示しています。このコード例を実行するには、SMTP 認証情報を取得する必要があります。詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。さらに、[JavaMail API](#) もダウンロードする必要があります。

```
import java.util.Properties;

import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class AmazonSESSample {

    // Replace sender@example.com with your "From" address.
    // This address must be verified.
    static final String FROM = "sender@example.com";
    static final String FROMNAME = "Sender Name";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    static final String TO = "recipient@example.com";

    // Replace smtp_username with your Amazon SES SMTP user name.
    static final String SMTP_USERNAME = "smtp_username";

    // Replace smtp_password with your Amazon SES SMTP password.
    static final String SMTP_PASSWORD = "smtp_password";

    // The name of the Configuration Set to use for this message.
    // If you comment out or remove this variable, you will also need to
    // comment out or remove the header below.
    static final String CONFIGSET = "ConfigSet";

    // Amazon SES SMTP host name. This example uses the US West (Oregon) region.
    // See https://docs.aws.amazon.com/ses/latest/DeveloperGuide/regions.html#region-
    endpoints
    // for more information.
    static final String HOST = "email-smtp.us-west-2.amazonaws.com";

    // The port you will connect to on the Amazon SES SMTP endpoint.
    static final int PORT = 587;
```

```
static final String SUBJECT = "Amazon SES test (SMTP interface accessed using
Java)";

static final String BODY = String.join(
    System.getProperty("line.separator"),
    "<h1>Amazon SES SMTP Email Test</h1>",
    "<p>This email was sent with Amazon SES using the ",
    "<a href='https://github.com/javaee/javamail'>Javamail Package</a>",
    " for <a href='https://www.java.com'>Java</a>."
);

public static void main(String[] args) throws Exception {

    // Create a Properties object to contain connection configuration information.
    Properties props = System.getProperties();
    props.put("mail.transport.protocol", "smtp");
    props.put("mail.smtp.port", PORT);
    props.put("mail.smtp.starttls.enable", "true");
    props.put("mail.smtp.auth", "true");

    // Create a Session object to represent a mail session with the specified
    properties.
    Session session = Session.getDefaultInstance(props);

    // Create a message with the specified information.
    MimeMessage msg = new MimeMessage(session);
    msg.setFrom(new InternetAddress(FROM, FROMNAME));
    msg.setRecipient(Message.RecipientType.TO, new InternetAddress(TO));
    msg.setSubject(SUBJECT);
    msg.setContent(BODY, "text/html");

    // Add a configuration set header. Comment or delete the
    // next line if you are not using a configuration set
    msg.setHeader("X-SES-CONFIGURATION-SET", CONFIGSET);

    // Create a transport.
    Transport transport = session.getTransport();

    // Send the message.
    try
    {
        System.out.println("Sending...");

        // Connect to Amazon SES using the SMTP username and password you specified
        above.
        transport.connect(HOST, SMTP_USERNAME, SMTP_PASSWORD);

        // Send the email.
        transport.sendMessage(msg, msg.getAllRecipients());
        System.out.println("Email sent!");
    }
    catch (Exception ex) {
        System.out.println("The email was not sent.");
        System.out.println("Error message: " + ex.getMessage());
    }
    finally
    {
        // Close and terminate the connection.
        transport.close();
    }
}
```

## JavaScript

次のコード例は、Amazon SES SMTP インターフェイスでのメール送信の完全なソリューションを Node.js の [NodeMailer](#) モジュールで示しています。

このコード例を実行するには、まず SMTP 認証情報を取得する必要があります。詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。また、[NodeMailer](#) モジュールをインストールする必要があります。

```
/*
This code uses callbacks to handle asynchronous function responses.
It currently demonstrates using an async-await pattern.
AWS supports both the async-await and promises patterns.
For more information, see the following:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/
async_function
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises
https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/calling-services-
asynchronously.html
https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-handler.html
*/

"use strict";
const nodemailer = require("nodemailer");

// If you're using Amazon SES in a region other than US West (Oregon),
// replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
// endpoint in the appropriate AWS Region.
const smtpEndpoint = "email-smtp.us-west-2.amazonaws.com";

// The port to use when connecting to the SMTP server.
const port = 587;

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
const senderAddress = "Mary Major <sender@example.com>";

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified. To specify
// multiple addresses, separate each address with a comma.
var toAddresses = "recipient@example.com";

// CC and BCC addresses. If your account is in the sandbox, these
// addresses have to be verified. To specify multiple addresses, separate
// each address with a comma.
var ccAddresses = "cc-recipient0@example.com,cc-recipient1@example.com";
var bccAddresses = "bcc-recipient@example.com";

// Replace smtp_username with your Amazon SES SMTP user name.
const smtpUsername = "AKIAIOSFODNN7EXAMPLE";

// Replace smtp_password with your Amazon SES SMTP password.
const smtpPassword = "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY";

// (Optional) the name of a configuration set to use for this message.
var configurationSet = "ConfigSet";

// The subject line of the email
var subject = "Amazon SES test (Nodemailer)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon SES Test (Nodemailer)
-----
This email was sent through the Amazon SES SMTP interface using Nodemailer.
`;
```

```
// The body of the email for recipients whose email clients support HTML content.
var body_html = `
```

## Perl

次のコード例は、Amazon SES SMTP インターフェイスでのメール送信の完全なソリューションを Perl で示しています。このコード例を実行するには、SMTP 認証情報を取得する必要があります。詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。また、`Email::Sender`、`Email::MIME`、および `Try::Tiny` モジュールを [CPAN](#) からインストールする必要があります。

```
#!/usr/bin/perl
use warnings;
```

```
use strict;
use Email::Sender::Simple qw(sendmail);
use Email::Sender::Transport::SMTP;
use Email::MIME;
use Try::Tiny;

# Replace sender@example.com with your "From" address.
# This address must be verified.
my $sender = 'Sender name <sender@example.com>';

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
my $recipient = 'recipient@example.com';

# Replace smtp_username with your Amazon SES SMTP user name.
my $smtp_username = "smtp_username";

# Replace smtp_password with your Amazon SES SMTP password.
my $smtp_password = "smtp_password";

# (Optional) the name of a configuration set to use for this message.
# If you comment out this line, you also need to remove or comment out
# the "X-SES-CONFIGURATION-SET:" header below.
my $configset = "ConfigSet";

# If you're using Amazon SES in an AWS Region other than US West (Oregon),
# replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
# endpoint in the appropriate region.
my $host = "email-smtp.us-west-2.amazonaws.com";
my $port = 587;

# The subject line of the email.
my $subject = "Amazon SES Test (Perl)";

# The HTML body for the email.
my $htmlbody = <<'END_HTML';
<html>
  <head></head>
  <body>
    <h1>Amazon SES SMTP Email Test</h1>
    <p>This email was sent with Amazon SES using the
      <a href='https://www.perl.org/'>Perl</a>
      <a href='http://search.cpan.org/~rjbs/Email-Sender-1.300031/'>
        Email::Sender</a> library.</p>
  </body>
</html>
END_HTML

# The email body for recipients with non-HTML email clients.
my $textbody = "Amazon SES Test\r\n"
  . "This message was sent with Amazon SES using the Perl "
  . "Email::Sender module.";

# Create the SMTP transport.
my $transport = Email::Sender::Transport::SMTP->new(
  host      => "$host",
  port     => "$port",
  ssl      => 'starttls',
  sasl_username => "$smtp_username",
  sasl_password => "$smtp_password",
);

# Build a multipart MIME message with an HTML part and a text part.
my $message = Email::MIME->create(
  attributes => {
    content_type => 'multipart/alternative',
```

```
        charset      => 'UTF-8',
    },
    header_str => [
        From      => "$sender",
        To        => "$recipient",
        Subject   => "$subject",
    ],
    parts => [
        Email::MIME->create(
            attributes => { content_type => 'text/plain' },
            body       => "$textbody",
        ),
        Email::MIME->create(
            attributes => { content_type => 'text/html' },
            body       => "$htmlbody",
        )
    ],
);

# Add the configuration set header to the MIME message.
$message->header_str_set( 'X-SES-CONFIGURATION-SET' => "$configset" );

# Try to send the email using the sendmail function from
# Email::Sender::Simple.
try {
    sendmail($message, { transport => $transport });
# If something goes wrong, print an error message.
} catch {
    die "Error sending email: $_";
};
```

## PHP

次のコード例は、Amazon SES SMTP インターフェイスでのメール送信の完全なソリューションを PHP で示しています。このコード例を実行するには、SMTP 認証情報を取得する必要があります。詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。また、[Composer](#) を使用して、[PHPMailer パッケージ](#) をインストールする必要があります。

```
<?php

// Import PHPMailer classes into the global namespace
// These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

// If necessary, modify the path in the require statement below to refer to the
// location of your Composer autoload.php file.
require 'vendor/autoload.php';

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
$sender = 'sender@example.com';
$senderName = 'Sender Name';

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
$recipient = 'recipient@example.com';

// Replace smtp_username with your Amazon SES SMTP user name.
$usernameSmtplib = 'smtp_username';

// Replace smtp_password with your Amazon SES SMTP password.
$passwordSmtplib = 'smtp_password';
```

```
// Specify a configuration set. If you do not want to use a configuration
// set, comment or remove the next line.
$configurationSet = 'ConfigSet';

// If you're using Amazon SES in a region other than US West (Oregon),
// replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
// endpoint in the appropriate region.
$host = 'email-smtp.us-west-2.amazonaws.com';
$port = 587;

// The subject line of the email
$subject = 'Amazon SES test (SMTP interface accessed using PHP)';

// The plain-text body of the email
$bodyText = "Email Test\r\nThis email was sent through the
    Amazon SES SMTP interface using the PHPMailer class.";

// The HTML-formatted body of the email
$bodyHtml = '<h1>Email Test</h1>
    <p>This email was sent through the
    <a href="https://aws.amazon.com/ses">Amazon SES</a> SMTP
    interface using the <a href="https://github.com/PHPMailer/PHPMailer">
    PHPMailer</a> class.</p>';

$mail = new PHPMailer(true);

try {
    // Specify the SMTP settings.
    $mail->isSMTP();
    $mail->setFrom($sender, $senderName);
    $mail->Username = $usernameSmtp;
    $mail->Password = $passwordSmtp;
    $mail->Host = $host;
    $mail->Port = $port;
    $mail->SMTPAuth = true;
    $mail->SMTPSecure = 'tls';
    $mail->addCustomHeader('X-SES-CONFIGURATION-SET', $configurationSet);

    // Specify the message recipients.
    $mail->addAddress($recipient);
    // You can also add CC, BCC, and additional To recipients here.

    // Specify the content of the message.
    $mail->isHTML(true);
    $mail->Subject = $subject;
    $mail->Body = $bodyHtml;
    $mail->AltBody = $bodyText;
    $mail->Send();
    echo "Email sent!" , PHP_EOL;
} catch (phpmailerException $e) {
    echo "An error occurred. {$e->errorMessage()}", PHP_EOL; //Catch errors from
    PHPMailer.
} catch (Exception $e) {
    echo "Email not sent. {$mail->ErrorInfo}", PHP_EOL; //Catch errors from Amazon SES.
}

?>
```

## Python

次のコード例は、Amazon SES SMTP インターフェイスでのメール送信の完全なソリューションを Python で示しています。このコード例を実行するには、SMTP 認証情報を取得する必要があります。詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。



```
import smtplib
import email.utils
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

# Replace sender@example.com with your "From" address.
# This address must be verified.
SENDER = 'sender@example.com'
SENDERNAME = 'Sender Name'

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
RECIPIENT = 'recipient@example.com'

# Replace smtp_username with your Amazon SES SMTP user name.
USERNAME_SMTP = "smtp_username"

# Replace smtp_password with your Amazon SES SMTP password.
PASSWORD_SMTP = "smtp_password"

# (Optional) the name of a configuration set to use for this message.
# If you comment out this line, you also need to remove or comment out
# the "X-SES-CONFIGURATION-SET:" header below.
CONFIGURATION_SET = "ConfigSet"

# If you're using Amazon SES in an AWS Region other than US West (Oregon),
# replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
# endpoint in the appropriate region.
HOST = "email-smtp.us-west-2.amazonaws.com"
PORT = 587

# The subject line of the email.
SUBJECT = 'Amazon SES Test (Python smtplib)'

# The email body for recipients with non-HTML email clients.
BODY_TEXT = ("Amazon SES Test\r\n"
             "This email was sent through the Amazon SES SMTP "
             "Interface using the Python smtplib package."
             )

# The HTML body of the email.
BODY_HTML = """<html>
<head></head>
<body>
  <h1>Amazon SES SMTP Email Test</h1>
  <p>This email was sent with Amazon SES using the
    <a href='https://www.python.org/'>Python</a>
    <a href='https://docs.python.org/3/library/smtplib.html'>
      smtplib</a> library.</p>
</body>
</html>
"""

# Create message container - the correct MIME type is multipart/alternative.
msg = MIMEMultipart('alternative')
msg['Subject'] = SUBJECT
msg['From'] = email.utils.formataddr((SENDERNAME, SENDER))
msg['To'] = RECIPIENT
# Comment or delete the next line if you are not using a configuration set
msg.add_header('X-SES-CONFIGURATION-SET', CONFIGURATION_SET)

# Record the MIME types of both parts - text/plain and text/html.
part1 = MIMEText(BODY_TEXT, 'plain')
part2 = MIMEText(BODY_HTML, 'html')
```

```
# Attach parts into message container.
# According to RFC 2046, the last part of a multipart message, in this case
# the HTML message, is best and preferred.
msg.attach(part1)
msg.attach(part2)

# Try to send the message.
try:
    server = smtplib.SMTP(HOST, PORT)
    server.ehlo()
    server.starttls()
    #smtplib docs recommend calling ehlo() before & after starttls()
    server.ehlo()
    server.login(USERNAME_SMTP, PASSWORD_SMTP)
    server.sendmail(SENDER, RECIPIENT, msg.as_string())
    server.close()
# Display an error message if something goes wrong.
except Exception as e:
    print ("Error: ", e)
else:
    print ("Email sent!")
```

## Ruby

次のコード例は、Amazon SES SMTP インターフェイスでのメール送信の完全なソリューションを Ruby で示しています。このコード例を実行するには、SMTP 認証情報を取得する必要があります。詳細については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。

```
require 'net/smtp'

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = "sender@example.com"
senderName = "Sender Name"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
recipient = "recipient@example.com"

# Replace smtp_username with your Amazon SES SMTP user name.
smtp_username = "smtp_username"

# Replace smtp_password with your Amazon SES SMTP password.
smtp_password = "smtp_password"

# (Optional) the name of a configuration set to use for this message.
# If you comment out this line, you also need to remove or comment out
# the "X-SES-CONFIGURATION-SET" header below.
configSet = "ConfigSet"

# If you're using Amazon SES in an AWS Region other than US West (Oregon),
# replace email-smtp.us-west-2.amazonaws.com with the Amazon SES SMTP
# endpoint in the appropriate region.
server = "email-smtp.us-west-2.amazonaws.com"
port = 587

# The subject line of the email.
subject = "Amazon SES Test (Ruby Net::SMTP library)"

# Specify the headers and body of the message as a variable.
message = [
    #Remove the next line if you are not using a configuration set
    "X-SES-CONFIGURATION-SET: #{configSet}",
    "Content-Type: text/html; charset=UTF-8",
```

```
"Content-Transfer-Encoding: 7bit",
"From: #{senderName} <#{sender}>",
"To: #{recipient}",
"Subject: #{subject}",
"",
"<h1>Amazon SES Test (Ruby Net::SMTP library)</h1>",
"<p>This email was sent with \
<a href='https://aws.amazon.com/ses/'>\
Amazon SES</a> using the Ruby Net::SMTP library.</p>"
].join("\n")

# Create a new SMTP object called "smtp."
smtp = Net::SMTP.new(server, port)

# Tell the smtp object to connect using TLS.
smtp.enable_starttls

# Open an SMTP session and log in to the server using SMTP authentication.
smtp.start(server, smtp_username, smtp_password, :login)

# Try to send the message.
begin
  smtp.send_message(message, sender, recipient)
  puts "Email sent!"
# Print an error message if something goes wrong.
rescue => e
  puts e
end
```

## AWS SDK を使用して raw E メールを送信する

AWS SDK には、Amazon SES および 他の AWS のサービスを実行するためのメソッドが組み込まれています。Amazon SES を他の AWS のサービスと併用する場合は、SDK の使用をお勧めします。AWS SDK の詳細については、「[Amazon Web Services のツール](#)」を参照してください。

このセクションでは、AWS SDK を使用した Amazon SES 経由での raw E メール送信プロセスを各種プログラミング言語のコード例で示します。

### Java

次のコード例は、[JavaMail](#) ライブラリと [AWS SDK for Java](#) を使用して、HTML 部分、テキスト部分、および添付ファイルを含む raw E メールを作成して送信する方法を示しています。

このコード例では、AWS SDK for Java をインストールしていること、および共有認証情報ファイルを作成しているものとします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

#### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2 つの環境変数 (AWS\_ACCESS\_KEY\_ID と AWS\_SECRET\_ACCESS\_KEY) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
package com.amazonaws.samples;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.PrintStream;
```

```
import java.nio.ByteBuffer;
import java.util.Properties;

// JavaMail libraries. Download the JavaMail API
// from https://javaee.github.io/javamail/
import javax.activation.DataHandler;
import javax.activation.DataSource;
import javax.activation.FileDataSource;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.internet.AddressException;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;

// AWS SDK libraries. Download the AWS SDK for Java
// from https://aws.amazon.com/sdk-for-java
import com.amazonaws.regions.Regions;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailService;
import com.amazonaws.services.simpleemail.AmazonSimpleEmailServiceClientBuilder;
import com.amazonaws.services.simpleemail.model.RawMessage;
import com.amazonaws.services.simpleemail.model.SendRawEmailRequest;

public class AmazonSESSample {

    // Replace sender@example.com with your "From" address.
    // This address must be verified with Amazon SES.
    private static String SENDER = "Sender Name <sender@example.com>";

    // Replace recipient@example.com with a "To" address. If your account
    // is still in the sandbox, this address must be verified.
    private static String RECIPIENT = "recipient@example.com";

    // Specify a configuration set. If you do not want to use a configuration
    // set, comment the following variable, and the
    // ConfigurationSetName=CONFIGURATION_SET argument below.
    private static String CONFIGURATION_SET = "ConfigSet";

    // The subject line for the email.
    private static String SUBJECT = "Customer service contact info";

    // The full path to the file that will be attached to the email.
    // If you're using Windows, escape backslashes as shown in this variable.
    private static String ATTACHMENT = "C:\\Users\\sender\\customers-to-contact.xlsx";

    // The email body for recipients with non-HTML email clients.
    private static String BODY_TEXT = "Hello,\r\n"
        + "Please see the attached file for a list "
        + "of customers to contact.";

    // The HTML body of the email.
    private static String BODY_HTML = "<html>"
        + "<head></head>"
        + "<body>"
        + "<h1>Hello!</h1>"
        + "<p>Please see the attached file for a "
        + "list of customers to contact.</p>"
        + "</body>"
        + "</html>";

    public static void main(String[] args) throws AddressException, MessagingException,
    IOException {

        Session session = Session.getDefaultInstance(new Properties());
```

```
// Create a new MimeMessage object.
MimeMessage message = new MimeMessage(session);

// Add subject, from and to lines.
message.setSubject(SUBJECT, "UTF-8");
message.setFrom(new InternetAddress(SENDER));
message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(RECIPIENT));

// Create a multipart/alternative child container.
MimeMultipart msg_body = new MimeMultipart("alternative");

// Create a wrapper for the HTML and text parts.
MimeBodyPart wrap = new MimeBodyPart();

// Define the text part.
MimeBodyPart textPart = new MimeBodyPart();
textPart.setContent(BODY_TEXT, "text/plain; charset=UTF-8");

// Define the HTML part.
MimeBodyPart htmlPart = new MimeBodyPart();
htmlPart.setContent(BODY_HTML, "text/html; charset=UTF-8");

// Add the text and HTML parts to the child container.
msg_body.addBodyPart(textPart);
msg_body.addBodyPart(htmlPart);

// Add the child container to the wrapper object.
wrap.setContent(msg_body);

// Create a multipart/mixed parent container.
MimeMultipart msg = new MimeMultipart("mixed");

// Add the parent container to the message.
message.setContent(msg);

// Add the multipart/alternative part to the message.
msg.addBodyPart(wrap);

// Define the attachment
MimeBodyPart att = new MimeBodyPart();
DataSource fds = new FileDataSource(ATTACHMENT);
att.setDataHandler(new DataHandler(fds));
att.setFileName(fds.getName());

// Add the attachment to the message.
msg.addBodyPart(att);

// Try to send the email.
try {
    System.out.println("Attempting to send an email through Amazon SES "
        +"using the AWS SDK for Java...");

    // Instantiate an Amazon SES client, which will make the service
    // call with the supplied AWS credentials.
    AmazonSimpleEmailService client =
        AmazonSimpleEmailServiceClientBuilder.standard()
        // Replace US_WEST_2 with the AWS Region you're using for
        // Amazon SES.
        .withRegion(Regions.US_WEST_2).build();

    // Print the raw email content on the console
    PrintStream out = System.out;
    message.writeTo(out);
}
```

```
// Send the email.
ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
message.writeTo(outputStream);
RawMessage rawMessage =
    new RawMessage(ByteBuffer.wrap(outputStream.toByteArray()));

SendRawEmailRequest rawEmailRequest =
    new SendRawEmailRequest(rawMessage)
        .withConfigurationSetName(CONFIGURATION_SET);

client.sendRawEmail(rawEmailRequest);
System.out.println("Email sent!");
// Display an error if something goes wrong.
} catch (Exception ex) {
    System.out.println("Email Failed");
    System.err.println("Error message: " + ex.getMessage());
    ex.printStackTrace();
}
}
```

## PHP

次のコード例は、[PHPMailer](#) パッケージと [AWS SDK for PHP](#) を使用して、HTML 部分、テキスト部分、および添付ファイルを含む raw E メールを作成して送信する方法を示しています。

このコード例では [Composer](#) を使用して [PHPMailer](#) パッケージをインストールしていることを前提としています。さらに、AWS SDK for PHP をインストールしていること、および共有認証情報ファイルを作成しているものとしします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2つの環境変数 (`AWS_ACCESS_KEY_ID` と `AWS_SECRET_ACCESS_KEY`) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
<?php
require 'vendor/autoload.php';
use PHPMailer\PHPMailer\PHPMailer;
use Aws\Ses\SesClient;
use Aws\Ses\Exception\SesException;

// Replace sender@example.com with your "From" address.
// This address must be verified with Amazon SES.
$sender = 'sender@example.com';
$sendername = 'Sender Name';

// Replace recipient@example.com with a "To" address. If your account
// is still in the sandbox, this address must be verified.
$recipient = 'recipient@example.com';

// Specify a configuration set.
$configset = 'ConfigSet';

// Replace us-west-2 with the AWS Region you're using for Amazon SES.
$region = 'us-west-2';

$subject = 'List of customers to contact';

$htmlbody = <<<EOD
```

```
<html>
<head></head>
<body>
<h1>Hello!</h1>
<p>Please see the attached file for a list of customers to contact.</p>
</body>
</html>
EOD;

$textbody = <<<EOD
Hello,
Please see the attached file for a list of customers to contact.
EOD;

// The full path to the file that will be attached to the email.
$att = 'path/to/customers-to-contact.xlsx';

// Create an SesClient.
$client = SesClient::factory(array(
    'version' => 'latest',
    'region' => $region
));

// Create a new PHPMailer object.
$mail = new PHPMailer;

// Add components to the email.
$mail->setFrom($sender, $sendername);
$mail->addAddress($recipient);
$mail->Subject = $subject;
$mail->Body = $htmlbody;
$mail->AltBody = $textbody;
$mail->addAttachment($att);
$mail->addCustomHeader('X-SES-CONFIGURATION-SET', $configset);

// Attempt to assemble the above components into a MIME message.
if (!$mail->preSend()) {
    echo $mail->ErrorInfo;
} else {
    // Create a new variable that contains the MIME message.
    $message = $mail->getSentMIMEMessage();
}

// Try to send the message.
try {
    $result = $client->sendRawEmail([
        'RawMessage' => [
            'Data' => $message
        ]
    ]);
    // If the message was sent, show the message ID.
    $messageId = $result->get('MessageId');
    echo("Email sent! Message ID: $messageId."\n");
} catch (SesException $error) {
    // If the message was not sent, show a message explaining what went wrong.
    echo("The email was not sent. Error message: "
        . $error->getAwsErrorMessage()."\n");
}

?>
```

## Python

次のコード例は、Python [email](#) パッケージと [AWS SDK for Python \(Boto\)](#) を使用して、HTML 部分、テキスト部分、および添付ファイルを含む raw E メールを作成して送信する方法を示しています。

このコード例では、AWS SDK for Python (Boto) をインストールしていること、および共有認証情報ファイルを作成しているものとしてします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2 つの環境変数 (AWS\_ACCESS\_KEY\_ID と AWS\_SECRET\_ACCESS\_KEY) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
import os
import boto3
from botocore.exceptions import ClientError
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.application import MIMEApplication

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
SENDER = "Sender Name <sender@example.com>"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
RECIPIENT = "recipient@example.com"

# Specify a configuration set. If you do not want to use a configuration
# set, comment the following variable, and the
# ConfigurationSetName=CONFIGURATION_SET argument below.
CONFIGURATION_SET = "ConfigSet"

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# The subject line for the email.
SUBJECT = "Customer service contact info"

# The full path to the file that will be attached to the email.
ATTACHMENT = "path/to/customers-to-contact.xlsx"

# The email body for recipients with non-HTML email clients.
BODY_TEXT = "Hello,\r\nPlease see the attached file for a list of customers to
contact."

# The HTML body of the email.
BODY_HTML = """\
<html>
<head></head>
<body>
<h1>Hello!</h1>
<p>Please see the attached file for a list of customers to contact.</p>
</body>
</html>
""

# The character encoding for the email.
CHARSET = "utf-8"

# Create a new SES resource and specify a region.
client = boto3.client('ses', region_name=AWS_REGION)

# Create a multipart/mixed parent container.
msg = MIMEMultipart('mixed')
```



```
# Add subject, from and to lines.
msg['Subject'] = SUBJECT
msg['From'] = SENDER
msg['To'] = RECIPIENT

# Create a multipart/alternative child container.
msg_body = MIMEMultipart('alternative')

# Encode the text and HTML content and set the character encoding. This step is
# necessary if you're sending a message with characters outside the ASCII range.
textpart = MIMEText(BODY_TEXT.encode(CHARSET), 'plain', CHARSET)
htmlpart = MIMEText(BODY_HTML.encode(CHARSET), 'html', CHARSET)

# Add the text and HTML parts to the child container.
msg_body.attach(textpart)
msg_body.attach(htmlpart)

# Define the attachment part and encode it using MIMEApplication.
att = MIMEApplication(open(ATTACHMENT, 'rb').read())

# Add a header to tell the email client to treat this part as an attachment,
# and to give the attachment a name.
att.add_header('Content-
Disposition', 'attachment', filename=os.path.basename(ATTACHMENT))

# Attach the multipart/alternative child container to the multipart/mixed
# parent container.
msg.attach(msg_body)

# Add the attachment to the parent container.
msg.attach(att)
#print(msg)
try:
    #Provide the contents of the email.
    response = client.send_raw_email(
        Source=SENDER,
        Destinations=[
            RECIPIENT
        ],
        RawMessage={
            'Data':msg.as_string(),
        },
        ConfigurationSetName=CONFIGURATION_SET
    )
# Display an error if something goes wrong.
except ClientError as e:
    print(e.response['Error']['Message'])
else:
    print("Email sent! Message ID:"),
    print(response['MessageId'])
```

## Ruby

次のコード例は、Ruby [MIME gem](#) と [AWS SDK for Ruby](#) を使用して、HTML 部分、テキスト部分、および添付ファイルを含む raw E メールを作成して送信する方法を示しています。

このコード例では、AWS SDK for Ruby と MIME gem をインストールしていること、および共有認証情報ファイルを作成しているものとします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

### Important

共有認証情報ファイルでは、AWS アクセスキー ID とシークレットアクセスキーを渡します。共有認証情報ファイルを使用しない場合は、2 つの環境変数 (AWS\_ACCESS\_KEY\_ID と

AWS\_SECRET\_ACCESS\_KEY) を設定することで AWS アクセスキー ID とシークレットアクセスキーを指定できます。この例が機能するには、以下のいずれかの方法で認証情報を指定する必要があります。

```
require 'base64' #standard library
require 'aws-sdk' #gem install aws-sdk
require 'mime' #gem install mime

# Replace sender@example.com with your "From" address.
# This address must be verified with Amazon SES.
sender = "sender@example.com"
sendername = "Sender Name"

# Replace recipient@example.com with a "To" address. If your account
# is still in the sandbox, this address must be verified.
recipient = "recipient@example.com"

# Specify a configuration set.
configsetname = "ConfigSet"

# Replace us-west-2 with the AWS Region you're using for Amazon SES.
awsregion = "us-west-2"

# The subject line for the email.
subject = "Customer service contact info"

# The full path to the file that will be attached to the email.
attachment = "path/to/customers-to-contact.xlsx"

# The email body for recipients with non-HTML email clients.
textbody = ""
Hello,
Please see the attached file for a list of customers to contact.
""

# The HTML body of the email.
htmlbody = ""
<html>
<head></head>
<body>
<h1>Hello!</h1>
<p>Please see the attached file for a list of customers to contact.</p>
</body>
</html>
""

# Create a new MIME text object that contains the base64-encoded content of the
# file that will be attached to the message.
file = MIME::Application.new(Base64::encode64(open(attachment, "rb").read))

# Specify that the file is a base64-encoded attachment to ensure that the
# receiving client handles it correctly.
file.transfer_encoding = 'base64'
file.disposition = 'attachment'

# Create a MIME Multipart Mixed object. This object will contain the body of the
# email and the attachment.
msg_mixed = MIME::Multipart::Mixed.new

# Create a MIME Multipart Alternative object. This object will contain both the
# HTML and plain text versions of the email.
msg_body = MIME::Multipart::Alternative.new

# Add the plain text and HTML content to the Multipart Alternative part.
msg_body.add(MIME::Text.new(textbody, 'plain'))
```

```
msg_body.add(MIME::Text.new(htmlbody, 'html'))

# Add the Multipart Alternative part to the Multipart Mixed part.
msg_mixed.add(msg_body)

# Add the attachment to the Multipart Mixed part.
msg_mixed.attach(file, 'filename' => attachment)

# Create a new Mail object that contains the entire Multipart Mixed object.
# This object also contains the message headers.
msg = MIME::Mail.new(msg_mixed)
msg.to = { recipient => nil }
msg.from = { sender => sendername }
msg.subject = subject
msg.headers.set('X-SES-CONFIGURATION-SET', configsetname)

# Create a new SES resource and specify a region
ses = Aws::SES::Client.new(region: awsregion)

# Try to send the email.
begin

  # Provide the contents of the email.
  resp = ses.send_raw_email({
    raw_message: {
      data: msg.to_s
    }
  })
  # If the message was sent, show the message ID.
  puts "Email sent! Message ID: " + resp[0].to_s

# If the message was not sent, show a message explaining what went wrong.
rescue Aws::SES::Errors::ServiceError => error
  puts "Email not sent. Error message: #{error}"

end
```

## 複数の E メールアドレスの検証

別の E メール送信ソリューションから Amazon SES に移行する場合は、すでに E メールを送信するために使用する E メールアドレスのリストが長い場合があります。この例の Python スクリプトは、JSON 形式の E メールアドレスのリストを入力として受け取ります。次の例は、入力ファイルの構造を示しています。

```
[
  {
    "email": "carlos.salazar@example.com"
  },
  {
    "email": "mary.major@example.co.uk"
  },
  {
    "email": "wei.zhang@example.cn"
  }
]
```

次のスクリプトは入力ファイルを読み込み、ファイルに含まれるすべての E メールアドレスの検証を試みます。このコード例では、AWS SDK for Python (Boto) をインストールしていること、および共有認証情報ファイルを作成しているものとします。共有認証情報ファイルの作成の詳細については、「[認証情報の共有ファイルの作成 \(p. 31\)](#)」を参照してください。

```
import json      #Python standard library
import boto3     #sudo pip install boto3
from botocore.exceptions import ClientError

# The full path to the file that contains the identities to be verified.
# The input file must be JSON-formatted. See
# https://docs.aws.amazon.com/ses/latest/DeveloperGuide/sample-code-bulk-verify.html
# for a sample input file.
FILE_INPUT = '/path/to/identities.json'

# If necessary, replace us-west-2 with the AWS Region you're using for Amazon SES.
AWS_REGION = "us-west-2"

# Create a new SES resource specify a region.
client = boto3.client('ses', region_name=AWS_REGION)

# Read the file that contains the identities to be verified.
with open(FILE_INPUT) as data_file:
    data = json.load(data_file)

# Iterate through the array from the input file. Each time an object named
# 'email' is found, run the verify_email_identity operation against the value
# of that object.
for i in data:
    try:
        response = client.verify_email_identity(
            EmailAddress=i['email']
        )
        # Display an error if something goes wrong.
    except ClientError as e:
        print(e.response['Error']['Message'])
    # Otherwise, show the request ID of the verification message.
    else:
        print('Verification email sent to ' + i['email'] + '. Request ID: ' +
              response['ResponseMetadata']['RequestId'])
```

## AWS リージョンで E メールアイデンティティをレプリケートする

複数の AWS リージョンで同じ ID から E メールを送信する場合は、各リージョンの ID を検証する必要があります。この例の Python スクリプトは、別のリージョンですでに検証済みの場合、新しいリージョンの ID を検証するプロセスが簡素化されます。

このスクリプトを使用して、E メールアドレスまたはドメイン ID をレプリケートできます。このスクリプトを使用して E メールアドレスをレプリケートする場合は、Amazon SES によりそのアドレスに送信される E メールにある確認リンクをクリックする必要があります。ドメインの DNS 設定が Route 53 によって管理されている場合にのみ、このスクリプトを使用してドメイン ID をレプリケートできます。

### Note

サンドボックスおよびアカウントのクォータ設定は、各 AWS リージョンに固有のものです。つまり、アカウントが 1 つのリージョンの Amazon SES サンドボックスから削除された場合、他のリージョンでも削除する必要があります。アカウントを Amazon SES サンドボックスから削除する方法については、「[Amazon SES サンドボックス外への移動 \(p. 74\)](#)」を参照してください。Amazon SES Service Quotas の増加の詳細については、「[Amazon SES 送信クォータの引き上げ \(p. 151\)](#)」を参照してください。

```
import argparse
```

```
import json
import logging
from pprint import pprint
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def get_identities(ses_client):
    """
    Gets the identities for the current Region. The Region is specified in the
    Boto3 Amazon SES client object.

    :param ses_client: A Boto3 Amazon SES client.
    :return: The list of email identities and the list of domain identities.
    """
    email_identities = []
    domain_identities = []
    try:
        identity_paginator = ses_client.get_paginator('list_identities')
        identity_iterator = identity_paginator.paginate(
            PaginationConfig={'PageSize': 20})
        for identity_page in identity_iterator:
            for identity in identity_page['Identities']:
                if '@' in identity:
                    email_identities.append(identity)
                else:
                    domain_identities.append(identity)
        logger.info(
            "Found %s email and %s domain identities.", len(email_identities),
            len(domain_identities))
    except ClientError:
        logger.exception("Couldn't get identities.")
        raise
    else:
        return email_identities, domain_identities

def verify_emails(email_list, ses_client):
    """
    Starts verification of a list of email addresses. Verification causes an email
    to be sent to each address. To complete verification, the recipient must follow
    the instructions in the email.

    :param email_list: The list of email addresses to verify.
    :param ses_client: A Boto3 Amazon SES client.
    :return: The list of emails that were successfully submitted for verification.
    """
    verified_emails = []
    for email in email_list:
        try:
            ses_client.verify_email_identity(EmailAddress=email)
            verified_emails.append(email)
            logger.info("Started verification of %s.", email)
        except ClientError:
            logger.warning("Couldn't start verification of %s.", email)
    return verified_emails

def verify_domains(domain_list, ses_client):
    """
    Starts verification for a list of domain identities. This returns a token for
    each domain, which must be registered as a TXT record with the DNS provider for
    the domain.
```

```

:param domain_list: The list of domains to verify.
:param ses_client: A Boto3 Amazon SES client.
:return: The generated domain tokens to use to completed verification.
"""
domain_tokens = {}
for domain in domain_list:
    try:
        response = ses_client.verify_domain_identity(Domain=domain)
        token = response['VerificationToken']
        domain_tokens[domain] = token
        logger.info("Got verification token %s for domain %s.", token, domain)
    except ClientError:
        logger.warning("Couldn't get verification token for domain %s.", domain)
return domain_tokens

def get_hosted_zones(route53_client):
    """
    Gets the Amazon Route 53 hosted zones for the current account.

    :param route53_client: A Boto3 Route 53 client.
    :return: The list of hosted zones.
    """
    zones = []
    try:
        zone_paginator = route53_client.get_paginator('list_hosted_zones')
        zone_iterator = zone_paginator.paginate(PaginationConfig={'PageSize': 20})
        zones = [
            zone for zone_page in zone_iterator for zone in zone_page['HostedZones']]
        logger.info("Found %s hosted zones.", len(zones))
    except ClientError:
        logger.warning("Couldn't get hosted zones.")
    return zones

def find_domain_zone_matches(domains, zones):
    """
    Finds matches between Amazon SES verified domains and Route 53 hosted zones.
    Subdomain matches are taken when found, otherwise root domain matches are taken.

    :param domains: The list of domains to match.
    :param zones: The list of hosted zones to match.
    :return: The set of matched domain-zone pairs. When a match is not found, the
             domain is included in the set with a zone value of None.
    """
    domain_zones = {}
    for domain in domains:
        domain_zones[domain] = None
        # Start at the most specific sub-domain and walk up to the root domain until a
        # zone match is found.
        domain_split = domain.split('.')
        for index in range(0, len(domain_split) - 1):
            sub_domain = '.'.join(domain_split[index:])
            for zone in zones:
                # Normalize the zone name from Route 53 by removing the trailing '..'
                zone_name = zone['Name'][:-1]
                if sub_domain == zone_name:
                    domain_zones[domain] = zone
                    break
            if domain_zones[domain] is not None:
                break
    return domain_zones

def add_route53_verification_record(domain, token, zone, route53_client):
    """

```

Adds a domain verification TXT record to the specified Route 53 hosted zone. When a TXT record already exists in the hosted zone for the specified domain, the existing values are preserved and the new token is added to the list.

```
:param domain: The domain to add.
:param token: The verification token for the domain.
:param zone: The hosted zone where the domain verification record is added.
:param route53_client: A Boto3 Route 53 client.
"""
domain_token_record_set_name = f'_amazonses.{domain}'
record_set_paginator = route53_client.get_paginator(
    'list_resource_record_sets')
record_set_iterator = record_set_paginator.paginate(
    HostedZoneId=zone['Id'], PaginationConfig={'PageSize': 20})
records = []
for record_set_page in record_set_iterator:
    try:
        txt_record_set = next(
            record_set for record_set
            in record_set_page['ResourceRecordSets']
            if record_set['Name'][:-1] == domain_token_record_set_name and
            record_set['Type'] == 'TXT')
        records = txt_record_set['ResourceRecords']
        logger.info(
            "Existing TXT record found in set %s for zone %s.",
            domain_token_record_set_name, zone['Name'])
        break
    except StopIteration:
        pass
records.append({'Value': json.dumps(token)})
changes = [{
    'Action': 'UPSERT',
    'ResourceRecordSet': {
        'Name': domain_token_record_set_name,
        'Type': 'TXT',
        'TTL': 1800,
        'ResourceRecords': records}}]
try:
    route53_client.change_resource_record_sets(
        HostedZoneId=zone['Id'], ChangeBatch={'Changes': changes})
    logger.info(
        "Created or updated the TXT record in set %s for zone %s.",
        domain_token_record_set_name, zone['Name'])
except ClientError as err:
    logger.warning(
        "Got error %s. Couldn't create or update the TXT record for zone %s.",
        err.response['Error']['Code'], zone['Name'])

def generate_dkim_tokens(domain, ses_client):
    """
    Generates DKIM tokens for a domain. These must be added as CNAME records to the
    DNS provider for the domain.

    :param domain: The domain to generate tokens for.
    :param ses_client: A Boto3 Amazon SES client.
    :return: The list of generated DKIM tokens.
    """
    dkim_tokens = []
    try:
        dkim_tokens = ses_client.verify_domain_dkim(Domain=domain)['DkimTokens']
        logger.info("Generated %s DKIM tokens for domain %s.", len(dkim_tokens), domain)
    except ClientError:
        logger.warning("Couldn't generate DKIM tokens for domain %s.", domain)
    return dkim_tokens
```

```

def add_dkim_domain_tokens(hosted_zone, domain, tokens, route53_client):
    """
    Adds DKIM domain token CNAME records to a Route 53 hosted zone.

    :param hosted_zone: The hosted zone where the records are added.
    :param domain: The domain to add.
    :param tokens: The DKIM tokens for the domain to add.
    :param route53_client: A Boto3 Route 53 client.
    """
    try:
        changes = [{
            'Action': 'UPSERT',
            'ResourceRecordSet': {
                'Name': f'{token}._domainkey.{domain}',
                'Type': 'CNAME',
                'TTL': 1800,
                'ResourceRecords': [{'Value': f'{token}.dkim.amazonses.com'}]
            } for token in tokens]
        route53_client.change_resource_record_sets(
            HostedZoneId=hosted_zone['Id'], ChangeBatch={'Changes': changes})
        logger.info(
            "Added %s DKIM CNAME records to %s in zone %s.", len(tokens),
            domain, hosted_zone['Name'])
    except ClientError:
        logger.warning(
            "Couldn't add DKIM CNAME records for %s to zone %s.", domain,
            hosted_zone['Name'])

def configure_sns_topics(identity, topics, ses_client):
    """
    Configures Amazon Simple Notification Service (Amazon SNS) notifications for
    an identity. The Amazon SNS topics must already exist.

    :param identity: The identity to configure.
    :param topics: The list of topics to configure. The choices are Bounce, Delivery,
        or Complaint.
    :param ses_client: A Boto3 Amazon SES client.
    """
    for topic in topics:
        topic_arn = input(
            f"Enter the Amazon Resource Name (ARN) of the {topic} topic or press "
            f"Enter to skip: ")
        if topic_arn != '':
            try:
                ses_client.set_identity_notification_topic(
                    Identity=identity, NotificationType=topic, SnsTopic=topic_arn)
                logger.info("Configured %s for %s notifications.", identity, topic)
            except ClientError:
                logger.warning(
                    "Couldn't configure %s for %s notifications.", identity, topic)

def replicate(source_client, destination_client, route53_client):
    logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')

    print('-'*88)
    print(f"Replicating Amazon SES identities and other configuration from "
          f"{source_client.meta.region_name} to {destination_client.meta.region_name}.")
    print('-'*88)

    print(f"Retrieving identities from {source_client.meta.region_name}.")
    source_emails, source_domains = get_identities(source_client)
    print("Email addresses found:")
    print(*source_emails)

```



```

print("Domains found:")
print(*source_domains)

print("Starting verification for email identities.")
dest_emails = verify_emails(source_emails, destination_client)
print("Getting domain tokens for domain identities.")
dest_domain_tokens = verify_domains(source_domains, destination_client)

# Get Route 53 hosted zones and match them with Amazon SES domains.
answer = input(
    "Is the DNS configuration for your domains managed by Amazon Route 53 (y/n)? ")
use_route53 = answer.lower() == 'y'
hosted_zones = get_hosted_zones(route53_client) if use_route53 else []
if use_route53:
    print("Adding or updating Route 53 TXT records for your domains.")
    domain_zones = find_domain_zone_matches(dest_domain_tokens.keys(), hosted_zones)
    for domain in domain_zones:
        add_route53_verification_record(
            domain, dest_domain_tokens[domain], domain_zones[domain],
            route53_client)
else:
    print("Use these verification tokens to create TXT records through your DNS "
          "provider:")
    pprint(dest_domain_tokens)

answer = input("Do you want to configure DKIM signing for your identities (y/n)? ")
if answer.lower() == 'y':
    # Build a set of unique domains from email and domain identities.
    domains = {email.split('@')[1] for email in dest_emails}
    domains.update(dest_domain_tokens)
    domain_zones = find_domain_zone_matches(domains, hosted_zones)
    for domain, zone in domain_zones.items():
        answer = input(
            f"Do you want to configure DKIM signing for {domain} (y/n)? ")
        if answer.lower() == 'y':
            dkim_tokens = generate_dkim_tokens(domain, destination_client)
            if use_route53 and zone is not None:
                add_dkim_domain_tokens(zone, domain, dkim_tokens, route53_client)
            else:
                print(
                    "Add the following DKIM tokens as CNAME records through your "
                    "DNS provider:")
                print(*dkim_tokens, sep='\n')

answer = input(
    "Do you want to configure Amazon SNS notifications for your identities (y/n)? ")
if answer.lower() == 'y':
    for identity in dest_emails + list(dest_domain_tokens.keys()):
        answer = input(
            f"Do you want to configure Amazon SNS topics for {identity} (y/n)? ")
        if answer.lower() == 'y':
            configure_sns_topics(
                identity, ['Bounce', 'Delivery', 'Complaint'], destination_client)

print(f"Replication complete for {destination_client.meta.region_name}.")
print('-'*88)

def main():
    boto3_session = boto3.Session()
    ses_regions = boto3_session.get_available_regions('ses')
    parser = argparse.ArgumentParser(
        description="Copies email address and domain identities from one AWS Region to "
        "another. Optionally adds records for domain verification and DKIM "
        "signing to domains that are managed by Amazon Route 53, "
        "and sets up Amazon SNS notifications for events of interest.")

```

```
parser.add_argument(
    'source_region', choices=ses_regions, help="The region to copy from.")
parser.add_argument(
    'destination_region', choices=ses_regions, help="The region to copy to.")
args = parser.parse_args()
source_client = boto3.client('ses', region_name=args.source_region)
destination_client = boto3.client('ses', region_name=args.destination_region)
route53_client = boto3.client('route53')
replicate(source_client, destination_client, route53_client)

if __name__ == '__main__':
    main()
```

# リージョンと Amazon SES

Amazon SES は、世界中のいくつかの AWS リージョンで利用できます。各リージョンで、AWS は複数のアベイラビリティゾーンを維持しています。これらのアベイラビリティゾーンは物理的に相互に分離されていますが、低レイテンシーで高スループットの冗長性に優れたプライベートネットワーク接続で統合されています。これらのアベイラビリティゾーンにより、非常に高いレベルの可用性と冗長性を提供しながら、レイテンシーを最小限に抑えています。

Amazon SES リージョンのすべてのエンドポイントのリストについては、アマゾン ウェブ サービス全般リファレンスの「[Amazon Simple Email サービスエンドポイントとクォータ](#)」を参照してください。各リージョンで利用できるアベイラビリティゾーンの数の詳細については、「[AWS グローバルインフラストラクチャ](#)」を参照してください。

このセクションには、複数の AWS リージョンで Amazon SES を使用する場合に知っている必要がある情報が含まれています。必要事項は以下のとおりです。

- [Amazon SES のリージョンとエンドポイント \(p. 467\)](#)
- [サンドボックスのステータスと送信制限の引き上げ \(p. 468\)](#)
- [E メールアドレスおよびドメインの検証 \(p. 468\)](#)
- [Easy DKIM \(p. 468\)](#)
- [サブプレッションリスト \(p. 468\)](#)
- [フィードバック通知 \(p. 468\)](#)
- [SMTP 認証情報 \(p. 469\)](#)
- [送信認可 \(p. 470\)](#)
- [カスタムの MAIL FROM ドメイン \(p. 469\)](#)
- [Eメールの受信 \(p. 470\)](#)

AWS リージョンの概要については、[AWS 全般リファレンス](#)の「[AWS リージョンとエンドポイント](#)」を参照してください。

## Amazon SES のリージョンとエンドポイント

Amazon Simple Email Service (Amazon SES) を使用して E メールを送信する場合は、Amazon SES API または SMTP インターフェイス用のエンドポイントを提供する URL に接続します。AWS 全般リファレンスには、Amazon SES 経由でメールを送受信するために使用するエンドポイントの完全なリストが含まれています。詳細については、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。

Amazon SES 経由で E メールを送信する際に、API (HTTPS) エンドポイント列の URL を使用して SES API に対して HTTPS リクエストを行うことができます。また、SMTP エンドポイント列の URL を使用して、SMTP インターフェイスを使用して E メールを送信することもできます。

ドメインに送信された E メールを受信するように Amazon SES を設定している場合は、[ドメインの DNS 設定でメールエクスチェンジャ \(MX\) レコードをセットアップする \(p. 220\)](#)際に、インバウンド SMTP エンドポイント URL (「inbound-smtp」で始まる URL) を使用できます。

### Note

インバウンド SMTP URL は、IMAP サーバーアドレスではありません。つまり、Outlook などのアプリケーションを使用した Eメールの受信に使用することはできません。受信メール用の IMAP サーバーを提供するサービスについては、「[Amazon WorkMail](#)」を参照してください。

## サンドボックスのステータスと送信制限の引き上げ

アカウントのサンドボックスのステータスは AWS リージョン間で異なる状態にできます。つまり、アカウントが 米国西部 (オレゴン) リージョンのサンドボックスから削除されても、米国東部 (バージニア北部) リージョンのサンドボックスには、そのリージョンのサンドボックスからも削除しない限り残っています。

送信制限も、AWS リージョンごとに別のものにすることができます。例えば、ヨーロッパ (アイルランド) リージョンではアカウントが 1 秒あたり 10 件のメッセージを送信できる場合に、別のリージョンでは送信できるメッセージをそれよりも多くすることも少なくすることもできます。

サンドボックスからアカウントを削除するリクエストを送信する場合 (p. 74)、またはアカウントの送信クォータを増やすリクエストを送信する場合 (p. 152)、リクエストが適用されるすべての AWS リージョンを選択していることを確認してください。単一のサポートセンターのケースで複数のリクエストを送信できます。

## E メールアドレスおよびドメインの検証

Amazon SES を使用してメールを送信する前に、Eメールの送信元となるアドレスまたはドメインを所有していることを検証する必要があります。Eメールアドレスおよびドメインの検証ステータスも、AWS リージョン間で異なります。たとえば、米国西部 (オレゴン) リージョンでドメインを検証する場合、米国東部 (バージニア北部) リージョンでドメインの検証プロセスを再び完了するまで、そのリージョンではそのドメインを Eメールの送信に使用することはできません。Eメールアドレスとドメインの検証については、「[Amazon SES の ID の検証 \(p. 48\)](#)」を参照してください。

## Easy DKIM

Easy DKIM を使用する各リージョンで、Easy DKIM のセットアッププロセスを実行する必要があります。つまり、各リージョンで、Amazon SES コンソールまたは Amazon SES API を使用して TXT レコードを生成する必要があります。次に、すべての TXT レコードを、ドメインの DNS 設定に追加する必要があります。Easy DKIM のセットアップの詳細については、「[Amazon SES の Easy DKIM \(p. 135\)](#)」を参照してください。

## サブレクションリスト

サブレクションリストはリージョンごとに別個に用意されますが、1つのリージョンのグローバルサブレクションリストからアドレスを削除すると、Amazon SES はすべてのリージョンのグローバルサブレクションリストからそのアドレスを削除します。サブレクションリストからアドレスを削除するには、Amazon SES コンソールを使用します。サブレクションリストからアドレスを削除する方法の詳細については、「[Amazon SES グローバルサブレクションリストの使用 \(p. 199\)](#)」を参照してください。

## フィードバック通知

複数のリージョンでのフィードバック通知のセットアップについては、重要な点が 2 つあります。

- フィードバックを Eメールで受信するか Amazon Simple Notification Service (Amazon SNS)を通じて受信するかなど、検証されたアイデンティティ設定は、それを設定したリージョンのみに適用されます。たとえば、user@example.com を 米国西部 (オレゴン) および 米国東部 (バージニア北部) リージョンで検証し、返送されるバウンスメールを Amazon SNS 通知によって受信する場合、Amazon SNS API まで

または Amazon SES コンソールを使用して、user@example.com の Amazon SNS フィードバック通知を両方のリージョンで設定する必要があります。

- フィードバック転送に使用する Amazon SNS トピックは、Amazon SES を使用するのと同じリージョンにある必要があります。

## SMTP 認証情報

Amazon SES SMTP インターフェイスを介して E メールを送信するために使用する認証情報は、各 AWS リージョンに固有のもので、Amazon SES SMTP インターフェイスを使用して複数のリージョンに E メールを送信する場合は、各リージョンで [SMTP 認証情報のセットを生成する \(p. 89\)](#) 必要があります。

### Note

2019 年 1 月 10 日以前に SMTP 認証情報を作成した場合、古いバージョンの AWS 署名を使用して SMTP 認証情報が作成されました。セキュリティのため、この日付よりも前に作成した認証情報を削除して、新しい認証情報に置き換える必要があります。[古い認証情報は、IAM コンソールを使用して削除](#)できます。

## カスタムの MAIL FROM ドメイン

異なる複数の AWS リージョン内の検証済みのアイデンティティに対して、同じカスタムの MAIL FROM ドメインを使用することができます。このように使用する場合も、MAIL FROM ドメインの DNS サーバーに MX レコードを 1 件のみ公開する必要があります。このような状況では、バウンス通知は MX レコードで最初に指定されたリージョンの Amazon SES フィードバックエンドポイントに送信されます。次に、Amazon SES はバウンスを Eメールの送信元のリージョン内の検証済みアイデンティティにリダイレクトします。

いずれかのリージョン内のアイデンティティに対して、カスタムの MAIL FROM の設定プロセス中に Amazon SES から提供される MX レコード設定を使用します。カスタム MAIL FROM のセットアッププロセスについては、「[カスタムの MAIL FROM ドメインの設定 \(p. 68\)](#)」を参照してください。参照用として、すべてのリージョンのフィードバックエンドポイントを次の表に示します。

### Note

SMTP エンドポイントは現在、中東 (バーレーン) では使用できません。

リージョン名	カスタム MAIL FROM 送信設定のフィードバックエンドポイント
米国東部 (オハイオ)	feedback-smtp.us-east-2.amazonses.com
米国東部 (バージニア北部)	feedback-smtp.us-east-1.amazonses.com
米国西部 (北カリフォルニア)	feedback-smtp.us-west-1.amazonses.com
米国西部 (オレゴン)	feedback-smtp.us-west-2.amazonses.com
アジアパシフィック (ムンバイ)	feedback-smtp.ap-south-1.amazonses.com
アジアパシフィック (ソウル)	feedback-smtp.ap-northeast-2.amazonses.com
アジアパシフィック (シンガポール)	feedback-smtp.ap-southeast-1.amazonses.com
アジアパシフィック (シドニー)	feedback-smtp.ap-southeast-2.amazonses.com
アジアパシフィック (東京)	feedback-smtp.ap-northeast-1.amazonses.com

リージョン名	カスタム MAIL FROM 送信設定のフィードバックエンドポイント
カナダ (中部)	feedback-smtp.ca-central-1.amazonses.com
欧州 (フランクフルト)	feedback-smtp.eu-central-1.amazonses.com
欧州 (アイルランド)	feedback-smtp.eu-west-1.amazonses.com
欧州 (ロンドン)	feedback-smtp.eu-west-2.amazonses.com
欧州 (パリ)	feedback-smtp.eu-west-3.amazonses.com
欧州 (ストックホルム)	feedback-smtp.eu-north-1.amazonses.com
南米 (サンパウロ)	feedback-smtp.sa-east-1.amazonses.com
AWS GovCloud (US)	feedback-smtp.us-gov-west-1.amazonses.com

## 送信認可

代理送信者は、アイデンティティ所有者のアイデンティティが検証された AWS リージョンからのみ、Eメールを送信できます。代理送信者にアクセス許可を付与する承認の送信ポリシーは、そのリージョンのアイデンティティにアタッチする必要があります。送信承認の詳細については、「[Amazon SES での送信承認の使用 \(p. 154\)](#)」を参照してください。

## Eメールの受信

Amazon S3 バケットを除き、Amazon SES で Eメールの受信に使用するすべての AWS リソースは、Amazon SES エンドポイントと同じ AWS リージョンに存在する必要があります。たとえば、米国西部 (オレゴン) リージョンで Amazon SES を使用する場合は、使用する Amazon SNS トピック、AWS KMS キー、Lambda 関数も米国西部 (オレゴン) リージョン内に存在する必要があります。同様に、リージョン内で Amazon SES により Eメールを受信するには、そのリージョン内にアクティブな受信ルールセットを作成する必要があります。

次の表は、Eメールの受信が Amazon SES でサポートされているすべての AWS リージョンでの Eメール受信エンドポイントの一覧です。

リージョン名	Eメール受信エンドポイント
米国東部 (バージニア北部)	inbound-smtp.us-east-1.amazonaws.com
米国西部 (オレゴン)	inbound-smtp.us-west-2.amazonaws.com
欧州 (アイルランド)	inbound-smtp.eu-west-1.amazonaws.com

### Note

Amazon SES は、米国東部 (オハイオ)、米国西部 (北カリフォルニア)、アジアパシフィック (ムンバイ)、アジアパシフィック (ソウル)、アジアパシフィック (シンガポール)、アジアパシフィック (シドニー)、アジアパシフィック (東京)、カナダ (中部)、欧州 (フランクフルト)、欧州 (ロンドン)、欧州 (パリ)、欧州 (ストックホルム)、中東 (バーレーン)、南米 (サンパウロ)、および AWS GovCloud (US) の各リージョンでの Eメール受信をサポートしていません。

# Amazon SES の Service Quotas

以下のセクションでは、Amazon SES のリソースとオペレーションに適用されるクォータについて、一覧を示して説明します。クォータによっては、引き上げることができないものもあります。クォータの引き上げをリクエストできるかどうかを判断するには、各セクションの「増加の対象」列を参照してください。

## E メール送信クォータ

以下のクォータは、Amazon SES 経由の E メール送信に適用されます。

### 送信クォータ

#### Note

クォータは、メッセージ数ではなく、受信者の数に基づきます。

リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
24 時間ごとに送信可能な E メール数	アカウントがサンドボックスにある場合は、24 時間あたり最大 200 通の E メールを送信できます。  アカウントがサンドボックスの外にある場合、この数字は具体的なユースケースによって異なります。  Note  この値は過去に「送信クォータ」と呼ばれていました。	はい (p. 151)
1 秒あたりに送信できる E メール数 (送信レート)	アカウントがサンドボックスにある場合、1 秒あたり 1 通の E メールを送信できます。  アカウントがサンドボックスの外にある場合、このレートは具体的なユースケースによって異なります。	はい (p. 151)

### メッセージのクォータ

リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
最大メッセージサイズ (添付ファイルを含む)	メッセージあたり 10 MB (base64 エンコーディング後)	いいえ

## 送信者と受信者のクォータ

リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
メッセージあたりの受取人の最大数	メッセージあたり 50 人の受取人。  Note  受取人は「To」、「CC」、または「BCC」アドレスのいずれかです。	いいえ
検証できるアイデンティティの最大数	AWS リージョンあたり 10,000 ID。  Note  ID とは、Amazon SES 経由で E メールを送信するために使用するドメインまたは E メールアドレスです。	いいえ

## イベント発行に関連するクォータ

リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
設定セットの最大数	10,000	いいえ
設定セット名の最大長	設定名には、最大で 64 文字までの英数字を含めることができます。この名前には、ハイフン (-) とアンダースコア (_) も使用できます。名前にスペース、アクセント文字、またはその他の特殊文字を含めることはできません。	いいえ
設定セットあたりのイベント送信先の最大数	10	いいえ
CloudWatch イベント送信先あたりのディメンションの最大数	10	いいえ

## E メールテンプレートのクォータ

リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
各 AWS リージョンの E メールテンプレートの最大数	10,000	いいえ



リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
最大テンプレートサイズ	500 KB	いいえ
各テンプレートの置換値の最大数	無制限	該当なし
テンプレート化された各 E メール の受信者の最大数	送信先 50 件。送信先は、 「To」、「CC」、「BCC」行に あるすべての E メールアドレス です。  Note  1 回の API の呼び出しで 連絡できる送信先がアカ ウントの最大送信レート によって制限される場合 があります。	いいえ

## Eメールの受信に関連するクォータ

次の表に、Amazon SES による Eメールの受信に関連するクォータを示します。

リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
受信ルールセットあたりのルールの最大数	200	いいえ
受信ルールあたりのアクションの最大数	10	いいえ
受信ルールあたりの受取人の最大数	100	いいえ
AWS アカウントあたりの受信ルールセットの最大数	40	いいえ
AWS アカウントあたりの IP アドレスフィルターの最大数	100	いいえ
Amazon S3 バケットに保存できる Eメールの最大サイズ (ヘッダーを含む)	30 MB	いいえ
Amazon SNS 通知を使用して公開できる Eメールの最大サイズ (ヘッダーを含む)	150 KB	いいえ

## 一般的なクォータ

次の表は、Amazon SES 経由の Eメールの送受信に適用されるクォータの一覧です。

## Amazon SES API のクォータ

リソース	デフォルトのクォータ	引き上げの対象かどうかの確認
Amazon SES API アクションを呼び出すことができるレート	すべてのアクション (SendEmail および SendRawEmail を除く) は、1 秒あたり 1 リクエストに絞られます。Amazon SES API の詳細については、「 <a href="#">Amazon Simple Email Service API リファレンス</a> 」を参照してください。	いいえ

# Amazon SES を使用した E メール送信のベストプラクティス

お客様とのメール通信を管理する方法は、メールプログラムと呼ばれます。メールプログラムの成否を分けるいくつかの要因があります。これらの要因に対しては、当初、複雑さや戸惑いを感じることもあるかもしれませんが、メールの配信方法を理解し、特定のベストプラクティスに従うことで、受取人の受信トレイにメールが正常に届くようになります。

## トピック

- [E メールプログラムの成功のメトリクス \(p. 475\)](#)
- [ヒントとベストプラクティス \(p. 477\)](#)

## E メールプログラムの成功のメトリクス

メールプログラムの成功度は、いくつかのメトリクスを使って測定することができます。

このセクションでは、以下のメトリクスについて説明します。

- [Bounces \(p. 475\)](#)
- [Complaints \(p. 476\)](#)
- [メッセージの品質 \(p. 477\)](#)

## Bounces

メールが目的の受取人に届かないと、バウンスが発生します。ハードバウンスとソフトバウンスの 2 種類があります。ハードバウンスは、メールの不達の原因が、E メールアドレスが存在しないなどの永続的な問題である場合に発生します。ソフトバウンスは、メールの不達の原因が一時的な問題である場合に発生します。たとえば、受信者の受信トレイが満杯である場合や受信サーバーが一時的に使用できない場合にソフトバウンスが発生します。Amazon SES では、ソフトバウンスの対処方法として、一定期間、メールの再配信を試行します。

メールプログラムでハードバウンス数をモニタリングすること、および受信者リストからハードバウンスの原因である E メールアドレスを削除することが重要です。E メールレシーバーは、ハードバウンス率が高いことを検出すると、送信者が受取人をよく知らないものとみなします。その結果、高いハードバウンス率によりメールメッセージの配信性能が低下する場合があります。

バウンスをなくし、送信者の評価を向上させるために、以下のガイドラインを参考にしてください。

- ハードバウンス率は 5% 未満に抑える。メールプログラムのハードバウンス数が少ないほど、ISP がメッセージを正当で重要なものと判断する確率が高まります。このレートは妥当で十分に達成可能な目標ですが、すべての ISP に採用されている統一ルールというわけではありません。
- メールリストのレンタルや購入は絶対に避ける。レンタルや購入したリストには、大量の無効なアドレスが含まれている場合があります。ハードバウンス率が大幅に増える原因となります。さらに、これらのリストには、スパムトラップが含まれている可能性があります。それは違法な送信者を捕まえるために限定的に使用される E メールアドレスです。メッセージがスパムトラップに陥ると、配信率と送信者の評価は決定的に損なわれる場合があります。
- リストを最新状態に保つ。受取人に久しぶりにメールを送る場合は、他の手段 (ウェブサイトのログインアクティビティや購入履歴など) で受取人のステータスを確認します。

- お客様のステータスを確認する方法がない場合は、ウインバックメールの送信を検討する。一般的なウインバックメールでは、お客様からの連絡が途絶えていることをお客様に伝え、引き続きメールの受信を望む場合は確認の返信をしてもらいます。ウインバックメールの送信後に、返信がなかったすべての受取人をリストから消去します。

バウンスを受信したら、以下のルールに従って適切に対応する必要があります。

- ハードバウンスの原因がメールアドレスである場合は、そのアドレスをリストから即座に削除します。ハードバウンスしたアドレスにメッセージを再送信しないでください。ハードバウンスの繰り返しは加算されて、最終的に受取人の ISP から悪い評価を受けることになります。
- バウンス通知の受信に使用するアドレスで E メールを受信できることを確認します。バウンスと苦情通知の設定の詳細については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。
- 受信メールが独自の社内サーバーではなく ISP から届いている場合は、大量のバウンス通知がスパムフォルダに入るか、完全に廃棄される可能性があります。バウンスの受信にはホステッド型の E メールアドレスを使用しないのが理想的です。それでも使用する必要がある場合は、スパムフォルダを頻繁に確認してください。また、バウンスメッセージをスパムとして報告しないでください。Amazon SES で、バウンス通知の送信先のアドレスを指定できます。
- 通常、バウンスでは配信を拒否しているメールボックスのアドレスが提供されます。ただし、受取人のアドレスを特定の E メールキャンペーンにマッピングするための詳細なデータが必要な場合は、自社の追跡システムまでたどれるように X ヘッダーに適切な値を含めます。詳細については、「[ヘッダーフィールド \(p. 523\)](#)」を参照してください。

## Complaints

メールの受取人がウェブベースの E メールクライアントで [スパムとしてマーク] (または同等) のボタンをクリックすると、苦情が発生します。これらの苦情が大量に累積すると、ISP からスパムの送信元とみなされます。これにより、配信性能と送信者の評価が低下します。いくつか、全てではありませんが、苦情が報告されたときに通知する ISP があります。これはフィードバックループとして知られています。Amazon SES は、フィードバックループを提供する ISP からの苦情を自動的に転送します。

苦情をなくし、送信者の評価を向上させるために、以下のガイドラインを参考にしてください。

- 苦情率は 0.1% 未満に抑える。メールプログラムの苦情数が少ないほど、ISP がメッセージを正当で重要なものと判断する確率が高まります。このレートは妥当で十分に達成可能な目標ですが、すべての ISP に採用されている統一ルールというわけではありません。
- マーケティングメールに関してお客様から苦情が寄せられた場合は、そのお客様に対するマーケティングメールの送信を即座に停止します。ただし、メールプログラムに他のタイプの E メール (通知メールやトランザクションメールなど) が含まれている場合、これらのタイプのメッセージは引き続き苦情元の受取人に送信することが許容されます。
- ハードバウンスと同様に、久しぶりにメールを送信する受取人のリストがある場合は、メッセージを送信する理由を確実に受取人に伝えます。受取人に送信者を思い出してもらうためと、連絡する理由を知らせるための「ようこそ」メッセージを送信することをお勧めします。

苦情を受けたら、以下のルールに従って適切に対応する必要があります。

- 苦情通知の受信に使用するアドレスで E メールを受信できることを確認します。バウンスと苦情通知の設定の詳細については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。
- 苦情通知が ISP またはメールシステムによってスパムとして報告されていないことを確認します。
- 通常、苦情通知には E メール本文が含まれます。この点は、Eメールのヘッダーのみが含まれるバウンス通知とは異なります。ただし、苦情通知では、苦情元の個人の E メールアドレスは削除されます。苦情元の E メールアドレスを特定できるように E メール本文に X ヘッダーや特別な識別子を埋め込む

ことをお勧めします。この方法を使用すると、苦情元のアドレスをより簡単に識別して、受信者リストから削除できるようになります。

## メッセージの品質

E メールレシーバーは、メッセージの真偽を見分けるために、コンテンツフィルタを使用してメッセージ内の特定の属性を検出します。これらのコンテンツフィルタは、メッセージの内容を自動的に確認し、悪意あるメッセージの一般的な特性を識別します。Amazon SES では、コンテンツのフィルタリングテクノロジーを使用し、スパムやマルウェアを含むメッセージを送信前に検出してブロックします。

E メールレシーバーのコンテンツフィルタで、メッセージ内にスパムや悪意のある E メールの特徴が検出されると、フラグが付けられて受取人の受信トレイから除外されます。

E メールを設計するときは、以下の点に留意してください。

- 最近のコンテンツフィルタはインテリジェントであり、絶えず適応し、変わり続けています。定義済みのルールのセットには依存しません。[ReturnPath](#) や [Litmus](#) などのサードパーティーのサービスを使用すると、どのような内容の E メールがコンテンツフィルタをトリガーする可能性があるかを確認できます。
- E メールにリンクが含まれている場合は、それらの URL が [URIBL.com](#) や [SURBL.org](#) などにある DNS ブラックリストに登録されていないかチェックします。
- リンク短縮ツールの使用を避けます。悪意のある送信者は、リンク短縮ツールを使用して実際のリンク先を隠す場合があります。ISP は、不正な目的でリンク短縮サービス - 最も評判の高いものであっても - が使用されていると判断すると、これらのサービス全体へのアクセスを拒否する場合があります。拒否リストに登録されたリンク短縮サービスへのリンクが含まれている E メールは、お客様の受信トレイに届かず、E メールキャンペーンの成功は危くなります。
- E メール内のすべてのリンクをテストし、目的のページを参照することを確認します。
- ウェブサイトにプライバシーポリシーと利用規約のドキュメントが含まれていること、これらのドキュメントが最新の内容であることを確認します。これらのドキュメントへのリンクは、送信する E メールごとに提供するのが適切です。これらのドキュメントへのリンクを提供することで、顧客に対する隠し事がないことを示し、信頼関係を築くのに役立ちます。
- 頻繁に変わるコンテンツ（「今日のお買い得情報」メッセージなど）を送信する場合は、E メールの内容がデプロイごとに異なることを確認します。頻繁に内容が変わるメッセージを送信する場合は、これらのメッセージがタイムリーで関連性が高いことを確認し、同じ繰り返しで煩わしさを感じさせないようにします。

## ヒントとベストプラクティス

お客様の利益を最優先する姿勢でも、依然として、メッセージの配信性能に影響するような状況が発生する場合があります。以下のセクションでは、E メールメッセージを目的のユーザーに確実に届けるための推奨事項について説明します。

### 全般的な推奨事項

- お客様の立場で考えます。送信するメッセージが自分の受信トレイで受け取りたい内容であるかどうか自問します。その答えがどんなものでも「Yes!」以外であれば、おそらく送るべきではありません。
- 一部の業界は、メールメッセージの内容が低品質または悪意があるとさえ評価されています。以下の業界の関係者は、自社の評価を厳密にモニタリングし、早急に問題を解決する必要があります。
  - 住宅担保ローン
  - クレジット
  - 薬品およびサプリメント

- アルコールおよびタバコ
- アダルト向けエンターテインメント
- カジノおよびギャンブル
- 在宅就業プログラム

## ドメインおよび「From」アドレスに関する考慮事項

- Eメールの差出人のアドレスについてよく考えます。「From」アドレスは、受取人が最初に接する情報の1つであり、第一印象は長く残る場合があります。さらに、「From」アドレスと評価の関連付けを行うISPもあります。
- 通信のタイプ別に異なるサブドメインを使用することを検討します。たとえば、ドメイン example.com を使用してマーケティングメッセージとトランザクションメッセージの両方を送信する場合を考えます。この場合、すべてのメッセージを example.com から送信するのではなく、マーケティングメッセージは marketing.example.com などのサブドメインから、トランザクションメッセージは orders.example.com などのサブドメインから送信します。評価はサブドメイン別に確立されます。たとえば、マーケティングメッセージがスパムトラップにかかったり、コンテンツフィルタをトリガーしたりした場合、サブドメインを使用していると、自社の評価が損なわれるリスクを軽減できます。
- 大量のメッセージを送信する場合は、sender@hotmail.com などの ISP ベースのアドレスから送信しません。sender@hotmail.com から大量のメッセージが送信されていることをISPが気付くと、そのEメールは自社所有の送信メール用ドメインから送信されるEメールとは別の方法で処理されます。
- ドメインレジストラと協力してドメインのWHOIS情報が正しいことを確認します。正直かつ最新のWHOISレコードを維持することは、透過性を重要視していることを示します。また、ユーザーはドメインの正当性を簡単に確認できます。
- 「From」アドレスまたは「Reply-to」アドレスとして no-reply@example.com などの no-reply アドレスを使用しないようにします。no-reply@ メールアドレスを使用すると、受取人に返信する方法を封じていること、受取人のフィードバックに関心がないことが明確なメッセージとして伝わります。

## Authentication

- ドメインを [SPF \(p. 134\)](#) および SenderID で認証します。これらの認証方法により、各Eメールの送信元のドメインが真正のものであることを受取人が確認できます。
- 送信メールに [DKIM \(p. 135\)](#) で署名します。このステップにより、コンテンツが送受信の当事者間での転送中に変更されていないことを受取人が確認できます。
- SPF と DKIM の両方の認証設定をテストするには、個人の Gmail アカウントや Hotmail アカウントなどの ISP ベースの個人のメールアドレスにEメールを送信し、メッセージのヘッダーを確認します。ヘッダーを見て、メッセージの認証と署名が成功したかどうかわかります。

## リストの構築とメンテナンス

- ダブルオプトイン戦略を実装します。ユーザーがサインアップしてEメールを受信する場合は、確認リンクを記載したメッセージを送信し、ユーザーがそのリンクをクリックしてアドレスを確認するまではEメールの送信を開始しません。ダブルオプトイン戦略は、入力ミスによるハードバウンスの数を減らすのに役立ちます。
- ウェブベース形式でメールアドレスを集める場合は、収集時にそれらのアドレスに対して最小限の検証を実施します。たとえば、収集したアドレスの形式が正しいこと (recipient@example.com の形式になっていること)、および有効な MX レコードのドメインを参照していることを確認します。
- ユーザー定義の入力を無検査で Amazon SES に渡すことを許可するときは注意が必要です。フォーラムの登録やフォームの提出には、それなりのリスクがあります。コンテンツ全体をユーザーが生成し、スパム送信者が独自の内容をフォームに記入する可能性があるためです。高品質のコンテンツのEメールのみを送信することは、お客様の責任です。

- 標準エイリアス (postmaster@、abuse@、noc@ など) が意図的に E メールにサインアップするようなことは考えられません。メッセージは、実際に希望する相手にものみ送信します。このルールは、Eメールのウォッチドッグとして通例予約されている標準エイリアスに特に該当します。これらのエイリアスは、評価を損なう目的で一種の悪意のある破壊活動としてリストに追加される場合があります。

## Compliance

- Eメールの送信先の国や地域で適用される Eメールマーケティングとスパム対策に関する法律および規制に注意してください。送信する Eメールがこれらの法律に準拠していることを確認するのは、お客様の責任です。このガイドではこれらの法律については触れていませんので、お客様側で調べるのが重要です。法律のリストについては、Wikipedia の「[Email Spam Legislation by Country](#)」を参照してください。
- 法的なアドバイスを受けるには、弁護士に相談してください。

# Amazon SES のトラブルシューティング

このセクションは、問題が発生した場合に役に立つ次のトピックから構成されます。

- ドメインの確認時に発生する可能性がある問題の詳細については、「[E メールアドレスおよびドメイン認証の問題 \(p. 481\)](#)」を参照してください。
- DKIM 関連の問題に対する解決策については、「[Amazon SES の DKIM 問題のトラブルシューティング \(p. 483\)](#)」を参照してください。
- E メール送信時に発生する可能性がある一般的な配信に関連する問題のリストとその対処方法については、「[Amazon SES 配信の問題 \(p. 484\)](#)」を参照してください。
- Amazon SES から送信された Eメールの受信者側での表示に関する問題については、「[Amazon SES から受け取った Eメールに関する問題 \(p. 485\)](#)」を参照してください。
- バウンス、苦情、および配信通知に関する問題の解決方法については、「[Amazon SES 通知の問題 \(p. 486\)](#)」を参照してください。
- Amazon SES で Eメールを送信するときに発生する可能性があるエラーのリストについては、「[Amazon SES の Eメール送信エラー \(p. 486\)](#)」を参照してください。
- API または SMTP インターフェイスを使用して Amazon SES への複数の呼び出しを行うときに Eメール送信速度を向上させるためのヒントについては、「[Amazon SES のスループットを上げる \(p. 488\)](#)」を参照してください。
- Simple Mail Transfer Protocol (SMTP、簡易メール転送プロトコル) インターフェイスを介して Amazon SES を使用するとき発生する可能性がある一般的な問題の解決方法と、Amazon SES から返される SMTP 応答コードの一覧については、「[Amazon SES SMTP の問題 \(p. 489\)](#)」を参照してください。
- Amazon SES の クエリ (HTTPS) API から返されるエラーコードのリストについては、「[Amazon SES API から返されるエラーコード \(p. 422\)](#)」を参照してください。
- レビューの送信プロセスに関連する一般的な問題の説明とその対処方法については、「[Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)」を参照してください。
- DNS ベースのブラックホールリスト (DNSBL) が Amazon SES を使用した送信にどのように影響を与えるかについては、「[DNS ブラックホールリスト \(DNSBL\) に関するよくある質問 \(p. 515\)](#)」を参照してください。

Amazon SES API を直接呼び出している場合は、「[Amazon Simple Email Service API リファレンス](#)」を参照して、発生する可能性がある HTTP エラーを確認してください。

## 一般的な Amazon SES の問題

このページで情報は、Amazon SES を使用する上で遭遇する可能性のある問題を説明しており、診断に役立ちます。

### 行った変更がすぐに表示されない

世界中のデータセンター内のコンピュータを介してアクセスされるサービスとして、Amazon SES は、**結果整合性**と呼ばれる分散コンピューティングモデルを採用しています。Amazon SES (または他の AWS サービス) で行った変更は、すべての可能なエンドポイントから認識されるまでに時間がかかります。この遅延は、サーバー間および世界中のリージョン間でのデータ送信にかかる時間から発生している場合があります。ほとんどのケースでは、この遅延が数分以上かかることはありません。

遅延が生じる可能性のあるエリアには以下が含まれます。



- 設定セットの作成と変更 - 設定セットを作成または変更すると (例えば、[専用 IP プールを既存の設定セットと関連付ける場合 \(p. 264\)](#))、作成や変更の時点から変更が有効になるまでに少しの間があります。
- イベント発生先の作成と変更 - イベント発生先の作成または変更をすると (たとえば、[Amazon SES に Eメールの送信データを別の AWS のサービスに送信するように伝える場合 \(p. 296\)](#))、イベント発生先の作成または変更をした時点から Eメールの送信イベントが実際に指定した先に届くまでの間に遅延が生じることがあります。

## Eメールアドレスおよびドメイン認証の問題

Amazon SES で Eメールアドレスまたはドメインを認証するには、Amazon SES コンソールまたは Amazon SES API を使用してプロセスを開始します。このセクションには、認証プロセスに関する問題の解決に役立つ情報が含まれます。

### Eメール認証のよくある問題

- 確認 Eメールが届かない - [Amazon SES Eメールアドレスの確認 \(p. 49\)](#) の手順を完了しても数分以内に確認 Eメールが届かない場合は、以下のステップを実行します。
  - 認証しようとしている Eメールアドレスのスパムまたは迷惑メールフォルダを確認します。
  - 認証しようとしている Eメールアドレスで Eメールを受信できることを確かめます。別の Eメール アドレス (個人の Eメール アドレスなど) を使用して、認証しようとしているアドレスに、テスト Eメールを送信します。
  - [Amazon SES コンソールの認証されたアドレスのリスト](#) をチェックします。認証しようとしている Eメール アドレスに間違いがないことを確認します。

### ドメインの検証に関する一般的な問題

「[Amazon SES でのドメインの検証 \(p. 61\)](#)」の手順を使用してドメインを確認しようとしたときに問題が発生した場合は、次に示す考えられる原因と解決策を確認してください。

- 自分が所有していないドメインを確認しようとしています - 自分が所有していないドメインを確認することはできません。たとえば、Amazon SES 経由で、gmail.com ドメインから Eメールを送信する場合は、[その電子メールアドレスを明確に確認する \(p. 49\)](#) 必要があります。gmail.com ドメイン全体を確認することはできません。
- DNS プロバイダーにより、TXT レコード名にアンダースコアを含めることが許可されない - 一部の DNS プロバイダーは、ドメインの DNS レコード名にアンダースコア文字を含めることを許可していません。これがお客様のプロバイダーに当てはまる場合、TXT レコード名から \_amazonses を削除できます。
- DNS プロバイダーによって TXT レコードの末尾にドメイン名が追加された - 一部の DNS プロバイダーは、ドメインの名前を自動的に TXT レコードの属性名に追加します。たとえば、属性名が \_amazonses.example.com のレコードを作成した場合、プロバイダーはドメイン名を追加して \_amazonses.example.com.example.com となります。ドメイン名の重複を避けるために、TXT レコードの作成時にドメイン名の末尾にピリオドを追加します。この手順では、ドメイン名を TXT レコードに追加する必要はないことを DNS プロバイダーに伝えます。
- DNS プロバイダーが DNS レコードの値を変更した場合 - 一部のプロバイダーは、小文字のみを使用するように DNS レコードの値を自動的に変更します。Amazon SES がドメインを検証するのは、ドメインの所有権の検証プロセスをスタートしたときに Amazon SES が提供した値と属性値が正確に一致する検証レコードが検出された場合のみです。ドメインの DNS プロバイダーが小文字のみを使用するように TXT レコード値を変更した場合、追加のサポートを DNS プロバイダーまでお問い合わせください。
- 同じドメインを複数回確認したい - 異なるリージョンで送信しようとしている場合や、同じリージョンの同じドメインの複数の AWS アカウントから送信しようとしている場合は、複数回ドメインを確認す

る必要があります。DNS プロバイダーが同じ属性名の複数の TXT レコードを持つことを許可しない場合、2つのドメインを確認できることがあります。DNS プロバイダーによって許可される場合、同じ TXT レコードに複数の属性値を割り当てることができます。たとえば、DNS が Amazon Route 53 によって管理されている場合、以下の手順を実行して、同じ TXT レコードに対して複数の値をセットアップできます。

1. Route 53 コンソールで、最初のリージョンのドメインを検証したときに作成した TXT レコードを選択します。
2. [Value (値)] ボックスで、既存の属性値の末尾に移動し、Enter キーを押します。
3. 追加のリージョンの属性値を追加し、レコードセットを保存します。

お客様の DNS プロバイダーでは、同じ TXT レコードに複数の値を割り当てることができない場合、TXT レコードの属性名の `_amazonses` で 1 回、属性名から削除された `_amazonses` で再度ドメインを確認することができます。このソリューションの欠点は、同じドメインを 2 回しか確認できないことです。

## ドメイン検証設定の確認

次の手順を使用して、Amazon SES ドメイン確認 TXT レコードが適切に DNS サーバーに発行されたことをチェックできます。この手順では、Windows および Linux で使用できる `nslookup` ツールを使用します。Linux では、`dig` を使用することもできます。

これらの手順に示すコマンドは、Windows 7 で実行されています。使用されているサンプルのドメインは、`ses-example.com` です。

この手順では、最初にドメインにサービスを提供する DNS サーバーを見つけます。次に、これらのサーバーに対して、TXT レコードを表示するためのクエリを実行します。ドメインにサービスを提供する DNS サーバーに対してクエリを実行する理由は、これらのサーバーには他の DNS サーバーに伝達されるまでに時間がかかるドメインの最新情報が格納されているためです。

ドメイン確認 TXT レコードが DNS サーバーに公開されていることを確認するには

1. 次のステップを実行して、ドメインのネームサーバーを見つけます。
  - a. コマンドラインに移動します。Windows 7 でコマンドラインに移動するには、[Start] を選択し、`cmd` と入力します。Linux ベースのオペレーティングシステムでは、ターミナルウィンドウを開きます。
  - b. コマンドプロンプトで、次のように入力します。ここで、`<domain>` はドメインを示します。これにより、ドメインにサービスを提供しているすべてのネームサーバーが表示されます。

```
nslookup -type=NS <domain>
```

ドメインが `ses-example.com` の場合、このコマンドは次のようになります。

```
nslookup -type=NS ses-example.com
```

コマンドの出力に、ドメインにサービスを提供しているネームサーバーのリストが表示されます。次のステップでは、これらのサーバーの 1 つに対してクエリを実行します。

2. 次のステップを実行して、TXT レコードが適切に発行されていることを確認します。
  - a. コマンドプロンプトで、次のように入力します。ここで、`<domain>` はドメインを示し、`<name server>` はステップ 1 で見つけたネームサーバーの 1 つを示します。

```
nslookup -type=TXT _amazonses.<domain> <name server>
```

ses-example.com の例で、ステップ 1 で見つけたネームサーバーが ns1.name-server.net の場合は、次のように入力します。

```
nslookup -type=TXT _amazonses.ses-example.com ns1.name-server.net
```

- b. コマンドの出力の text = に続く文字列が、Amazon SES コンソールのアイデンティティリストでドメインを選択すると表示される TXT 値と一致することを確認します。

この例では、\_amazonses.ses-example.com の下で、値が fmxqxT/icOYx4aA/bEUrDPMeax9/s3frblS+niixmqk= の TXT レコードを探しています。レコードが正しく発行されている場合、次のようなコマンド出力が得られます。

```
_amazonses.ses-example.com text = "fmxqxT/icOYx4aA/bEUrDPMeax9/s3frblS+niixmqk="
```

## Amazon SES の DKIM 問題のトラブルシューティング

このセクションでは、Amazon SES で DKIM 認証を設定するときに発生する可能性のある問題について説明します。DKIM を設定しようとしたときに問題が発生した場合は、以下の考えられる原因と解決策を確認してください。

DKIM を正常にセットアップしたが、メッセージが DKIM 署名されていない

[Easy DKIM \(p. 135\)](#) または [BYODKIM \(p. 143\)](#) を使用してドメインの DKIM を設定したが、送信するメッセージが DKIM 署名されていない場合は、次の手順を実行します。

- 適切な ID に対して DKIM が有効になっていることを確認します。Amazon SES コンソールで ID に対して DKIM を有効にするには、ID リストで E メールドメインを選択します。ドメインの詳細ページで、[DKIM] を展開し、[Enable] を選択して DKIM を有効にします。
- 同じドメインの確認済み E メールアドレスから送信していないことを確認します。ドメインに DKIM を設定した場合は、そのドメインから送信するすべてのメッセージが DKIM 署名されます。ただし、個別に確認した E メールアドレスは除外されます。個別に確認された E メールアドレスには、別の設定が使用されます。たとえば、ドメイン example.com に DKIM を設定し、E メールアドレス mary@example.com を個別に検証した場合（ただし、アドレスに DKIM を設定していない場合）、mary@example.com から送信する E メールは DKIM 認証なしで送信されます。この問題は、アカウントの ID リストから E メールアドレス ID を削除することで解決できます。
- 複数の AWS リージョンで同じ ID を使用する場合は、リージョンごとに個別に DKIM を設定する必要があります。同様に、同じドメインを複数の AWS アカウントで使用する場合は、アカウントごとに DKIM を設定する必要があります。特定のリージョンやアカウントに必要な DNS レコードを削除すると、Amazon SES はそのリージョンやアカウントの DKIM 署名を無効にします。DKIM 署名が無効になると、Amazon SES は E メールで通知を送信します。

Amazon SES コンソールのユーザードメインの DKIM 詳細は DKIM: waiting on sender verification... を示します。DKIM 検証ステータス: 検証の保留中。

[Easy DKIM \(p. 135\)](#) または [独自の DKIM 認証トークンを提供する \(p. 143\)](#) の手順を実行してドメインの DKIM を設定しても、Amazon SES コンソールに DKIM 検証が保留中であることが示される場合は、次の手順を実行します。

- 最長で 72 時間待ちます。まれに、DNS レコードが Amazon SES に表示されるまでに時間がかかることがあります。
- CNAME レコード (Easy DKIM の場合) または TXT レコード (BYODKIM の場合) に正しい名前が使用されていることを確認します。DNS プロバイダーによっては、作成するレコードにドメイン

名が自動的に付加される場合があります。たとえば、`example._domainkey.example.com` の [Name] でレコードを作成した場合、DNS プロバイダーはこの文字列の末尾にドメインの名前を追加して、`example._domainkey.example.com.example.com` という名前にします。詳細については、DNS プロバイダーのドキュメントを参照してください。

Amazon SES から、DKIM の設定が取り消された (または取り消される) という E メールが届きます。

これは、Amazon SES が DNS サーバー上で必要な CNAME レコード (Easy DKIM を使用している場合) または必要な TXT レコード (BYODKIM を使用している場合) を検出できなくなっていることを意味します。通知 E メールには、DNS レコードを再発行することで DKIM のセットアップステータスが失効して DKIM 署名が無効化されるのを回避できる残りの期間が記載されています。DKIM のセットアップが失効した場合は、DKIM のセットアップ手順を最初から繰り返す必要があります。

BYODKIM を設定しようとする、DKIM 検証プロセスに失敗します。

プライベートキーが正しい形式を使用していることを確認してください。プライベートキーは PKCS #1 形式で、1024 ビットの RSA 暗号化を使用する必要があります。さらに、プライベートキーには base64 エンコードを適用する必要があります。

BYODKIM のセットアップ中にドメインのパブリックキーを指定しようすると `BadRequestException` エラーが表示されます。

`BadRequestException` エラーが発生した場合は、次の操作を行います。

- パブリックキーに指定するセクタに、1 ~ 63 文字の英数字が含まれていることを確認します。セクタに、ピリオド、その他の記号、句読点を含めることはできません。
- パブリックキーからヘッダー行とフッター行を削除し、パブリックキーからすべての改行を削除したことを確認します。

Easy DKIM を使用すると、DNS サーバーは Amazon SES DKIM CNAME レコードを正常に返しますが、ドメイン検証 TXT レコードに対しては `SERVFAIL` を返します。

DNS プロバイダーが CNAME レコードをリダイレクトできない場合があります。TXT レコードの Amazon SES と ISP クエリ。DKIM の仕様準拠するには、DNS サーバーが CNAME レコードに対するクエリだけでなく、TXT レコードに対するクエリにも応答できる必要があります。DNS プロバイダーが TXT レコードのクエリに応答できない場合は、DNS ホスティングプロバイダーとして Route 53 を使用することもできます。

E メールに 2 つの DKIM 署名が含まれている

追加の DKIM 署名 (d=amazonses.com を含む) は、Amazon SES によって自動的に追加されます。この署名は無視できます。

## Amazon SES 配信の問題

Amazon SES へのリクエストが成功すると、メッセージは多くの場合、すぐに送信されます。ただし場合によっては、短い遅延が生じることがあります。いずれにせよ、E メールが送信されることは保証されません。

ただし、Amazon SES からメッセージが送信されるときにいくつかの要因で正常な配信が妨げられることがあります。場合によっては、送信したメッセージが到着しないことを知って初めて配信が失敗したことに気付く結果になります。この状況を解決するには、次のプロセスを使用します。

E メールが到着しない場合は、次の操作を試してください。

- 対象の E メールに `SendEmail` または `SendRawEmail` リクエストを行っていて、成功を示す応答を受け取っていることを確認します。これらのリクエストをプログラムで行っている場合は、ソフトウェアのログを調べて、プログラムがリクエストを実行し、成功を示す応答を受け取っていることを確認します。

- ブログ記事「[SES を使用してメールを送信したときに遅延が発生する可能性がある 3 つの場所](#)」を参照してください。実際は、E メールが配信されていないのではなく、遅延が生じている可能性があります。
- 送信者の E メールアドレス (「From」アドレス) が有効であることを確認します。バウンスメッセージが送信される「Return-Path」アドレスも確認します。メールがバウンスされた場合は、説明のエラーメッセージが返されます。
- [\[AWS Service Health Dashboard\]](#)で、Amazon SES に既知の問題がないことをチェックするため確認してください。
- Eメールの受取人または受取人の ISP に問い合わせます。受取人が正しい E メールアドレスを使用していることを確認し、受取人の ISP において配信に関する既知の問題があるかどうかを問い合わせます。また、実際は到着した E メールがスパムとしてフィルタリングされていないかどうかを確認します。
- 有償の [AWS サポートプラン](#)を契約している場合は、新しい技術サポートケースを開くことができます。お問い合わせの際は、SendEmail または SendRawEmail 応答から返されたリクエスト ID またはメッセージ ID に加えて、該当する受取人のアドレスをお知らせください。
- 原因が実際は配信時の遅延であり永続的なエラーでないか、しばらく観察します。スパム発信者への対策として、一部の ISP は不明な送信メールサーバーからの受信メッセージを一時的に拒否します。グレーリストと呼ばれるこのプロセスにより配信時に遅延が生じることがあります。Amazon SES はこれらのメッセージを再試行します。グレーリストが原因となっている場合は、再試行される E メールを ISP が許可することができます。

## Amazon SES から受け取った E メールに関する問題

このセクションでは、Amazon SES から送信された E メールを受信するときに発生する可能性がある一般的な問題について説明します。

E メールクライアントでは、Eメールのソースとして「amazonses.com 経由で送信されました」と表示されます

一部のメールクライアントでは、送信者のドメインが Eメールの送信元のドメイン (この場合は amazonses.com) と一致しない場合、「via」ドメインが表示されます。詳細については、Gmail サポートウェブサイトの「[送信者の名前の横にあるその他の情報](#)」を参照してください。または、[ドメインキーアイデンティファイドメール \(p. 135\) \(DKIM\)](#)を設定することもできます。DKIM を使用してメールを認証する場合、通常、Eメールクライアントには "via" ドメインが表示されません。これは、DKIM の署名には、送信元として示されているドメインがメールの送信元として示されるためです。DKIM の設定の詳細については、「[Amazon SES における DKIM を使った Eメールの認証 \(p. 135\)](#)」を参照してください。

メッセージに文字化けまたは意味のない文字が含まれています

メッセージに ASCII 文字セットに含まれていない文字 (アクセント付きラテン文字、中国文字、アラビア文字など) が含まれている場合は、HTML 文字エンコーディングを使用してこれらの文字をエンコードする必要があります。ウェブベースのツールを使用して、Eメール内の文字をエンコードできます。たとえば、Email On Acid ウェブサイトの [HTML 文字コンバーター](#) など。

または、MIME メッセージを自分で組み立てることもできます。MIME メッセージでは、メッセージに UTF-8 エンコーディングを使用するように指定できます。UTF-8 エンコーディングを使用する場合、非 ASCII 文字をメッセージに直接使用できます。MIME メッセージの作成が完了したら、[SendRawEmail API](#) を使用して送信できます。SendRawEmail API を使用したメッセージの送信の詳細については、「[AWS SDK を使用した Raw Eメールの送信 \(p. 451\)](#)」を参照してください。

この問題の一般的な原因の 1 つは、Microsoft Word のスマート引用符機能です。Word からコンテンツを頻繁にコピーして Eメールに貼り付ける場合は、この問題が発生する可能性があります。スマート引用符機能は、ストレート引用文字 (「...」) を中引用符文字 (「...」) に置き換えます。中引用符文

字は標準 ASCII 文字ではありません。その結果、一部の E メールクライアントでは「??」として表記されます。あるいは「â€œ」などの文字群として表記されます。この問題を解決するには、Word でスマート引用符機能を無効にすることができます。または、前の段落の SendRawEmail ソリューションを使用することもできます。この機能を無効にする方法については、Microsoft Office サポートウェブサイトの「[Word のスマート引用符](#)」を参照してください。

## Amazon SES 通知の問題

バウンス、苦情、または配信通知に関する問題が発生した場合は、以下の考えられる原因と解決方法を確認します。

- バウンス通知を Amazon SNS 経由で受け取ったが、どの受信者が通知と対応しているかわからない - 今後、バウンス通知を特定の受信者に関連付けるには、以下のオプションを使用できます。
  - Amazon SES は追加したカスタムメッセージ ID を保持しないので、Amazon SES が E メールを受け付けるときにお客様に渡す Amazon SES メッセージ ID と識別子のマッピングを保存します。
  - Amazon SES を呼び出すたびに、単一のメッセージを複数の受信者に送信する代わりに、単一の受取人に送信します。
  - バウンスに関するメッセージ全文をお客様に転送する、E メールによるフィードバックの転送を有効にすることができます。
- 苦情通知を Amazon SNS 経由または E メールによるフィードバックの転送で受け取ったが、どの受信者が通知と対応しているかわからない - 一部の ISP は、苦情通知を Amazon SES に渡す前に、苦情を送信した受信者の E メールアドレスを編集します。受信者の E メールアドレスを特定できるようにする最善の方法は、Amazon SES が E メールを受け付けるときにお客様に渡す Amazon SES メッセージ ID と識別子のマッピングを保存することです。Amazon SES は追加したカスタムメッセージ ID を保持しないことに注意してください。
- 自分が所有していない Amazon SNS トピックに通知が送信されるように設定したい - このトピックの所有者は、お客様のアカウントがこの所有者のトピックで SNS:Publish アクションを呼び出すことを許可するアクセスポリシーを設定する必要があります。IAM ポリシーを使用して Amazon SNS トピックへのアクセスをコントロールする方法については、Amazon Simple Notification Service デベロッパーガイドの「[Amazon SNS トピックへのアクセスの管理](#)」を参照してください。

## Amazon SES の E メール送信エラー

このトピックでは、Amazon SES 経由で E メールを送信するときに発生する可能性がある E メール送信に固有のエラーのタイプについて説明します。Amazon SES 経由でメールを送信しようとしたときに Amazon SES の呼び出しに失敗すると、Amazon SES からアプリケーションにエラーメッセージが返され、E メールは送信されません。このエラーメッセージがどのように表示されるかは、Amazon SES を呼び出す方法によって異なります。

- Amazon SES API を直接呼び出す場合は、Query アクションによってエラーが返されます。エラーは「MessageRejected」あるいは、Amazon Simple Email Service API リファレンスの「[一般的なエラー](#)」トピック内で指定されたエラーの一つである可能性があります。
- 例外をサポートするプログラミング言語を使用する AWS SDK を使用して Amazon SES を呼び出した場合、Amazon SES は例外をスローします。例外のタイプは、SDK とエラーによって異なります。例えば、Amazon SES MessageRejectedException の例外 (実際の名前は SDK によって異なります) または一般的な AWS の例外がスローされます。例外のタイプにかかわらず、例外のエラータイプとエラーメッセージからより多くの情報が得られます。
- SMTP インターフェイスを介して Amazon SES を呼び出した場合、エラーがどのように示されるかはアプリケーションによって異なります。アプリケーションによって、特定のエラーメッセージが表示される場合もあれば、表示されない場合もあります。Amazon SES から返される SMTP 応答コードのリストについては、「[Amazon SES から返される SMTP 応答コード \(p. 490\)](#)」を参照してください。

## Note

E メール送信のための Amazon SES 呼び出しが失敗した場合、対象の E メールについては課金されません。

E メールを送信しようとしたときに Amazon SES がエラーを返す原因となる、Amazon SES に固有の問題のタイプを次に示します。これらのエラーは、Amazon Simple Email Service API Reference の「[一般的なエラー](#)」トピックに指定されている MalformedQueryString などの一般的な AWS エラーとは別のエラーです。

- E メールアドレスが検証されていません。以下のアイデンティティが、リージョン内でチェックできませんでした。リージョン: identity1、identity2、identity3。- [Amazon SES で検証 \(p. 48\)](#)されていない E メールアドレスまたはドメインから E メールを送信しようとしています。このエラーは、「From」、「Source」、「Sender」、または「Return-Path」のアドレスに該当する場合があります。アカウントが [Amazon SES サンドボックス \(p. 74\)](#)にまだある場合は、[Amazon SES のメールボックスシミュレーター \(p. 187\)](#)から提供されているアドレスを除く、すべての受信者のアドレスも検証する必要があります。Amazon SES が失敗した ID をすべて表示できない場合は、エラーメッセージが省略符号で終了します。

## Note

Amazon SES は複数の [AWS リージョン \(p. 467\)](#)にエンドポイントを持ち、E メールアドレスの検証ステータスは AWS のリージョンごとに別個に扱われます。使用する AWS リージョンの各送信者について、検証プロセスを完了する必要があります。

- Account is paused - アカウントによる E メール送信機能を一時停止します。Amazon SES コンソールには引き続きアクセスでき、ほとんどの操作を実行できます。ただし、E メールを送信しようとした場合、このメッセージが表示されます。

アカウントの E メール送信機能を一時停止する場合、AWS アカウントに関連付けられている E メールアドレスに通知を送信します。詳細については、「[the section called “送信レビュープロセス” \(p. 500\)](#)」を参照してください。

- スロットリング - アプリケーションが 1 秒あたりに送信しようとしているメッセージが多すぎるか、過去 24 時間に送信した E メールが多すぎる可能性があります。このような場合は、エラーメッセージは次の例のようになります。
  - Daily message quota exceeded - 24 時間の期間に送信することが許可されたメッセージの最大数に達しました。日次クォータを超過した場合、次の 24 時間の期間にならないとそれ以上 E メールを送信できません。
  - Maximum sending rate exceeded - 送信を試みた 1 秒あたりの Eメールの件数が、許可された最大送信レートを超過しました。送信レートを超過した場合、Eメールの送信を継続できますが、送信レートを引き下げる必要があります。詳細については、「AWS メッセージングとターゲティングブログ」の「["Throttling – Maximum sending rate exceeded \(スロットリング – 最大送信レートの超過\)" エラーの対処法](#)」を参照してください。
  - Sigv2 SMTP の最大送信速度を超過しました — 2019 年 1 月 10 日より前に作成された SMTP 認証情報を使用してメッセージを送信しようとしています。SMTP 認証情報は古いバージョンの AWS 署名を使用して作成されています。セキュリティのため、この日付よりも前に作成した認証情報を削除して、新しい認証情報に置き換える必要があります。古い認証情報は、IAM コンソールを使用して削除できます。認証情報ファイルの作成の詳細については、「[the section called “SMTP 認証情報の取得” \(p. 89\)](#)」を参照してください。

送信アクティビティを定期的に監視して、送信クォータにどれだけ近づいているかを確認する必要があります。詳細については、「[Amazon SES 送信クォータのモニタリング \(p. 150\)](#)」を参照してください。送信クォータに関する一般的な情報については、「[Amazon SES 送信クォータの管理 \(p. 149\)](#)」を参照してください。送信クォータを引き上げる方法については、「[Amazon SES 送信クォータの引き上げ \(p. 151\)](#)」を参照してください。

## Important

スロットリングエラーを説明するエラーテキストが日次クォータまたは最大送信レートの超過に関係ない場合は、システム全体の問題が原因で送信機能が制限されている可能性があります。サービスステータスについては、「[AWS Service Health Dashboard](#)」を参照してください。

- There are no recipients specified - 受信者が指定されていません。
- There are non-ASCII characters in the email address - E メールアドレス文字列は 7 ビット ASCII である必要があります。送信先または送信元の E メールアドレス内で、ドメインの部分に Unicode 文字が含まれる場合は、Punycode を使用してドメインをエンコードする必要があります。Punycode は E メールアドレスのローカル部分 (@ 記号の前の部分) では許可されていません。また、「差出人」名にも許可されていません。「差出人」名に Unicode 文字を使用する場合は、[Amazon SES API を使用して raw E メールを送信する \(p. 114\)](#) に説明されているとおりに MIME encoded-word 構文を使用して「差出人」名をエンコードする必要があります。Punycode の詳細については、[RFC 3492](#) を参照してください。
- Mail FROM domain is not verified - Amazon SES は、指定された MAIL FROM ドメインを使用するために必要な MX レコードを読み取ることができませんでした。カスタム MAIL FROM ドメインの設定については、[カスタムの MAIL FROM ドメインの設定 \(p. 68\)](#) を参照してください。
- Configuration set does not exist - 指定した設定セットが存在しません。設定セットは、メール送信イベントの発行に使用するオプションのパラメータです。詳細については、「[Amazon SES イベント発行を使用して E メール送信をモニタリングする \(p. 296\)](#)」を参照してください。

## Amazon SES のスループットを上げる

E メールを送信するときは、最大送信レートで許可される頻度で Amazon SES を呼び出すことができます。(最大送信レートの詳細については、「[Amazon SES 送信クォータの管理 \(p. 149\)](#)」を参照してください)。ただし、それぞれの Amazon SES の呼び出しが完了するまで時間がかかります。

Amazon SES API または SMTP インターフェイスを使用して Amazon SES を複数回呼び出す場合は、次のヒントを考慮するとスループットを高めることができます。

- 現在のパフォーマンスを測定してボトルネックを識別する - 考えられるパフォーマンステストには、アプリケーションのコードループ内で複数のテスト E メールをできる限り早く送信する操作が含まれます。各 SendEmail リクエストのラウンドトリップレイテンシーを測定します。次に、同じマシン上でアプリケーションの追加インスタンスを増分的に起動して、ネットワークレイテンシーへの影響を調べます。また、このテストを複数のマシンおよび異なるネットワークで実行すると、考えられるあらゆるマシンリソースボトルネックや存在する可能性があるネットワークボトルネックを特定するのに役立ちます。
- (API のみ) 永続的な HTTP 接続の使用を検討する - 永続的な HTTP 接続を使用して、API リクエストごとに別個の新しい HTTP 接続を確立するオーバーヘッドを回避します。つまり、複数の API リクエストに対して同じ HTTP 接続を再利用します。
- 複数のスレッドの使用を検討する - アプリケーションが単一のスレッドを使用する場合、アプリケーションコードは Amazon SES API を呼び出し、API 応答を同期的に待機します。通常、E メール送信では I/O 負荷が高いため、複数のスレッドからこの操作をする方がよりスループットが向上します。任意の数の実行スレッドを使用して、同時にメールを送信できます。
- 複数のプロセスを使用することを検討する - 複数のプロセスを使用すると、Amazon SES に対してより多くの同時アクティブ接続を持つことになるため、スループットが向上します。たとえば、目的のメールを複数のバケットにセグメント化した後、E メール送信スクリプトの複数のインスタンスを同時に実行できます。
- ローカルメールリレーを使用することを検討する - アプリケーションは、ローカルメールサーバーにメッセージを迅速に送信できます。次に、ローカルメールサーバーを使用して、メッセージをバッファ処理し、非同期的に Amazon SES に送信します。一部のメールサーバーでは、同時配信がサポートされています。つまり、アプリケーションがメールサーバーに送る E メールがシングルスレッドで生成される場合でも、メールサーバーから Amazon SES に E メールを送信するときは複数のスレッドが使用さ



れます。詳細については、「[Amazon SES を既存の E メールサーバーと統合する \(p. 96\)](#)」を参照してください。

- Amazon SES API エンドポイントに近い場所でアプリケーションをホストすることを検討する - Amazon SES API エンドポイントに近いデータセンターが、Amazon SES API エンドポイントと同じ AWS リージョンの Amazon EC2 インスタンスでアプリケーションをホストすることを検討してください。これにより、アプリケーションと Amazon SES の間のネットワークレイテンシーが小さくなり、スループットが向上する可能性があります。Amazon SES を使用できるリージョンのリストについては、AWS 全般のリファレンスの「[Amazon Simple Email Service \(Amazon SES\)](#)」を参照してください。
- 複数のマシンを使用することを検討する - ホストマシンのシステム構成によっては、単一 IP アドレスへの同時 HTTP 接続の数に制限がある場合があります。つまり、単一マシンでの特定の同時接続数を越えた場合に並列処理の利点が制限されることとなります。これがボトルネックとなる場合は、複数のマシンを使用して同時 Amazon SES リクエストを実行することを検討してください。
- SMTP エンドポイントではなく Amazon SES クエリ API を使用することを検討する - Amazon SES SMTP エンドポイントとのやり取りには複数のネットワークリクエスト (たとえば、EHLO、MAIL FROM、RCPT TO、DATA、QUIT) から構成される SMTP 対話が含まれるのに対し、クエリ API を使用すると、単一のネットワーク呼び出しを使用して E メール送信リクエストを送信できます。Amazon SES クエリ API の詳細については、[Amazon SES API を使用して E メールを送信する \(p. 113\)](#) を参照してください。
- Amazon SES メールボックスシミュレーターを使用して最大スループットをテストする - 実装した変更をテストするために、メールボックスシミュレーターを使用できます。メールボックスシミュレーターを使用すると、日次送信クォータを使い果たすことなくシステムの最大スループットを判定できます。メールボックスシミュレータの詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

SMTP インターフェイスを介して Amazon SES にアクセスする場合は、「[Amazon SES SMTP の問題 \(p. 489\)](#)」スループットに影響を与える可能性がある特定の SMTP 関連の問題」を確認してください。

## Amazon SES SMTP の問題

このセクションでは、SMTP (Amazon SES Simple Mail Transfer Protocol) インターフェイスを使用した Eメールの送信に関連するいくつかの一般的な問題の解決策について説明します。また、Amazon SES から返される SMTP 応答コードも一覧表示します。

Amazon SES SMTP インターフェイス経由の Eメール送信の詳細については、「[Amazon SES SMTP インターフェイスを使用した Eメールの送信 \(p. 88\)](#)」を参照してください。

- Amazon SES SMTP エンドポイントに接続できない。

Amazon SES SMTP エンドポイントへの接続の問題は、ほとんどの場合以下の問題が関係しています。

- 認証情報の誤り - SMTP エンドポイントへの接続に使用する認証情報が AWS 認証情報と異なっています。SMTP 認証情報を取得する方法については、「[Amazon SES SMTP 認証情報を取得 \(p. 89\)](#)」を参照してください。認証情報の詳細については、「[Amazon SES 認証情報の種類 \(p. 17\)](#)」を参照してください。
- ネットワークまたはファイアウォールの問題 - ネットワークにより、Eメールを送信しようとしているポートでのアウトバウンド接続がブロックされている可能性があります。ローカルネットワークが原因で接続の問題が発生しているかどうかを確認するには、コマンドラインに次のコマンドを入力します。`port` は、使用するポート (通常は 465、587、2465 または 2587) に置き換えてください。`telnet email-smtp.us-west-2.amazonaws.com port`

このコマンドを使用して SMTP サーバーに接続でき、さらに TLS ラッパーまたは STARTTLS を使用して Amazon SES に接続しようとしている場合は、[コマンドラインを使用して Amazon SES SMTP インターフェイスへの接続をテストする \(p. 107\)](#) に示されている手順を実行してください。

telnet または openssl を使用して Amazon SES SMTP エンドポイントに接続できない場合は、ネットワークのどこか (ファイアウォールなど) で、使用しようとしているポートでのアウトバウンド接続がブロックされているということです。ネットワーク管理者と協力して、問題を診断および解決してください。

- ポート 25 を使用して Amazon EC2 インスタンスから Amazon SES に送信しているときに、タイムアウトエラーを受信する。

Amazon EC2 はデフォルトでポート 25 を制限します。これらの制限を削除するためには、[E メール送信制限解除申請](#)に Amazon EC2 リクエストを送信します。制限されないポート 465 または 587 を使用して Amazon SES に接続することもできます。

- ネットワークエラーが原因で E メールのドロップが発生している。

アプリケーションが Amazon SES SMTP エンドポイントに接続するときに再試行ロジックを使用していること、およびアプリケーションがネットワークエラーを検出しメッセージ配信を再試行できることを確認します。SMTP は冗長プロトコルで、このプロトコルを使用して E メールを送信する場合は複数のネットワークラウンドトリップが必要になります。SMTP の性質上、ネットワークエラーの可能性は高くなります。

- SMTP エンドポイントとの接続が失われる。

ほとんどの場合、接続の切断は以下の問題によって発生します。

- MTU サイズ - タイムアウトエラーメッセージが表示された場合、Amazon SES SMTP インターフェイスとの接続に使用しているコンピュータのネットワークインターフェイスの最大送信単位 (MTU) が大きすぎる可能性があります。この問題を解決するには、そのコンピュータの MTU サイズを 1500 バイトに設定します。

Windows、Linux、macOS の各オペレーティングシステムで MTU サイズを設定する方法の詳細については、Amazon Redshift クラスター管理ガイドの「[クライアントでクエリがハングし、クラスターに到達しないように見える](#)」を参照してください。

Amazon EC2 の MTU サイズ設定の詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「[EC2 インスタンスのネットワーク最大送信単位 \(MTU\)](#)」を参照してください。

- 存続時間の長い接続 - Amazon SES SMTP エンドポイントは、Elastic Load Balancer (ELB) の背後の複数の Amazon EC2 インスタンスのフリートで実行されます。システムの最新の状態と耐障害性を維持するために、アクティブな Amazon EC2 インスタンスは定期的に終了され、新しいインスタンスと置き換えられます。アプリケーションは ELB を介して Amazon EC2 インスタンスに接続するため、Amazon EC2 インスタンスが終了すると接続が無効になります。単一の SMTP 接続を使用して所定の数のメッセージを配信した後、または SMTP 接続がある程度の時間にわたってアクティブであった場合は、新しい SMTP 接続を確立する必要があります。アプリケーションがホストされている場所やアプリケーションがどのように E メールを Amazon SES に送信するかに応じて、いろいろな値を試しながら適切なしきい値を見つける必要があります。
- 本稼働環境のネットワークのホワイトリストに登録できるように、Amazon SES の SMTP メールサーバーの IP アドレスを知りたい。

Amazon SES SMTP エンドポイントの IP アドレスは、ロードバランサーの背後に存在します。その結果、これらの IP アドレスは頻繁に変更されます。Amazon SES エンドポイントのすべての IP アドレスの明確なリストを提供することはできません。個々の IP アドレスをホワイトリストに登録するのではなく、amazonses.com ドメインをホワイトリストに登録することをお勧めします。

## Amazon SES から返される SMTP 応答コード

このセクションでは、Amazon SES SMTP インターフェイスから返される応答コードを一覧表示します。

400 エラーを受け取った SMTP リクエストは、再試行する必要があります。待機時間を少しずつ増やしながらリクエストを再試行するシステムを実装することをお勧めします (たとえば、待機時間を 5 秒、10 秒、30 秒と増やしながら再試行します)。3 回目の再試行が失敗した場合は、20 分待ってからプロセス

を繰り返してください。段階的な再試行ポリシーを使用する実装例については、AWS メッセージングとターゲティングブログで「[Throttling – Maximum sending rate exceeded \(スロットリング – 最大送信レートの超過\)](#)」エラーの対処法」を参照してください。

#### Note

AWS SDK は、再試行ロジックを自動的に実装しますが、SMTP の代わりに HTTPS インターフェイスを使用します。

500 エラーが発生した場合は、もう一度リクエストを送信する前に、リクエストを修正して問題を修正する必要があります。たとえば、AWS 認証情報が無効な場合は、要求を再度送信する前に正しい認証情報を使用するようにアプリケーションを更新する必要があります。

説明	Response Code (レスポンスコード)	詳細情報
認証に成功	235 Authentication successful	SMTP クライアントは SMTP サーバーに正常に接続してサインインしました。
正常に配信	250 Ok <i>MessageID</i>	<b>##### ID</b> は、Amazon SES がメッセージを一意に識別するために使用する文字列です。
Service unavailable	421 Too many concurrent SMTP connections	SMTP サーバーへの接続が現在多すぎるため、Amazon SES はリクエストを処理できません。
ローカル処理エラー	451 Temporary service failure	Amazon SES はリクエストを処理できませんでした。リクエストを処理できないという問題がある可能性があります。
タイムアウト	451 Timeout waiting for data from client	リクエストの間の経過時間が長すぎるため、SMTP サーバーが接続を閉じました。
日次送信クォータの超過	454 Throttling failure: Daily message quota exceeded	24 時間の期間に送信することが Amazon SES によって許可されたメッセージの最大数を超えました。詳細については、「 <a href="#">Amazon SES 送信クォータの管理 (p. 149)</a> 」を参照してください。
最大送信レートの超過	454 Throttling failure: Maximum sending rate exceeded	Amazon SES によって許可された、1 秒あたりに送信できる Eメールの最大数を超えました。詳細については、「 <a href="#">Amazon SES 送信クォータの管理 (p. 149)</a> 」を参照してください。
SMTP 認証情報を検証するときの Amazon SES の問題	454 Temporary authentication failure	この問題の原因となる可能性のある問題には、以下が含まれます (ただし、これらに限定されません)。 <ul style="list-style-type: none"> <li>E メール送信アプリケーションと Amazon SES の間で暗号化に関する問題が発生しています。Amazon SES に接続する</li> </ul>

説明	Response Code (レスポンスコード)	詳細情報
		<p>ときは暗号化された接続を使用する必要があることに注意してください。詳細については、「<a href="#">Amazon SES SMTP エンドポイントへの接続 (p. 94)</a>」を参照してください。</p> <ul style="list-style-type: none"> <li>Amazon SES に問題が発生している可能性があります。更新のために、<a href="#">AWS Service Health Dashboard</a>をチェックします。</li> </ul>
リクエストの受信に関する問題	454 Temporary service failure	Amazon SES はリクエストを受信できませんでした。その結果、メッセージは送信されませんでした。
認証情報の誤り	530 Authentication required	Eメールの送信に使用するアプリケーションは、Amazon SES SMTP インターフェイスに接続したときに認証を試みませんでした。
無効な認証情報	535 Authentication Credentials Invalid	Eメールを送信するのに使用するアプリケーションは Amazon SES に正しい SMTP 認証情報を提供しませんでした。SMTP 認証情報は AWS 認証情報と同じではありません。詳細については、「 <a href="#">Amazon SES SMTP 認証情報を取得 (p. 89)</a> 」を参照してください。
Amazon SES にサブスクライブされないアカウント	535 Account not subscribed to SES	SMTP 認証情報を所有する AWS アカウントは、Amazon SES にサインアップされていません。
メッセージが長すぎます	552 Message is too long.	送信しようとしているメッセージのサイズが 10 MB を超えています。
Amazon SES にサブスクライブされないアカウント	535 Account not subscribed to SES	SMTP 認証情報を所有する AWS アカウントは、Amazon SES にサインアップされていません。
Amazon SES SMTP エンドポイントを呼び出す許可が与えられていないユーザー	554 Access denied: User <i>UserARN</i> is not authorized to perform ses:SendRawEmail on resource <i>IdentityARN</i>	SMTP 認証情報を所有するユーザーの AWS Identity and Access Management (IAM) ポリシーまたは Amazon SES 送信承認ポリシーでは、SMTP エンドポイントを呼び出すことが許可されません。

説明	Response Code (レスポンスコード)	詳細情報
未確認の E メールアドレス	554 Message rejected: Email address is not verified. The following identities failed the check in region <b>region</b> : <b>identity0</b> , <b>identity1</b> , <b>identity2</b>	<p><a href="#">Amazon SES アカウントからのメール送信が確認されていない (p. 48)</a>メールアドレスまたはドメインからメールを送信しようとしています。このエラーは、「From」、「Source」、「Sender」、または「Return-Path」のアドレスに該当する場合があります。アカウントがサンドボックスにまだある場合は、<a href="#">Amazon SES メールボックスシミュレーター (p. 187)</a>から提供されているアドレスを除く、すべての受信者のアドレスも確認する必要があります。Amazon SES が検証チェックに失敗したすべての ID を表示できない場合、エラーメッセージは 3 つのピリオド (...) で終わります。</p> <p>Note</p> <p><a href="#">Amazon SES は複数の AWS リージョン (p. 467)</a>にエンドポイントを持ち、E メールアドレスの検証ステータスは AWS のリージョンごとに別個に扱われます。使用する AWS リージョンの各送信者について、検証プロセスを完了する必要があります。</p>

# Amazon SES から個人データを削除する

使用方法によっては、個人用と見なされる可能性のあるデータが Amazon SES に保存される場合があります。例えば、Amazon SES を使用して E メールを送信するには、少なくとも 1 つの検証済み ID を指定する必要があります (E メールアドレスまたはドメイン)。Amazon SES コンソールまたは Amazon SES API を使用して、この個人データを完全に削除できます。

この章では、個人用と見なされる可能性があるさまざまなタイプのデータを削除する手順について説明します。

このセクションのトピック:

- [アカウントレベルのサブプレッジョンリストから E メールアドレスを削除する \(p. 494\)](#)
- [Amazon SES を使用して送信された E メールに関するデータを削除する \(p. 495\)](#)
- [ID に関するデータを削除する \(p. 496\)](#)
- [送信者認証データを削除する \(p. 496\)](#)
- [受信ルールに関連するデータを削除する \(p. 497\)](#)
- [IP アドレスフィルターに関連するデータを削除する \(p. 497\)](#)
- [E メールテンプレートのデータを削除する \(p. 498\)](#)
- [カスタム検証 E メールテンプレートのデータを削除する \(p. 498\)](#)
- [AWS アカウントを解約することですべての個人データを削除する \(p. 499\)](#)

## アカウントレベルのサブプレッジョンリストから E メールアドレスを削除する

Amazon SES には、オプションのアカウントレベルのサブプレッジョンリストが含まれています。この機能を有効にすると、バウンスや苦情が発生したときに、E メールアドレスがサブプレッジョンリストに自動的に追加されます。E メールアドレスは、削除するまでこのリストに残ります。アカウントレベルのサブプレッジョンリストの詳細については、「[アカウントレベルのサブプレッジョンリストの使用 \(p. 190\)](#)」を参照してください。

アカウントレベルのサブプレッジョンリストからメールアドレスを削除するには [Amazon SES API v2](#) で `DeleteSuppressedDestination` オペレーションを実行します。ここでは、AWS CLI を使用して E メールアドレスを削除する手順について説明します。AWS CLI のインストールおよび設定の詳細については、「[AWS Command Line Interface ユーザーガイド](#)」を参照してください。

AWS CLI を使用してアカウントレベルのサブプレッジョンリストからアドレスを削除するには

- コマンドラインで以下のコマンドを入力します。

```
aws sesv2 delete-suppressed-destination --email-address recipient@example.com
```

前のコマンドで、*recipient@example.com* を、アカウントレベルのサブプレッジョンリストから削除する E メールアドレスに置き換えます。

## Amazon SES を使用して送信された E メールに関するデータを削除する

Amazon SES を使用して E メールを送信するとき、E メールに関するデータを他の AWS サービスに送信することができます。たとえば、E メールイベントについての情報 (配信、オープン、クリックなど) を Kinesis Data Firehose に送信することができます。このイベントデータには通常、E メールアドレスと、Eメールの送信元の IP アドレスが含まれます。さらに、Eメールが送信されたすべての受信者の Eメールアドレスも含まれます。

Kinesis Data Firehose を使用して Eメールのイベントデータを Amazon Simple Storage Service、Amazon Elasticsearch Service、Amazon Redshift など、複数の送信先にストリーミングすることができます。このデータを削除するには、まず Kinesis Data Firehose へのデータのストリーミングを停止した後、既にストリーミングされているデータを削除する必要があります。Kinesis Data Firehose への Amazon SES イベントデータのストリーミングを停止するには、Kinesis Data Firehose イベントの送信先を削除する必要があります。

Amazon SES コンソールを使用して Kinesis Data Firehose イベントの送信先を削除するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. [Email Sending] で、[Configuration Sets] を選択します。
3. 設定セットのリストで、Kinesis Data Firehose イベント送信先が含まれている設定セットを選択します。
4. 削除する Kinesis Data Firehose イベントの送信先の横にある、[delete (削除)] (🗑️) ボタンを選択します。
5. 必要に応じて、Kinesis Data Firehose が他のサービスに書き込んだデータを削除します。詳細については、「[the section called “保存されたイベントデータを削除する” \(p. 495\)](#)」を参照してください。

Amazon SES API を使用してイベントの送信先を削除することもできます。次の手順では、AWS Command Line Interface (AWS CLI) を使用して Amazon SES API を操作します。AWS SDK を使用したり、HTTP リクエストを直接行ったりすることで API を操作することもできます。

AWS CLI を使用して Kinesis Data Firehose イベントの送信先を削除するには

1. コマンドラインから、以下のコマンドを入力します。

```
aws ses delete-configuration-set-event-destination --configuration-set-name configSet \
--event-destination-name eventDestination
```

このコマンドで、*configSet* を、Kinesis Data Firehose イベントの送信先が含まれる設定セットの名前に置き換えます。*eventDestination* を Kinesis Data Firehose イベントの送信先の名前に置き換えます。

2. 必要に応じて、Kinesis Data Firehose が他のサービスに書き込んだデータを削除します。詳細については、「[the section called “保存されたイベントデータを削除する” \(p. 495\)](#)」を参照してください。

## 保存されたイベントデータを削除する

他の AWS サービスから情報を削除する方法については、次のドキュメントを参照してください。

- Amazon Simple Storage Service 入門ガイドの「[オブジェクトとバケットの削除](#)」
- Amazon Elasticsearch Service デベロッパーガイドの「[Amazon ES ドメインの削除](#)」
- 『Amazon Redshift クラスター管理ガイド』の「[クラスターの削除](#)」

Kinesis Data Firehose を使用して E メールデータを Splunk (AWS によってサポートされていないが、AWS Management Console で管理されないサードパーティーサービス) にストリーミングすることもできます。Splunk からデータを削除する方法の詳細については、システム管理者に問い合わせるか、[Splunk ウェブサイト](#)でドキュメントを参照してください。

## ID に関するデータを削除する

ID には、Amazon SES を使用して E メールを送信するために使用する E メールアドレスとドメインが含まれています。管轄区域によっては、E メールアドレスやドメインが個人を特定できるデータと見なされることがあります。

Amazon SES コンソールを使用して ID を削除するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. [Identity Management] で、以下のいずれかを行います。
  - ドメインを削除する場合は、[Domains] を選択します。
  - E メールアドレスを削除する場合は、[Email Addresses] を選択します。
3. 削除する ID を選択し、[Remove] を選択します。
4. 確認ダイアログボックスで、[Yes, Delete Identity] を選択します。

Amazon SES API を使用して ID を削除することもできます。次の手順では、AWS Command Line Interface (AWS CLI) を使用して Amazon SES API を操作します。AWS SDK を使用したり、HTTP リクエストを直接行ったりすることで API を操作することもできます。

AWS CLI を使用して ID を削除するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses delete-identity --identity sender@example.com
```

このコマンドで、*sender@example.com* を、削除する ID に置き換えます。

## 送信者認証データを削除する

送信者認証とは、自分の代わりに別のユーザーが E メールを送信できるように Amazon SES を設定するプロセスを指します。送信者認証を有効にするには、「[Amazon SES での送信承認の使用 \(p. 154\)](#)」で説明されているようにポリシーを作成する必要があります。これらのポリシーには、(自分の代わりに E メールを送信するユーザーまたはグループに関連付けられている) AWS ID に加えて、(自分に属する) ID が含まれています。この個人データは、送信者認証ポリシーを変更または削除することで削除することができます。次の手順では、これらのポリシーを削除する方法を示します。

Amazon SES コンソールを使用して送信者認証ポリシーを削除するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. [Identity Management] で、以下のいずれかを行います。
  - 削除する送信者認証ポリシーがドメインと関連付けられている場合は、[Domains] を選択します。
  - 削除する送信者認証ポリシーが E メールアドレスと関連付けられている場合は、[Email Addresses] を選択します。



3. [Identity Policies] で、削除するポリシーを選択し、[Remove Policy] を選択します。

Amazon SES API を使用して送信者認証ポリシーを削除することもできます。次の手順では、AWS Command Line Interface (AWS CLI) を使用して Amazon SES API を操作します。AWS SDK を使用したり、HTTP リクエストを直接行ったりすることで API を操作することもできます。

AWS CLI を使用して送信者認証ポリシーを削除するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses delete-identity-policy --identity example.com --policy-name samplePolicy
```

このコマンドで、*example.com* を、送信者認証ポリシーが含まれている ID に置き換えます。*samplePolicy* を送信者認証ポリシーの名前に置き換えます。

## 受信ルールに関連するデータを削除する

Amazon SES を使用して受信 E メールを受信する場合、1 つ以上の ID (E メールアドレスやドメイン) に適用される受信ルールを作成できます。これらのルールにより、指定された ID に送信された受信メールに対して Amazon SES が行う処理が決まります。

Amazon SES コンソールを使用して受信ルールを削除するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. [Email Receiving] で [Rule Sets] を選択します。
3. 受信ルールがアクティブなルールセットの一部である場合、[View Active Rule Set] を選択します。それ以外の場合は、削除する受信ルールセットが含まれるルールセットを選択します。
4. 受信ルールのリストで、削除するルールを選択します。
5. [Actions] メニューで、[Delete] を選択します。
6. 確認ダイアログボックスで、[Delete] を選択します。

Amazon SES API を使用して受信ルールを削除することもできます。次の手順では、AWS Command Line Interface (AWS CLI) を使用して Amazon SES API を操作します。AWS SDK を使用したり、HTTP リクエストを直接行ったりすることで API を操作することもできます。

AWS CLI を使用して受信ルールを削除するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses delete-receipt-rule --rule-set myRuleSet --rule-name myReceiptRule
```

このコマンドで、*myRuleSet* を、受信ルールが含まれる受信ルールセットの名前に置き換えます。*myReceiptRule* を、削除する受信ルールの名前に置き換えます。

## IP アドレスフィルターに関連するデータを削除する

Amazon SES を使用して受信メールを受信する場合、特定の IP アドレスから送信されるメッセージを明示的に許可またはブロックするフィルターを作成できます。

Amazon SES コンソールを使用して IP アドレスフィルターを削除するには

1. Amazon SES コンソール (<https://console.aws.amazon.com/ses/>) を開きます。
2. [Email Receiving] で [IP Address Filters] を選択します。
3. IP アドレスフィルターのリストで、削除するフィルターを選択し、[Delete] を選択します。

Amazon SES API を使用して IP アドレスフィルターを削除することもできます。次の手順では、AWS Command Line Interface (AWS CLI) を使用して Amazon SES API を操作します。AWS SDK を使用したり、HTTP リクエストを直接行ったりすることで API を操作することもできます。

AWS CLI を使用して IP アドレスフィルターを削除するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses delete-receipt-filter --filter-name IPfilter
```

このコマンドで、*IPfilter* を、削除する IP アドレスフィルターの名前に置き換えます。

## E メールテンプレートのデータを削除する

E メールを送信するために E メールテンプレートを使用する場合、テンプレートを設定した方法によっては、テンプレートに個人データが含まれている可能性があります。たとえば、受信者が詳細を問い合わせることができるように E メールアドレスをテンプレートに追加した可能性があります。

E メールテンプレートを削除するには、Amazon SES API を使用する必要があります。

AWS CLI を使用して E メールテンプレートを削除するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses delete-template --template-name sampleTemplate
```

このコマンドで、*sampleTemplate* を、削除する E メールテンプレートの名前に置き換えます。

## カスタム検証 E メールテンプレートのデータを削除する

新しい E メール送信アドレスを検証するためにカスタムテンプレートを使用する場合、テンプレートを設定した方法によっては、テンプレートに個人データが含まれている可能性があります。たとえば、受信者が詳細を問い合わせることができるように E メールアドレスを検証 E メールテンプレートに追加した可能性があります。

カスタム検証 E メールテンプレートを削除するには、Amazon SES API を使用する必要があります。

AWS CLI を使用してカスタム検証 E メールテンプレートを削除するには

- コマンドラインから、以下のコマンドを入力します。

```
aws ses delete-custom-verification-email-template --template-name verificationEmailTemplate
```

このコマンドで、`verificationEmailTemplate` を、削除するカスタム検証 E メールテンプレートの名前に置き換えます。

## AWS アカウントを解約することですべての個人データを削除する

AWS アカウントを解約することにより、Amazon SES に保存されたすべての個人データを削除することもできます。ただし、このアクションでは、他のすべての AWS サービスに保存した他のすべてのデータ (個人データと非個人データ) も削除されます。

AWS アカウントを解約すると、AWS アカウントのデータは 90 日間保持されます。その保持期間の後、完全に削除されて元に戻すことはできません。

### Warning

すべての AWS サービスおよびリージョンにまたがる AWS アカウントに保存したすべてのデータを完全に削除してよいことがはっきりしている場合を除き、以下の手順を実行しないでください。

AWS Management Console を使用して AWS アカウントを解約することができます。

AWS アカウントを解約するには

1. <https://console.aws.amazon.com/kms> で AWS Management Console を開きます。
2. <https://console.aws.amazon.com/billing/home?#/account> の [Account Settings] (アカウント設定) ページに移動します。

### Warning

次の 2 つのステップでは、すべての AWS リージョンにまたがるすべての AWS サービスに保存したデータがすべて完全に削除されます。

3. [Close Account] で、AWS アカウントを解約した場合の結果について説明する免責事項を読みます。条項に同意する場合、チェックボックスをオンにして、[Close Account] を選択します。
4. 確認ダイアログボックスで [Close Account] を選択します。

# Amazon SES に関するよくある質問 (FAQ)

このセクションでは、Amazon SES の使用に関するよくある質問に回答します。

## よくある質問

- [Amazon SES 送信レビュープロセスに関するよくある質問 \(p. 500\)](#)
- [DNS ブラックホールリスト \(DNSBL\) に関するよくある質問 \(p. 515\)](#)
- [Amazon SES E メール送信メトリクスに関するよくある質問 \(p. 518\)](#)

## Amazon SES 送信レビュープロセスに関するよくある質問

Amazon SES から送信された E メールをモニタリングして、このサービスが悪意のあるコンテンツや迷惑なコンテンツ、または質の低い Eメールの送信に使用されていないことを確認しています。ユーザーがこのようなカテゴリに分類されるコンテンツを送信していると判断した場合、そのアカウントに対してアクションを起こします。このプロセスを送信レビュープロセスと呼びます。

多くの場合、アカウントの問題を検出すると、そのアカウントを[レビュー \(p. 500\)](#)します。[アカウントの Eメール送信機能を一時停止 \(p. 503\)](#)する場合があります。当社では、各アカウントの送信者の評価を保護するため、また、他の Amazon SES ユーザーがサービスの中断や配信性能の問題を経験しないためにこれらのアクションを実行します。

このセクションには、以下のトピックについてのよくある質問が含まれています。

- [アカウントのレビューに関するよくある質問 \(p. 500\)](#)
- [送信一時停止に関するよくある質問 \(p. 503\)](#)
- [バウンスに関するよくある質問 \(p. 505\)](#)
- [苦情に関するよくある質問 \(p. 507\)](#)
- [スパムトラップに関するよくある質問 \(p. 512\)](#)
- [手動調査に関するよくある質問 \(p. 514\)](#)

## アカウントのレビューに関するよくある質問

**Q1. 「お客様のアカウントはレビュー対象です」というメッセージを受け取りました。これはどういう意味ですか？**

AWS はお客様のアカウントから送信された Eメールに関連する問題を検出すると、その修正までの猶予期間を与えます。引き続き通常どおり Eメールを送信できますが、お客様のアカウントがレビュー対象の原因となった問題を修正する必要があります。レビュー期間が終了するまでに問題を修正しない場合は、以後の Eメールの送信機能が一時停止される場合があります。

**Q2. アカウントがレビュー対象になるときは、必ず通知されますか？**

はい。AWS アカウントに関連付けられた Eメールアドレスに通知が送信されます。

### Q3. アカウントがレビュー対象であるという通知を受け取らなかったのはなぜですか？

アカウントがレビュー対象になると、AWS アカウントに関連付けられた E メールアドレスに通知が自動的に送信されます。この E メールアドレスは、AWS アカウントの作成時に指定したものです。Amazon SES を介した Eメールの送信に使用するものと異なる場合があります。

評価ダッシュボード (p. 377) を定期的に参照して、送信者の評価をモニタリングすることをお勧めします。また、[Amazon CloudWatch で自動アラームを設定 \(p. 390\)](#) できます。これらのアラームは、評価メトリクスが特定のしきい値を超えたときに通知を送信します。Amazon CloudWatch は、携帯電話にテキストメッセージを送信するなど、他の方法で連絡するように設定することもできます。

### Q4. Amazon SES アカウントがレビュー対象であることは、他の AWS のサービスの使用に影響を与えますか？

Amazon SES のアカウントがレビュー中でも、他の AWS のサービスを使用できます。ただし、アウトバウンド通信を送信する他の AWS のサービス (Amazon SNS など) の service quota の引き上げを申請した場合、Amazon SES アカウントのレビューが解除されるまでは、その申請は拒否されることがあります。

### Q5. アカウントがレビュー対象である場合、どうすればよいですか？

次のことを実行してください。

- お客様の状況で可能な場合は、問題が解決されるまでメールの送信を停止します。アカウントのレビュー中も Eメールの送信を続行できます。ただし、変更を加えずにメールを送信し続けると、不用意に問題を悪化させる可能性があります。
- 当社から受け取る Eメールをご覧になり、問題の概要について確認してください。
- 送信内容を調べて、どのような送信が原因で問題が発生したのかを具体的に特定します。
- 問題解決のために変更した後に、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージでは、問題を解決するために行ったステップの詳細情報を提供し、これらのステップでどのように今後問題の再発を防ぐことができるかを説明します。
- 当社が具体的に要求するすべての情報を必ず提供してください。お客様の状況を評価するために、これらの情報が必要になります。

### Q6. レビューとは何ですか？

レビュー対象にするという当社の決定を見直すようにリクエストできます。見直しのリクエストに関する詳細については、次の質問を参照してください。

### Q7. 見直しのリクエストをするにはどうすればよいですか？

見直しをリクエストするために、AWS コンソールにサインインし、サポートセンターに移動します。お客様に代わって AWS がオープンしたケースに返信してください。

リクエストで以下の情報を提供してください。

- アカウントがレビュー対象となった根本原因に関する情報。
- 問題修正のために行った変更のリスト。実行済みのステップのみを含め、今後実行する予定のステップは含めないでください。

- これらの変更により今後どのように同じ問題の再発が防止されるかに関する情報。

アカウントがレビュー対象となったイベントの性質によっては、追加情報を提出していただく場合があります。リクエストに含める情報のリストについては、発生した問題に関する、よくある質問のトピックを参照してください。

## Q8. 見直しリクエストが受け入れられない場合どうなりますか？

当社はお客様にリクエストを受け入れなかった理由に関する情報を回答します。問題を解決し、その変更により今後問題の再発を防げることを示すことができれば、場合によっては、別のリクエストを送信することができます。

## Q9. 問題を診断する際にサポートを受けることはできますか？

通常は、問題の概要だけが伝えられます (たとえば、バウンスに関する問題があるなど)。根本原因は、お客様の側で調査する必要があります。

## Q10. アカウントがレビュー対象でなくなったことを確認するには、どうすればよいですか？

評価ダッシュボードにはアカウントの現在のステータスに関する情報が含まれます。詳細については、「[評価ダッシュボードを使用してバウンス率と苦情率を追跡する \(p. 377\)](#)」を参照してください。

## Q11. 問題が存在する場合は必ずアカウントがレビュー対象になりますか。

いいえ。状況によっては、アカウントをまずレビュー対象にするのではなく、Eメール送信機能を一時停止する場合があります。以下に例を示します。

- 問題が非常に深刻な場合。
- お客様のアカウントが過去に何度も同じ問題でレビュー対象になっている場合。そのため、レビュー対象の原因となった個別の問題を解決するだけでなく、根本的な問題に対処することが重要となります。たとえば、特定のキャンペーンが原因でアカウントがレビュー対象になっているのであれば、そのキャンペーンを単に停止する以上のことを実行する必要があります。キャンペーンのどのような特徴が問題となっていたかを特定し、今後のキャンペーンで同じ問題が発生しないようにするプロセスを構築する必要があります。

いずれの場合でも、アカウントのEメール送信機能を一時停止する際には、自動的に通知を送信します。

## Q12. レビュー対象の期限が切れる直前に問題を解決した場合どうなりますか？

AWS コンソールにサインインし、サポートセンターに移動します。お客様に代わって AWS がオープンしたケースに返信してください。ケースへの返信で、問題を解決したことを AWS に通知します。

## Q13. AWS の担当者やプレミアムサポートからサポートを受けることはできますか？

AWS のアカウント担当者がすでに決まっている場合は、お客様のアカウントがレビュー対象になったことが、その担当者に自動的に通知されます。アカウント担当者は、問題の把握に役立つ追加情報をお客様に提供できる場合があります。プレミアムサポートを利用している場合は、そのチームに連絡してさらなるサポートを受ける必要もあります。

## 送信一時停止に関するよくある質問

### Q1. 「アカウントの E メール送信機能を一時停止します」というメッセージを受信しました。これはどういう意味ですか？

お客様が送信した E メールに重大な問題があるため、お客様のアカウントの E メール送信機能を一時停止しました。ほとんどの場合、アカウントの一時停止は、次のいずれかの理由によります。

- 以前にアカウントがレビュー対象になっていた。アカウントがレビュー対象になった原因がレビュー期間の終了までに修正されなかったため、アカウントの E メール送信を一時停止しました。
- アカウントは同じ問題で何度もレビュー対象になっている。
- アカウントが [AWS のサービス利用条件](#) に違反した E メールを送信した。これらの違反が深刻な場合、アカウントをまずレビュー対象にするのではなく、E メール送信機能を一時停止する可能性があります。

### Q2. アカウントの E メール送信機能が一時停止されるときは、必ず通知されますか。

はい。AWS アカウントに関連付けられた E メールアドレスに通知が送信されます。

### Q3. アカウントの E メール送信機能が一時停止されています。なぜ通知がなかったのでしょうか？

アカウントの E メール送信機能を一時停止する場合は、そのアカウントに関連付けられている E メールアドレスに通知を送信します。

#### Note

AWS アカウントを作成するときに、E メールアドレスを指定する必要があります。このアドレスはいつでも変更できます。AWS アカウントに関連付けられているアドレスを変更する詳しい方法については、「[AWS アカウントの管理](#)」(AWS Billing and Cost Management ユーザーガイド)を参照してください。

バウンス率や苦情率が高すぎるときに通知するアラームを Amazon CloudWatch で作成できます。アラームを作成すると、アカウントの E メール送信機能の一時停止に至るような要因について早い段階で警告を受けることができます。ただし、バウンスや苦情以外にも E メール送信機能を一時停止する要因があります。CloudWatch アラームの作成の詳細については、「[CloudWatch を使用して評価モニタリングアラームを作成する \(p. 390\)](#)」を参照してください。

[配信性能ダッシュボード \(p. 377\)](#) を使用してアカウントの現在のステータスを確認することもできます。たとえば、アカウントの E メール送信機能が現在一時停止になっている場合、配信性能ダッシュボードの [Account status (アカウントステータス)] セクションに、ステータスとして [SENDING PAUSE (送信一時停止)] と表示されます。アカウントで正常に E メールを送信できる場合は、[HEALTHY (正常)] と表示されます。

最後に、<https://phd.aws.amazon.com/> で AWS Personal Health Dashboard (PHD) をチェックし、アカウントの E メール送信機能が現在一時停止されているかどうかを確認できます。アカウントの E メール送信機能が一時停止されると、PHD の [Event log (イベントログ)] セクションに [SES sending paused (SES 送信一時停止)] イベントが自動的に追加されます。[SES sending paused (SES 送信一時停止)] イベントのステータスは、アカウントの E メール送信機能が現在一時停止されているかどうかに関係なく、常に [Closed (クローズ)] です。イベントログには、送信一時停止イベントの発生時に AWS アカウントに関連付けられていた E メールアドレスに送信された Eメールのコピーも含まれます。

個人用 Health ダッシュボードに新しいイベントが表示されたときにアラートするアラームを CloudWatch で作成できます。詳細については、AWS Health ユーザーガイドの「[CloudWatch Eventsを使用した AWS Health イベントのモニタリング](#)」を参照してください。

## Q4. アカウントの E メール送信機能が一時停止されています。これによって、他の AWS のサービスの使用に影響がありますか。

アカウントの E メール送信機能の一時停止中にも他の AWS のサービスを使用することもできます。ただし、アウトバウンド通信を送信する他の AWS のサービス (Amazon SNS など) で service quota の引き上げを申請した場合、アカウントの E メール送信機能が回復するまで、その申請は拒否されることがあります。

## Q5. アカウントの E メール送信機能が一時停止されるときは、どうすればよいですか？

次のことを実行してください。

- 当社から受け取る E メールをご覧になり、問題の概要について確認してください。
- 送信内容を調べて、どのような送信が原因で問題が発生したのかを具体的に特定します。
- 問題解決のために変更した後に、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージでは、問題を解決するために行ったステップの詳細情報を提供し、これらのステップでどのように今後問題の再発を防ぐことができるかを説明します。
- 当社が具体的に要求するすべての情報を必ず提供してください。お客様の状況を評価するために、これらの情報が必要になります。

## Q6. レビューとは何ですか？

レビュー対象にするという当社の決定を見直すようにリクエストできます。見直しのリクエストに関する詳細については、次の質問を参照してください。

## Q7. 見直しのリクエストをするにはどうすればよいですか？

見直しをリクエストするために、AWS コンソールにサインインし、サポートセンターに移動します。お客様に代わって AWS がオープンしたケースに返信してください。

リクエストで以下の情報を提供してください。

- この問題の原因に関する情報。
- 問題修正のために行った変更のリスト。実行済みのステップのみを含め、今後実行する予定のステップは含めないでください。
- これらの変更により今後どのように同じ問題の再発を防ぐことができるかに関する情報。

アカウントの E メール送信機能が一時停止となったイベントの性質によっては、追加情報を提出していただく場合があります。リクエストに含める情報のリストについては、発生した問題に関する、よくある質問のトピックを参照してください。

## Q8. リクエストが受け入れられない場合どうなりますか？

当社のお客様にリクエストを受け入れなかった理由に関する情報を回答します。問題を解決し、その変更により今後問題の再発を防げることを示すことができれば、場合によっては、別のリクエストを送信することができます。



## Q9. 問題を診断する際にサポートを受けることはできますか？

通常は、問題の概要だけが伝えられます (たとえば、バウンスに関する問題があるなど)。問題を解決するのは、お客様の責任です。

## Q10. アカウントの E メール送信機能が回復したことはどうすればわかりますか？

評価ダッシュボードにはアカウントの現在のステータスに関する情報が含まれます。詳細については、「[評価ダッシュボードを使用してバウンス率と苦情率を追跡する \(p. 377\)](#)」を参照してください。

## Q11. AWS の担当者やプレミアムサポートからサポートを受けることはできますか？

AWS のアカウント担当者がすでに決まっている場合は、お客様のアカウントの E メール送信機能が一時停止になると、その担当者に自動的に通知されます。アカウント担当者は、問題の把握に役立つ追加情報をお客様に提供できる場合があります。プレミアムサポートを利用している場合は、そのチームに連絡してさらなるサポートを受ける必要もあります。

## バウンスに関するよくある質問

### Q1. なぜバウンスを重視しているのですか？

高いバウンス率は、E メールプロバイダーやアンチ スパム組織などのエンティティで、不良な E メール送信プラクティスに参与している送信者を検出するためによく使用されます。高いバウンス率は、E メールが受信トレイではなく迷惑メールフォルダに送信される原因になる可能性があります。

### Q2. アカウントのバウンス率により、アカウントがレビュー対象になっている、または送信が一時停止になっている、という通知を受信した場合、どうすればよいですか？

問題の原因を特定し、修正します。問題解決のために変更した後に、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージでは、問題を解決するために行ったステップの詳細情報を提供し、これらのステップでどのよう今後問題の再発を防ぐことができるかを説明します。次の情報も含めてください。

- バウンスの追跡で使用する方法。
- 新しい受信者へ送信する前に、新しい受信者の E メールアドレスが有効であることを検証する方法。たとえば、「[Q11. バウンスを最小限に抑えるにはどうすればよいですか? \(p. 507\)](#)」で実行する推奨事項などがあります。

### Q3. どのような種類のバウンスがバウンス率に含まれますか？

バウンス率には、まだ確認していないドメインに対するハードバウンスのみが含まれます。ハードバウンスは、「アドレスは存在しません」などの永続的な配信障害です。「メールボックスがいっぱいです」などの一時的かつ断続的な障害や、IP アドレスのブロックによるバウンスは、バウンス率にカウントされません。

## Q4. アカウントのレビューや、送信の一時停止の原因になりえるバウンス率は公開されていますか？

最良の結果を得るには、バウンス率を 2% 未満に維持する必要があります。これより高いバウンス率は、Eメールの配信に影響する可能性があります。

バウンス率が 5% 以上になると、アカウントはレビュー対象になります。バウンス率が 10% 以上の場合は、高いバウンス率の原因となった問題が解決するまで、以後の Eメール送信を一時停止することがあります。

## Q5. バウンス率の計算対象となる期間はどれくらいですか？

バウンス率の計算は固定期間に基づいて行われるわけではありません。これは、さまざまな送信者が異なる割合で送信を行うためです。代わりに、代表ボリューム (典型的な通常の Eメール送信量) を調べます。大量のメールの送信者と少量のメールの送信者を公平に扱うために、代表的ボリュームはユーザーごとに異なっており、ユーザーの送信パターンの変化に伴い代表的ボリュームも変わります。

## Q6. Amazon SES コンソールや GetSendStatistics API の情報を使用して、自分でバウンス率を計算できますか？

いいえ。バウンス率は、代表ボリュームを使用して計算されます (「[Q5. バウンス率の計算対象となる期間はどれくらいですか? \(p. 506\)](#)」を参照)。送信割合によっては、Amazon SES コンソールや GetSendStatistics で取得できる期間よりも過去にさかのぼった期間のバウンス率になる場合があります。また、バウンス率を計算するときには考慮されるのは、検証されていないドメインへの Eメールのみです。ただし、これらのメソッドを使用して定期的にバウンス率をモニタリングしている場合は、問題が発生してアカウントがレビュー対象になる、または、アカウントの Eメール送信機能が一時停止される前に問題を見つける適切な指標になります。

## Q7. どの Eメールアドレスがバウンスされたかを確認するには、どうすればよいですか？

Amazon SES から送信されたバウンス通知を調べます。Amazon SES が通知を転送する先の Eメールアドレスは、「[Eメールで送信された Amazon SES 通知 \(p. 274\)](#)」に説明されているように、元のメッセージの送信方法によって異なります。また、「[通知を使用した Amazon SES Eメール送信のモニタリング \(p. 273\)](#)」で説明されているように、Amazon Simple Notification Service (Amazon SNS) からバウンス通知をセットアップすることもできます。バウンスされたアドレスを詳しく調べずにリストから削除すると、根本的な問題が解決できなくなる場合があるので、注意してください。バウンスを減らすために実行できる作業については、「[Q11. バウンスを最小限に抑えるにはどうすればよいですか? \(p. 507\)](#)」を参照してください。

## Q8. バウンスをモニタリングしていない場合、バウンスされたアドレスのリストは提供されますか？

いいえ。バウンスされたアドレスの完全なリストを提供することはできません。アカウントのバウンスをモニタリングし、それに応じて行動するのは、お客様の責任です。

## Q9. バウンスはどのように処理すればよいですか？

バウンスされたアドレスをメーリングリストから削除し、それらのアドレスへのメール送信をすぐに停止する必要があります。送信メールが少量であれば、場合によっては、Eメールを使用してバウンスをモニタリングし、バウンスされたアドレスをメーリングリストから手動で削除するだけで十分です。もしボリュームが多ければ、場合によっては、このプロセスのオートメーションをセットアップする必要があります。

まず、そのためには、バウンスを受信したメールボックスをプログラムで処理するか、Amazon SNS を経由するようにバウンス通知を設定します。詳細については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。

## Q10. 送信クォータに達したために、E メールがバウンスされる場合がありますか？

いいえ。バウンスは送信クォータとは関係ありません。送信クォータを超えて E メールを送信しようとすると、Amazon SES API または SMTP インターフェイスからエラーが送信されます。

## Q11. バウンスを最小限に抑えるにはどうすればよいですか？

まず、バウンスの状況について把握してください（「[Q7. どの E メールアドレスがバウンスされたかを確認するには、どうすればよいですか? \(p. 506\)](#)」を参照）。そして、次のガイドラインに従ってください。

- E メールアドレスの購入、貸与、共有は行わないでください。Eメールの受信を明示的にリクエストした受信者にのみ E メールを送信します。
- バウンスされた E メールアドレスをリストから削除します。
- ユーザーは、ウェブのフォームで、E メールアドレスを 2 回入力してフォーム送信前に両方のアドレスが一致していることをチェックするように求められます。
- ダブルオプトインを使用して、新しいユーザーをサインアップします。つまり、新しいユーザーは、サインアップ後に確認メールを受信します。その確認メールで確認のクリックをすると、その他のメールを受け取ることができます。これにより、他のユーザーのサインアップをしたり、誤ってサインアップしたりすることを回避できます。
- 最近メールを送信していないアドレスに送信する必要がある場合（つまり、そのアドレスがまだ有効であるかどうかを確認できていない場合）、すべての送信内容のごく一部だけを、そのアドレスに送信します。詳細については、ブログ投稿「[古いアドレスには送信しないこと。ただし、送信する必要がある場合はどうするか?](#)」を参照してください。
- 架空のアドレスの使用を推奨するようなサインアップを構築していないことを確認します。たとえば、付加価値や特典は受信者の確認済みアドレスにのみ与えるようにしてください。
- 「友だちにメールを送る」機能を取り入れている場合は、CAPTCHA や同様のメカニズムを使用して、この機能が自動的に使用されることを防いでください。また、ユーザーが自由にコンテンツを挿入することを許可しないでください。
- システム通知用に Amazon SES を使用している場合は、メールを受信できる実在のアドレスに通知を送信していることを確認してください。また、不要な通知は無効にすることを検討してください。
- 新しいシステムをテストしている場合は、E メールを受信できる実在のアドレスに送信していること、または Amazon SES メールボックスシミュレーターを使用していることを確認してください。詳細については、「[Amazon SES での E メール送信のテスト \(p. 187\)](#)」を参照してください。

## 苦情に関するよくある質問

### Q1. 苦情とは何ですか？

受信者が Eメールの受信をリクエストしていないことを報告した場合に、苦情が発生します。受信者が Eメールクライアントの [Report spam (スパムの報告)] のボタンをクリックした場合、Eメールプロバイダーに苦情を報告した場合、Amazon SES に直接または他の手段で通知した場合、などです。このトピックには、苦情に関する全般的な情報が含まれています。通知に、苦情のソースに関する特定の情報が含まれている場合は、関連トピック「[フィードバックループ経由で Amazon SES に寄せられた苦情に関するよくある質問 \(p. 508\)](#)」、「[受信者から直接 Amazon SES に寄せられた苦情に関するよくある質問 \(p. 510\)](#)」、または「[Eメールプロバイダー経由で Amazon SES に寄せられた苦情に関するよくある質問 \(p. 511\)](#)」も参照してください。

## Q2. なぜ苦情を重視しているのですか？

高い苦情率は、E メールプロバイダー、アンチスパム組織などの団体によって頻繁に使用されます。このような苦情率は、Eメールの受信にサインアップしていない受信者に送信者がメールを送ったことや、送信者が送ったコンテンツは受信者がサインアップしたコンテンツのタイプと異なっていることを示す指標として利用されます。

## Q3. 苦情が絡む問題により、アカウントがレビュー対象になるという通知や、送信が一時停止になるという通知を受信した場合、どうすればよいですか？

お客様のリスト取得プロセスと Eメールの内容を確認して、受取人がお客様からの Eメールを受け取りたくない理由を確認します。問題の原因を特定し、修正します。問題解決のために変更した後に、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージでは、問題を解決するために行ったステップの詳細情報を提供し、これらのステップでどのように今後問題の再発を防ぐことができるかを説明します。

## Q4. 苦情を最小限に抑えるにはどうすればよいですか？

最初に、Amazon SES から通知される苦情を必ずモニタリングします。Amazon SES は、フィードバックループ (「[フィードバックループ経由で Amazon SES に寄せられた苦情に関するよくある質問 \(p. 508\)](#)」を参照) を通じて苦情を受け取ります。そして、次のガイドラインに従ってください。

- Eメールアドレスの購入、貸与、共有は行わないでください。お客様からのメールを具体的にリクエストしたアドレスのみを使用します。
- ダブルオプトインを使用して、新しいユーザーをサインアップします。つまり、ユーザーは、サインアップ後に確認メールを受信します。その確認メールで確認のクリックをすると、その他のメールを受け取ることができます。これにより、他のユーザーのサインアップをしたり、誤ってサインアップしたりすることを回避できます。
- 送信したメールへの対応状況をモニタリングし、メッセージを開かなかった受信者やクリックしなかった受信者への送信を停止します。
- 新しいユーザーがサインアップするとき、ユーザーが受信する Eメールの種類を明確にします。また、ユーザーがサインアップしたときに指定した種類のメールだけを送信するようにします。たとえば、ユーザーがニュースの更新にサインアップした場合は、広告を送信しないでください。
- メールが適切な形式になっており、見た目の質も高いことを確認します。
- お客様からのメールであることが明確で、他のメールと混同されないことを確認します。
- ユーザーがメールの受信登録を解除するとき、わかりやすく簡単な方法で実行できるようにします。

## フィードバックループ経由で Amazon SES に寄せられた苦情に関するよくある質問

このトピックでは、フィードバックループを通じて Eメールプロバイダーから Amazon SES が受け取った苦情に関する情報を提供します。すべての種類の苦情に適用される一般的情報については、「[苦情に関するよくある質問 \(p. 507\)](#)」を参照してください。

### Q1. この種類の苦情はどのように報告されていますか？

ほとんどの Eメールクライアントプログラムには、メッセージをスパムフォルダに移動して Eメールプロバイダーに転送するためのボタン ([Mark as Spam (スパムとしてマーク)] など) が用意されています。また、ほとんどのプロバイダーでは、ユーザーが不要な Eメールを転送して Eメールプロバイダーによる防止策をリクエストできる迷惑メール用アドレス (abuse@example.com など) を用意しています。Eメール

ルプロバイダーとのフィードバックループ (FBL) を Amazon SES で設定している場合、苦情は Amazon SES に転送されます。

## Q2. これらの苦情は、Amazon SES コンソールにある苦情率の統計に含まれていて、GetSendStatistics API で返されますか？

はい。ただし、苦情率の統計には、Amazon SES にフィードバックを提供しない E メールプロバイダーからの苦情は含まれません。フィードバックを提供するドメインでの苦情率は、他の送信を代表しているものと考えることができます。

## Q3. これらの苦情はどのようにして通知されますか？

E メールまたは Amazon SNS 通知を通じて通知を受けることができます。セットアップ手順については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。

## Q4. E メールまたは Amazon SNS を通じて苦情の通知を受け取った場合はどうすればよいですか？

最初に、苦情が発生したアドレスをメーリングリストから削除し、それらのアドレスへのメール送信をすぐに停止する必要があります。受信解除のリクエストを受け取ったことを示す Eメールの送信も行わないでください。このプロセスのオートメーションのセットアップを検討してください。そのためには、苦情を受信したメールボックスをプログラムで処理するか、Amazon SNS を経由して苦情通知を設定するかします。詳細については、「[通知を使用した Amazon SES E メール送信のモニタリング \(p. 273\)](#)」を参照してください。

次に、送信状況を詳細に調べて、お客様から送信されたメールを受信者が快く思わない理由を判断し、根本的な問題に対処します。苦情を持つすべてのユーザーの中には、お客様のメールを快く思わなかったけれども、苦情を申し出なかった (または苦情を申し出ることができなかった) 受信者が多数いる可能性があります。実際に苦情を申し出た受信者を削除しただけでは、根本的な問題に対処したことにはなりません。

## Q5. アカウントがレビュー対象になったり、アカウントの E メール送信機能が一時停止されたりする原因となり得る Amazon SES 苦情率は公開されていますか？

最良の結果を得るには、苦情率を 0.1% 未満に維持する必要があります。これより高い苦情率は、Eメールの配信に影響する可能性があります。

苦情率が 0.1% 以上になると、アカウントはレビュー対象になります。苦情率が 0.5% 以上の場合は、高い苦情率の原因となった問題が解決するまで、以後の Eメール送信を一時停止することがあります。

## Q6. 苦情率の計算対象となる期間はどれくらいですか？

苦情率の計算は固定期間に基づいて行われるわけではありません。これは、さまざまな送信者が異なる割合で送信を行うためです。代わりに、代表ボリューム (典型的な通常のメール送信量) を調べます。大量のメールの送信者と少量のメールの送信者を公平に扱うために、代表的ボリュームはユーザーごとに異なっており、ユーザーの送信パターンの変化に伴い代表的ボリュームも変わります。また、苦情率はすべての Eメールに基づいて計算されるわけではありません。苦情率は、苦情のフィードバックを Amazon SES に送っているドメインを対象としており、それらのドメインに送信されたメールに対する苦情の割合として計算されます。

## Q7. Amazon SES コンソールや GetSendStatistics API のメトリクスを使用して、自分で苦情率を計算できますか？

いいえ。その主な理由には、次の 2 つがあります。

- 苦情率は、代表ボリュームを使用して計算されます (「[Q6. 苦情率の計算対象となる期間はどれくらいですか? \(p. 509\)](#)」を参照)。送信割合によっては、Amazon SES コンソールや GetSendStatistics API で取得できる期間よりも過去にさかのぼった期間の苦情率になる場合があります。

ます。このため、これらのメソッドを定期的を使用して、アカウントの苦情率をモニタリングすることをお勧めします。このように苦情率をモニタリングすることで、Eメールの配信に影響を及ぼす前に、問題を特定するための必要な情報を取得できます。

- 苦情率を計算するとき、すべてのEメールが計算対象になるわけではありません。苦情率は、苦情のフィードバックをAmazon SESに送るドメインを対象としており、それらのドメインに送信されたメールに対する苦情の割合として計算されます。

## Q8. どのEメールアドレスに苦情があったかを確認するには、どうすればよいですか？

EメールまたはAmazon SNSを通じてAmazon SESが送信する苦情の通知を確認します(「[通知を使用した Amazon SES Eメール送信のモニタリング \(p. 273\)](#)」を参照)。ただし、Eメールプロバイダーごとに提供される情報の量は異なり、一部のプロバイダーは苦情を送信した受信者のEメールアドレスを編集してから苦情通知をAmazon SESに渡します。今後、受信者のEメールアドレスを特定できるようにする最善の方法は、Amazon SESがEメールを受け付けるときにお客様に渡すAmazon SESメッセージIDと識別子のマッピングを保存することです。カスタムメッセージIDを追加した場合は、Amazon SESに保持されないことに注意してください。

## Q9. 苦情をモニタリングしていない場合、苦情があったアドレスのリストは提供されますか？

残念ながら、弊社では、包括的なリストを提供することはできません。ただし、EメールやAmazon SNSを利用して、将来の苦情をモニタリングすることはできます。

## Q10. Eメールのサンプルを入手できますか？

リクエストに応じてEメールのサンプルをお送りすることはできませんが、この情報は苦情通知に含まれている場合があります。詳細については、「[Q8. どのEメールアドレスに苦情があったかを確認するには、どうすればよいですか? \(p. 510\)](#)」を参照してください。

## 受信者から直接 Amazon SES に寄せられた苦情に関するよくある質問

このトピックでは、受信者が直接Amazon SESに送付した苦情に関する情報を示します。すべての種類の苦情に適用される一般的な情報については、「[苦情に関するよくある質問 \(p. 507\)](#)」を参照してください。

### Q1. この種類の苦情はどのように報告されていますか？

Eメールまたはその他の方法を通じて、複数の受信者がお客様のメールに関してAmazon SESに連絡しました。

### Q2. これらの苦情は、Amazon SES コンソールにある苦情率の統計に含まれていて、GetSendStatistics API で返されますか？

いいえ。Amazon SES コンソールまたはGetSendStatistics APIを使用して取得する苦情率の統計には、フィードバックループを通じてAmazon SESが受け取った苦情のみが含まれます。これらの苦情のタイプの詳細については、「[フィードバックループ経由で Amazon SES に寄せられた苦情に関するよくある質問 \(p. 508\)](#)」を参照してください。

### Q3. Eメールのフィードバック通知またはAmazon SNSを通じて、これらの苦情について通知されないのはなぜですか？

Eメールのフィードバック転送およびAmazon SNSの通知には、フィードバックループを通じてAmazon SESが受け取る苦情のみが含まれます。受信者から直接Amazon SESに寄せられた苦情については通知されません。

#### Q4. どの E メールアドレスに苦情があったかを確認するには、どうすればよいですか？

苦情元の受信者の ID を保護するために、お客様の E メールについて苦情を申し出た受信者の E メールアドレスのリストを提供することはできません。

お客様のリストからそのような受信者を削除することに注力するのではなく、苦情が寄せられる原因となった問題を特定することをお勧めします。まず、顧客獲得プロセスを見直し、お客様からの Eメールの受信を明示的にリクエストしていない顧客をリストから削除することをお勧めします。Eメールの内容を分析して、受信者が苦情を申し出ている理由の把握に努めてください。

#### Q5. Eメールのサンプルを入手できますか？

苦情元の受信者の ID を保護するために、受信者が苦情を申し出る原因となった Eメールのコピーを提供することはできません。

#### Q6. 直接的な苦情により、アカウントがレビュー対象である、または送信が一時停止になるという通知を受信した場合、どうすればよいですか？

送信プロセスをすぐに変更して、お客様からのメッセージの受信にサインアップした受信者にのみメッセージを送信するようにします。また、受信者がサインアップした受信コンテンツタイプを送信するようにしてください。問題解決のために変更した後に、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージでは、問題を解決するために行ったステップの詳細情報を提供し、これらのステップでどのように今後問題の再発を防ぐことができるかを説明します。

お客様が3週間以内に見直しをリクエストせず、かつ受信者から直接苦情が送信され続ける場合、お客様のアカウントの Eメール送信機能を一時停止することがあります。

## Eメールプロバイダー経由で Amazon SES に寄せられた苦情に関するよくある質問

このトピックでは、Eメールプロバイダー(メールボックスプロバイダーとも呼ばれる)を介して Amazon SES に寄せられる苦情に関する情報を提供します。すべての種類の苦情に適用される一般的な情報については、「[苦情に関するよくある質問 \(p. 507\)](#)」を参照してください。

#### Q1. この種類の苦情はどのように報告されていますか？

非常に多くの数の顧客が、お客様の Eメールをスパムとマークしたことが Eメールプロバイダーから Amazon SES に報告されました。この報告は、「[フィードバックループ経由で Amazon SES に寄せられた苦情に関するよくある質問 \(p. 508\)](#)」で説明しているフィードバックループ以外の方法で Amazon SES に提供されました。

#### Q2. これらの苦情は、Amazon SES コンソールにある苦情率の統計に含まれていて、GetSendStatistics API で返されますか？

いいえ。Amazon SES コンソールまたは GetSendStatistics API を使用して取得する苦情率の統計には、フィードバックループを通じて Amazon SES が受け取った苦情のみが含まれます。

#### Q3. Eメールのフィードバック通知または Amazon SNS を通じて、これらの苦情について通知されないのはなぜですか？

Eメールのフィードバック転送および Amazon SNS の通知には、フィードバックループを通じて Amazon SES が受け取る苦情のみが含まれます。

#### Q4. どの E メールアドレスに苦情があったかを確認するには、どうすればよいですか？

通常、E メールプロバイダーはこの情報を公開していません。ただし、リストから個別の受信者を削除することではなく、根本的な問題の発見と修正に重点を置く必要があります。お客様のリスト取得プロセスと E メールの内容を確認して、受取人がお客様の E メールを受け取りたくない理由を特定することから開始します。

#### Q5. Eメールのサンプルを入手できますか？

いいえ。通常、E メールプロバイダーはサンプルの E メールを公開していません。

#### Q6. E メールプロバイダーの苦情により、アカウントがレビュー対象である、または送信が一時停止になるという通知を受信した場合、どうすればよいですか？

問題の原因を特定し、修正します。問題解決のために変更した後に、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージでは、問題を解決するために行ったステップの詳細情報を提供し、これらのステップでどのように今後問題の再発を防ぐことができるかを説明します。お客様が3週間以内に見直しをリクエストせず、かつ引き続き E メールプロバイダーから苦情が送信され続ける場合、お客様のアカウントの今後の E メール送信を一時停止することがあります。

## スパムトラップに関するよくある質問

### Q1. スпамトラップとは何ですか？

スパムトラップは、インターネットサービスプロバイダー (ISP)、E メールプロバイダー、またはアンチスパム組織によって管理される特別な E メールアドレスです。アドレスが E メール受信のサインアップを正当に行っていないため、スパムトラップを管理する組織は、アドレスにメールを送信している者が不審なメール行為を実施しているようであると判断しています。

### Q2. スпамトラップはどのようにセットアップされますか？

スパムトラップのアドレスは複数の方法でセットアップされます。これらのアドレスは、以前は有効であったアドレスから変換されたものですが、長期間にわたって未使用 (およびバウンスの状態) になっています。また、これらのアドレスはスパムトラップとして使用されることのみを目的としてセットアップされています。スパムトラップのアドレスは、推測しにくい通常とは異なるアドレスで、実際のアドレスに近いアドレスになっている場合もあります (たとえば、一般的なドメイン名のスペルを変えたアドレスなど)。常にというわけではありませんが、スパムトラップがさまざまな方法でインターネット上に配置され、世界中に「仕掛けられる」ことがよくあります。

### Q3. スпамトラップに送信したかどうかを、Amazon SES はどのようにして把握することができるのですか？

Amazon SES の送信者によってスパムトラップがヒットされると、スパムトラップを操作している特定の組織が Amazon SES に通知を送信します。

### Q4. Amazon SES はスパムトラップに関するレポートをどのように使用しますか？

レポートを確認します。アカウントがスパムトラップに E メールを送信していると判断された場合、アカウントはレビュー対象になり、根本的な問題を解決するよう求められます。レビュー期間が終了するまで



に問題を修正しない場合、アカウントの E メール送信機能が一時停止されることがあります。スパムトラップの問題が非常に深刻である場合は、アカウントをまずレビュー対象にするのではなく、ただちに E メール送信機能を一時停止する可能性があります。

## Q5. スпамトラップの問題により、アカウントがレビュー対象である、または送信が一時停止になるという通知を受信した場合、どうすればよいですか？

まず、アカウントがレビュー対象になる、または、アカウントの E メール送信機能が一時停止される原因となった問題に対処します。次に、AWS コンソールにサインインし、サポートセンターに移動します。お客様に代わって AWS がオープンしたケースに返信してください。メッセージでは、問題を解決するために行ったステップの詳細情報を提供し、これらのステップでどのように今後問題の再発を防ぐことができるかを説明します。お客様が加えた変更により問題が適切に対処されたと見なされた場合は、アカウントのレビュー期間をキャンセル、または一時停止を解除します。

スパムトラップのヒットが報告される方法に起因して、お客様の加えた変更で問題が解決されたかどうかを判断するのに 3 週間以上かかる場合があります。

## Q6. アカウントがレビュー対象になる、または、アカウントの E メール送信機能が一時停止されるまで許容されるスパムトラップのヒットは何件ですか？

アカウントに対してアクションを実行するまでのスパムトラップの特定のヒット件数は公開されていません。ただし、重要な注意点として、スパムトラップのヒットは少数でも、お客様の送信者としての評価に悪影響を及ぼす可能性があります。そのため、スパムトラップレポートを重視する必要があります。

## Q7. スпамトラップのアドレスは公開されていますか？

いいえ。スパムトラップが有効であるためには、スパムトラップが公開されていないことが重要です。スパムトラップの組織は、スパムトラップのヒットが発生したことだけを公開しており、実際のスパムトラップのアドレスは公開していません。

## Q8. スпамトラップへの送信を回避するにはどうすればよいですか？

スパムトラップに送信するリスクを軽減するには、次のガイドラインに従ってください。

- E メールアドレスの購入、貸与、共有は行わないでください。お客様からのメールを具体的にリクエストしたアドレスのみを使用します。
- ユーザーは、ウェブフォームで、E メールアドレスを 2 回入力してフォーム送信前に両方のアドレスが一致していることを確認するように求められます。
- ダブルオプトインを使用して、新しいユーザーをサインアップします。つまり、ユーザーは、サインアップ後に確認メールを受信します。その確認メールで確認のクリックをすると、その他のメールを受け取ることができます。
- ハードバウンスが発生するアドレスをリストから削除します。これにより、それらのアドレスは、スパムトラップに変換されるよりもかなり前に削除されることになります。
- 受信者の関与の状況をモニタリングし、最近 E メールやウェブサイトを使用していない受信者への送信を停止します。「使用しているユーザー」を決める期間はお客様のユースケースによって異なりますが、一般的に、ユーザーが数か月にわたって E メールを開いていないまたはクリックしていない場合は、そのようなユーザーを削除することを検討してください (ただし、そのユーザーが実際にはメールを希望していることが明確な場合は除きます)。

- リエンゲージメントキャンペーンとして、最近やり取りが途絶えている相手に意図的に連絡する場合は、細心の注意が必要です。こうした活動はリスクが高く、スパムトラップへの送信だけでなく、バウンスや苦情に関連した問題を引き起こす場合が多くあります。
- メーリングリスト全体にオプトインメッセージを送信し、検証リンクをクリックした受取人のみを維持します。この手順では、リストからアクティブでない受取人を削除することに加え、スパムトラップアドレスを削除するのにも役立ちます。ただし、メーリングリストに多くの不正なアドレスが含まれているか、バウンスの問題がすでに発生していると思われる場合、この手法の使用はお勧めしません。アカウントのバウンス率が上昇する可能性があるためです。

## 手動調査に関するよくある質問

### Q1. マニュアルでの調査のため、アカウントがレビュー対象である、または送信が一時停止になるという通知を受信した場合、どうすればよいですか？

Amazon SES の調査で、お客様の送信について重大な問題が発生したことが確認されました。代表的な問題を以下に示します (ただし、これらに限定されません)。

- お客様の送信が、[AWS の適正利用ポリシー \(AUP\)](#) に違反している。
- お客様の E メールが相手にとって迷惑メールになっているようである。
- お客様のコンテンツは、フィッシング (シミュレートされたフィッシングを含む) に関連しています。
- お客様のコンテンツは、Amazon SES でサポートしていないユースケースに関連しています。

問題が解決可能であるとみなされた場合、お客様のアカウントは一定時間レビュー対象になります。アカウントのレビューが行われている間に、E メール送信状況を変更して、問題を修正する必要があります。

問題が解決不能とみなされた、または問題が非常に深刻である場合は、アカウントをまずレビュー対象にするのではなく、E メール送信機能を一時停止する場合があります。

### Q2. どのような問題により、E メール送信においてマニュアルのレビューを実行することになりますか？

アカウントのマニュアルでのレビューを開始する原因となる問題は複数あります。以下の理由がありますが、これらに限定されるものではありません。

- 受信者が Amazon SES に連絡して、お客様のアカウントから送信された E メールについて苦情を申し出た。
- E メール送信のパターンに異常な変更が検出された。
- 当社のスパムフィルタで一般的な未承諾コンテンツまたは、質の低いコンテンツの Eメールの特性が検出された。

お客様のアカウントがレビュー対象になる、または、アカウントの E メール送信機能が一時停止される場合、通知が送信されます。ほとんどの場合、この通知には、問題に関する情報、および実行すべき次のステップに関する情報が含まれています。

### Q3. 「未承諾」 E メールとは何ですか？

未承認 E メールは、受信者が受信を明示的にリクエストしなかった E メールです。これには、受信者がある種類のメール (通知など) にサインアップしたにもかかわらず、異なる種類のメール (広告など) が送信された場合が含まれます。

お客様のアカウントがレビュー対象になる、または、アカウントの E メール送信機能が一時停止される場合、通知が送信されます。「未承諾メールの問題により、これらの内いずれかのアクションが実行されます」という通知を受信した場合、AWS コンソールにサインインし、サポートセンターに移動します。お客様に代わって AWS がオープンしたケースに返信してください。メッセージには、次に示す情報を含めてください。

- お客様が送信したすべてのメッセージは受信者が受信を明示的にリクエストしたものであるかどうか、そして、それらのメッセージが [AWS の適正利用ポリシー](#) に準拠しているかどうか。
- ユーザーがお客様またはウェブサイトと具体的にやり取りする、またはユーザーが E メールを要求するのは異なる方法で E メールアドレスを取得したかどうか。メーリングリストを入手した方法について説明する必要があります。
- 受信登録や受信登録の解除に関するプロセスは、どのように機能しているか。オプトインやオプトアウトのリンクを含める必要があります。

## Q4. 手動レビューのため、アカウントがレビュー対象である、または送信が一時停止になるという通知を受信した場合、どうすればよいですか？

問題の原因を特定し、修正します。問題解決のために変更した後に、AWS コンソールにサインインし、サポートセンターにアクセスします。お客様に代わって AWS がオープンしたケースに返信してください。メッセージでは、問題を解決するために行ったステップの詳細情報を提供し、これらのステップでどのように今後問題の再発を防ぐことができるかを説明します。お客様が加えた変更により問題が適切に対処されたと思われた場合は、アカウントのレビュー期間をキャンセルします。

## Q5. どのような種類の問題が「修正可能」と見なされますか？

一般的に、適切な送信手続きの履歴がある場合、または大部分の送信を継続しながら問題のある送信を削除するステップがある場合を、修正可能な状況と考えています。たとえば、3 つの異なる種類の E メールを送信しており、そのうち 1 種類だけが問題のあるメールであるとき、場合によっては、問題のある送信を停止するだけで、残りの送信を継続することができます。

## Q6. 問題の原因がわからない場合、どうなりますか？

AWS コンソールにサインインし、サポートセンターに移動することができます。お客様に代わって当社がオープンしたケースに返信し、問題の原因となったメールのサンプルをリクエストします。

# DNS ブラックホールリスト (DNSBL) に関するよくある質問

ドメインネームシステムベースのブラックホールリスト(DNSBLs) Realtime List(RBL)、拒否リスト、ブロックリスト、またはブラックリストと呼ばれます。これは、不要な E メールを送信した疑いのある IP アドレスを E メールプロバイダーに知らせるためのものです。

異なる DNSBL は、Eメールの配信性能にさまざまな影響を与えます。このトピックでは、DNSBL が Amazon SES からの Eメールの配信に与える影響と、DNSBL から Amazon SES IP アドレスを削除するためのポリシーについて説明します。

### Note

このトピックは、Eメールプロバイダーが着信メッセージをブロックするために使用する DNSBL に関するものです。以前にバウンスされた Eメールアドレスを所有する受取人に対する Eメール

送信を Amazon SES でブロックする方法については、「[Amazon SES グローバルサブプレッショ  
ンリストの使用 \(p. 199\)](#)」を参照してください。

## Q1. DNSBL は E メール配信にどのような影響を与えますか？

異なる DNSBL は、メッセージの正常な配信に異なる影響を与えます。主要な E メールプロバイダー (Gmail、Hotmail、AOL、Yahoo など) は、Spamhaus が提供する DNSBL など、評価の高いごく少数のブラックリストを認識しているようです。当社の経験によれば、他の DNSBL の影響は少ない傾向がありますが、一部のメールシステムでは特定の DNSBL を他の DNSBL より重視しています。

また、多くの E メールプロバイダーは独自の内部拒否リストを持っています。E メールプロバイダーは、このようなリストを厳重に保護し、一般と共有することはほとんどありません。このようなリストに IP アドレスが追加されると、そのプロバイダーを使用する受信者に E メールを送信する能力に大きな影響を与える可能性があります。

## Q2. IP アドレスは、どのようにして DNSBL に追加されるのですか？

IP アドレスは、いくつかの経路で DNSBL に追加されます。IP アドレスは、スパムトラップに E メールを送信すると、DNSBL に追加されます。スパムトラップとは、人間のユーザーに属さない E メールアドレスです。スパムトラップは、迷惑メールを収集しスパム発信者を特定するためだけに存在します。いくつかの DNSBL は、個別のユーザーからの IP アドレスの送信を受け入れます。IP アドレス範囲全体の送信を受け入れる DNSBL さえあります。他の DNSBL は E メール管理者の協力によって維持され、管理者が自分のシステムを不正使用していると思われる IP アドレスを含めることができます。

## Q3. Amazon SES は、IP アドレスが DNSBL に追加されるのをどのようにして防ぐのですか？

当社のシステムは、不正使用の兆候を探します。IP アドレスが DNSBL に追加される可能性のある送信パターンやその他の特性を検出した場合は、送信者に通知を送信します。状況が深刻な場合、または通知を送信した後に送信者が問題を修正しない場合、送信者が問題を解決するまで E メールを送信する機能が一時停止されます。このように送信ポリシーを適用すると、Amazon の IP アドレスが DNSBL に載せられる可能性が低くなります。

## Q4. Amazon SES は IP アドレスを DNSBL から削除できますか？

Amazon では、Amazon SES サービス全体の配信に影響を与える可能性のあるブラックリストや、Gmail、Yahoo、AOL、Hotmail などの主要な E メールプロバイダーを使用する受信者に電子メールを送信する機能に影響を与える可能性のある DNSBL を積極的にモニタリングします。Spamhaus が提供する DNSBL はこのカテゴリに分類されます。これらのいずれかの条件を満たすリストに当社の IP アドレスの 1 つが表示された場合、そのアドレスが DNSBL からできるだけ早く削除されるように迅速に対応します。

Amazon SES サービス全体の配信に影響を与える可能性の低い DNSBL や、主要な E メールプロバイダーへの配信に大きな影響を与えない DNSBL はモニタリングしません。SORBS と UCEPROTECT が提供する DNSBL は、このカテゴリに分類されます。これらのリストを運営するベンダーはリストへの追加やリストからの削除に関して特定の方法に従っているため、これらのリストから当社の IP アドレスを削除することはできません。

## Q5. 送信側 IP アドレスが Spamhaus 以外の DNSBL に追加されているという理由で、E メールプロバイダーから E メールを拒否されています。どうすればよいですか？

まず、メッセージがブロックされている原因が確かに DNSBL であるかを確認します。送信側 IP アドレスが DNSBL に追加されていることが E メール拒否の原因である場合は、通常、次の例のように DNSBL プロバイダー名が記載されたバウンス通知が届きます。

```
554 5.7.1 Service unavailable; Client host [192.0.2.0] blocked using DNSBLName;  
See: http://www.example.com/query/ip/192.0.2.0
```

バウンス通知を受信しても上の例に示すようなメッセージとは異なる場合、E メールプロバイダーがメッセージを拒否した理由は DNSBL 追加とは関係ないとほぼ断定できます。

送信側 IP アドレスが DNSBL に追加されているために E メールプロバイダーから E メールをブロックされたことが確認できる場合、いくつかの対処方法があります。

- メッセージを拒否したドメインのポストマスターに連絡し、スパムフィルタリングポリシーからの例外をリクエストします。一部のポストマスターにはサポートプロセスがあり、このプロセスを説明するポストマスターページを公開していることがあります。連絡しようとしているドメインがポストマスターサポートポリシーを公開していない場合は、[postmaster@example.com](mailto:postmaster@example.com) に E メールを送信してポストマスターに連絡できる場合があります。ここで、[example.com](http://www.example.com) は問題となっているドメインです。ドメインは、ポストマスターのメールボックスを持つことが「RFC 5321」で要求されています。

ポストマスターに連絡するときは、受信したバウンスコード、送信しようとしている Eメールのヘッダー、DNSBL が Eメールの配信に及ぼす影響の測定値、Eメールが誤って DNSBL されていると信じる理由の情報を説明します。正当な Eメールを送信していることを示すためにポストマスターに提供できる情報が多くなればなるほど、ポストマスターは例外を作成する可能性が高くなります。

- Eメールプロバイダーが対応しない場合、またはポリシーを変更する意思がない場合は、[専用 IP アドレス \(p. 178\)](#)の使用を検討します。専用 IP アドレスはお客様専用のアドレスです。適切な送信プラクティスを実装することで、高いエンゲージメント率を維持し、バウンス、苦情、およびスパムトラップのヒット数を低く抑えることができます。適切な送信プラクティスは、アドレスが DNSBL に追加されるのを阻止できます。

## Q6. Gmail、Yahoo、Hotmail などの主要なプロバイダーに送信した E メールがスパムフォルダに追加されます。これは、送信側 IP アドレスが DNSBL に追加されていることが原因ですか？

おそらくそうではありません。大きな影響を与える DNSBL (Spamhaus のいずれかの DNSBL など) に IP アドレスが含まれている場合、主要な Eメールプロバイダーは、この IP アドレスをスパムメールフォルダに送信せずに、この IP アドレスからの Eメールを完全に拒否します。

通常、主要な Eメールプロバイダーは、Eメールを拒否せずに受け入れる場合、ユーザーエンゲージメントを考慮して Eメールを受信トレイに入れるか、スパムフォルダに入れるかを判断します。ユーザーエンゲージメントとは、ユーザーが以前に受信したメッセージを操作した方法を指します。

メッセージがユーザーの受信トレイに届く確率を高めるには、以下のすべてのベストプラクティスを実行してください。

- E メールアドレスのリストは、絶対にレンタルまたは購入しません。リストのレンタルや購入は [AWS 利用ポリシー](#) (AUP) 違反であり、状況の如何を問わず、Amazon SES では許可されません。
- Eメールの受信を明示的に希望したユーザーにのみ Eメールを送信します。世界中の多くの国や管轄区域では、Eメールの受信に明示的に同意していない受取人に Eメールを送信することは違法です。
- 過去 30~90 日間に送信したメッセージを開いていないユーザーやメッセージ内のリンクをクリックしていないユーザーには、Eメールの送信を停止します。このステップは、エンゲージメント率を高く保つために役立ちます。これにより、今後送信するメッセージが受取人の受信トレイに到着する可能性が高くなります。
- 送信する各メッセージの設計要素と記述スタイルを統一し、ユーザーがメッセージの送信元を簡単に識別できるようにします。
- [SPF \(p. 134\)](#) や [DKIM \(p. 135\)](#) などの Eメール認証メカニズムを使用します。
- ユーザーがウェブフォームを使用してコンテンツにサブスクライブする場合、Eメールをユーザーに送信し、今後の Eメールの受信をユーザーが希望することを確認します。ユーザーが Eメールの受信を希望することを確認するまでは、追加の Eメールを送信しません。このプロセスは、確定オプトインまたはダブルオプトインとして知られています。
- ユーザーによるサブスクリプション解除手続きを容易にし、サブスクリプション解除リクエストを即座に了承します。
- リンクを含む Eメールを送信する場合は、リンクを Spamhaus Domain Block List (DBL) に対してテストして確認します。リンクをテストするには、Spamhaus ウェブサイトの [Domain Lookup Tool](#) を使用します。

これらのプラクティスを実装することで、送信者の評価を向上させることができます。これにより、送信する Eメールが受信者の受信トレイに届く可能性が高まります。これらのプラクティスを実装することで、アカウントのバウンス率や苦情率を低く抑えることができ、Eメールがスパムトラップに送信されるリスクを低減できます。

## Amazon SES Eメール送信メトリクスに関するよくある質問

Amazon SES は、Eメール送信に関するいくつかのメトリクスを収集します。これらのメトリクスを使用して、Eメールプログラムの効果分析およびバウンス率や苦情率などの重要な統計のモニタリングができます。

このセクションでは、Eメール送信メトリクスに関する以下のトピックについてよくある質問を示します。

- [一般的な質問 \(p. 518\)](#)
- [オープンの追跡 \(p. 519\)](#)
- [クリックの追跡 \(p. 520\)](#)

### 一般的な質問

#### Q1. Eメールの配信後、いつまで Amazon SES はオープンとクリックのメトリクスを収集しますか？

Amazon SES は各 Eメールの送信後 60 日間、オープンとクリックのメトリクスを収集します。

## Q2. ユーザーが特定の E メールを複数回開くか、特定の E メール内のリンクを複数回クリックした場合、これらの各イベントが別個に追跡されるのですか？

受信者が E メールを複数回開いた場合、Amazon SES によって各オープンが一意のオープンイベントとしてカウントされます。同様に、受信者が同じリンクを複数回クリックすると、Amazon SES によって各クリックは一意なクリックイベントとしてカウントされます。

## Q3. オープンとクリックのメトリクスは集約されますか？ または受取人レベルで測定できますか？

オープンとクリックは、受取人レベルで追跡されます。オープンとクリックの追跡により、E メールを開いた受信者や E メール内のリンクをクリックした受信者を判断できます。

## Q4. Amazon SES API を使用して開封とクリックのメトリクスを取得できますか？

Amazon SES API は、開封とクリックのメトリクスを取得するメソッドを提供していません。ただし、CloudWatch API を使用して Amazon SES のオープンとクリックのメトリクスを取得できます。たとえば、AWS CLI を使用して、以下のコマンドを発行することで、CloudWatch API を使用してクリックメトリクスを取得できます。

```
aws cloudwatch get-metric-statistics --namespace AWS/SES --metric-name Click \  
  --statistics Sum --period 86400 --start-time 2017-01-01T00:00:00Z \  
  --end-time 2017-12-31T23:59:59Z
```

上に示したコマンドは、2017 年の各日のクリックイベントの総数を取得します。開封メトリクスを取得するには、`metric-name` パラメータの値を `Open` に変更します。また、`start-time` および `end-time` パラメータを変更して分析期間を変更したり、または、`period` パラメータを変更してよりきめ細かい分析をしたりできます。

## オープンの追跡

### Q1. オープンの追跡はどのように行われますか？

Amazon SES で送信される各 E メール下部に 1 ピクセル四方の透明な GIF イメージが挿入されます。E メールごとに、このイメージファイルへの一意な参照があります。このイメージが開かれると、どのメッセージをだれが開いたかを Amazon SES で正確に判断できます。

この追跡用のピクセルを追加しても、Eメールの外観は変わりません。

### Q2. オープンの追跡はデフォルトで有効になりますか？

オープンの追跡は、デフォルトですべての Amazon SES ユーザーに利用可能です。オープンの追跡を使用するには、以下の操作を行う必要があります。

1. 設定セットを作成します。
2. 設定セットに、イベント送信先を作成する。
3. 送信先にオープンイベント通知を発行するようにイベント送信先を設定する。
4. オープンの追跡対象である E メールごとにステップ 1 で作成した設定セットを指定する。

このプロセスの詳細な説明については、「[the section called “イベント発行を使用した送信のモニタリング” \(p. 296\)](#)」を参照してください。

### Q3. オープンの追跡用ピクセルを特定の E メールから除外できますか？

オープン追跡用のピクセルを E メールから除外するには 2 つの方法があります。最初の方法では、設定セットを指定しないで E メールを送信します。別の方法として、オープンイベントに関するデータを発行するように設定されていない設定セットを指定できます。

### Q4. プレーンテキストメールのオープンは追跡されますか？

オープン追跡の対象となるのは HTML メールのみです。オープン追跡にはイメージの挿入が必要であるため、ユーザーがテキスト専用 (HTML 以外) の E メールクライアントで E メールを開くと、オープンメトリクスを収集することはできません。

## クリックの追跡

### Q1. クリックの追跡はどのように行われますか？

クリックを追跡するには、Amazon SES で E メール本文の各リンクを変更します。受信者がリンクをクリックすると、クリックは Amazon SES サーバーに送信され、即座に送信先アドレスに転送されます。オープン追跡と同様に、各転送先リンクは一意です。これにより、リンクをクリックした受信者、クリックした時間、リンク元の E メールを Amazon SES で判断できます。

#### Important

単一のメッセージを複数の宛先に送信した場合、各受取人によって同じクリック追跡リンクが保存されます。個々の受取人のクリックアクティビティを追跡するには、1 回の送信操作につき 1 人の受取人にメールを送信します。

### Q2. クリックの追跡を無効化できますか？

E メール HTML 本文のアンカータグに `ses:no-track` 属性を追加することで、個々のリンクのクリック追跡を無効にできます。たとえば、AWS ホームページにリンクする場合、通常のアンカーリンクは次のようになります。

```
<a href="https://aws.amazon.com">Amazon Web Services</a>
```

そのリンクのクリック追跡を無効にするには、以下のように変更します。

```
<a ses:no-track href="aws.amazon.com">Amazon Web Services</a>
```

`ses:no-track` は標準の HTML 属性ではないため、受信者の受信トレイに届く E メールバージョンから Amazon SES によって自動的に削除されます。

特定の設定セットを使用して送信するすべてのメッセージに対してクリック追跡を無効にすることもできます。クリック追跡を無効にするには、設定セットのイベント送信先を変更して、クリックイベントがキャプチャされないようにします。詳細については、「[Amazon SES でのイベント送信先の管理 \(p. 262\)](#)」を参照してください。

### Q3. 各 E メールで追跡できるリンクの数はいくつでしょうか？

クリック追跡システムでは、最大 250 のリンクを追跡できます。



## Q4. プレーンテキストの E メールへのリンクに対するクリックメトリクスは収集されますか？

HTML E メールへのリンクのみを追跡できます。

## Q5. リンクに一意的な識別子をタグ付けできますか？

`ses:tags` 属性を使用すると、メール内のリンクにキーと値のペアとしてタグを追加できます。追加できるタグの数に制限はありません。この属性を使用する場合は、CSS インラインプロパティを渡す場合と同じ形式を使用してキーと値を指定します。この形式では、キーの後にコロン (:)、その後続けて値を入力します。複数のキー値ペアを渡す必要がある場合は、各ペアをセミコロン (;) で区切ります。

たとえば、リンクにタグとして `product:book, genre:fiction, subgenre:scifi, type:newrelease` を追加するとします。この場合、結果のリンクは次のようになります。

```
<a ses:tags="product:book;genre:fiction;subgenre:scifi;type:newrelease;" href="http://www.amazon.com/.../">New Releases in Science Fiction</a>
```

これらのタグはイベント発行先に渡され、ユーザーがクリックした特定のリンクに対する追加の分析を実行できます。

### Note

リンクタグには、0~9 の数字、A~Z の文字 (大文字と小文字)、ハイフン (-)、およびアンダースコア (\_) を使用できます。

## Q6. 追跡されるリンクは、HTTP または HTTPS プロトコルを使用しますか？

リンクの追跡には E メール内のリンクと同じプロトコルを使用します。

たとえば、E メールに `https://www.amazon.com` へのリンクが含まれる場合、リンクは HTTPS プロトコルを使用する追跡リンクに置き換えられます。E メールに `http://www.example.com` へのリンクが含まれる場合、リンクは HTTP プロトコルを使用する追跡リンクに置き換えられます。E メールに前述のリンクの両方が含まれる場合、HTTPS リンクは HTTPS プロトコルを使用する追跡リンクに、HTTP リンクは HTTP プロトコルを使用する追跡リンクに置き換えられます。

## Q7. 追跡されている E メール内のリンクはありません。理由

Amazon SES では、E メール内のリンクに適切にエンコードされた URL が含まれています。具体的には、リンク内の URL は、RFC 3986 に準拠している必要があります。E メール内のリンクが適切にエンコードされていない場合は、受信者は E メールにリンクを表示できますが、Amazon SES はそのリンクのクリックイベントを追跡しません。

不適切なエンコードに関連する問題は、通常、クエリ文字列が含まれている URL で発生します。例えば、E メール内のリンクの URL のクエリ文字列にエンコードされていないスペース文字が含まれている場合 (次の例の「John」と「Doe」の間のスペースなど `http://www.example.com/path/to/page?name=John Doe`)、Amazon SES はリンクを追跡しません。ただし URL でエンコードされた空白文字 (次の例の「%20」など `http://www.example.com/path/to/page?name=John%20Doe`) が代わりに使用されている場合、Amazon SES は想定どおりに追跡します。

# Amazon SES のリソース

次の表に、Amazon Simple Email Service (Amazon SES) を使用するときに関係するリソースを示します。

リソース	説明
<a href="#">Amazon Simple Storage Service API リファレンス</a>	Amazon SES API リファレンス API アクション、パラメータ、データ型について詳しく説明します。サービスから返されるエラーのリストも含まれています。
<a href="#">Amazon SES 料金表</a>	Amazon SES の価格情報です。
<a href="#">SES 送信制限引き上げの場合</a>	送信クォータを引き上げるようリクエストし、サンドボックスから移動するサポートセンターのフォームです。
<a href="#">Amazon SES フォーラム</a>	Amazon SES ユーザーが質問を投稿して、Amazon SES に関する多様なトピックについて議論できるフォーラムです。
<a href="#">AWS メッセージングとターゲティングのブログ</a>	Amazon SES チームの投稿やお知らせを含むブログです。
<a href="#">AWS 開発者用ツール</a>	開発者用ツールと、AWS を利用した革新的なアプリケーションの構築に役立つ資料、コード例、リリースノート、その他の情報を含むリソースへのリンクです。
<a href="#">AWS Support Center</a>	AWS サポートケースを作成および管理するためのハブ。フォーラム、技術上のよくある質問、サービス状態ステータス、AWS Trusted Advisor などの便利なリソースへのリンクも含まれています。
<a href="#">お問い合わせ</a>	AWS の支払、アカウント設定その他に関する連絡先です。
<a href="#">AWS の用語集</a>	AWS の用語集です。Amazon SES や他の AWS のサービスでよく使用される用語の定義が含まれます。
<a href="#">利用規約</a>	AWS 利用規定ポリシー。Amazon Web Services, Inc. が提供するウェブサービスでの迷惑 E メールや、その他の禁止されている使用方法について説明します。
<a href="mailto:email-abuse@amazon.com">email-abuse@amazon.com</a>	Amazon SES から送信された悪質なメールや迷惑メール (スパム) を報告する E メールアドレス。  このアドレスに連絡する際は、次の情報をお知らせください。  <ul style="list-style-type: none"> <li>E メールメッセージの完全なヘッダー。Eメールのヘッダーの取得の詳細については、「<a href="https://support.google.com/mail/answer/22454?hl=en">https://support.google.com/mail/answer/22454?hl=en</a>」を参照してください。</li> <li>発生している迷惑行為のタイプ。たとえば、オプトアウトの方法が提供されていない未承諾の E メールなどです。</li> </ul>

# Amazon SES デベロッパーガイドの 付録

この付録には、Amazon Simple Email Service (Amazon SES) を使用した E メール送信に関する補足情報が記載されています。

- Amazon SES で送信する Eメールのヘッダーフィールドの要件については、「[ヘッダーフィールド \(p. 523\)](#)」を参照してください。
- Amazon SES が対応していない添付ファイルの種類については、「[サポート対象外の添付ファイルのタイプ \(p. 525\)](#)」を参照してください。

## ヘッダーフィールド

Amazon SES は、[RFC 822](#) で規定された形式に準拠するあらゆる E メールヘッダーに対応できます。

以下のフィールドをメッセージのヘッダーセクションに複数回含めることはできません。

- Accept-Language
- AcceptLanguage

### Note

このフィールドは標準ではありません。可能であれば、代わりに Accept-Language ヘッダーを使用してください。

- Archived-At
- Auto-Submitted
- Bounces-to
- コメント
- Content-Alternative
- Content-Base
- Content-Class
- Content-Description
- Content-Disposition
- Content-Duration
- Content-ID
- Content-Language
- Content-Length
- Content-Location
- Content-MD5
- Content-Transfer-Encoding
- Content-Type
- 日付

### Note

Date ヘッダーを指定した場合、Amazon SES は、Amazon SES がメッセージを受け取ったときに、UTC タイムゾーンの日付と時刻に対応するタイムスタンプでヘッダーを上書きします。

- Delivered-To
- Disposition-Notification-Options
- Disposition-Notification-To
- DKIM-Signature
- DomainKey-Signature
- Errors-To
- From
- Importance
- In-Reply-To
- Keywords
- List-Archive
- List-Help
- List-Id
- List-Owner
- List-Post
- List-Subscribe
- List-Unsubscribe
- Message-Context
- Message-ID

#### Note

Message-ID ヘッダーを指定すると、Amazon SES はヘッダーを独自の値で上書きします。

- MIME-Version
- 組織
- Original-From
- Original-Message-ID
- Original-Recipient
- Original-Subject
- 優先順位
- 優先度
- 参照
- Reply-To
- Return-Path

#### Note

リターンパス ヘッダーを指定した場合、Amazon SES は指定されたアドレスにバウンスと苦情の通知を送信します。ただし、受取人が受け取るメッセージの Return-Path ヘッダーには別の値が含まれています。

- Return-Receipt-To
- 送信者
- Solicitation
- Sensitivity
- Subject
- Thread-Index
- Thread-Topic
- User-Agent
- VBR-Info

## サポート対象外の添付ファイルのタイプ

多目的インターネットメール拡張 (MIME) スタンドアードを使用して、Amazon SES 経由で送信メッセージにファイルを添付できます。Amazon SES は、すべての添付ファイルのタイプを受け入れます。しかし次のリストのファイル拡張子を持つ添付ファイルは例外です。

### Note

ISP によっては他にも制限があります (添付ファイルのアーカイブに関する制限など)。実際に本稼働する前に大手 ISP を使って E メール送信をテストするようお勧めします。

.ade	.hta	.mau	.mst	.psc1
.adp	.inf	.mav	.ops	.psc2
.app	.ins	.maw	.pcd	.tmp
.asp	.isp	.mda	.pif	.url
.bas	.its	.mdb	.plg	.vb
.bat	.js	.mde	.prf	.vbe
.cer	.jse	.mdt	.prg	.vbs
.chm	.ksh	.mdw	.reg	.vps
.cmd	.lib	.mdz	.scf	.vsmacros
.com	.lnk	.msc	.scr	.vss
.cpl	.mad	.msh	.sct	.vst
.crt	.maf	.msh1	.shb	.vsw
.csh	.mag	.msh2	.shs	.vxd
.der	.mam	.mshxml	.sys	.ws
.exe	.maq	.msh1xml	.ps1	.wsc
.fxp	.mar	.msh2xml	.ps1xml	.wsf
.gadget	.mas	.msi	.ps2	.wsh
.hlp	.mat	.msp	.ps2xml	.xnk

# Amazon SES デベロッパーガイドの ドキュメント履歴

次の表は、『Amazon Simple Email Service (Amazon SES) デベロッパーガイド』の主な変更を示しています。

変更	説明	変更日
新機能	これで確認済みの各 ID を通じてメールを送信するときに、「 <a href="#">デフォルト構成セット (p. 260)</a> 」を指定できるようになりました。	2021 年 1 月 27 日
新機能	E メール配信が一時的に遅れる原因となるイベントが発生したときに <a href="#">通知 (p. 296)</a> を受信できるようになりました。	2020 年 6 月 19 日
新機能	Amazon SES が <a href="#">AWS GovCloud (US) (p. 467)</a> で利用可能に。	2020 年 4 月 30 日
新機能	<a href="#">Virtual Private Cloud (Amazon VPC) で Amazon SES エンドポイントを作成する (p. 84)</a> (Amazon VPC) ことが可能になりました。	2020 年 4 月 29 日
新機能	Amazon SES が <a href="#">追加の 3 つの AWS リージョン (p. 467)</a> で利用可能に: カナダ (中部)、欧州 (ロンドン)、および南米 (サンパウロ)。	2020 年 4 月 1 日
新機能	自分の IP 範囲を使用して E メールを送信できるようになりました。詳細については、「 <a href="#">Amazon SES での自分の IP アドレスを使用した Eメールの送信 (p. 186)</a> 」を参照してください。	2019 年 12 月 23 日
新機能	独自のパブリック/プライベートのキーペアを使用して、ドメインの DKIM 認証プロセスを完了できるようになりました。詳細については、「 <a href="#">」を参照してください独自の DKIM 認証トークンを Amazon SES で提供する (p. 143)</a>	2019 年 12 月 13 日
新機能	<a href="#">アカウントレベルのサプレッションリスト (p. 190)</a> を使用して、以前にバウンスや苦情が発生した E メールアドレスへのメッセージの送信を自動的に防止できるようになりました。	2019 年 11 月 25 日
新機能	アカウントの状態に問題がなく、アカウントの送信クォータに近づいている場合は、Amazon SES はクォータを自動的に引き上げます。詳細については、「 <a href="#">」を参照してくださいAmazon SES 送信クォータの引き上げ (p. 151)</a>	
ドキュメントの更新	<a href="#">Amazon SES から個人データを削除する方法 (p. 494)</a> を追加しました。	2018 年 3 月 13 日
オープンソースドキュメント	Amazon SES のドキュメントを <a href="#">GitHub</a> で入手できるようになりました。GitHub リポジトリで問題を送信したり、変更をリクエストしたり、変更を直接加えてプルリクエストを送信したりできます。	2018 年 2 月 22 日
ドキュメントの更新	Amazon SES に保存された <a href="#">個人データを削除する (p. 494)</a> 方法について説明するセクションを追加しました。	2018 年 2 月 28 日
ドキュメントの更新	<a href="#">Amazon SNS イベント発行フィールドの定義 (p. 328)</a> を改訂し、 <a href="#">レタリング障害イベントの例 (p. 347)</a> を追加しました。	2018 年 1 月 22 日

変更	説明	変更日
ドキュメントの更新	配信性能ダッシュボードの付録を更新し、IAM および Lambda コンソールの変更を反映しました。  Note  使用されていたコンポーネントがサポートされなくなったため、2019年5月3日にこの付録を削除しました。	2018年1月18日
ドキュメントの更新	<a href="#">CloudWatch へのイベントの発行に関するコンテンツ (p. 299)</a> を更新し、ブロックされたフィールドについて説明しました。	2018年1月15日
ドキュメントの更新	<a href="#">OpenSSL を使用して E メールを送信する手順 (p. 107)</a> を更新し、わかりやすくしました。	2018年1月11日
ドキュメントの更新	AWS SDK for Ruby を使用して raw E メールを送信するためのコード例を追加しました。	2018年1月2日 2014年1月3日
ドキュメントの更新	AWS SDK for PHP を使用して raw E メールを送信するためのコード例を追加しました。	2017年12月29日
新機能	カスタム確認 E メールに関連するコンテンツを追加しました。	2017年12月7日
新機能	E メール送信の一時停止と、設定セットの評価メトリクスのエクスポートに関連するコンテンツを追加しました。	2017年11月15日
ドキュメントの更新	AWS SDK for Java を使用して raw E メールを送信するためのコード例を追加しました。	2017年10月23日
ドキュメントの更新	AWS SDK for Python (Boto) を使用して raw E メールを送信するためのコード例を追加しました。	2017年10月20日
新機能	E メールテンプレートとパーソナライズされた E メール機能に関連するコンテンツを追加しました。	2017年10月11日
新機能	オープンとクリックに関するカスタムドメイン機能に関連するコンテンツを追加しました。	2017年9月18日
新機能	評価ダッシュボードに関連するコンテンツを追加しました。	2017年8月24日
新機能	専用 IP プール機能に関するコンテンツを追加しました。	2017年8月17日
新機能	追跡機能のオープンとクリックに関するコンテンツを追加しました。	2017年8月1日
ドキュメントの更新	コード例のインデックスを追加しました。	2017年6月26日
ドキュメントの更新	Amazon SES の配信性能ダッシュボードの作成プロセスを示す付録を追加しました。  Note  使用されていたコンポーネントがサポートされなくなったため、2019年5月3日にこの付録を削除しました。	2017年6月22日

変更	説明	変更日
ドキュメントの更新	E メール送信のコード例を更新しました。	2017 年 6 月 6 日
新機能	専用 IP について更新しました。	2016 年 11 月 21 日
新機能	メール送信イベントの発行について更新しました。	2016 年 11 月 2 日
サービスの更新	DKIM レコードの生成後にユーザーが Easy DKIM 署名を明示的に有効化する必要がなくなったことを反映するように更新しました。	2016 年 9 月 15 日
ドキュメントの更新	メール受信の開始チュートリアルを追加しました。	2016 年 7 月 12 日
新機能	拡張通知を更新しました。	2016 年 6 月 14 日
新機能	カスタムの MAIL FROM ドメインについて更新しました。	2016 年 3 月 14 日
新機能	インバウンド E メールについて更新しました。	2015 年 9 月 28 日
新機能	承認の送信について更新しました。	2015 年 7 月 8 日
新機能	AWS CloudTrail ログを更新しました。	2015 年 5 月 7 日
サービスの更新	Amazon SES の送信クォータ引き上げフォームの統合を反映するように更新。「本稼働アクセス」という用語を削除。	2015 年 4 月 8 日
サービスの更新	ドメイン検証 TXT レコードに関する新しい要件に更新しました。	2015 年 2 月 25 日
ドキュメントの更新	適用に関するよくある質問を追加しました。	2014 年 12 月 15 日
新機能	配信通知を更新しました。	2014 年 6 月 23 日
新機能	サブドメインサポートを更新しました。	2014 年 3 月 19 日
新機能	Amazon SES の米国西部 (オレゴン) リージョンへの拡張用に更新されました。	2014 年 1 月 29 日
新機能	Amazon SES の欧州 (アイルランド) リージョンへの拡張用に更新されました。	2014 年 1 月 15 日
新機能	ヘッダーフィールドおよび MIME タイプの検証における変更を反映するように更新しました。	2013 年 11 月 6 日
ドキュメントの更新	送信者 ID のコンテンツを削除しました。	2013 年 8 月 22 日
新機能	Amazon SES コンソールの再設計を反映するように更新しました。	2013 年 6 月 19 日



変更	説明	変更日
新機能	ブラックリスト機能をグローバルサブプレッションリストに変更しました。	2013 年 5 月 8 日
新機能	グローバルサブプレッションリストの削除機能を反映するように更新されました。	2013 年 3 月 4 日
ドキュメントの更新	MIME タイプを追加しました。	2013 年 2 月 4 日
ドキュメントの更新	使用開始セクションを追加し、スタンドアロンの使用開始ガイドを置き換えました。目次を再構成し、Sendmail の統合手順を更新しました。	2013 年 1 月 21 日
ドキュメントの更新	スループットの増加や SMTP 問題に関するトラブルシューティングセクションを追加しました。	2012 年 12 月 12 日
ドキュメントの更新	クォータの送信に関する情報を再構成しました。	2012 年 11 月 9 日
新機能	Amazon SES メールボックスシミュレーターを更新しました。	2012 年 10 月 3 日
新機能	DKIM 署名を使用して確認済みのアイデンティティから E メールに署名するように更新しました。	2012 年 7 月 17 日
新機能	Amazon Simple Notification Service (Amazon SNS) からバウンスや苦情のフィードバック通知を受け取るように更新しました。	2012 年 6 月 26 日
新機能	ドメインの確認について更新しました。	2012 年 5 月 15 日
新機能	追加ヘッダーおよび添付ファイルのタイプを反映するように更新しました。	2012 年 4 月 25 日
新機能	STARTTLS 拡張子を SMTP に変更しました。	2012 年 3 月 7 日
新機能	可変エンベロープリターンパス (VERP) を更新しました。	2012 年 2 月 22 日
新機能	SMTP サポートを更新しました。	2011 年 12 月 13 日
新機能	AWS マネジメントコンソールのサポートを更新しました。	2011 年 11 月 17 日
新機能	添付ファイルサポートを更新しました。	2011 年 7 月 18 日
初回リリース	Amazon Simple Email Service デベロッパーガイドの初回リリース。	2011 年 1 月 25 日