aws

Panduan Developerr

Layanan Email Sederhana Amazon

Layanan Email Sederhana Amazon: Panduan Developerr

Table of Contents

Apa itu Amazon SES? 1
Manfaat 1
Layanan terkait 1
Harga 2
Wilayah 2
SESwilayah dan titik akhir 3
Penghapusan sandbox dan batas pengiriman meningkat 4
Verifikasi alamat email dan domain 4
Mudah DKIM 4
Daftar penindasan tingkat akun 5
Notifikasi umpan balik 5
SMTPkredensialnya 5
Titik akhir umpan balik yang digunakan untuk domain kustom MAIL FROM 6
Otorisasi pengiriman 6
Penerimaan email 6
Kuota 7
Kuota pengiriman email 7
Email menerima kuota 11
Kuota Manajer Surat 12
Kuota umum 14
Tipe kredensial 14
Bagaimana Amazon SES bekerja 18
Setelah pengirim mengirim permintaan email ke SES 19
Setelah Amazon SES mengirim email 20
Format email 22
Memahami kemampuan pengiriman 26
Praktik terbaik email 32
Bekerja dengan AWS SDK 38
Memulai 40
Pengaturan 40
Mendaftar untuk AWS 40
Siapkan akun SES Anda 41
Berikan akses terprogram (Untuk berinteraksi dengan SES di luar konsol) 41
Unduh AWS SDK (Untuk menggunakan SESAPIs) 43

Migrasi ke Amazon SES 43
Langkah 1. Verifikasikan domain Anda 43
Langkah 2. Minta akses produksi 43
Langkah 3. Konfigurasikan sistem autentikasi domain 44
Langkah 4. Hasilkan SMTP kredensi Anda 44
Langkah 5. Connect ke SMTP endpoint 44
Langkah selanjutnya 44
Minta akses produksi 45
Batas pengiriman 49
Meningkatkan kuota pengiriman Anda 50
Secara otomatis meningkatkan kuota pengiriman 51
Pengguna diminta untuk meningkatkan kuota pengiriman 52
Pemantauan kuota pengiriman Anda 53
Memantau kuota pengiriman Anda menggunakan konsol Amazon SES 53
Memantau kuota pengiriman Anda menggunakan Amazon SES API 54
Kesalahan kuota pengiriman 55
Mencapai batas pengiriman dengan API Amazon SES 55
Mencapai batas pengiriman dengan SMTP 55
Siapkan pengiriman email 56
Menggunakan SMTP antarmuka 56
Persyaratan untuk mengirim email SMTP 57
Metode untuk mengirim email SMTP 57
Informasi email yang akan disediakan 58
Memperoleh SMTP kredensi 58
Menghubungkan ke titik SMTP akhir 64
Menggunakan paket perangkat lunak untuk mengirim email 65
Mengirim email secara terprogram 67
Mengintegrasikan dengan server email yang ada 77
Menguji koneksi Anda ke SES SMTP antarmuka Amazon 80
Menggunakan API 89
Mengirim email terformat 90
Mengirim email mentah 91
Menggunakan template untuk mengirim email 103
Mengirim email menggunakan AWS SDK 120
Pengkodean konten 139
Protokol keamanan yang didukung 140

Pengirim email ke Amazon SES 140
Amazon SES ke penerima 141
nd-to-end Enkripsi E 141
Bidang header yang didukung 142
Tipe lampiran yang tidak didukung 145
Penerimaan email 147
Konsep penerimaan email & kasus penggunaan 148
Kontrol berbasis penerima menggunakan aturan penerimaan 148
Kontrol berbasis IP menggunakan filter alamat IP 150
Proses penerimaan email 151
Kasus penggunaan & pembatasan 152
Otentikasi email dan deteksi malware 155
Menyiapkan penerimaan email 156
Memverifikasi domain Anda 157
Menerbitkan catatan MX 157
Memberikan izin 160
Panduan konsol penerimaan email 168
Membuat aturan penerimaan 169
Buat filter IP 208
Metrik penerimaan email 210
Identitas terverifikasi 214
Membuat & memverifikasi identitas 214
Membuat identitas domain 218
Memverifikasi identitas domain 221
Membuat identitas alamat email 226
Memverifikasi identitas alamat email 227
Buat & verifikasi identitas dan tetapkan set konfigurasi default pada saat yang sama (API) .. 228
Menggunakan templat email verifikasi kustom 229
Mengelola identitas 241
Melihat identitas menggunakan konsol 242
Hapus identitas menggunakan konsol 243
Mengedit identitas menggunakan konsol 243
Mengedit identitas untuk menggunakan set konfigurasi default menggunakan SES API 244
Ambil set konfigurasi default yang digunakan oleh identitas menggunakan SES API 246
Ganti set konfigurasi default saat ini yang digunakan oleh identitas menggunakan SES API 246

Mengonfigurasi identitas 247
Metode autentikasi email 248
Menyiapkan notifikasi peristiwa 293
Menggunakan otorisasi identitas 330
Menggunakan otorisasi pengiriman 345
Mengirim email dengan simulator percobaan dengan simulator percobaan dengan simulator percobaan dengan simulator percobaan 376
Menggunakan simulator kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari 376
Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual 378
Set konfigurasi 383
Buat set konfigurasi 384
Buat set konfigurasi 384
Buat set konfigurasi. (AWS CLI) 388
Kelola set konfigurasi 390
Lihat, edit, & hapus set konfigurasi (konsol) 390
Daftarkan set konfigurasi (AWS CLI) 392
Dapatkan detail set konfigurasi (AWS CLI) 393
Hapus set konfigurasi (AWS CLI) 393
Hentikan pengiriman email dari set konfigurasi (AWS CLI) 393
Memahami set konfigurasi default 393
Buat tujuan acara 395
Tetapkan kolam IP 400
Konfigurasikan domain buka dan klik kustom 401
Tentukan set konfigurasi dalam email 409
Lihat dan ekspor metrik reputasi 410
Mengaktifkan ekspor metrik reputasi 410
Menonaktifkan ekspor metrik reputasi 410
Alamat IP khusus 412
Kemudahan penyiapan 414
Manajemen reputasi 414
Prediktabilitas pola pengiriman 415
Volume email keluar 416
Biaya tambahan 416
Kontrol atas reputasi pengirim 416

Kemampuan untuk mengisolasi reputasi pengirim 416
Diketahui, alamat IP yang tidak berubah 417
Standar 417
Meminta & melepaskan 418
Pemanasan 422
Membuat kolam 425
Dikelola 427
Manfaat dan fitur 428
Pentingnya pemanasan 429
Membuat kumpulan IP terkelola 430
Melihat pengiriman dan kapasitas kolam 434
Menghapus kumpulan IP terkelola 436
Bawa alamat IP Anda sendiri 437
Persyaratan 437
Pertimbangan-pertimbangan 438
Menggunakan alamat IP Anda sendiri dengan Amazon SES 438
Manajer Pengiriman Virtual 440
Memulai 441
Memulai (konsol) 442
Memulai (AWS CLI) 443
Dasbor 445
Menggunakan dasbor (konsol) 447
Mengakses data metrik ()AWS CLI 451
Memfilter dan mengekspor data metrik ()AWS CLI 452
Menemukan pesan, statusnya, & mengekspor hasil ()AWS CLI 454
Mengelola lowongan kerja ekspor (AWS CLI) 458
Melihat detail pesan (AWS CLI) 460
Bagaimana metrik dasbor dihitung 461
Penasihat 463
Apa yang dicari penasihat 464
Menggunakan penasihat (konsol) 467
Mengakses rekomendasi ()AWS CLI 468
Pengaturan 469
Mengubah pengaturan Virtual Deliverability Manager (konsol) 469
Mengubah pengaturan Virtual Deliverability Manager ()AWS CLI 470
BARU - Manajer Surat 473

Memulai 474
Memulai 475
Titik akhir masuknya 476
Mengkonfigurasi lingkungan Anda 477
Membuat titik akhir ingress (konsol) 479
Kebijakan lalu lintas & pernyataan kebijakan 481
Membuat kebijakan lalu lintas & pernyataan kebijakan (konsol) 482
Kondisi pernyataan kebijakan 483
Set aturan & aturan 484
Membuat set aturan & aturan (konsol) 485
Kondisi & tindakan aturan 488
SMTPestafet 490
Membuat SMTP relay (konsol) 491
Menyiapkan Google Workspaces 495
Menyiapkan Microsoft Office 365 497
Pengarsipan email 503
Menggunakan pengarsipan email (konsol) 503
Email Tambah Ons 508
Berlangganan Add Ons (konsol) 509
Kebijakan izin 511
Kebijakan titik akhir Ingress 511
SMTPkebijakan relay 513
Kebijakan pengarsipan email 514
Kebijakan tindakan aturan 520
Daftar dan langganan 524
Daftar penekanan global 526
Pertimbangan daftar penekanan global 526
Menggunakan daftar penekanan tingkat akun 527
Pertimbangan daftar penekanan tingkat akun 528
Mengaktifkan daftar penekanan tingkat akun 529
Mengaktifkan daftar penekanan tingkat akun Anda untuk set konfigurasi 530
Menambahkan alamat email individual ke daftar penekanan tingkat akun Anda 533
Menambahkan alamat email secara massal ke daftar penekanan tingkat akun Anda 534
Melihat daftar alamat yang ada di daftar penindasan tingkat akun Anda 538
Menghapus alamat email individual dari daftar penindasan tingkat akun Anda 541
Menghapus alamat email secara massal dari daftar penindasan tingkat akun Anda 543

Melihat daftar tugas impor untuk akun 546
Mendapatkan informasi tentang tugas impor untuk akun 548
Menonaktifkan daftar penekanan tingkat akun 550
Menggunakan penekanan tingkat konfigurasi 551
Mengaktifkan penekanan tingkat konfigurasi 553
Menggunakan pengelolaan daftar 554
Gambaran umum manajemen daftar 554
Mengonfigurasi pengelolaan daftar 555
Panduan manajemen daftar dengan contoh 561
Menggunakan manajemen berlangganan 563
Gambaran umum manajemen berlangganan 564
Pertimbangan header berhenti berlangganan 565
Menambahkan tautan footer berhenti berlangganan 566
Memantau aktivitas pengiriman 567
Pemantauan menggunakan konsol 573
Dasbor akun 574
Metrik reputasi 575
Pengaturan SMTP 576
Menggunakan konsol untuk memantau metrik 577
Pantau menggunakan API 578
Memanggil operasi API `GetSendStatistics` menggunakan AWS CLI 579
Memanggil pemrograman operasi `GetSendStatistics` 579
Pantau pengiriman email menggunakan penerbitan acara 583
Cara kerja penerbitan acara dengan set konfigurasi dan tag pesan 583
Umpan balik halus untuk kampanye email 584
Cara menggunakan penerbitan peristiwa 586
Terminologi penerbitan peristiwa 586
Menyiapkan penerbitan peristiwa 587
Bekerja dengan data peristiwa 603
Pemantauan reputasi pengirim 675
Menggunakan metrik reputasi 675
Pesan metrik reputasi 677
Pesan Status Umum 678
Notifikasi Tingkat Pentalan 679
Notifikasi Tingkat Aduan 681
Notifikasi Organisasi Anti-Spam 682

Pemberitahuan Listbombing 684
Notifikasi umpan balik langsung 685
Notifikasi Daftar Blokir Domain 687
Notifikasi Peninjauan Internal 688
Notifikasi Penyedia Kotak Surat 690
Notifikasi Umpan Balik Penerima 691
Notifikasi Akun Terkait 692
Notifikasi Jebakan Spam 693
Notifikasi Situs Rentan 695
Kredensi yang Dikompromikan 696
Notifikasi lainnya 697
Membuat alarm menggunakan CloudWatch 697
Metrik SNDS untuk IP khusus 700
Pertanyaan terkait pemecahan masalah 702
Menjeda pengiriman email secara otomatis 702
Untuk seluruh akun Anda 703
Untuk satu set konfigurasi 710
Pemantauan menggunakan EventBridge 719
SESacara 719
Referensi skema acara 721
Skema status penasihat Manajer Pengiriman Virtual 722
SESskema status pengiriman email 723
Menggunakan EventBridge 726
Tentukan contoh peristiwa di EventBridge 726
Pola acara untuk SES acara 727
EventBridgeSumber daya tambahan 729
Contoh kode 731
Amazon SES 733
Hal-hal mendasar 735
Skenario 849
Amazon SES API v2 888
Hal-hal mendasar 889
Skenario 943
Keamanan 985
Perlindungan data 986
Enkripsi data saat istirahat 987

Enkripsi bergerak 997
Menghapus data pribadi 997
Pengelolaan identitas dan akses 1004
Membuat Kebijakan IAM untuk Akses ke SES 1005
Contoh Kebijakan IAM untuk SES 1008
AWS kebijakan terkelola 1014
Menggunakan peran terkait layanan 1016
Pencatatan dan pemantauan 1019
Mencatat panggilan API 1020
Validasi kepatuhan 1023
Ketahanan 1024
Keamanan infrastruktur dalam SES 1025
Titik akhir VPC 1025
Contoh panduan pengaturan SES di Amazon VPC 1026
Pemecahan Masalah 1030
Masalah umum 1031
Perubahan yang saya buat tidak selalu langsung terlihat 1031
Masalah verifikasi 1032
Masalah verifikasi domain 1032
Memeriksa pengaturan verifikasi domain 1034
Masalah verifikasi email 1035
Masalah DKIM 1036
Masalah pengiriman 1038
Masalah dengan email yang diterima 1039
Masalah notifikasi 1040
Kesalahan pengiriman email 1041
Meningkatkan throughput 1044
Masalah SMTP 1046
Kode respons SMTP 1048
FAQ 1055
FAQ proses peninjauan Pengiriman 1055
Akun Dalam Peninjauan 1056
Penjedaan Pengiriman 1059
Pentalan 1063
Aduan 1066
Jebakan Spam 1073

Investigasi manual 1076
FAQ DNS Blackhole List (DNSBL) 1078
Q1 FAQ DNSBL 1078
Q2 FAQ DNSBL 1079
Q3 FAQ DNSBL 1079
T4 FAQ DNSBL 1079
T5 FAQ DNSBL 1080
T6 FAQ DNSBL 1081
FAQ metrik email 1082
Umum 1083
Pelacakan Buka 1084
Pelacakan Klik 1085
Indeks Cari Cepat 1088
Cara & konsep 1088
..... mxcv

Apa itu Amazon SES?

Amazon Simple Email Service (SES) adalah platform email yang menyediakan cara mudah dan hemat biaya bagi Anda untuk mengirim dan menerima email menggunakan alamat email dan domain Anda sendiri.

Misalnya, Anda dapat mengirim email pemasaran seperti penawaran khusus, email transaksional seperti konfirmasi pesanan, dan tipe korespondensi lainnya seperti buletin. Ketika Anda menggunakan Amazon SES untuk menerima email, Anda dapat mengembangkan solusi perangkat lunak seperti autoresponders email, email berhenti berlangganan sistem, dan aplikasi yang menghasilkan tiket dukungan pelanggan dari email yang masuk.

Untuk informasi selengkapnya tentang topik yang terkait dengan Amazon SES, lihat Blog Pesan dan Target AWS.

Manfaat

Membangun solusi email berskala besar seringkali merupakan tantangan yang rumit dan mahal bagi suatu bisnis. Anda harus menghadapi tantangan infrastruktur seperti manajemen server email, konfigurasi jaringan, dan reputasi alamat IP. Selain itu, banyak solusi email pihak ketiga yang memerlukan kontrak dan negosiasi harga, serta biaya di muka yang signifikan. Amazon SES menghilangkan tantangan ini dan memungkinkan Anda untuk mendapatkan manfaat dari pengalaman bertahun-tahun dan infrastruktur email canggih Amazon.com telah dibangun untuk melayani basis pelanggan berskala besar sendiri.

Layanan terkait

Amazon SES terintegrasi secara mulus dengan produk lain AWS . Sebagai contoh, Anda dapat:

- Menambah kemampuan pengiriman email ke aplikasi apa pun.
- Anda dapat mengirim email dari Amazon EC2 dengan menggunakanSDK AWS, dengan menggunakan Antarmuka SMTP Amazon SES, atau dengan melakukan panggilan langsung ke API Amazon SES.
- Gunakan AWS Elastic Beanstalk untuk membuat aplikasi yang mengaktifkan email seperti program yang menggunakan Amazon SES untuk mengirim buletin kepada pelanggan.
- Menyiapkan Amazon Simple Notification Service (Amazon SNS) untuk memberi tahu Anda email yang terpental, mengajukan aduan, atau yang berhasil dikirim ke server email penerima. Bila Anda

menggunakan Amazon SES untuk menerima email, konten email Anda dapat dipublikasikan ke topik Amazon SNS.

- Gunakan AWS Management Console untuk mengatur Easy DKIM, yang merupakan cara untuk mengautentikasi email Anda. Meskipun Anda dapat menggunakan Easy DKIM dengan penyedia DNS apa pun, Easy DKIM sangat mudah diatur ketika Anda mengelola domain Anda dengan Route 53.
- Kontrol akses pengguna ke pengiriman email Anda dengan menggunakan AWS Identity and Access Management (IAM).
- Menyimpan email yang Anda terima di Amazon Simple Storage Service (Amazon S3).
- Lakukan tindakan pada email yang Anda terima dengan memicu fungsi AWS Lambda.
- Gunakan AWS Key Management Service (AWS KMS) untuk secara opsional mengenkripsi email yang Anda terima di bucket Amazon S3.
- Gunakan AWS CloudTrail untuk mencatat panggilan API Amazon SES yang Anda buat menggunakan konsol atau API Amazon SES.
- Publikasikan acara pengiriman email Anda ke Amazon CloudWatch atau Amazon Data Firehose. Jika Anda mempublikasikan acara pengiriman email ke Firehose, Anda dapat mengaksesnya di Amazon Redshift, Amazon Service, atau OpenSearch Amazon S3.

Harga

Dengan Amazon SES, Anda membayar berdasarkan volume email yang dikirim dan yang diterima. Untuk informasi lebih lanjut, lihat Harga Amazon SES.

Wilayah dan Amazon SES

SESTersedia di beberapa Wilayah AWS di seluruh dunia. Di setiap wilayah, AWS pertahankan beberapa Availability Zone. Availability Zone ini secara fisik terisolasi satu sama lain, tetapi disatukan oleh koneksi jaringan privat, latensi rendah, throughput tinggi, dan sangat redundan. Availability Zone ini memungkinkan kami untuk menyediakan tingkat ketersediaan dan redundansi yang sangat tinggi, sekaligus meminimalkan latensi.

Untuk daftar semua titik akhir SES regional, lihat titik akhir Amazon Simple Email Service dan kuota di. Referensi Umum AWS Untuk mempelajari lebih lanjut tentang jumlah Availability Zone yang tersedia di setiap wilayah, lihat Infrastruktur AWS Global.

Bagian ini berisi informasi yang perlu Anda ketahui jika Anda berencana untuk menggunakannya SES dalam beberapa Wilayah AWS. Bagian ini membahas subjek berikut:

- SESwilayah dan titik akhir
- Penghapusan sandbox dan batas pengiriman meningkat
- Verifikasi alamat email dan domain
- Mudah DKIM
- Daftar penindasan tingkat akun
- Notifikasi umpan balik
- SMTPkredensialnya
- Otorisasi pengiriman
- Titik akhir umpan balik yang digunakan untuk domain kustom MAIL FROM
- Penerimaan email
- Menyiapkan catatan (MX)

Untuk informasi umum tentang Wilayah AWS, lihat titik akhir AWS layanan di Referensi AWS Umum.

SESwilayah dan titik akhir

Ketika Anda menggunakan SES untuk mengirim email, Anda terhubung ke URL yang menyediakan titik akhir untuk SES API atau SMTP antarmuka. Referensi Umum AWSIni berisi daftar lengkap titik akhir yang Anda gunakan untuk mengirim dan menerima email melaluiSES. Untuk informasi selengkapnya, lihat titik akhir Amazon Simple Email Service dan kuota di bagian Referensi Umum AWS—spesifik direferensikan di bawah ini:

- APIendpoints — Ketika Anda mengirim email melaluiSES, Anda dapat menggunakan yang URLs tercantum dalam tabel ini untuk membuat HTTPS permintaan ke SESAPI.
- SMTPtitik akhir - Anda dapat menggunakan yang URLs tercantum dalam tabel ini untuk mengirim email saat menggunakan SMTP antarmuka.
- Titik akhir Penerimaan Email - Jika Anda telah mengonfigurasi SES untuk menerima email yang dikirim ke domain Anda, Anda dapat menggunakan SMTP titik akhir masuk yang URLs tercantum dalam tabel ini saat menyiapkan catatan penukar email (MX) di pengaturan untuk domain Anda. DNS

> **Note**
>
> Inbound SMTP URLs bukan alamat IMAP server. Dengan kata lain, Anda tidak dapat menggunakannya untuk menerima email dengan menggunakan aplikasi seperti Outlook. Untuk layanan yang menyediakan IMAP server untuk email masuk, lihat Amazon WorkMail.

Penghapusan sandbox dan batas pengiriman meningkat

Status kotak pasir untuk akun Anda dapat berbeda antara keduanya Wilayah AWS. Dengan kata lain, jika akun Anda telah dihapus dari kotak pasir di wilayah AS Barat (Oregon), mungkin masih berada di kotak pasir di wilayah AS Timur (Virginia N.), kecuali jika Anda juga telah menghapusnya dari kotak pasir di wilayah itu.

Batas pengiriman juga bisa berbeda tergantung pada Wilayah AWS. Misalnya, jika akun Anda dapat mengirim 10 pesan per detik di wilayah Eropa (Irlandia), Anda mungkin dapat mengirim lebih banyak atau lebih sedikit pesan di wilayah lain.

Ketika Anda mengajukan permintaan agar akun Anda dihapus dari kotak pasir, atau ketika Anda mengirimkan permintaan agar kuota pengiriman akun Anda ditingkatkan, pastikan untuk memilih semua permintaan Anda berlaku. Wilayah AWS Anda dapat mengirimkan beberapa permintaan dalam satu kasus Pusat Dukungan.

Verifikasi alamat email dan domain

Sebelum Anda dapat mengirim email menggunakanSES, Anda harus memverifikasi bahwa Anda memiliki alamat email atau domain yang Anda rencanakan untuk dikirim. Status verifikasi alamat email dan domain juga berbeda. Wilayah AWS Misalnya, jika Anda memverifikasi domain di wilayah AS Barat (Oregon), Anda tidak dapat menggunakan domain tersebut untuk mengirim email di wilayah AS Timur (Virginia Utara) hingga Anda menyelesaikan proses verifikasi lagi untuk wilayah tersebut. Untuk informasi selengkapnya tentang memverifikasi alamat email dan domain, lihat Identitas terverifikasi di Amazon SES.

Mudah DKIM

Anda harus melakukan proses DKIM pengaturan Mudah untuk setiap Wilayah AWS tempat Anda ingin menggunakan EasyDKIM. Artinya, di setiap wilayah, Anda harus menggunakan SES konsol

atau SES API untuk menghasilkan CNAME catatan. Selanjutnya, Anda harus menambahkan semua CNAME catatan ke DNS konfigurasi untuk domain Anda. Untuk informasi selengkapnya tentang pengaturan MudahDKIM, lihatEasy DKIM di Amazon SES.

Tidak semua Wilayah AWS menggunakan SES DKIM domain default, `dkim.amazonses.com` — untuk melihat apakah wilayah Anda menggunakan DKIM domain spesifik wilayah, periksa tabel DKIM domain di. Referensi Umum AWS

Daftar penindasan tingkat akun

Daftar penekanan SES tingkat akun Anda Akun AWS hanya berlaku untuk Anda saat ini. Wilayah AWS Anda dapat secara manual menambah atau menghapus, secara individu atau massal, alamat dari daftar penindasan tingkat akun Anda dengan menggunakan SES API v2 atau konsol. Untuk informasi selengkapnya tentang menggunakan daftar penekanan tingkat akun Anda, lihat. Menggunakan daftar SES penindasan tingkat akun Amazon

Notifikasi umpan balik

Ada dua poin penting yang perlu diperhatikan tentang pengaturan pemberitahuan umpan balik dalam beberapa Wilayah AWS:

- Pengaturan identitas terverifikasi, seperti apakah Anda menerima umpan balik melalui email atau melaluiSNS, hanya berlaku untuk wilayah tempat Anda mengaturnya. Misalnya, jika Anda memverifikasi user@example.com di wilayah AS Barat (Oregon) dan AS Timur (Virginia N.) dan Anda ingin menerima email yang dipantulkan melalui SNS pemberitahuan, Anda harus menggunakan SES API atau SES konsol untuk mengatur pemberitahuan SNS umpan balik untuk user@example.com di kedua wilayah.
- SNStopik yang Anda gunakan untuk penerusan umpan balik harus berada di wilayah yang sama dengan tempat Anda menggunakan. SES

Untuk informasi selengkapnya tentang memantau aktivitas pengiriman Anda melalui pemberitahuan umpan balik, lihatMenyiapkan pemberitahuan acara untuk Amazon SES.

SMTPkredensialnya

Kredensi yang Anda gunakan untuk mengirim email melalui SES SMTP antarmuka unik untuk masing-masing. Wilayah AWS Jika Anda menggunakan SES SMTP antarmuka untuk mengirim email di lebih dari satu wilayah, Anda harus menghasilkan satu set SMTP kredensi untuk setiap wilayah.

> **Note**
>
> Jika Anda membuat SMTP kredensil sebelum 10 Januari 2019, SMTP kredensil Anda dibuat menggunakan versi Tanda Tangan yang lebih lama. AWS Demi keamanan, Anda harus menghapus kredensial yang Anda buat sebelum tanggal ini, dan menggantinya dengan kredensial yang lebih baru. Anda dapat menghapus kredensil lama dengan menggunakan konsol. IAM

Titik akhir umpan balik yang digunakan untuk domain kustom MAIL FROM

Jika Anda menggunakan MAIL FROM domain khusus, SES Anda harus mempublikasikan catatan MX sehingga domain Anda dapat menerima pemberitahuan pentalan dan keluhan yang dikirimkan oleh penyedia email kepada Anda. Anda dapat menggunakan MAIL FROM domain kustom yang sama untuk identitas terverifikasi secara berbeda Wilayah AWS karena pemberitahuan pentalan dan keluhan dikirim ke titik akhir umpan balik khusus wilayah.

Saat Anda mengonfigurasi MAIL FROM domain kustom, SES secara otomatis menentukan titik akhir umpan balik yang benar untuk wilayah tempat kustom MAIL FROM dikonfigurasi. Titik akhir ini disediakan di bidang nilai catatan MX untuk Anda publikasikan (tambahkan) ke konfigurasi domain Anda. DNS

Proses MAIL FROM pengaturan khusus dijelaskan dalamMenggunakan domain MAIL FROM kustom. Sebagai referensi, titik akhir umpan balik yang SES digunakan untuk perbedaan Wilayah AWS tercantum dalam tabel titik akhir Umpan Balik di. Referensi Umum AWS

Otorisasi pengiriman

Pengirim delegasi hanya dapat mengirim email dari Wilayah AWS tempat identitas pemilik identitas diverifikasi. Kebijakan otorisasi pengiriman yang memberikan izin kepada pengirim delegasi harus dilampirkan ke identitas di wilayah tersebut. Untuk informasi selengkapnya tentang otorisasi pengiriman, lihat Menggunakan otorisasi pengiriman dengan Amazon SES.

Penerimaan email

Dengan pengecualian bucket Amazon S3, semua AWS sumber daya yang Anda gunakan untuk menerima email SES harus Wilayah AWS sama dengan titik akhir. SES Misalnya, jika Anda menggunakan SES di wilayah AS Barat (Oregon), maka SNS topik, KMS kunci, dan fungsi Lambda

apa pun yang Anda gunakan juga harus berada di wilayah AS Barat (Oregon). Demikian pula, untuk menerima email dengan SES dalam suatu wilayah, Anda harus membuat aturan tanda terima aktif yang ditetapkan di wilayah tersebut. Konsep penerimaan email dan proses penyiapan dijelaskan dalamPenerimaan email dengan Amazon SES.

Tabel titik akhir Penerimaan Email dalam Referensi Umum AWS daftar titik akhir penerima email untuk semua Wilayah AWS tempat SES mendukung penerimaan email.

Service quotas di Amazon SES

Bagian berikut daftar dan menggambarkan kuota yang berlaku untuk sumber daya dan operasi Amazon SES. Beberapa kuota dapat ditingkatkan, sementara yang lain tidak bisa. Untuk menentukan apakah Anda dapat meminta kenaikan kuota, lihat kolom Adjustable.

> **Note**
>
> Kuota SES adalah untuk setiap Wilayah AWS yang Anda gunakan dalam kuota Anda Akun AWS.

Kuota pengiriman email

Kuota berikut berlaku untuk mengirim email melalui SES.

Kuota pengiriman

Kuota didasarkan pada jumlah penerima, bukan pada jumlah pesan.

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah email yang dapat dikirim per periode 24 jam	Jika akun Anda berada di sandbox, Anda dapat mengirim hingga 200 email per periode 24 jam. Jika akun Anda berada di luar sandbox, jumlah ini	Ya

Sumber Daya	Kuota Default	Dapat Disesuaikan
	bervariasi berdasarkan kasus penggunaan spesifik Anda.	
Jumlah email yang dapat dikirim per detik (laju pengiriman)	Jika akun Anda berada di sandbox, Anda dapat mengirim 1 email per detik. Jika akun Anda berada di luar sandbox, lajur ini bervariasi berdasarkan kasus penggunaan spesifik Anda.	Ya

Kuota pesan

Sumber Daya	Kuota Default	Dapat Disesuaikan
Menggunakan SES v1 API - Ukuran pesan maksimum (termasuk lampiran)	10 MB per pesan (setelah pengodean base64).	Tidak (Untuk beban kerja dengan ukuran pesan lebih dari 10MB, pertimbangkan untuk bermigrasi ke SES v2 API.)
Menggunakan SES v2 API atau SMTP - Ukuran pesan maksimum (termasuk lampiran)	40 MB per pesan (setelah pengkodean base64).	Tidak

> **Note**
>
> Pesan yang lebih besar dari 10MB tunduk pada pembatasan bandwidth, dan tergantung pada tingkat pengiriman Anda, Anda mungkin dibatasi hingga serendah 40MB/s. Misalnya, Anda dapat mengirim pesan 40MB dengan kecepatan 1 pesan per detik, atau dua pesan 20MB per detik.

Kuota pengirim dan penerima

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah maksimum penerima per pesan	50 penerima per pesan. **Note** Penerima adalah alamat "Kepada", "CC", atau "BCC".	Batas penerima tidak dapat disesuaikan. Silakan hubungi Manajer AWS Akun Anda untuk meminta fitur ini setelah membaca catatan di bawah ini.
Jumlah maksimum identitas yang dapat Anda verifikasi	10.000 identitas per. Wilayah AWS **Note** Identitas adalah domain atau alamat email yang Anda gunakan untuk mengirim email melalui SES.	Silakan hubungi Manajer AWS Akun Anda untuk mendiskus ikan kasus penggunaan Anda.
Jumlah maksimum kumpulan IP khusus (termasuk kolam IP terkelola dan standar)	50	Tidak

Note

Sebelum meminta peningkatan batas penerima per pesan, silakan baca blog ini dan bersiaplah untuk menjelaskan secara rinci mengapa kasus penggunaan Anda tidak dapat dipenuhi menggunakan batas default 50 penerima per pesan atau dengan mengirim pesan ke penerima individu. Mendefinisikan beberapa penerima dalam tujuan pesan dapat

menyebabkan observabilitas yang buruk serta kemampuan pengiriman yang buruk dan tidak boleh digunakan kecuali kasus penggunaan Anda secara khusus mengharuskannya.

Kuota yang berkaitan dengan penerbitan kejadian

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah maksimum set konfigurasi	10.000	Tidak
Panjang maksimum nama set konfigurasi	Nama set konfigurasi dapat berisi hingga 64 karakter alfanumerik. Mereka juga dapat berisi tanda hubung (-) dan garis bawah (_). Nama tidak boleh berisi spasi, karakter beraksen, atau karakter khusus lainnya.	Tidak
Jumlah maksimum tujuan kejadian per set konfigurasi	10	Tidak
Jumlah maksimum dimensi per tujuan CloudWatch acara	10	Tidak

Kuota templat email

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah maksimum template email di masing-masing Wilayah AWS	20.000	Tidak
Ukuran maksimum templat	500 KB	Tidak

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah maksimum nilai penggantian di setiap templat	Tidak terbatas	N/A
Jumlah maksimum penerima untuk setiap email yang ditemplat	50 tujuan. Tujuan adalah alamat email apa pun di baris "Kepada", "CC", atau "BCC". **Note** Jumlah tujuan yang dapat Anda hubungi dalam satu panggilan ke API mungkin dibatasi oleh laju pengiriman maksimum akun Anda.	Tidak

Email menerima kuota

Tabel berikut mencantumkan kuota yang terkait dengan menerima email melalui SES.

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah maksimum aturan per set aturan penerimaan	200	Tidak
Jumlah maksimum tindakan per aturan penerimaan	10	Tidak
Jumlah maksimum penerima per aturan penerimaan	100	Tidak
Jumlah maksimum set aturan penerimaan per Akun AWS	40	Tidak

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah maksimum filter alamat IP per Akun AWS	100	Tidak
Ukuran email maksimum (termasuk header) yang dapat disimpan di bucket Amazon S3	40 MB	Tidak
Ukuran email maksimum (termasuk header) yang dapat dipublikasikan menggunakan notifikasi Amazon SNS	150 KB	Tidak

Kuota Manajer Surat

Tabel berikut mencantumkan kuota yang terkait dengan Mail Manager.

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah maksimum titik akhir masuknya terbuka	10	Tidak
Jumlah maksimum titik akhir masuk resmi	50	Tidak
Jumlah maksimum penerima per pesan	100	Tidak
Ukuran email maksimum (termasuk header)	40 MB	Tidak
Jumlah maksimum pernyataan kebijakan lalu lintas	20	Tidak

Sumber Daya	Kuota Default	Dapat Disesuaikan
Jumlah maksimum kondisi pernyataan kebijakan lalu lintas	10	Tidak
Jumlah maksimum kebijakan lalu lintas per wilayah	100	Tidak
Jumlah maksimum relay SMTP	100	Tidak
Jumlah maksimum set aturan	40	Tidak
Jumlah maksimum eksekusi aturan per pesan	200	Tidak
Jumlah maksimum kondisi per aturan	10	Tidak
Jumlah maksimum tindakan per aturan	10	Tidak
Jumlah maksimum relai atau kirim tindakan per set aturan	10	Tidak
Jumlah maksimum arsip aktif	10	Tidak
Jumlah maksimum permintaan pencarian yang berjalan secara paralel	1	Tidak
Jumlah maksimum permintaan ekspor yang berjalan secara paralel	1	Tidak
Jumlah maksimum perubahan retensi untuk arsip per minggu	1	Tidak

Kuota umum

Tabel berikut mencantumkan kuota yang berlaku untuk mengirim dan menerima email melalui SES.

SES API mengirim kuota

Sumber Daya	Kuota Default	Dapat Disesuaikan
Laju ketika Anda dapat memanggil tindakan Amazon SES API	Semua tindakan (kecuali untuk`SendEmail` ,`SendRawEmail` , dan`SendTemplatedEmail`) dibatasi pada satu permintaan per detik.	Tidak
Bagian MIME	500	Tidak

Tipe kredensial Amazon SES

Untuk berinteraksi dengan Amazon Simple Email Service (Amazon SES), Anda menggunakan kredensial keamanan untuk memverifikasi siapa Anda dan apakah Anda memiliki izin untuk berinteraksi dengan Amazon SES. Ada berbagai tipe kredensial, dan kredensial yang Anda gunakan tergantung pada apa yang ingin Anda lakukan. Misalnya, Anda menggunakan AWS access key ketika Anda mengirim email menggunakan Amazon SES API, dan kredensial SMTP ketika Anda mengirim email menggunakan antarmuka SMTP Amazon SES.

Tabel berikut mencantumkan tipe kredensial yang mungkin Anda gunakan dengan Amazon SES, tergantung pada apa yang Anda lakukan.

Jika Anda ingin mengakses...	Gunakan kredensial ini	Kredensialnya terdiri dari	Cara mendapatkan kredensial
API Amazon SES (Anda mungkin mengakses Amazon SES API	Access key AWS	Access key ID dan secret access key	Lihat Access Keys di Referensi Umum AWS.

Jika Anda ingin mengakses...	Gunakan kredensial ini	Kredensialnya terdiri dari	Cara mendapatkan kredensial
secara langsung, atau tidak langsung melalui SDK AWS, AWS Command Line Interface, atau AWS Tools for Windows PowerShell.)			**Note** Untuk praktik terbaik keamanan, gunakan access key pengguna AWS Identity and Access Management (IAM), bukan access key Akun AWS. Akun AWSKredensi Anda memberikan akses penuh ke semuaAWS sumber daya Anda, sehingga Anda harus menyimpan nya di tempat yang aman dan daripada menggunakan kredensi pengguna IAM untuk day-to-day interaksi denganAWS . Untuk informasi selengkapnya, lihat Kredensi Akun Root vs. Kredensi Pengguna IAM di. Kredensi Akun Root vs. Kredensi Pengguna IAM di Referensi Umum AWS. Kredensi Akun Root vs

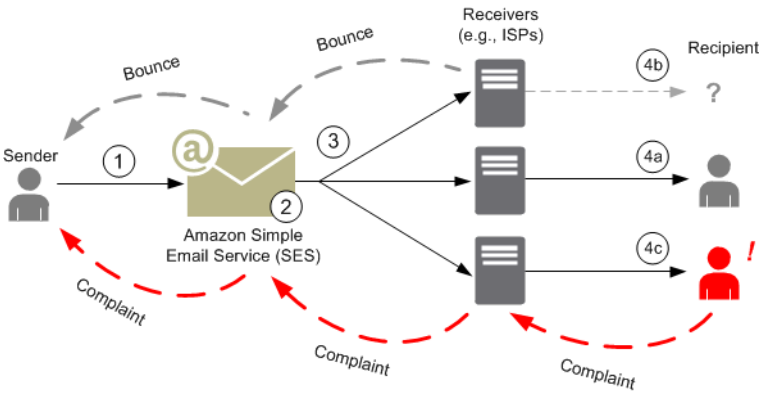
Jika Anda ingin mengakses...	Gunakan kredensial ini	Kredensialnya terdiri dari	Cara mendapatkan kredensial
Antarmuka Amazon SES SMTP	Kredensial SMTP	Nama pengguna dan kata sandi	Lihat Memperoleh SES SMTP kredensi Amazon. **Note** Meskipun kredensial SMTP Amazon SES Anda berbeda dari access key AWS Anda dan access key pengguna IAM, kredensial SMTP Amazon SES sebenarnya adalah tipe kredensial IAM. Pengguna IAM dapat membuat kredensi Amazon SES SMTP, tetapi pemilik akun akar harus memastikan bahwa kebijakan pengguna IAM memberi mereka izin untuk mengakses tindakan IAM berikut: “iam: “,ListUsers “iam: “iam:CreateUser “iam:CreateAccessKey “, dan “iam:PutUserPolicy”.

Jika Anda ingin mengakses...	Gunakan kredensial ini	Kredensialnya terdiri dari	Cara mendapatkan kredensial
Konsol Amazon SES	Nama pengguna dan kata sandi IAM ATAU Alamat email dan kata sandi	Nama pengguna dan kata sandi IAM ATAU Alamat email dan kata sandi	Lihat IAM User Name dan Password dan Alamat Email dan Password dari Referensi Umum AWS. **Note** Untuk praktik terbaik keamanan, gunakan nama pengguna dan kata sandi IAM bukan alamat email dan kata sandi. Kombinasi alamat email dan kata sandi adalah untuk AndaAkun AWS, sehingga Anda harus menyimpannya di tempat yang aman daripada menggunak annya untuk day-to-day interaksi denganAWS. Untuk informasi selengkap nya, lihat Kredensi Akun Root vs. Kredensi Pengguna IAM di. Kredensi Akun Root vs. Kredensi Pengguna IAM di Referensi Umum AWS. Kredensi Akun Root vs

Untuk informasi selengkapnya tentang berbagai tipe kredensiAWS keamanan (kecuali untuk kredensi SMTP, yang hanya digunakan untuk Amazon SES), lihat KredensiAWS Keamanan di Referensi Umum AWS.

Cara kerja pengiriman email di Amazon SES

Topik ini menjelaskan apa yang terjadi ketika Anda mengirim email denganSES, dan berbagai hasil yang dapat terjadi setelah email dikirim. Gambar berikut adalah gambaran umum tingkat tinggi dari proses pengiriman:



1. Aplikasi klien, bertindak sebagai pengirim email, membuat permintaan untuk mengirim email SES ke satu atau lebih penerima.
2. Jika permintaan valid, SES terima email.
3. SESmengirimkan pesan melalui Internet ke penerima penerima. Setelah pesan diteruskan keSES, biasanya dikirim segera, dengan upaya pengiriman pertama biasanya terjadi dalam milidetik.
4. Pada titik ini, ada kemungkinan yang berbeda. Sebagai contoh:

    a. ISPBerhasil mengirimkan pesan ke kotak masuk penerima.

    b. Alamat email penerima tidak ada, sehingga ISP mengirimkan pemberitahuan bouncing keSES. SESkemudian meneruskan notifikasi ke pengirim.

    c. Penerima menerima pesan tetapi menganggapnya sebagai spam dan mendaftarkan keluhan dengan. ISP TheISP, yang memiliki loop umpan balik yang disiapkanSES, mengirimkan keluhan keSES, yang kemudian meneruskannya ke pengirim.

Bagian berikut meninjau hasil individu yang mungkin setelah pengirim mengirim permintaan email ke SES dan setelah SES mengirim pesan email ke penerima.

Setelah pengirim mengirim permintaan email ke SES

Ketika pengirim membuat permintaan SES untuk mengirim email, panggilan mungkin berhasil atau gagal. Bagian berikut menjelaskan hal yang terjadi dalam setiap kasus.

Permintaan pengiriman berhasil

Jika permintaan untuk SES berhasil, SES mengembalikan respon sukses kepada pengirim. Pesan ini mencakup ID pesan, string karakter yang secara unik mengidentifikasi permintaan. Anda dapat menggunakan ID pesan untuk mengidentifikasi email terkirim atau untuk melacak masalah yang dihadapi selama pengiriman (Anda harus menyimpan pemetaan Anda sendiri antara pengenal dan ID SES pesan yang SES diteruskan kembali kepada Anda saat menerima email). SESkemudian merakit pesan email berdasarkan parameter permintaan, memindai pesan untuk konten dan virus yang dipertanyakan dan kemudian mengirimkannya melalui Internet menggunakan Simple Mail Transfer Protocol (). SMTP Pesan Anda biasanya segera dikirim; upaya pengiriman pertama biasanya terjadi dalam milidetik.

> **Note**
>
> Jika SES menerima permintaan pengirim dan kemudian menentukan bahwa pesan berisi virus, SES berhenti memproses pesan dan tidak mencoba mengirimkannya ke server email penerima.

Gagal mengirim permintaan

Jika permintaan pengiriman email pengirim SES gagal, SES balas pengirim dengan kesalahan dan hapus email. Permintaan bisa gagal karena beberapa alasan. Misalnya, permintaan mungkin tidak diformat dengan benar atau alamat email mungkin belum diverifikasi oleh pengirim.

Metode di mana Anda dapat menentukan apakah permintaan gagal tergantung pada bagaimana Anda meneleponSES. Berikut ini adalah contoh cara kesalahan dan pengecualian dikembalikan:

- Jika Anda menelepon SES melalui Query API (HTTPS) (`SendEmail`atau`SendRawEmail`), tindakan akan mengembalikan kesalahan. Untuk informasi selengkapnya, lihat APIReferensi Layanan Email Sederhana Amazon.
- Jika Anda menggunakan bahasa pemrograman AWS SDK untuk bahasa pemrograman yang menggunakan pengecualian, panggilan ke SES akan melempar a MessageRejectedException. (Nama pengecualian mungkin sedikit berbeda tergantung padaSDK.)

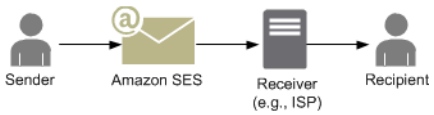
- Jika Anda menggunakan SMTP antarmuka, maka pengirim menerima kode SMTP respons, tetapi bagaimana kesalahan disampaikan tergantung pada klien pengirim. Beberapa klien mungkin menampilkan kode kesalahan; orang lain mungkin tidak.

Untuk informasi tentang kesalahan yang dapat terjadi ketika Anda mengirim email denganSES, lihatKesalahan pengiriman email Amazon SES.

Setelah Amazon SES mengirim email

Jika permintaan pengirim untuk SES berhasil, maka SES kirimkan email dan salah satu hasil berikut terjadi:

- Pengiriman berhasil dan penerima tidak keberatan dengan email — Email diterima olehISP, dan ISP mengirimkan email ke penerima. Pengiriman yang berhasil ditunjukkan pada gambar berikut.



- Hard bounce — Email ditolak oleh ISP karena kondisi persisten atau ditolak oleh SES karena alamat email ada di daftar SES penindasan. Alamat email ada di daftar SES penindasan jika baru-baru ini menyebabkan pantulan keras bagi pelanggan mana punSES. Hard bounce dengan an ISP dapat terjadi karena alamat penerima tidak valid. Pemberitahuan hard bounce dikirim dari ISP belakang keSES, yang memberi tahu pengirim melalui email atau melalui Amazon Simple Notification Service (AmazonSNS), tergantung pada pengaturan pengirim. SESmemberi tahu pengirim daftar penindasan memantul dengan cara yang sama. Jalur pantulan keras dari an ISP ditunjukkan pada gambar berikut.

- Soft bounce — ISP Tidak dapat mengirimkan email ke penerima karena kondisi sementara, seperti ISP terlalu sibuk untuk menangani permintaan atau kotak surat penerima penuh. Sebuah pentalan lunak juga dapat terjadi jika domain tidak ada. ISPMengirim notifikasi bouncing lunak kembali keSES, atau, dalam kasus domain yang tidak ada, SES tidak dapat menemukan server email untuk domain tersebut. Dalam kedua kasus tersebut, SES coba ulang email untuk jangka waktu yang lama. Jika SES tidak dapat mengirimkan email dalam periode waktu tersebut, email akan mengirimkan pemberitahuan bouncing melalui email atau melalui AmazonSNS. Jika SES dapat mengirimkan email ke penerima selama percobaan ulang, pengiriman berhasil. Sebuah pentalan

lunak ditunjukkan pada gambar berikut. Dalam hal ini, SES coba lagi mengirim email, dan akhirnya dapat mengirimkannya ke penerima. ISP

- Keluhan — Email diterima oleh ISP dan dikirim ke penerima, tetapi penerima menganggap email sebagai spam dan mengklik tombol seperti “Tandai sebagai spam” di klien emailnya. Jika SES memiliki loop umpan balik yang diatur denganISP, maka pemberitahuan keluhan dikirim keSES, yang meneruskan pemberitahuan keluhan ke pengirim. Sebagian besar ISPs tidak memberikan alamat email penerima yang mengajukan keluhan, sehingga pemberitahuan keluhan dari SES memberikan pengirim daftar penerima yang mungkin telah mengirim keluhan, berdasarkan penerima pesan asli dan ISP dari mana SES menerima keluhan. Jalur aduan ditunjukkan dalam gambar berikut.

- Respons otomatis — Email diterima olehISP, dan ISP mengirimkannya ke penerima. ISPKemudian mengirimkan respons otomatis seperti out-of-the-office (OOTO) pesan keSES. SESmeneruskan notifikasi respons otomatis ke pengirim. Respons otomatis ditunjukkan pada gambar berikut.

Pastikan program SES yang diaktifkan tidak mencoba lagi mengirim pesan yang menghasilkan respons otomatis.

> **Tip**
>
> Anda dapat menggunakan simulator SES kotak surat untuk menguji pengiriman yang berhasil, pentalan, keluhanOOTO, atau apa yang terjadi ketika alamat ada di daftar penindasan. Untuk informasi selengkapnya, lihat Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual.

Format email dan Amazon SES

Ketika klien membuat permintaan ke Amazon SES, Amazon SES membangun pesan email yang patuh dengan spesifikasi Format Pesan Internet (RFC 5322). Email terdiri dari header, isi, dan envelope, seperti yang dijelaskan di bawah ini.

- Header—Berisi petunjuk perutean dan informasi tentang pesan. Contohnya adalah alamat pengirim, alamat penerima, subjek, dan tanggal. Header analog dengan informasi di bagian atas surat pos, meskipun dapat berisi banyak tipe informasi lainnya, seperti format pesan.
- Isi—Berisi teks pesan itu sendiri.
- Envelope—Berisi informasi perutean aktual yang dikomunikasikan antara klien email dan server surat selama sesi SMTP. Informasi envelope email ini analog dengan informasi pada envelope pos. Informasi perutean dari envelope email biasanya sama dengan informasi perutean di header email, tetapi tidak selalu. Misalnya, ketika Anda mengirim salinan tembusan (BCC), alamat penerima sebenarnya (berasal dari envelope) tidak sama dengan alamat "Kepada" yang ditampilkan di klien email penerima, yang berasal dari header.

Berikut ini adalah contoh sederhana dari email. Header diikuti oleh baris kosong lalu isi email. Envelope tidak ditampilkan karena dikomunikasikan antara klien dan server surat selama sesi SMTP, bukan bagian dari email itu sendiri.

```
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
```

Bagian berikut meninjau header dan isi email dan mengidentifikasi informasi yang Anda perlu berikan ketika Anda menggunakan Amazon SES.

Header email

Ada satu header per pesan email. Setiap baris header berisi bidang diikuti oleh titik dua yang diikuti oleh badan bidang. Ketika Anda membaca email di klien email, klien email biasanya menampilkan nilai-nilai bidang header berikut:

- Kepada—Alamat email penerima pesan.
- CC—Alamat email penerima tembusan pesan.
- Dari—Alamat email dari tempat email dikirim.
- Subjek—Ringkasan topik pesan.
- Tanggal—Waktu dan tanggal email dikirim.

Ada banyak bidang header tambahan yang menyediakan informasi perutean dan menjelaskan isi dari pesan. Klien email biasanya tidak menampilkan bidang ini ke pengguna. Untuk daftar lengkap bidang header yang diterima Amazon SES, lihat Bidang SES header Amazon. Bila Anda menggunakan Amazon SES, hal paling utama Anda perlu memahami perbedaan antara bidang header "Dari," "Balas Ke," dan "Jalur Kembali". Seperti yang disebutkan sebelumnya, alamat "Dari" adalah alamat email pengirim pesan, sedangkan "Balas Ke" dan "Jalur Kembali" adalah sebagai berikut:

- Balas Ke—Alamat email yang akan dikirim balasan. Secara default, balasan akan dikirim ke alamat email pengirim asli.
- Jalur Kembali—Alamat email tempat pentalan dan aduan pesan yang harus dikirim. "Jalur Kembali" terkadang dipanggil "envelope dari," "pengirim envelope," atau "MAIL FROM."

> **Note**
>
> Bila Anda menggunakan Amazon SES, kami merekomendasikan Anda untuk selalu mengatur parameter "Jalur Kembali" sehingga Anda dapat menyadari pentalan dan mengambil tindakan korektif jika terjadi.

Untuk dengan mudah mencocokkan pesan pentalan dengan penerima yang dimaksudkan, Anda dapat menggunakan Variable Envelope Return Path (VERP). Dengan VERP, Anda menetapkan "Jalur Kembali" berbeda untuk setiap penerima, sehingga jika pesan terpental kembali, Anda secara

otomatis mengetahui tempat penerima itu terpental, daripada harus membuka pesan pentalan dan mengurainya.

Isi email

Isi email berisi teks pesan. Isi dapat dikirim dalam format berikut:

- HTML—Jika klien email penerima dapat menafsirkan HTML, isi dapat mencakup teks dan hyperlink yang diformat
- Teks biasa—Jika klien email penerima berbasis teks, isi tidak harus berisi karakter yang tidak dapat dicetak.
- Kedua HTML dan teks biasa—Saat Anda menggunakan kedua format untuk mengirim konten yang sama dalam satu pesan, klien email penerima memutuskan yang akan ditampilkan, berdasarkan kemampuannya.

Jika Anda mengirim pesan email ke sejumlah besar penerima, maka masuk akal untuk mengirimkannya dalam kedua HTML dan teks. Beberapa penerima akan memiliki klien email yang diaktifkan HTML, sehingga mereka dapat mengeklik hyperlink yang tersemat dalam pesan. Penerima yang menggunakan klien email berbasis teks akan meminta Anda untuk menyertakan URL yang dapat disalin dan dibuka menggunakan peramban web.

Informasi email yang Anda butuhkan untuk diberikan ke Amazon SES

Ketika Anda mengirim email dengan Amazon SES, informasi email yang Anda butuhkan untuk disediakan tergantung pada cara Anda memanggil Amazon SES. Anda dapat menyediakan jumlah minimal informasi dan memiliki Amazon SES untuk mengurus semua format Anda. Atau, jika Anda ingin melakukan sesuatu yang lebih lanjut seperti mengirim lampiran, Anda dapat menyediakan pesan mentah itu sendiri. Bagian berikut meninjau hal yang Anda butuhkan untuk disediakan ketika Anda mengirim email dengan menggunakan API Amazon SES, antarmuka SMTP Amazon SES, atau konsol Amazon SES.

API Amazon SES

Jika Anda memanggil API Amazon SES secara langsung, Anda memanggil API `SendEmail` atau `SendRawEmail`. Jumlah informasi yang Anda butuhkan untuk disediakan tergantung pada API yang Anda panggil.

- `SendEmail API` mengharuskan Anda untuk hanya menyediakan alamat sumber, alamat tujuan, subjek pesan, dan isi pesan. Anda dapat menyediakan alamat "Balas Ke" secara opsional. Ketika

Anda memanggil API ini, Amazon SES secara otomatis merakit pesan email Multipurpose Internet Mail Extensions (MIME) beberapa bagian yang diformat dengan benar yang dioptimalkan untuk ditampilkan oleh perangkat lunak klien email. Untuk informasi lebih lanjut, lihat Mengirim email yang diformat menggunakan Amazon SES API.

- API `SendRawEmail` menyediakan Anda fleksibilitas untuk memformat dan mengirim pesan email mentah Anda sendiri dengan menentukan header, bagian MIME, dan tipe konten. `SendRawEmail` biasanya digunakan oleh pengguna tingkat lanjut. Anda perlu menyediakan isi pesan dan semua bidang header yang ditentukan seperti yang diperlukan dalam spesifikasi Format Pesan Internet (RFC 5322). Untuk informasi lebih lanjut, lihat Mengirim email mentah menggunakan Amazon SES API v2.

Jika Anda menggunakan AWS SDK untuk memanggil API Amazon SES, Anda memberikan informasi yang tercantum di atas ke fungsi yang sesuai (misalnya, `SendEmail` dan `SendRawEmail` untuk Java).

Untuk informasi selengkapnya tentang mengirim email menggunakan API Amazon SES, lihat Menggunakan Amazon SES API untuk mengirim email.

Antarmuka SMTP Amazon SES

Ketika Anda mengakses Amazon SES melalui antarmuka SMTP, aplikasi klien SMTP Anda merakit pesan, sehingga informasi yang Anda butuhkan untuk disediakan tergantung pada aplikasi yang Anda gunakan. Minimal, pertukaran SMTP antara klien dan server memerlukan alamat sumber, alamat tujuan, dan data pesan.

Untuk informasi selengkapnya tentang mengirim email menggunakan antarmuka SMTP Amazon SES, lihat Menggunakan SES SMTP antarmuka Amazon untuk mengirim email.

Konsol Amazon SES

Ketika Anda mengirim email dengan menggunakan konsol Amazon SES, jumlah informasi yang Anda butuhkan untuk diberikan tergantung pada yang Anda pilih untuk mengirim email berformat atau email mentah.

- Untuk mengirim email berformat, Anda harus menyediakan alamat sumber, alamat tujuan, subjek pesan, dan isi pesan. Amazon SES secara otomatis merakit pesan email MIME beberapa bagian yang diformat dengan benar yang dioptimalkan untuk ditampilkan oleh perangkat lunak klien email. Anda juga dapat menentukan bidang balasan ke dan jalur kembali.

- Untuk mengirim email mentah, Anda menyediakan alamat sumber, alamat tujuan, dan konten pesan, yang harus berisi isi pesan dan semua bidang header yang ditentukan seperti yang diperlukan dalam spesifikasi Format Pesan Internet (RFC 5322).

Memahami pengiriman email di Amazon SES

Anda ingin penerima membaca email Anda, menganggap mereka berharga, dan tidak memberi label mereka sebagai spam. Dengan kata lain, Anda ingin memaksimalkan kemampuan pengiriman email—persentase email Anda yang masuk di kotak masuk penerima. Topik ini mengulas konsep pengiriman email yang harus Anda ketahui saat menggunakan AmazonSES.

Untuk memaksimalkan kemampuan pengiriman email, Anda perlu memahami masalah pengiriman email, secara proaktif mengambil langkah-langkah untuk mencegahnya, memiliki informasi tentang status email yang Anda kirim, lalu meningkatkan program pengiriman email Anda, jika perlu, untuk lebih meningkatkan kemungkinan keberhasilan pengiriman. Bagian berikut meninjau konsep di balik langkah-langkah ini dan bagaimana Amazon SES membantu Anda melalui prosesnya.

Pahami masalah pengiriman email

Dalam kebanyakan kasus, pesan Anda berhasil dikirim ke penerima yang mengharapkan pesan tersebut. Di beberapa kasus, bagaimanapun, pengiriman mungkin gagal, atau penerima mungkin tidak ingin menerima surat yang Anda kirim. Pentalan, aduan, dan daftar penekanan terkait dengan masalah pengiriman ini dan dijelaskan dalam bagian berikut.

Pantulan

Jika penerima penerima Anda (misalnya, penyedia email) gagal mengirimkan pesan Anda ke penerima, penerima akan memantulkan pesan kembali ke Amazon. SES Amazon SES kemudian memberi tahu Anda tentang email yang dipantulkan melalui email atau melalui Amazon Simple Notification Service SNS (Amazon), tergantung pada bagaimana Anda mengatur sistem Anda. Untuk informasi selengkapnya, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.

Ada pentalan keras dan pentalan lunak, sebagai berikut:

- Pentalan keras – Kegagalan pengiriman email secara terus-menerus. Misalnya, kotak surat tidak ada. Amazon SES tidak mencoba lagi pantulan keras, dengan pengecualian kegagalan pencarian. DNS Kami sangat merekomendasikan agar Anda tidak membuat upaya pengiriman berulang ke alamat email yang mengalami pentalan keras.
- Pentalan lunak – Kegagalan pengiriman email sementara. Misalnya, kotak surat penuh, ada terlalu banyak sambungan (juga disebut throttling), atau waktu sambungan habis. Amazon SES mencoba kembali soft bounce beberapa kali. Jika email masih tidak dapat dikirimkan, Amazon SES berhenti mencobanya lagi.

Amazon SES memberi tahu Anda tentang pantulan keras dan pantulan lunak yang tidak akan lagi dicoba lagi. Namun, hanya pantulan keras yang dihitung terhadap rasio pentalan Anda dan metrik bouncing yang Anda ambil menggunakan konsol Amazon SES atau. `GetSendStatistics` API

Pentalan juga dapat sinkron atau asinkron. Sebuah pentalan sinkron terjadi sementara server email dari pengirim dan penerima secara aktif berkomunikasi. Sebuah pentalan asinkron terjadi ketika penerima awalnya menerima pesan email untuk pengiriman dan kemudian gagal mengirimkannya ke penerima.

Keluhan

Sebagian besar program klien email menyediakan tombol berlabel "Tandai sebagai Spam," atau serupa, yang memindahkan pesan ke folder spam, dan meneruskannya ke penyedia email.

Selain itu, sebagian besar penyedia email mempertahankan alamat penyalahgunaan (misalnya, abuse@example.net), tempat pengguna dapat meneruskan pesan email yang tidak diinginkan dan meminta penyedia email mengambil tindakan untuk mencegahnya. Dalam kedua kasus ini, penerima membuat aduan. Jika penyedia email menyimpulkan bahwa Anda adalah spammer, dan Amazon SES memiliki loop umpan balik yang diatur dengan penyedia email, maka penyedia email akan mengirim keluhan kembali ke Amazon. SES Ketika Amazon SES menerima keluhan seperti itu, Amazon meneruskan keluhan kepada Anda baik melalui email atau dengan menggunakan SNS pemberitahuan Amazon, tergantung pada bagaimana Anda mengatur sistem Anda. Untuk informasi selengkapnya, lihat Menyiapkan pemberitahuan acara untuk Amazon SES. Kami merekomendasikan agar Anda tidak membuat upaya pengiriman berulang ke alamat email yang menghasilkan aduan.

Daftar penekanan global

Daftar penindasan SES global Amazon, yang dimiliki dan dikelola oleh SES untuk melindungi reputasi alamat di kumpulan IP SES bersama, berisi alamat email penerima yang baru-baru ini menyebabkan pantulan keras bagi pelanggan mana punSES. Jika Anda mencoba mengirim email melalui SES alamat yang ada di daftar penindasan, panggilan untuk SES berhasil, tetapi SES memperlakukan email sebagai pantulan keras alih-alih mencoba mengirimnya. Seperti setiap pentalan keras, daftar penekanan mementalkan hitungan terhadap kuota pengiriman dan tingkat pentalan Anda. Alamat email dapat tetap berada dalam daftar penekanan hingga 14 hari. Jika Anda yakin bahwa alamat email yang Anda coba kirim valid, Anda dapat mengganti daftar penindasan global dengan memastikan alamat tersebut tidak tercantum dalam daftar penekanan tingkat akun Anda dan masih SES akan mencoba pengiriman, tetapi jika memantul, pantulan akan memengaruhi reputasi Anda sendiri, tetapi tidak ada orang lain yang akan memantul karena mereka tidak dapat mengirim ke alamat email itu jika mereka tidak menggunakan suplevel akun mereka sendiri daftar penindasan. Untuk memahami lebih lanjut tentang daftar penekanan tingkat akun, lihat. Menggunakan daftar SES penindasan tingkat akun Amazon

Jadilah proaktif

Salah satu masalah terbesar dengan email di Internet adalah email dalam jumlah besar yang tidak diminta (spam). Penyedia email mengambil langkah-langkah ekstensif untuk mencegah pelanggan mereka menerima spam. Amazon SES juga mengambil langkah-langkah untuk mengurangi kemungkinan penyedia email menganggap email Anda sebagai spam. Amazon SES menggunakan verifikasi, otentikasi, pengiriman kuota, dan pemfilteran konten. Amazon SES juga mempertahankan reputasi tepercaya dengan penyedia email dan mengharuskan Anda mengirim email berkualitas tinggi. Amazon SES melakukan beberapa hal tersebut untuk Anda secara otomatis (misalnya, pemfilteran konten); dalam kasus lain, Amazon menyediakan alat (seperti otentikasi), atau memandu

Anda ke arah yang benar (mengirim kuota). Bagian berikut menyediakan informasi selengkapnya tentang setiap konsep.

Verifikasi

Sayangnya, spammer dapat memalsukan header email dan memalsukan alamat email asal sehingga muncul seolah-olah email berasal dari sumber yang berbeda. Untuk menjaga kepercayaan antara penyedia email dan AmazonSES, Amazon SES perlu memastikan bahwa pengirimnya adalah siapa yang mereka katakan. Oleh karena itu Anda diminta untuk memverifikasi semua alamat email dari mana Anda mengirim email melalui Amazon SES untuk melindungi identitas pengiriman Anda. Anda dapat memverifikasi alamat email dengan menggunakan SES konsol Amazon atau dengan menggunakan Amazon SESAPI. Anda juga dapat memverifikasi seluruh domain. Untuk informasi selengkapnya, silakan lihat Membuat identitas alamat email dan Membuat identitas domain.

Jika akun Anda masih di SES kotak pasir Amazon, Anda juga perlu memverifikasi semua alamat penerima kecuali alamat yang disediakan oleh simulator SES kotak surat Amazon. Untuk informasi tentang keluar dari sandbox, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon). Untuk informasi selengkapnya tentang simulator kotak surat, lihat Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual.

Autentikasi

Autentikasi adalah cara lain yang dapat Anda tunjukkan kepada penyedia email bahwa Anda jujur seperti yang Anda katakan. Ketika Anda mengautentikasi email, Anda memberikan bukti bahwa Anda adalah pemilik akun dan email Anda belum diubah saat transit. Di beberapa kasus, penyedia email menolak untuk meneruskan email yang tidak diautentikasi. Amazon SES mendukung dua metode otentikasi: Kerangka Kebijakan Pengirim (SPF) dan DomainKeys Identified Mail (DKIM). Untuk informasi selengkapnya, lihat Mengonfigurasi identitas di Amazon SES.

Kuota pengiriman

Jika penyedia email mendeteksi lonjakan volume atau laju email secara tiba-tiba dan tak terduga, penyedia email mungkin menduga Anda adalah spammer dan memblokir email Anda. Oleh karena itu, setiap SES akun Amazon memiliki serangkaian kuota pengiriman. Kuota ini membatasi jumlah email yang dapat Anda kirim dalam jangka waktu 24 jam, dan jumlah yang dapat Anda kirim per detik. Kuota pengiriman ini membantu melindungi kepercayaan Anda dengan penyedia email.

Dalam kebanyakan kasus, jika Anda adalah pengguna baru, Amazon SES memungkinkan Anda mengirim sejumlah kecil email setiap hari. Jika surat yang Anda kirim dapat diterima oleh penyedia email, kami akan secara otomatis meningkatkan kuota ini. Kuota pengiriman Anda terus meningkat

seiring berjalannya waktu sehingga Anda dapat mengirim email dalam jumlah yang lebih besar dengan laju yang lebih cepat. Anda juga dapat membuat kasus Peningkatan Batas SES Pengiriman untuk meminta peningkatan kuota tambahan.

Untuk informasi selengkapnya tentang kuota pengiriman, dan cara meningkatkan kuota, lihat Mengelola batas pengiriman Amazon SES Anda.

Pemfilteran konten

Banyak penyedia email menggunakan pemfilteran konten untuk menentukan jika email masuk adalah spam. Filter konten mencari konten yang dipertanyakan dan memblokir email jika email sesuai dengan profil spam. Amazon juga SES menggunakan filter konten. Saat aplikasi Anda mengirim permintaan ke AmazonSES, Amazon SES mengumpulkan pesan email atas nama Anda dan kemudian memindai header dan badan pesan untuk menentukan apakah mereka berisi konten yang mungkin dianggap sebagai spam oleh penyedia email. Jika pesan Anda terlihat seperti spam ke filter konten yang SES digunakan Amazon, reputasi Anda dengan Amazon SES akan terpengaruh secara negatif.

Amazon SES juga memindai semua pesan dari virus. Jika pesan berisi virus, Amazon SES tidak mencoba mengirimkan pesan ke server email penerima.

Reputasi

Untuk pengiriman email, reputasi—ukuran kepercayaan bahwa alamat IP, alamat email, atau domain pengiriman bukanlah sumber spam—adalah hal yang penting. Amazon SES mempertahankan reputasi yang kuat dengan penyedia email sehingga mereka mengirimkan email Anda ke kotak masuk penerima Anda. Demikian pula, Anda perlu mempertahankan reputasi tepercaya dengan AmazonSES. Anda membangun reputasi Anda dengan Amazon SES dengan mengirimkan konten berkualitas tinggi. Ketika Anda mengirim konten berkualitas tinggi, reputasi Anda menjadi lebih dipercaya dari waktu ke waktu dan Amazon SES meningkatkan kuota pengiriman Anda. Pantulan dan keluhan yang berlebihan berdampak negatif pada reputasi Anda dan dapat menyebabkan Amazon SES mengurangi kuota pengiriman untuk akun Anda, atau mengakhiri akun Amazon Anda. SES

Salah satu cara untuk membantu mempertahankan reputasi Anda adalah dengan menggunakan simulator kotak surat saat Anda menguji sistem Anda, alih-alih mengirim ke alamat email yang telah Anda buat sendiri. Email ke simulator kotak surat tidak dihitung terhadap metrik pentalan dan aduan Anda. Untuk informasi selengkapnya tentang simulator kotak surat, lihat Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual.

Email berkualitas tinggi

Email berkualitas tinggi adalah email yang dianggap berharga dan ingin terima oleh penerima. Nilai berarti hal yang berbeda untuk penerima yang berbeda dan dapat datang dalam bentuk penawaran, konfirmasi pesanan, penerimaan, buletin, dll. Pada akhirnya, kemampuan pengiriman Anda bergantung pada kualitas email yang Anda kirim karena penyedia email memblokir email yang dianggap berkualitas rendah.

Memiliki informasi di setiap waktu

Apakah pengiriman Anda gagal, penerima mengeluh tentang email Anda, atau Amazon SES berhasil mengirimkan email ke server email penerima, Amazon SES membantu Anda melacak masalah dengan memberikan pemberitahuan dan dengan memungkinkan Anda untuk dengan mudah memantau statistik penggunaan Anda.

Pemberitahuan

Saat email memantul, penyedia email memberi tahu Amazon, dan SES SES Amazon memberi tahu Anda. Amazon SES memberi tahu Anda tentang pantulan keras dan pantulan lunak yang tidak akan dicoba lagi oleh SES Amazon. Banyak penyedia email juga meneruskan keluhan, dan Amazon SES mengatur loop umpan balik keluhan dengan penyedia email utama sehingga Anda tidak perlu melakukannya. Amazon SES dapat memberi tahu Anda tentang pantulan, keluhan, dan pengiriman yang berhasil dengan dua cara: Anda dapat mengatur akun Anda untuk menerima pemberitahuan melalui AmazonSNS, atau Anda dapat menerima pemberitahuan melalui email (hanya pantulan dan keluhan). Untuk informasi selengkapnya, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.

Statistik penggunaan

Amazon SES menyediakan statistik penggunaan sehingga Anda dapat melihat pengiriman yang gagal untuk menentukan dan menyelesaikan akar penyebabnya. Anda dapat melihat statistik penggunaan Anda dengan menggunakan SES konsol Amazon atau dengan menelepon Amazon SESAPI. Anda dapat melihat seberapa banyak pengiriman, pentalan, aduan, dan email yang ditolak karena terinfeksi virus yang Anda miliki, dan Anda juga dapat melihat kuota pengiriman Anda untuk memastikan bahwa Anda tetap berada di dalamnya.

Tingkatkan program pengiriman email Anda

Jika Anda mendapatkan sejumlah besar pentalan dan aduan, sekarang saatnya untuk menilai kembali strategi pengiriman email Anda. Ingatlah bahwa bouncing, keluhan, dan upaya yang

berlebihan untuk mengirim email berkualitas rendah merupakan penyalahgunaan dan menempatkan Anda Akun AWS pada risiko penghentian. Pada akhirnya, Anda harus yakin bahwa Anda menggunakan Amazon SES untuk mengirim email berkualitas tinggi dan hanya mengirim email ke penerima yang ingin menerimanya.

t-least-oncePengiriman

Amazon SES menyimpan salinan pesan Anda di beberapa server untuk redundansi dan ketersediaan tinggi. Pada kesempatan yang jarang terjadi, salah satu server yang menyimpan salinan pesan mungkin tidak tersedia saat Anda menerima atau menghapus pesan.

Jika ini terjadi, salinan pesan tidak akan dihapus di server yang tidak tersedia tersebut, dan Anda mungkin mendapatkan salinan pesan itu lagi saat menerima pesan. Rancang aplikasi Anda menjadi idempoten (tidak boleh terpengaruh secara negatif saat memproses pesan yang sama lebih dari sekali).

Praktik terbaik untuk mengirim email menggunakan Amazon SES

Cara Anda mengelola komunikasi email dengan pelanggan disebut sebagai program email Anda. Ada beberapa faktor yang dapat menyebabkan keberhasilan atau kegagalan program email Anda; faktor-faktor ini mungkin tampak membingungkan atau pelik pada awalnya. Namun, dengan memahami cara pengiriman email, dan mengikuti praktik terbaik tertentu, Anda dapat meningkatkan peluang email Anda berhasil mencapai kotak masuk pelanggan.

Topik

- Metrik keberhasilan program email
- Mempertahankan reputasi pengirim yang positif

Metrik keberhasilan program email

Ada beberapa metrik yang membantu mengukur keberhasilan program email Anda.

Bagian ini menyediakan informasi tentang metrik berikut:

- Pentalan
- Aduan
- Kualitas pesan

Pentalan

Pentalan terjadi ketika email tidak dapat dikirim ke penerima yang dimaksudkan. Ada dua tipe pentalan: pentalan keras dan pentalan lunak. Pentalan keras terjadi ketika email tidak dapat dikirim karena masalah terus-menerus, seperti ketika alamat email tidak ada. Pentalan lunak terjadi ketika masalah sementara mencegah pengiriman email. Pentalan lunak dapat terjadi ketika penerima kotak masuk penuh, atau ketika server penerima sementara tidak tersedia. Amazon SES menangani soft bounce dengan mencoba mengirimkan kembali email soft bounce untuk jangka waktu tertentu.

Sangat penting bahwa Anda memantau jumlah pentalan keras dalam program email Anda, dan Anda menghapus alamat email pentalan keras dari daftar penerima Anda. Ketika penerima email mendeteksi pentalan keras tingkat tinggi, mereka menganggap bahwa Anda tidak tahu penerima Anda dengan baik. Akibatnya, tingkat pentalan keras yang tinggi dapat berdampak negatif terhadap kemampuan pengiriman pesan email Anda.

Panduan berikut dapat membantu Anda menghindari pentalan dan meningkatkan reputasi pengirim Anda:

- Cobalah untuk menjaga tingkat pentalan keras Anda di bawah 5%. Semakin sedikit hard bounce dalam program email Anda, semakin besar kemungkinan ISPs akan melihat pesan Anda sebagai sah dan berharga. Tingkat ini harus dianggap sebagai tujuan yang masuk akal dan dapat dicapai, tetapi bukan aturan universal di semua. ISPs
- Jangan pernah menyewa atau membeli daftar email. Daftar ini mungkin berisi sejumlah besar alamat yang tidak valid, yang dapat menyebabkan tingkat pentalan keras Anda meningkat secara drastis. Selain itu, daftar ini dapat berisi jebakan spam—alamat email yang secara khusus digunakan untuk menangkap pengirim yang tidak sah. Jika pesan Anda masuk dalam jebakan spam, tingkat pengiriman dan reputasi pengirim Anda dapat rusak tidak dapat dibatalkan.
- Buat daftar Anda tetap mutakhir. Jika Anda belum mengirim email ke penerima Anda dalam waktu lama, cobalah untuk memvalidasi status pelanggan Anda melalui beberapa cara lain (seperti aktivitas masuk situs web atau riwayat pembelian).
- Jika Anda tidak memiliki metode untuk memverifikasi status pelanggan Anda, pertimbangkan untuk mengirim email win-back. Sebuah email win-back yang khas menyebutkan bahwa Anda belum mendengar dari pelanggan dalam beberapa saat, dan mendorong pelanggan untuk mengonfirmasi bahwa mereka masih ingin menerima email Anda. Setelah mengirim email win-back, bersihkan semua penerima yang tidak merespons dari daftar Anda.

Ketika Anda menerima pentalan, sangat penting jika Anda merespons mereka dengan tepat dengan mengamati aturan berikut:

- Jika alamat email mengalami pentalan keras, segera hapus alamat tersebut dari daftar Anda. Jangan mencoba untuk mengirim ulang pesan ke alamat yang mengalami pentalan keras. Pantulan keras berulang bertambah, dan pada akhirnya merusak reputasi Anda dengan penerima. ISP
- Pastikan alamat yang Anda gunakan untuk menerima notifikasi pentalan dapat menerima email. Untuk informasi selengkapnya tentang pengaturan notifikasi pentalan dan aduan, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.
- Jika email masuk Anda datang kepada Anda dariISP, alih-alih melalui server internal Anda sendiri, masuknya pemberitahuan bouncing dapat mendarat di folder spam Anda atau dihapus sepenuhnya. Idealnya, Anda tidak harus menggunakan alamat email yang di-host untuk menerima pentalan. Namun, jika Anda perlu, sering periksa folder spam, dan jangan menandai pesan pentalan sebagai spam. Di AmazonSES, Anda dapat menentukan alamat tempat notifikasi bouncing dikirim.
- Biasanya, pentalan menyediakan alamat kotak surat yang menolak pengiriman. Namun, jika Anda memerlukan lebih banyak data terperinci untuk memetakan alamat penerima ke kampanye email tertentu, sertakan header X dengan nilai yang dapat Anda lacak kembali ke sistem pelacakan internal Anda. Untuk informasi lebih lanjut, lihat Bidang SES header Amazon.

Aduan

Aduan terjadi saat penerima email mengeklik tombol "Tandai sebagai Spam" (atau yang setara) di klien email berbasis web mereka. Jika Anda mengumpulkan sejumlah besar keluhan ini, ISP asumsi bahwa Anda mengirim spam. Hal ini berdampak negatif pada tingkat kemampuan pengiriman dan reputasi pengirim Anda. Beberapa, tetapi tidak semua, ISPs akan memberi tahu Anda ketika keluhan dilaporkan; ini dikenal sebagai loop umpan balik. Amazon SES secara otomatis meneruskan keluhan dari umpan balik penawaran ISPs itu kepada Anda.

Panduan berikut dapat membantu Anda menghindari aduan dan meningkatkan reputasi pengirim Anda:

- Cobalah untuk mempertahankan tingkat aduan Anda di bawah 0,1%. Semakin sedikit keluhan dalam program email Anda, ISPs semakin besar kemungkinan pesan Anda sah dan berharga. Tingkat ini harus dianggap sebagai tujuan yang masuk akal dan dapat dicapai, tetapi bukan aturan universal di semua. ISPs

- Jika pelanggan mengeluh tentang email pemasaran, Anda harus segera berhenti mengirim email pemasaran pelanggan. Namun, jika program email Anda juga mencakup tipe email lain (seperti notifikasi atau email transaksional), mungkin dapat diterima untuk terus mengirim tipe pesan tersebut ke penerima yang mengeluarkan aduan.
- Seperti pentalan keras, jika Anda memiliki daftar yang belum pernah Anda kirim email dalam beberapa saat, pastikan penerima memahami alasan mereka menerima pesan Anda. Kami merekomendasikan Anda mengirim pesan selamat datang yang mengingatkan mereka tentang Anda dan alasan Anda menghubungi mereka.

Ketika Anda menerima aduan, sangat penting jika Anda merespons mereka dengan tepat dengan mematuhi aturan berikut:

- Pastikan alamat yang Anda gunakan untuk menerima notifikasi aduan dapat menerima email. Untuk informasi selengkapnya tentang pengaturan notifikasi pentalan dan aduan, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.
- Pastikan pemberitahuan keluhan Anda tidak ditandai sebagai spam oleh sistem email ISP atau email Anda.
- Notifikasi aduan biasanya berisi isi email; hal ini berbeda dari notifikasi pentalan, yang hanya mencakup header email. Namun, dalam notifikasi aduan, alamat email dari individu yang mengeluarkan aduan dihapus. Gunakan header X kustom atau pengidentifikasi khusus yang disematkan di isi email sehingga Anda dapat mengidentifikasi alamat email yang mengeluarkan aduan. Teknik ini memudahkan untuk mengidentifikasi alamat yang mengadu sehingga Anda dapat menghapusnya dari daftar penerima Anda.

Kualitas pesan

Penerima email menggunakan filter konten untuk mendeteksi atribut tertentu dalam pesan Anda untuk mengidentifikasi pesan Anda sah atau tidak. Filter konten ini secara otomatis meninjau konten pesan Anda untuk mengidentifikasi sifat-sifat umum dari pesan berbahaya yang tidak diinginkan. Amazon SES menggunakan teknologi penyaringan konten untuk membantu mendeteksi dan memblokir pesan yang mengandung malware sebelum dikirim.

Jika filter konten penerima email menentukan jika pesan Anda berisi karakteristik spam atau email berbahaya, pesan Anda kemungkinan besar akan ditandai dan dialihkan dari kotak masuk penerima.

Ingat hal berikut saat merancang email Anda:

- Filter konten modern cerdas, terus beradaptasi dan berubah. Filter tidak bergantung pada set aturan yang telah ditetapkan. Layanan pihak ketiga seperti ReturnPathatau Litmus dapat membantu mengidentifikasi konten dalam email Anda yang dapat memicu filter konten.
- Jika email Anda berisi tautan, periksa tautan tersebut terhadap Blackhole Lists (DNSBLs) DNS berbasis, seperti yang ditemukan URIBLdi.com dan SURBL.org. URLs
- Hindari menggunakan penyingkat tautan. Pengirim berbahaya dapat menggunakan penyingkat tautan untuk menyembunyikan tujuan tautan yang sebenarnya. Ketika ISPs memperhatikan bahwa layanan pemendekan tautan — bahkan yang paling terkemuka — digunakan untuk tujuan jahat, mereka mungkin menolak akses ke layanan tersebut sama sekali. Jika email Anda berisi tautan ke layanan penyingkat tautan yang telah ditambahkan ke daftar penyangkalan, tautan tersebut tidak akan masuk ke kotak masuk pelanggan, dan keberhasilan kampanye email Anda akan terganggu.
- Uji setiap tautan di email Anda untuk memastikan jika tautan tersebut mengarah ke halaman yang dimaksud.
- Pastikan situs web Anda menyertakan dokumen Kebijakan Privasi dan Ketentuan Penggunaan, dan dokumen-dokumen tersebut mutakhir. Ini adalah praktik yang baik untuk menautkan ke dokumen ini dari setiap email yang Anda kirim. Menyediakan tautan ke dokumen-dokumen ini menunjukkan bahwa Anda tidak menyembunyikan apa pun dari pelanggan Anda, yang dapat membantu membangun hubungan kepercayaan.
- Jika Anda berencana untuk mengirim konten frekuensi tinggi (seperti pesan "transaksi harian"), pastikan konten email Anda berbeda dengan setiap deployment. Ketika Anda mengirim pesan dengan frekuensi tinggi, Anda harus memastikan bahwa pesan tersebut tepat waktu dan relevan, bukannya berulang-ulang dan mengganggu.

Mempertahankan reputasi pengirim yang positif

Di AmazonSES, reputasi pengirim mengacu pada kredibilitas dan kepercayaan pengirim email seperti yang dirasakan oleh penyedia email dan filter spam. Ini adalah ukuran seberapa besar kemungkinan email Anda dianggap sah dan berhasil dikirim ke kotak masuk penerima.

Bagian berikut memperkenalkan prinsip pengiriman email inti yang harus Anda perhatikan untuk memastikan bahwa komunikasi email Anda menjangkau audiens yang dituju sambil mempertahankan reputasi pengirim yang baik.

Pertimbangan alamat domain dan "Dari"

- Pikirkan baik-baik alamat tempat Anda mengirim email. Alamat "Dari" adalah salah satu bagian pertama dari informasi yang dilihat penerima Anda, dan karena itu dapat meninggalkan kesan

pertama yang abadi. Selain itu, beberapa ISPs mengaitkan reputasi Anda dengan alamat “Dari” Anda.

- Pertimbangkan untuk menggunakan subdomain untuk tipe komunikasi berbeda. Sebagai contoh, asumsikan Anda mengirim email dari domain example.com, dan Anda berencana untuk mengirim kedua pesan pemasaran dan transaksional. Daripada mengirim semua pesan Anda dari example.com, kirim pesan pemasaran Anda dari subdomain seperti marketing.example.com, dan pesan transaksional Anda dari subdomain seperti orders.example.com. Subdomain unik mengembangkan reputasinya sendiri. Menggunakan subdomain mengurangi risiko kerusakan reputasi Anda jika, misalnya, komunikasi pemasaran Anda mendarat di perangkap spam atau memicu filter konten.
- Jika Anda berencana untuk mengirim pesan dalam jumlah besar, jangan kirim pesan tersebut dari alamat ISP berbasis seperti sender@hotmail.com. Jika ada ISP pemberitahuan sejumlah besar pesan yang berasal dari sender@hotmail.com, email tersebut diperlakukan berbeda dari email yang berasal dari domain pengiriman email keluar yang Anda miliki.
- Bekerja dengan registrar domain Anda untuk memastikan bahwa WHOIS informasi untuk domain Anda akurat. Menjaga kejujuran dan up-to-date WHOIS catatan menunjukkan bahwa Anda menghargai transparansi, dan memungkinkan pengguna untuk dengan cepat mengidentifikasi apakah domain Anda sah atau tidak.
- Hindari menggunakan alamat tidak ada balasan, seperti no-reply@example.com, sebagai alamat "Dari" atau "Balasan ke" Anda. Menggunakan alamat email no-reply@ mengirimkan pesan yang jelas kepada penerima Anda: bahwa Anda tidak menawarkan cara untuk menghubungi Anda, dan Anda tidak tertarik dengan umpan balik mereka.

Autentikasi

- Otentikasi domain Anda dengan SPFdan senderId. Metode autentikasi ini mengonfirmasi ke penerima email bahwa setiap email yang Anda kirim sebenarnya dari domain yang diklaimnya berasal.
- Tanda tangani surat keluar Anda dengan DKIM. Langkah ini mengonfirmasi kepada penerima bahwa konten belum diubah saat transit antara pengirim dan penerima.
- Anda dapat menguji pengaturan otentikasi untuk keduanya SPF dan DKIM dengan mengirim email ke alamat email ISP berbasis yang Anda miliki, seperti akun Gmail atau Hotmail pribadi, dan kemudian melihat header pesan. Header menunjukkan upaya Anda untuk mengautentikasi dan menandatangani pesan berhasil.

Membangun dan mempertahankan daftar Anda

- Menerapkan strategi keikutsertaan ganda. Saat pengguna mendaftar untuk menerima email dari Anda, kirimi mereka pesan dengan tautan konfirmasi, dan jangan mulai mengirimi mereka email hingga mereka mengonfirmasi alamat mereka dengan mengeklik tautan tersebut. Strategi keikutsertaan ganda membantu mengurangi jumlah pentalan keras yang dihasilkan dari kesalahan ketik.
- Saat mengumpulkan alamat email dengan formulir berbasis web, lakukan validasi minimal pada alamat tersebut saat pengiriman. Sebagai contoh, pastikan bahwa alamat yang Anda kumpulkan terbentuk dengan baik (yaitu, mereka berada dalam format recipient@example.com), dan mereka merujuk ke domain dengan catatan MX yang valid.
- Berhati-hatilah saat mengizinkan input yang ditentukan pengguna diteruskan ke Amazon SES tanpa dicentang. Pendaftaran forum dan pengiriman formulir menghadirkan risiko unik karena konten benar-benar dibuat pengguna, dan spammer dapat mengisi formulir dengan konten mereka sendiri. Anda bertanggung jawab untuk memastikan bahwa Anda hanya mengirim email dengan konten berkualitas tinggi.
- Hal ini sangat tidak mungkin alias standar (seperti postmaster@, abuse@, atau noc@) akan pernah mendaftar untuk email Anda dengan sengaja. Pastikan jika Anda hanya mengirim pesan kepada orang-orang nyata yang benar-benar ingin menerima pesan Anda. Aturan ini berlaku terutama untuk alias standar, yang biasanya disediakan untuk pengawas email. Alias ini dapat ditambahkan ke daftar Anda sebagai bentuk sabotase, untuk merusak reputasi Anda.

Kepatuhan

- Perhatikan hukum dan peraturan pemasaran email dan anti-spam di negara dan wilayah tempat Anda mengirim email. Anda bertanggung jawab untuk memastikan bahwa email yang Anda kirim mematuhi hukum ini. Panduan ini tidak mencakup hukum-hukum ini, jadi penting bagi Anda untuk menelitinya. Untuk daftar hukum, lihat Undang-Undang Spam Email menurut Negara di Wikipedia.
- Selalu berkonsultasi dengan pengacara untuk mendapatkan nasihat hukum.

Menggunakan Amazon SES dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDK) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK for C++	AWS SDK for C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK for Go	AWS SDK for Go contoh kode
AWS SDK for Java	AWS SDK for Java contoh kode
AWS SDK for JavaScript	AWS SDK for JavaScript contoh kode
AWS SDK for Kotlin	AWS SDK for Kotlin contoh kode
AWS SDK for .NET	AWS SDK for .NET contoh kode
AWS SDK for PHP	AWS SDK for PHP contoh kode
AWS Tools for PowerShell	Alat untuk contoh PowerShell kode
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) contoh kode
AWS SDK for Ruby	AWS SDK for Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Untuk contoh khusus untuk Amazon SES, lihatContoh kode untuk Amazon SES menggunakanAWS SDK.

Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Memulai dengan Amazon Simple Email Service

Bab ini memandu Anda melalui tugas-tugas yang diperlukan untuk pengaturan awal Amazon SES serta tutorial untuk membantu Anda memulai.

Topik

- Menyiapkan Amazon Simple Email Service
- Bermigrasi ke Amazon SES dari solusi pengiriman email lain
- Minta akses produksi (Pindah dari SES kotak pasir Amazon)

Menyiapkan Amazon Simple Email Service

Sebelum Anda mulai menggunakan AmazonSES, Anda harus menyelesaikan tugas-tugas berikut.

Tugas

- Mendaftar untuk AWS
- Siapkan akun SES Anda
- Berikan akses terprogram (Untuk berinteraksi dengan SES di luar konsol)
- Unduh AWS SDK (Untuk menggunakan SESAPIs)

Mendaftar untuk AWS

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

Untuk mendaftar untuk Akun AWS

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

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

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

Siapkan akun SES Anda

Mulailah SES dengan memverifikasi alamat email dan mengirim domain sehingga Anda dapat mulai mengirim email SES dan meminta akses produksi untuk akun Anda dengan menggunakan panduan pengaturan SES akun.

Menggunakan panduan penyiapan SESakun untuk menyiapkan akun

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Pilih Mulai dari beranda SES konsol dan wizard akan memandu Anda melalui langkah-langkah pengaturan SES akun Anda.

Wizard pengaturan SES akun hanya akan ditampilkan jika Anda belum membuat identitas apa pun (alamat email atau domain) di SES.

Berikan akses terprogram (Untuk berinteraksi dengan SES di luar konsol)

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar. AWS Management Console Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses programatis, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
Identitas tenaga kerja (Pengguna dikelola di Pusat IAM Identitas)	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengkonfigurasi yang akan AWS CLI digunakan AWS IAM Identity Center dalam Panduan AWS Command Line Interface Pengguna.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
		• Untuk AWS SDKs, alat, dan AWS APIs, lihat otentikasi Pusat IAM Identitas di Panduan Referensi Alat AWS SDKs dan Alat.
IAM	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk dalam Menggunakan kredensil sementara dengan AWS sumber daya di IAMPanduan Pengguna.
IAM	(Tidak direkomendasikan) Gunakan kredensi jangka panjang untuk menandata ngani permintaan terprogra m ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk mengetahui AWS CLI, lihat Mengautentikasi menggunakan kredensil IAM pengguna di Panduan Pengguna.AWS Command Line Interface • Untuk AWS SDKs dan alat, lihat Mengautentikasi menggunakan kredensil jangka panjang di Panduan Referensi Alat AWS SDKs dan Alat. • Untuk AWS APIs, lihat Mengelola kunci akses untuk IAM pengguna di Panduan IAM Pengguna.

Unduh AWS SDK (Untuk menggunakan SESAPIs)

Untuk memanggil SES APIs tanpa harus menangani detail tingkat rendah seperti merakit HTTP permintaan mentah, Anda dapat menggunakan file. AWS SDK AWS SDKsMenyediakan fungsi dan tipe data yang merangkum fungsionalitas SES dan layanan lainnya. AWS Untuk mengunduh AWS SDK, buka SDKs. Setelah Anda mengunduhSDK, buat file kredensial bersama dan tentukan kunci AWS akses Anda.

Bermigrasi ke Amazon SES dari solusi pengiriman email lain

Topik ini memberikan ikhtisar tentang langkah-langkah yang harus diambil jika ingin memindahkan solusi pengiriman email ke Amazon SES dari solusi yang di-host di lokasi, atau dari solusi yang dihosting di instans Amazon. EC2

Topik di bagian ini:

- Langkah 1. Verifikasikan domain Anda
- Langkah 2. Minta akses produksi
- Langkah 3. Konfigurasikan sistem autentikasi domain
- Langkah 4. Hasilkan SMTP kredensi Anda
- Langkah 5. Connect ke SMTP endpoint
- Langkah selanjutnya

Langkah 1. Verifikasikan domain Anda

Sebelum Anda dapat menggunakan Amazon SES untuk mengirim email, Anda harus memverifikasi identitas yang Anda rencanakan untuk mengirim email. Di AmazonSES, identitas dapat berupa alamat email atau seluruh domain. Saat memverifikasi domain, Anda dapat menggunakan Amazon SES untuk mengirim email dari alamat mana pun di domain tersebut. Untuk informasi lebih lanjut tentang memverifikasi domain, lihat Membuat identitas domain.

Langkah 2. Minta akses produksi

Saat pertama kali mulai menggunakan AmazonSES, akun Anda berada di lingkungan kotak pasir. Saat akun Anda berada di sandbox, Anda hanya dapat mengirim email ke alamat yang telah diverifikasi. Selain itu, ada pembatasan jumlah pesan yang dapat Anda kirim per hari, dan nomor

yang dapat Anda kirim per detik. Untuk informasi selengkapnya tentang meminta akses produksi, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).

Langkah 3. Konfigurasikan sistem autentikasi domain

Anda dapat mengonfigurasi domain Anda untuk menggunakan sistem otentikasi seperti DKIM danSPF. Langkah ini secara teknis opsional. Namun, dengan menyiapkan salah satu DKIM atau SPF (atau keduanya) untuk domain Anda, Anda dapat meningkatkan pengiriman email Anda, dan meningkatkan jumlah kepercayaan yang dimiliki pelanggan Anda terhadap Anda. Untuk informasi selengkapnya tentang pengaturanSPF, lihatMengautentikasi Email dengan SPF di Amazon SES. Untuk informasi selengkapnya tentang pengaturanDKIM, lihatMengautentikasi Email dengan DKIM di Amazon SES.

Langkah 4. Hasilkan SMTP kredensi Anda

Jika Anda berencana untuk mengirim email menggunakan aplikasi yang menggunakanSMTP, Anda harus menghasilkan SMTP kredensi. SMTPKredensi Anda berbeda dari kredensi reguler AWS Anda. Kredensi ini juga unik di setiap AWS Wilayah. Untuk informasi selengkapnya tentang membuat SMTP kredensi Anda, lihat. Memperoleh SES SMTP kredensi Amazon

Langkah 5. Connect ke SMTP endpoint

Jika Anda menggunakan agen transfer pesan seperti postfix atau sendmail, Anda harus memperbarui konfigurasi aplikasi tersebut untuk merujuk ke titik akhir Amazon. SES SMTP Untuk daftar lengkap SMTP titik akhir, lihatMenghubungkan ke titik SES SMTP akhir Amazon. Perhatikan bahwa SMTP kredensional yang Anda buat pada langkah sebelumnya dikaitkan dengan Wilayah tertentu AWS . Anda harus terhubung ke SMTP titik akhir di wilayah tempat Anda membuat SMTP kredensialnya.

Langkah selanjutnya

Pada titik ini, Anda siap untuk mulai mengirim email menggunakan AmazonSES. Namun, ada beberapa langkah opsional yang dapat Anda ambil.

- Anda dapat membuat set konfigurasi, yang merupakan set aturan yang diterapkan ke email yang Anda kirim. Misalnya, Anda dapat menggunakan set konfigurasi untuk menentukan tempat notifikasi dikirim saat email disampaikan, saat penerima membuka pesan atau mengeklik tautan di dalamnya, saat email terpental, dan kapan penerima menandai email Anda sebagai spam. Untuk informasi selengkapnya, lihat Menggunakan set konfigurasi di Amazon SES.

- Saat Anda mengirim email melalui AmazonSES, penting untuk memantau pantulan dan keluhan untuk akun Anda. Amazon SES menyertakan halaman konsol metrik reputasi yang dapat Anda gunakan untuk melacak pantulan dan keluhan untuk akun Anda. Untuk informasi selengkapnya, lihat Menggunakan metrik reputasi untuk melacak tingkat pentalan dan aduan. Anda juga dapat membuat CloudWatch alarm yang mengingatkan Anda ketika tarif ini terlalu tinggi. Untuk informasi selengkapnya tentang membuat CloudWatch alarm, lihatMembuat alarm pemantauan reputasi menggunakan CloudWatch.
- Pelanggan yang mengirim email dengan volume besar, atau mereka yang hanya ingin memiliki kendali penuh atas reputasi alamat IP mereka, dapat menyewa alamat IP khusus untuk biaya bulanan tambahan. Untuk informasi selengkapnya, lihat Alamat IP khusus untuk Amazon SES.

Minta akses produksi (Pindah dari SES kotak pasir Amazon)

Untuk membantu mencegah penipuan dan penyalahgunaan, dan untuk membantu melindungi reputasi Anda sebagai pengirim, kami menerapkan batasan tertentu pada SES akun Amazon baru.

Kami menempatkan semua akun baru di SES kotak pasir Amazon. Status kotak pasir untuk akun Anda unik per masing-masing Wilayah AWS. Saat akun Anda ada di kotak pasir, Anda dapat menggunakan semua fitur AmazonSES. Namun, saat akun Anda berada di sandbox, kami menerapkan pembatasan berikut ke akun Anda:

- Anda hanya dapat mengirim email ke alamat email dan domain terverifikasi, atau ke simulator SESkotak surat Amazon.
- Anda dapat mengirim maksimal 200 pesan per periode 24 jam.
- Anda dapat mengirim maksimal 1 pesan per detik.
- Untuk mengirim otorisasi, baik Anda maupun pengirim delegasi tidak dapat mengirim email ke alamat email yang tidak diverifikasi.
- Untuk penekanan tingkat akun, tindakan massal dan SES API panggilan yang terkait dengan manajemen daftar penekanan dinonaktifkan.

Ketika akun Anda telah pindah dari kotak pasir dan masuk ke produksi, Anda dapat mengirim email ke penerima mana pun, terlepas dari apakah alamat atau domain penerima diverifikasi. Namun, Anda masih harus memverifikasi semua identitas yang Anda gunakan sebagai alamat “Dari”, “Sumber”, “Pengirim”, atau “Jalur Kembali”.

Selesaikan prosedur di bagian ini untuk meminta agar akun Anda dihapus dari kotak pasir dan dimasukkan ke dalam produksi.

> **Note**
>
> - Jika Anda belum membuat identitas apa pun (alamat email atau domain) diSES, Anda dapat melewati prosedur di halaman ini dan meminta akses produksi untuk akun Anda dengan menggunakan panduan pengaturan SES akun. Lihat Mengatur SES akun Anda untuk petunjuk tentang cara mengakses wizard.
> - Jika Anda menggunakan Amazon SES untuk mengirim email dari EC2 instans Amazon, Anda mungkin juga perlu meminta agar throttle dihapus dari port 25 pada instans Amazon EC2 Anda. Untuk informasi selengkapnya, lihat Bagaimana cara menghapus throttle pada port 25 dari instans saya? EC2 di pusat AWS pengetahuan.

Untuk meminta akses produksi (hapus akun Anda dari kotak pasir) menggunakan AWS Management Console

1. Buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, pilih Dasbor akun.
3. Di kotak peringatan di bagian atas konsol yang bertuliskan, “SESAkun Amazon Anda ada di kotak pasir”, di sisi kanan, pilih Lihat halaman pengaturan diikuti oleh Minta akses produksi.
4. Dalam detail akun modal, pilih salah satu tombol radio Pemasaran atau Transaksi yang paling menggambarkan sebagian besar email yang akan Anda kirim.
   - Email pemasaran - Dikirim one-to-many berdasarkan daftar prospek atau pelanggan yang ditargetkan yang berisi konten pemasaran dan promosi seperti untuk melakukan pembelian, mengunduh informasi, dll.
   - Email transaksional - Dikirim one-to-one secara unik untuk setiap penerima biasanya dipicu oleh tindakan pengguna seperti pembelian situs web, permintaan pengaturan ulang kata sandi, dll.
5. Di Situs Web URL, masukkan URL situs web Anda untuk membantu kami lebih memahami jenis konten yang Anda rencanakan untuk dikirim.
6. Di Kontak tambahan, beri tahu kami di mana Anda ingin menerima komunikasi tentang akun Anda. Hal ini dapat berupa daftar yang dipisahkan koma hingga 4 alamat email.

7. Dalam Bahasa kontak pilihan, pilih apakah Anda ingin menerima komunikasi di English (Bahasa Inggris) atau Japanese (Bahasa Jepang).
8. Dalam Pernyataan, centang kotak yang Anda setujui hanya mengirim email ke individu yang telah memintanya secara eksplisit dan konfirmasikan bahwa Anda memiliki proses untuk menangani notifikasi pentalan dan aduan.
9. Pilih tombol Kirimkan permintaan - banner akan ditampilkan untuk mengonfirmasi permintaan Anda telah dikirimkan dan saat ini sedang ditinjau.

Setelah mengirimkan ulasan detail akun, Anda tidak dapat mengedit detail hingga ulasan selesai. AWS Support Tim memberikan tanggapan awal atas permintaan Anda dalam waktu 24 jam.

Untuk mencegah sistem kami digunakan untuk mengirim konten yang tidak diinginkan atau berbahaya, kami harus mempertimbangkan setiap permintaan dengan hati-hati. Jika kami dapat melakukannya, kami akan memberikan permintaan Anda dalam periode 24 jam ini. Namun, jika kami perlu mendapatkan informasi tambahan dari Anda, mungkin diperlukan waktu lebih lama untuk menyelesaikan permintaan Anda.

Secara opsional, Anda juga dapat mengirimkan permintaan Anda untuk akses produksi menggunakan. AWS CLI Mengirimkan permintaan Anda menggunakan file AWS CLI sangat membantu ketika Anda ingin meminta akses produksi untuk sejumlah besar identitas, atau ketika Anda ingin mengotomatiskan proses pengaturan Amazon. SES

Untuk meminta agar akun Anda dihapus dari SES kotak pasir Amazon menggunakan AWS CLI

1. Prasyarat: Anda harus menginstal dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Command Line Interface.
2. Di baris perintah, masukkan perintah berikut:

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

Pada perintah sebelumnya, lakukan hal berikut:

a. Ganti *TRANSACTIONAL* dengan jenis email yang Anda rencanakan untuk dikirim melalui AmazonSES. Anda dapat menentukan TRANSACTIONAL atau PROMOTIONAL. Jika lebih dari satu nilai berlaku, tentukan pilihan yang berlaku untuk sebagian besar email yang akan dikirim.

b. Ganti *https://example.com* dengan situs URL web Anda. Menyediakan informasi ini membantu kami lebih memahami tipe konten yang Anda rencanakan untuk dikirim.

c. Ganti *info@example.com* dengan alamat email tempat Anda ingin menerima komunikasi tentang akun Anda. Hal ini dapat berupa daftar yang dipisahkan koma hingga 4 alamat email.

d. Ganti *EN* dengan bahasa pilihan Anda. Anda dapat menentukan EN untuk bahasa Inggris atau JA untuk bahasa Jepang.

Setelah mengirimkan ulasan detail akun, Anda tidak dapat mengedit detail hingga ulasan selesai. AWS Support Tim memberikan tanggapan awal atas permintaan Anda dalam waktu 24 jam.

Untuk mencegah sistem kami digunakan untuk mengirim konten yang tidak diinginkan atau berbahaya, kami harus mempertimbangkan setiap permintaan dengan hati-hati. Jika kami dapat melakukannya, kami akan memberikan permintaan Anda dalam periode 24 jam ini. Namun, jika kami perlu mendapatkan informasi tambahan dari Anda, mungkin diperlukan waktu lebih lama untuk menyelesaikan permintaan Anda.

Mengelola batas pengiriman Amazon SES Anda

Akun Amazon SES Anda telah menetapkan kuota pengiriman yang mengatur jumlah pesan email yang dapat Anda kirim dan laju untuk mengirim pesan-pesan tersebut. Kuota pengiriman memberikan manfaat untuk semua pelanggan Amazon SES karena kuota tersebut membantu untuk menjaga hubungan terpercaya antara Amazon SES dan penyedia email. Kuota pengiriman membantu Anda untuk meningkatkan aktivitas pengiriman secara bertahap dan mengurangi kemungkinan penyedia email memblokir email Anda karena lonjakan tiba-tiba dan tidak terduga di volume atau tingkat pengiriman email Anda.

Kuota berikut berlaku untuk mengirim email melalui Amazon SES:

- Kuota pengiriman—Jumlah email maksimum yang dapat Anda kirim dalam periode 24 jam. Kuota ini dihitung pada periode waktu bergulir. Setiap kali Anda mencoba untuk mengirim email, Amazon SES menentukan jumlah email yang Anda kirim dalam 24 jam sebelumnya. Selama jumlah total email yang telah Anda kirim dalam 24 jam terakhir kurang dari maksimum harian ini, permintaan pengiriman Anda diterima dan email Anda dikirim.

  Jika pengiriman pesan melebihi maksimum harian akun Anda, panggilan Anda ke Amazon SES ditolak.
- Tingkat pengiriman—Jumlah maksimum email yang dapat diterima Amazon SES dari akun Anda setiap detik. Anda dapat melebihi kuota ini untuk lonjakan singkat, tetapi tidak untuk periode waktu yang berkelanjutan.

  > **Note**
  >
  > Laju Amazon SES saat menerima pesan Anda dapat kurang dari laju pengiriman maksimum untuk akun Anda.

- Ukuran pesan maksimum (MB) —Ukuran email maksimum yang dapat Anda kirim. Ini termasuk gambar dan lampiran yang merupakan bagian dari email setelah pengkodean MIME. Misalnya, jika Anda melampirkan file 5MB, ukuran lampiran dalam email setelah pengkodean MIME akan menjadi ~ 6.85MB (sekitar 137% dari ukuran file asli).

  > **Note**
  >
  > Kami sarankan Anda mengunggah lampiran Anda ke drive cloud dan menyertakan URL lampiran drive cloud untuk mengurangi ukuran email dan meningkatkan kemampuan

pengiriman. SES tidak dapat menjamin bahwa email besar akan berakhir di kotak surat penerima karena server email yang berbeda akan memiliki kebijakan berdasarkan ukuran yang berbeda-beda.

Kuota pengiriman Amazon SES Anda terpisah untuk setiap Wilayah AWS. Untuk informasi tentang menggunakan Amazon SES di beberapa Wilayah AWS, lihat Wilayah dan Amazon SES.

Ketika akun Anda berada di sandbox Amazon SES, Anda hanya dapat mengirim 200 pesan per periode 24 jam, dan laju pengiriman maksimum Anda adalah satu pesan per detik. Ketika Anda mengirimkan permintaan agar akun Anda dihapus dari sandbox, Anda juga dapat meminta agar kuota Anda ditingkatkan pada saat yang bersamaan. Untuk informasi selengkapnya tentang menghapus akun dari sandbox, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).

Ketika akun Anda telah dihapus dari sandbox, Anda dapat meminta peningkatan kuota tambahan kapan saja dengan membuat kasus baru di Pusat Dukungan AWS. Untuk informasi lebih lanjut, lihat Meningkatkan kuota pengiriman Amazon SES Anda.

> **Note**
>
> Kuota pengiriman didasarkan pada penerima, bukan pada pesan. Misalnya, email yang memiliki 10 penerima dihitung sebagai 10 penerima terhadap kuota Anda. Namun, kami tidak merekomendasikan Anda untuk mengirim email ke beberapa penerima dalam satu panggilan ke operasi API `SendEmail`, karena jika panggilan gagal, seluruh email ditolak. Kami merekomendasikan agar Anda memanggil `SendEmail` sekali untuk setiap penerima.

- Untuk meningkatkan kuota pengiriman Anda, lihat Meningkatkan kuota pengiriman Amazon SES Anda.
- Untuk memantau kuota pengiriman Anda dengan menggunakan konsol Amazon SES atau Amazon SES API, lihat Memantau kuota SES pengiriman Amazon Anda.
- Untuk informasi tentang kesalahan yang diterima aplikasi saat Anda mencapai kuota pengiriman, lihat Kesalahan terkait kuota pengiriman untuk akun Amazon SES Anda.

Meningkatkan kuota pengiriman Amazon SES Anda

Akun Anda memiliki kuota berikut per wilayah Anda saat ini yang dapat ditingkatkan.

Resource	Kuota default	Deskripsi
Kuota pengiriman	200	Jumlah email maksimum yang dapat Anda kirim dalam periode 24 jam untuk akun ini saat iniWilayah AWS.
Jumlah pengiriman	1	Jumlah email maksimum yang dapat diterima Amazon SES setiap detik untuk akun ini saat iniWilayah AWS.

Secara otomatis meningkatkan kuota pengiriman

Ketika akun Anda keluar dari sandbox dan Anda mengirimkan email produksi berkualitas tinggi, kami mungkin secara otomatis meningkatkan kuota pengiriman untuk akun Anda. Seringkali, kami secara otomatis meningkatkan kuota ini sebelum Anda benar-benar ingin meningkatkan kuota tersebut.

Untuk memenuhi syarat untuk peningkatan laju otomatis, semua pernyataan berikut harus BETUL:

- Anda mengirim konten berkualitas tinggi yang ingin diterima penerima –Kirim konten yang diinginkan dan diharapkan oleh penerima. Hentikan pengiriman email ke pelanggan yang tidak membuka email Anda.
- Anda mengirim konten produksi aktual – Mengirimkan pesan percobaan ke alamat email palsu dapat memberikan efek negatif pada tingkat pentalan dan aduan Anda. Selain itu, mengirimkan pesan hanya kepada penerima internal mempersulit keputusan apakah Anda mengirim konten yang ingin diterima pelanggan. Namun, ketika Anda mengirim pesan produksi ke penerima noninternal, kami dapat menilai praktik pengiriman email Anda secara akurat.
- Pengiriman Anda mendekati kuota Anda saat ini – Untuk memenuhi syarat peningkatan kuota otomatis, volume email harian akun Anda harus mendekati batas maksimum harian secara teratur tanpa melebihinya.
- Anda memiliki tingkat pentalan dan aduan yang rendah – Minimalkan jumlah pentalan dan aduan yang Anda terima. Memiliki jumlah pentalan dan aduan yang tinggi dapat berdampak negatif pada kuota pengiriman Anda.

Pengguna diminta untuk meningkatkan kuota pengiriman

Jika kuota pengiriman Anda saat ini tidak memadai untuk kebutuhan Anda dan kami belum meningkatkannya secara otomatis, Anda dapat meminta peningkatan:

- Mengirim kuota atau tarif pengiriman— Meningkatkan permintaan untuk salah satu dari ini dapat disampaikan melaluiAWSService Quotas Konsol.

Untuk meminta peningkatan pada kuota pengiriman Amazon SES dengan menggunakan konsol Service Quotas.

1. Buka Konsol Service Quotas.
2. Pilih wilayah yang ingin Anda tingkatkan menggunakan dropdown pada bagian sudut kanan atas konsol tersebut (di samping nomor akun Anda).
3. Di panel navigasi, pilih services AWS.
4. Pilih Amazon Simple Email Service (SES).
5. Pilih kuota, dan ikuti petunjuk arahan untuk meminta peningkatan kuota.

AWS SupportSLA tim untuk meningkatkan jenis permintaan

Untuk mencegah sistem kami digunakan untuk mengirim konten yang tidak diinginkan atau berbahaya, kami harus mempertimbangkan setiap permintaan dengan hati-hati. Jika kami dapat melakukannya, kami akan mengabulkan permintaan Anda dalam waktu yang ditentukan di bawah ini untuk jenis peningkatan yang diminta. Namun, jika kami perlu mendapatkan informasi tambahan dari Anda, mungkin diperlukan waktu lebih lama untuk menyelesaikan permintaan Anda. Kami berhak untuk tidak mengabulkan permintaan Anda jika kasus penggunaan Anda tidak sesuai dengan kebijakan kami.

- Mengirim kuota atau tarif pengiriman: Hingga 24 jam.

Note

Meskipun konsol Service Quotas tersedia dalam berbagai bahasa, dukungan yang sebenarnya hanya tersedia dalam bahasa Inggris.

Memantau kuota SES pengiriman Amazon Anda

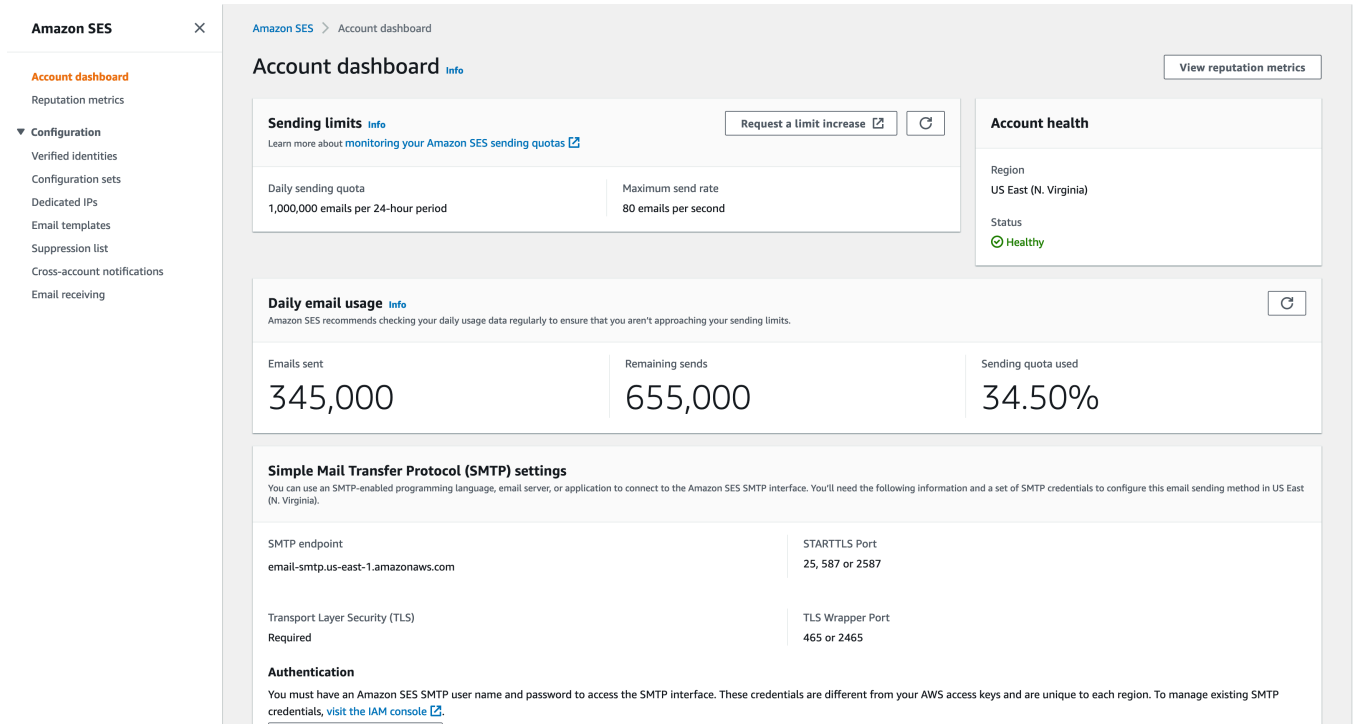
Anda dapat memantau kuota pengiriman Anda dengan menggunakan SES konsol Amazon atau melalui Amazon SESAPI, baik dengan memanggil antarmuka Query (HTTPS) secara langsung atau tidak langsung melalui AWS SDK, file AWS Command Line Interface, atau file. AWS Tools for Windows PowerShell

> **Important**
>
> Kami merekomendasikan Anda untuk sering memeriksa statistik pengiriman untuk memastikan Anda tidak mendekati kuota pengiriman Anda. Jika Anda mendekati kuota pengiriman, lihat Meningkatkan kuota pengiriman Amazon SES Anda untuk informasi tentang cara meningkatkannya. Jangan menunggu sampai Anda mencapai kuota pengiriman Anda untuk mempertimbangkan untuk meningkatkan kuota pengiriman.

Memantau kuota pengiriman Anda menggunakan konsol Amazon SES

Prosedur berikut menunjukkan kepada Anda cara melihat kuota pengiriman Anda menggunakan SES konsol Amazon.

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, pilih Dasbor Akun. Kuota pengiriman Anda ditampilkan di Batas Pengiriman. Total email yang dikirim, sisa pengiriman, dan persentase kuota pengiriman yang digunakan ditampilkan di bawah penggunaan email Harian.



3. Untuk memperbarui tampilan, pilih ikon segarkan di bagian sudut kanan atas kotak Penggunaan email harian.

Memantau kuota pengiriman Anda menggunakan Amazon SES API

Amazon SES API menyediakan `GetSendQuota` tindakan, yang mengembalikan kuota pengiriman Anda. Ketika Anda memanggil tindakan `GetSendQuota`, Anda akan menerima informasi berikut:

- Jumlah email yang telah Anda kirim selama 24 jam terakhir
- Kuota pengiriman untuk periode 24 jam saat ini
- Laju pengiriman maksimum

> **Note**
>
> Untuk penjelasannya`GetSendQuota`, lihat APIReferensi Layanan Email Sederhana Amazon.

Kesalahan terkait kuota pengiriman untuk akun Amazon SES Anda

Jika Anda mencoba untuk mengirim email setelah mencapai kuota pengiriman harian Anda (jumlah maksimum email yang dapat Anda kirim dalam jangka waktu 24 jam) atau laju pengiriman maksimum Anda (jumlah maksimum pesan yang dapat Anda kirim per detik), Amazon SES meninggalkan pesan dan tidak mencoba untuk mengirimnya ulang. Amazon SES juga menyediakan pesan kesalahan yang menjelaskan masalah. Cara Amazon SES menghasilkan pesan kesalahan ini tergantung pada bagaimana Anda mencoba untuk mengirim email. Topik ini mencakup informasi tentang pesan yang Anda terima melalui Amazon SES API dan melalui antarmuka SMTP.

Untuk teknik yang dapat Anda gunakan ketika Anda mencapai laju pengiriman maksimum, lihat Bagaimana menangani kesalahan “Throttling— Melebihi laju pengiriman maksimum” pada Blog Olah Pesan dan Target AWS.

Mencapai batas pengiriman dengan API Amazon SES

Jika Anda mencoba untuk mengirim email dengan menggunakan API Amazon SES (atau SDK AWS), tetapi Anda sudah melampaui batas pengiriman akun Anda, API menghasilkan kesalahan `ThrottlingException`. Pesan kesalahan mencakup salah satu pesan berikut:

- `Daily message quota exceeded`
- `Maximum sending rate exceeded`

Jika Anda mengalami kesalahan throttling, Anda harus memprogram aplikasi Anda untuk menunggu selang waktu hingga 10 menit, lalu mencoba mengirim permintaan lagi.

Mencapai batas pengiriman dengan SMTP

Jika Anda mencoba untuk mengirim email dengan menggunakan antarmuka Amazon SES SMTP, tetapi Anda sudah melampaui batas pengiriman akun Anda, klien SMTP Anda mungkin menampilkan salah satu kesalahan berikut:

- `454 Throttling failure: Maximum sending rate exceeded`
- `454 Throttling failure: Daily message quota exceeded`

Klien SMTP yang berbeda menangani kesalahan ini dengan cara yang berbeda.

Mengatur pengiriman email dengan Amazon SES

Anda dapat mengirim email dengan Amazon Simple Email Service (AmazonSES) menggunakan SES konsol Amazon, antarmuka Amazon SES Simple Mail Transfer Protocol (SMTP), atau Amazon SESAPI. Anda biasanya menggunakan konsol tersebut untuk mengirim dan mengelola aktivitas pengiriman Anda. Untuk mengirim email massal, Anda menggunakan SMTP antarmuka atau fileAPI. Untuk informasi tentang harga SES email Amazon, lihat SESHarga Amazon.

- Jika Anda ingin menggunakan paket perangkat lunak, aplikasi, atau bahasa pemrograman yang SMTP diaktifkan untuk mengirim email melalui AmazonSES, atau mengintegrasikan Amazon SES dengan server email yang ada, gunakan SES SMTP antarmuka Amazon. Untuk informasi selengkapnya, lihat Mengirim email secara terprogram melalui antarmuka Amazon SES SMTP.
- Jika Anda ingin menelepon Amazon SES dengan menggunakan HTTP permintaan mentah, gunakan Amazon SESAPI. Untuk informasi selengkapnya, lihat Menggunakan Amazon SES API untuk mengirim email.

> **Important**
>
> Saat Anda mengirim email ke beberapa penerima (penerima adalah alamat “Kepada”, “CC”, dan “BCC”) dan panggilan ke Amazon SES gagal, seluruh email ditolak dan tidak ada penerima yang akan menerima email yang dimaksud. Oleh karena itu, kami merekomendasikan Anda mengirim email ke satu penerima pada satu waktu.

Menggunakan SES SMTP antarmuka Amazon untuk mengirim email

Untuk mengirim email produksi melalui AmazonSES, Anda dapat menggunakan antarmuka Simple Mail Transfer Protocol (SMTP) atau Amazon SESAPI. Untuk informasi lebih lanjut tentang Amazon SESAPI, lihatMenggunakan Amazon SES API untuk mengirim email. Bagian ini menjelaskan SMTP antarmuka.

Amazon SES mengirim email menggunakanSMTP, yang merupakan protokol email paling umum di internet. Anda dapat mengirim email melalui Amazon SES dengan menggunakan berbagai bahasa pemrograman yang SMTP diaktifkan dan perangkat lunak untuk terhubung ke SES SMTP antarmuka

Amazon. Bagian ini menjelaskan cara mendapatkan SES SMTP kredensyal Amazon Anda, cara mengirim email dengan menggunakan SMTP antarmuka, dan cara mengonfigurasi beberapa perangkat lunak dan server email untuk menggunakan Amazon SES untuk pengiriman email.

Untuk solusi masalah umum yang mungkin Anda temui saat menggunakan Amazon SES melalui SMTP antarmukanya, lihatMasalah SMTP Amazon SES.

Persyaratan untuk mengirim email SMTP

Untuk mengirim email menggunakan SES SMTP antarmuka Amazon, Anda memerlukan yang berikut:

- Alamat SMTP titik akhir. Untuk daftar SES SMTP titik akhir Amazon, lihatMenghubungkan ke titik SES SMTP akhir Amazon.
- Nomor port SMTP antarmuka. Nomor port berbeda-beda menurut metode koneksinya. Untuk informasi selengkapnya, lihat Menghubungkan ke titik SES SMTP akhir Amazon.
- Nama SMTP pengguna dan kata sandi. SMTPkredensialnya unik untuk setiap AWS Wilayah. Jika Anda berencana untuk menggunakan SMTP antarmuka untuk mengirim email di beberapa AWS Wilayah, Anda memerlukan SMTP kredensi untuk setiap Wilayah.

  > **Important**
  >
  > SMTPKredensi Anda tidak identik dengan kunci AWS akses atau kredensil yang Anda gunakan untuk masuk ke konsol Amazon. SES Untuk informasi tentang cara menghasilkan SMTP kredensi Anda, lihat. Memperoleh SES SMTP kredensi Amazon

- Perangkat lunak klien yang dapat berkomunikasi menggunakan Transport Layer Security (TLS). Untuk informasi selengkapnya, lihat Menghubungkan ke titik SES SMTP akhir Amazon.
- Alamat email yang telah Anda verifikasi dengan AmazonSES. Untuk informasi selengkapnya, lihat Identitas terverifikasi di Amazon SES.
- Peningkatan kuota pengiriman, jika Anda ingin mengirim kuantitas email yang besar. Untuk informasi selengkapnya, lihat Mengelola batas pengiriman Amazon SES Anda.

Metode untuk mengirim email SMTP

Anda dapat mengirim email SMTP melalui salah satu metode berikut:

- Untuk mengonfigurasi perangkat lunak yang SMTP diaktifkan untuk mengirim email melalui SES SMTP antarmuka Amazon, lihatMengirim email melalui Amazon SES menggunakan paket perangkat lunak.
- Untuk memprogram aplikasi untuk mengirim email melalui AmazonSES, lihatMengirim email secara terprogram melalui antarmuka Amazon SES SMTP.
- Untuk mengonfigurasi server email yang ada untuk mengirim semua email keluar Anda melalui AmazonSES, lihatMengintegrasikan Amazon SES dengan server email yang ada.
- Untuk berinteraksi dengan SES SMTP antarmuka Amazon menggunakan baris perintah, yang dapat berguna untuk pengujian, lihatMenguji koneksi Anda ke SES SMTP antarmuka Amazon menggunakan baris perintah.

Untuk daftar kode SMTP respons, lihatKode respons SMTP dikembalikan oleh Amazon SES.

Informasi email yang akan disediakan

Saat Anda mengakses Amazon SES melalui SMTP antarmuka, aplikasi SMTP klien Anda menyusun pesan, sehingga informasi yang perlu Anda berikan bergantung pada aplikasi yang Anda gunakan. Minimal, SMTP pertukaran antara klien dan server membutuhkan yang berikut:

- alamat sumber
- alamat tujuan
- data pesan

Jika Anda menggunakan SMTP antarmuka dan mengaktifkan penerusan umpan balik, maka pantulan, keluhan, dan pemberitahuan pengiriman Anda dikirim ke alamat "". MAIL FROM Alamat "Balas Ke" yang Anda tentukan tidak digunakan.

Memperoleh SES SMTP kredensi Amazon

Anda memerlukan SES SMTP kredensi Amazon untuk mengakses antarmuka. SES SMTP

Kredensi yang Anda gunakan untuk mengirim email melalui SES SMTP antarmuka unik untuk setiap AWS Wilayah. Jika Anda menggunakan SES SMTP antarmuka untuk mengirim email di lebih dari satu Wilayah, Anda harus membuat satu set SMTP kredensil untuk setiap Wilayah yang Anda rencanakan untuk digunakan.

SMTPKata sandi Anda berbeda dari kunci akses AWS rahasia Anda. Untuk informasi selengkapnya tentang jenis kredensial, lihat Tipe kredensial Amazon SES.

> **Note**
>
> SMTPTitik akhir saat ini tidak tersedia di Afrika (Cape Town), Asia Pasifik (Jakarta), Eropa (Milan), Israel (Tel Aviv), dan Timur Tengah (Bahrain).

Memperoleh SES SMTP kredensil menggunakan konsol SES

Bila Anda menggunakan SES alur kerja di bawah ini untuk menghasilkan SMTP kredensyal menggunakan konsol, Anda akan dibawa ke IAM konsol untuk membuat pengguna dengan kebijakan yang sesuai untuk memanggil SES dan memberi Anda SMTP kredensyal yang terkait dengan pengguna tersebut.

Persyaratan

IAMPengguna dapat membuat SES SMTP kredensil, tetapi kebijakan pengguna harus memberi mereka izin untuk menggunakannya IAM sendiri, karena SES SMTP kredensil dibuat dengan menggunakan. IAM IAMKebijakan Anda harus memungkinkan Anda untuk melakukan IAM tindakan berikut:`iam:ListUsers`,`iam:CreateUser`,`iam:CreateAccessKey`, dan`iam:PutUserPolicy`. Jika Anda mencoba membuat SES SMTP kredensyal menggunakan konsol dan IAM pengguna Anda tidak memiliki izin ini, Anda akan melihat kesalahan yang menyatakan bahwa akun Anda “tidak diizinkan untuk melakukan iam:.” ListUsers

Untuk membuat SMTP kredensi Anda

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Pilih SMTPpengaturan di panel navigasi kiri - ini akan membuka halaman pengaturan Simple Mail Transfer Protocol (SMTP).
3. Pilih SMTPCreate Credentials di sudut kanan atas - konsol akan terbuka. IAM
4. (Opsional) Jika Anda perlu melihat, mengedit, atau menghapus SMTP pengguna yang telah Anda buat, pilih Kelola SMTP kredenal saya yang ada di sudut kanan bawah - konsol akan terbuka. IAM Rincian untuk mengelola SMTP kredensi diberikan mengikuti prosedur ini.

5. Untuk Buat Pengguna SMTP, ketik nama untuk SMTP pengguna Anda di bidang Nama Pengguna. Atau, Anda dapat menggunakan nilai default yang disediakan di bidang ini. Setelah selesai, pilih Buat pengguna di sudut kanan bawah.
6. Pilih Tampilkan di bawah SMTPkata sandi - SMTP kredensional Anda ditampilkan di layar.
7. Unduh kredensialnya dengan memilih Unduh file.csv atau salin dan simpan di tempat yang aman, karena Anda tidak dapat melihat atau menyimpan kredensialnya setelah menutup kotak dialog ini.
8. Pilih Kembali ke SES konsol.

Anda dapat melihat daftar SMTP kredensyal yang Anda buat menggunakan prosedur ini di IAM konsol di bawah Manajemen akses dan memilih Pengguna diikuti dengan menggunakan bilah pencarian untuk menemukan semua pengguna yang telah ditetapkan SMTP kredensialnya.

Anda juga dapat menggunakan IAM konsol untuk menghapus SMTP pengguna yang ada. Untuk mempelajari lebih lanjut tentang menghapus pengguna, lihat Mengelola IAM Pengguna di Panduan IAM Memulai.

Jika Anda ingin mengubah SMTP kata sandi, hapus SMTP pengguna yang ada di IAM konsol. Kemudian, untuk menghasilkan satu set SMTP kredensil baru, selesaikan prosedur sebelumnya.

Memperoleh SES SMTP kredensi dengan mengonversi kredensi yang ada AWS

Jika Anda memiliki pengguna yang Anda atur menggunakan IAM antarmuka, Anda dapat memperoleh kredensional pengguna dari SES SMTP kredensialnya. AWS

> **Important**
>
> Jangan gunakan kredensil sementara untuk mendapatkan AWS SMTP kredensil. SESSMTPAntarmuka tidak mendukung SMTP kredensil yang telah dihasilkan dari kredensil keamanan sementara.

Untuk memungkinkan IAM pengguna mengirim email menggunakan SES SMTP antarmuka, lakukan hal berikut.

- Dapatkan kredensi pengguna dari SMTP kredensialnya dengan menggunakan algoritme AWS yang disediakan di bagian ini. Karena Anda memulai dari AWS kredensialnya, nama SMTP pengguna sama dengan ID kunci AWS akses, jadi Anda hanya perlu membuat kata sandi. SMTP

- Terapkan kebijakan berikut kepada IAM pengguna:

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

Untuk informasi selengkapnya tentang menggunakan SES withIAM, lihatIdentity and access management di Amazon SES.

> **Note**
>
> Meskipun Anda dapat menghasilkan SES SMTP kredensil untuk setiap IAM pengguna, kami sarankan Anda membuat IAM pengguna terpisah ketika Anda membuat kredensil Anda. SMTP Untuk informasi tentang mengapa itu adalah praktik yang baik untuk membuat pengguna untuk tujuan tertentu, buka Praktik IAM Terbaik.

Pseudocode berikut menunjukkan algoritma yang mengubah kunci akses AWS rahasia ke kata sandi. SES SMTP

```
// 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);
```

Beberapa bahasa pemrograman termasuk perpustakaan yang dapat Anda gunakan untuk mengonversi kunci akses IAM rahasia menjadi SMTP kata sandi. Bagian ini mencakup contoh kode yang dapat Anda gunakan untuk mengonversi kunci akses AWS rahasia ke SES SMTP kata sandi menggunakan Python.

> **Note**
>
> Contoh berikut menggunakan f-string yang diperkenalkan di Python 3.6; jika menggunakan versi lama, maka tidak akan berfungsi.
> Saat ini, Python SDK (Boto3) secara resmi mendukung 2.7 dan 3.6 (atau lebih baru). Namun, dukungan 2.7 tidak lagi digunakan dan akan dihentikan pada 15/7/2021, sehingga Anda harus meningkatkan ke setidaknya versi 3.6.

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)
    "eu-south-1",  # Europe (Milan)
    "eu-north-1",  # Europe (Stockholm)
    "sa-east-1",  # South America (Sao Paulo)
    "us-gov-west-1",  # AWS GovCloud (US)
    "us-gov-east-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 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()
```

Untuk mendapatkan SMTP kata sandi Anda dengan menggunakan skrip ini, simpan kode sebelumnya sebagai. `smtp_credentials_generate.py` Kemudian, di baris perintah, jalankan perintah berikut:

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

Di perintah sebelumnya, lakukan hal berikut:

- Ganti *`path/to/`* dengan jalur ke lokasi tempat Anda menyimpan`smtp_credentials_generate.py`.
- Ganti *`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`* dengan kunci akses rahasia yang ingin Anda ubah menjadi SMTP kata sandi.
- Ganti *`us-east-1`* dengan AWS Wilayah di mana Anda ingin menggunakan SMTP kredensialnya.

Ketika skrip ini berjalan dengan sukses, satu-satunya output adalah SMTP kata sandi Anda.

Menghubungkan ke titik SES SMTP akhir Amazon

Untuk mengirim email menggunakan SES SMTP antarmuka Amazon, Anda terhubung ke SMTP titik akhir. Untuk daftar lengkap SES SMTP titik akhir Amazon, lihat titik akhir Amazon Simple Email Service dan kuota di. Referensi Umum AWS

SESSMTPEndpoint Amazon mengharuskan semua koneksi dienkripsi menggunakan Transport Layer Security (). TLS (Perhatikan bahwa sering TLS disebut dengan nama protokol pendahulunya,SSL.) Amazon SES mendukung dua mekanisme untuk membuat koneksi TLS -terenkripsi: STARTTLS dan Wrapper. TLS Periksa dokumentasi untuk perangkat lunak Anda untuk menentukan apakah itu mendukungSTARTTLS, TLS Wrapper, atau keduanya.

Amazon Elastic Compute Cloud (AmazonEC2) membatasi lalu lintas email melalui port 25 secara default. Untuk menghindari batas waktu saat mengirim email melalui SMTP titik akhir dariEC2, kirimkan Permintaan untuk Menghapus Batasan Pengiriman Email untuk menghapus throttle. Atau, Anda dapat mengirim email menggunakan port yang berbeda, atau menggunakan VPCendpoint Amazon.

Untuk masalah SMTP koneksi, lihatMasalah SMTP.

STARTTLS

STARTTLSadalah sarana untuk meningkatkan koneksi yang tidak terenkripsi ke koneksi terenkripsi. Ada versi STARTTLS untuk berbagai protokol; SMTP versi didefinisikan dalam RFC 3207.

Untuk mengatur STARTTLS koneksi, SMTP klien terhubung ke SES SMTP titik akhir Amazon pada port 25, 587, atau 2587, mengeluarkan EHLO perintah, dan menunggu server mengumumkan bahwa itu mendukung ekstensi. STARTTLS SMTP Klien kemudian mengeluarkan STARTTLS perintah, memulai TLS negosiasi. Ketika negosiasi selesai, klien mengeluarkan EHLO perintah atas koneksi terenkripsi baru, dan SMTP sesi berjalan normal.

TLSPembungkus

TLSWrapper (juga dikenal sebagai SMTPS atau Handshake Protocol) adalah sarana untuk memulai koneksi terenkripsi tanpa terlebih dahulu membuat koneksi yang tidak terenkripsi. Dengan TLS Wrapper, SES SMTP titik akhir Amazon tidak melakukan TLS negosiasi: klien bertanggung jawab untuk terhubung ke titik akhir menggunakanTLS, dan terus menggunakan TLS untuk seluruh percakapan. TLSWrapper adalah protokol yang lebih lama, tetapi banyak klien masih mendukungnya.

Untuk mengatur koneksi TLS Wrapper, SMTP klien terhubung ke SES SMTP titik akhir Amazon pada port 465 atau 2465. Server menyajikan sertifikatnya, klien mengeluarkan EHLO perintah, dan SMTP sesi berjalan normal.

Mengirim email melalui Amazon SES menggunakan paket perangkat lunak

Ada sejumlah paket perangkat lunak komersial dan open source yang mendukung pengiriman email melaluiSMTP. Berikut ini adalah beberapa contohnya:

- Platform blog
- RSSagregator

- Daftar perangkat lunak manajemen
- Sistem alur kerja

Anda dapat mengonfigurasi perangkat lunak SMTP berkemampuan seperti itu untuk mengirim email melalui SES SMTP antarmuka Amazon. Untuk petunjuk tentang cara mengkonfigurasi SMTP paket perangkat lunak tertentu, lihat dokumentasi untuk perangkat lunak tersebut.

Prosedur berikut menunjukkan cara mengatur SES pengiriman AmazonJIRA, solusi pelacakan masalah yang populer. Dengan konfigurasi ini, JIRA dapat memberitahu pengguna melalui email setiap kali ada perubahan dalam status masalah perangkat lunak.

Untuk mengonfigurasi JIRA untuk mengirim email menggunakan Amazon SES

1. Menggunakan browser web Anda, masuk JIRA dengan kredensi administrator.
2. Di jendela peramban, pilih Administrasi.
3. Di menu Sistem, pilih Surat.
4. Di halaman Administrasi surat, pilih Server Surat.
5. Pilih Konfigurasikan server SMTP email baru.
6. Pada formulir Add SMTP Mail Server, isi kolom berikut:

    a. Nama—Nama deskriptif untuk server ini.

    b. Alamat Dari—Alamat asal email akan dikirim. Anda harus memverifikasi alamat email ini dengan Amazon SES sebelum Anda dapat mengirim darinya. Untuk informasi selengkapnya tentang verifikasi, lihat Identitas terverifikasi di Amazon SES.

    c. Awalan email —String yang ditambahkan ke setiap baris subjek JIRA sebelum mengirim.

    d. Protokol —Pilih SMTP.

    > **Note**
    >
    > Jika Anda tidak dapat terhubung ke Amazon SES menggunakan pengaturan ini, coba SECURE_ SMTP.

    e. Nama host —Lihat Menghubungkan ke titik SES SMTP akhir Amazon daftar titik akhir Amazon SESSMTP. Misalnya, jika Anda ingin menggunakan SES titik akhir Amazon di Wilayah AS Barat (Oregon), nama host akan menjadi email-smtp.us-west-2.amazonaws.com.

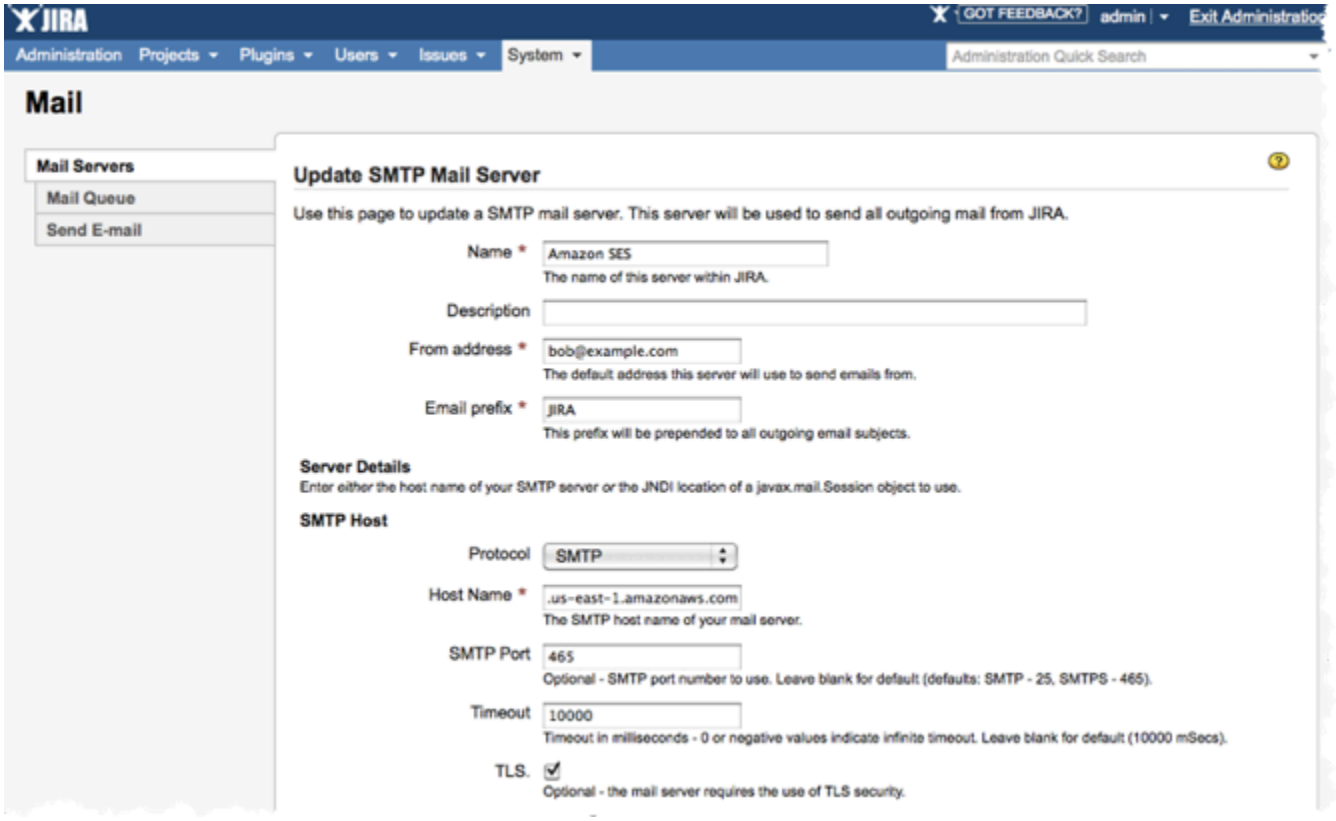
f. SMTPport —25, 587, atau 2587 (untuk terhubung menggunakanSTARTTLS), atau 465 atau 2465 (untuk terhubung menggunakan Wrapper). TLS

g. TLS—Pilih kotak centang ini.

h. Nama pengguna —Nama SMTP pengguna Anda.

i. Kata sandi —Kata SMTP sandi Anda.

Anda dapat melihat pengaturan untuk TLS Wrapper pada gambar berikut.



7. Pilih Uji Koneksi. Jika email pengujian yang JIRA dikirim melalui Amazon berhasil SES tiba, maka konfigurasi Anda selesai.

Mengirim email secara terprogram melalui antarmuka Amazon SES SMTP

Untuk mengirim email menggunakan SES SMTP antarmuka Amazon, Anda dapat menggunakan bahasa pemrograman, server email, atau aplikasi yang SMTP diaktifkan. Sebelum memulai, selesaikan tugas-tugas di Menyiapkan Amazon Simple Email Service. Anda juga harus mendapatkan informasi berikut:

- SESSMTPKredensi Amazon Anda, yang memungkinkan Anda terhubung ke titik akhir Amazon SESSMTP. Untuk mendapatkan SES SMTP kredensi Amazon Anda, lihat. Memperoleh SES SMTP kredensi Amazon

> **Important**
>
> SMTPKredensialnya berbeda dari kredensialmu AWS . Untuk informasi selengkapnya tentang jenis kredensial, lihat Tipe kredensial Amazon SES.

- Alamat SMTP titik akhir. Untuk daftar SES SMTP titik akhir Amazon, lihatMenghubungkan ke titik SES SMTP akhir Amazon.
- Nomor port SES SMTP antarmuka Amazon, yang tergantung pada metode koneksi. Untuk informasi selengkapnya, lihat Menghubungkan ke titik SES SMTP akhir Amazon.

Contoh kode

Anda dapat mengakses SES SMTP antarmuka Amazon dengan menggunakan bahasa pemrograman yang SMTP diaktifkan. Anda memberikan SES SMTP nama host Amazon dan nomor port beserta SMTP kredensialnya dan kemudian menggunakan SMTP fungsi generik bahasa pemrograman untuk mengirim email.

Amazon Elastic Compute Cloud (AmazonEC2) membatasi lalu lintas email melalui port 25 secara default. Untuk menghindari batas waktu saat mengirim email melalui SMTP titik akhir dari AmazonEC2, Anda dapat meminta agar pembatasan ini dihapus. Untuk informasi selengkapnya, lihat Bagaimana cara menghapus pembatasan pada port 25 dari EC2 instans atau AWS Lambda fungsi Amazon saya? di pusat AWS pengetahuan.

Contoh kode di bagian ini untuk Java dan PHP gunakan port 587 untuk menghindari masalah ini.

> **Note**
>
> Dalam tutorial ini, Anda mengirim email ke diri Anda sendiri sehingga Anda dapat memeriksa apakah Anda sudah menerimanya. Untuk eksperimen lebih lanjut atau pengujian beban, gunakan simulator SES kotak surat Amazon. Email yang Anda kirim ke simulator kotak surat tidak dihitung terhadap kuota pengiriman atau kecepatan pentalan dan aduan Anda. Untuk informasi lebih lanjut, lihat Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual.

Pilih bahasa pemrograman untuk melihat contoh bahasa tersebut:

> **Warning**
>
> Amazon SES tidak merekomendasikan penggunaan kredensi statis. Lihat AWS Secrets Manageruntuk mempelajari cara meningkatkan postur keamanan Anda dengan menghapus kredensi hard-code dari kode sumber Anda. Tutorial ini hanya disediakan untuk tujuan menguji SES SMTP antarmuka Amazon di lingkungan non-produksi.

Java

Contoh ini menggunakan Eclipse IDE dan JavaMail APIuntuk mengirim email melalui Amazon SES menggunakan antarmuka. SMTP

Sebelum Anda melakukan prosedur berikut, selesaikan tugas di Menyiapkan Amazon Simple Email Service.

Untuk mengirim email menggunakan SES SMTP antarmuka Amazon dengan Java

1. Di browser web, buka JavaMail GitHub halaman. Di bawah Assets, pilih javax.mail.jar untuk mengunduh versi terbaru. JavaMail

   > **Important**
   >
   > Tutorial ini membutuhkan JavaMail versi 1.5 atau yang lebih baru. Prosedur ini diuji menggunakan JavaMail versi 1.6.1.

2. Di web browser, buka GitHub halaman Aktivasi Jakarta, dan di bawah JavaBeans Activation Framework 1.2.1 Final Release, unduh jakarta.activation.jar
3. Buat proyek di Eclipse dengan melakukan langkah-langkah berikut:

   a. Mulai Eclipse.

   b. Di Eclipse, pilih File, pilih Baru, lalu pilih Proyek Java.

   c. Di kotak dialog Buat Proyek Java, ketik nama proyek lalu pilih Selanjutnya.

   d. Di kotak dialog Pengaturan Java, pilih tab Pustaka.

   e. Pilih Classpath dan tambahkan dua file jar eksternal javax.mail.jar dan jakarta.activation.jar menggunakan tombol Add External. JARs

f. Pilih Tambahkan Eksternal JARs.

g. Jelajahi folder tempat Anda mengunduh JavaMail. Pilih file `javax.mail.jar`, lalu pilih Buka.

h. Di kotak dialog Pengaturan Java, pilih Selesai.

4. Di Eclipse, di jendela Paket Explorer, perluas proyek Anda.
5. Di bawah proyek Anda, klik kanan direktori src, pilih Baru, lalu pilih Kelas.
6. Di kotak dialog Kelas Java Baru, di bidang Nama, ketik `AmazonSESSample` lalu pilih Selesai.
7. Ganti seluruh isi A mazonSESSample .java dengan kode berikut:

```
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";

    // 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();
```

```
        // Get the password
        String SMTP_PASSWORD = fetchSMTPPasswordFromSecureStorage();

        // 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();
        }
    }

    static String fetchSMTPPasswordFromSecureStorage() {
        /* IMPLEMENT THIS METHOD */
        // For example, you might fetch it from a secure location or AWS Secrets
 Manager: https://aws.amazon.com/secrets-manager/
    }
}
```

8. Di A mazonSESSample .java, ganti alamat email berikut dengan nilai Anda sendiri:

> **Important**
>
> Alamat email peka huruf besar kecil. Pastikan alamatnya sama persis dengan alamat yang Anda verifikasi.

- *sender@example.com* — Ganti dengan alamat email “Dari” Anda. Alamat ini harus diverifikasi sebelum Anda menjalankan program ini. Untuk informasi selengkapnya, lihat Identitas terverifikasi di Amazon SES.
- *recipient@example.com* — Ganti dengan alamat email “Ke” Anda. Jika akun Anda masih berada di sandbox, Anda harus memverifikasi alamat ini sebelum menggunakannya. Untuk informasi selengkapnya, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).

9. Dalam A mazonSESSample .java ganti yang berikut ini dengan nilai Anda sendiri:

   - *smtp_username* — Ganti dengan kredensi nama SMTP pengguna Anda. Perhatikan bahwa kredensi nama SMTP pengguna Anda adalah rangkaian huruf dan angka 20 karakter, bukan nama yang dapat dipahami.
   - *smtp_password* — Melaksanakan `fetchSMTPPasswordFromSecureStorage` untuk mengambil kata sandi.

10. (Opsional) Jika Anda ingin menggunakan SES SMTP titik akhir Amazon di Wilayah AWS selain *email-smtp.us-west-2.amazonaws.com*, ubah nilai variabel HOST ke titik akhir yang ingin Anda gunakan. Untuk daftar wilayah tempat Amazon SES tersedia, lihat Layanan Email Sederhana Amazon (AmazonSES) di Referensi Umum AWS.
11. (Opsional) Jika Anda ingin menggunakan set konfigurasi saat mengirim email ini, ubah nilai variabel *ConfigSet* dengan nama set konfigurasi. Untuk informasi selengkapnya tentang set konfigurasi, lihat Menggunakan set konfigurasi di Amazon SES.
12. Simpan A mazonSESSample .java.
13. Untuk membangun proyek, pilih Proyek lalu pilih Bangun Proyek. (Jika opsi ini dinonaktifkan, maka Anda mungkin memiliki pembangunan otomatis yang diaktifkan.)
14. Untuk memulai program dan mengirim email, pilih Jalankan lalu pilih Jalankan lagi.
15. Tinjau output. Jika email berhasil dikirim, konsol menampilkan “Email dikirim!” Jika tidak, ini akan menampilkan pesan kesalahan.
16. Masuk ke klien email dari alamat penerima. Anda akan menemukan pesan yang Anda kirim.

PHP

Contoh ini menggunakan PHPMailer kelas untuk mengirim email melalui Amazon SES menggunakan SMTP antarmuka.

Sebelum Anda melakukan prosedur berikut, selesaikan tugas di Menyiapkan Amazon Simple Email Service. Selain menyiapkan Amazon, SES Anda harus menyelesaikan prasyarat berikut untuk mengirim email dengan: PHP

Prasyarat:

- Instal PHP - PHP tersedia di http://php.net/downloads.php. Setelah Anda menginstalPHP, tambahkan path ke PHP dalam variabel lingkungan Anda sehingga Anda dapat menjalankan PHP dari prompt perintah apa pun.
- Instal manajer ketergantungan Composer - Setelah Anda menginstal manajer ketergantungan Composer, Anda dapat mengunduh dan menginstal PHPMailer kelas dan dependensinya. Untuk menginstal Composer, ikuti petunjuk instalasi di https://getcomposer.org/download.
- Instal PHPMailer kelas - Setelah Anda menginstal Composer, jalankan perintah berikut untuk menginstalPHPMailer:

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

  Pada perintah sebelumnya, ganti *path/to/* dengan jalur tempat Anda menginstal Komposer.

Untuk mengirim email menggunakan SES SMTP antarmuka Amazon dengan PHP

1. Buat file bernama amazon-ses-smtp-sample.php. Buka file dengan editor teks dan tempel di kode berikut:

   ```
   <?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.
$usernameSmtp = 'smtp_username';

// 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   = fetchSMTPPasswordFromSecureStorage();
    $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.
}
function fetchSMTPPasswordFromSecureStorage() {
/* IMPLEMENT THIS METHOD */
// For example, you might fetch it from a secure location or AWS Secrets
 Manager: https://aws.amazon.com/secrets-manager/
}

?>
```

2. Di amazon-ses-smtp-sample.php, ganti yang berikut ini dengan nilai Anda sendiri:

   - *sender@example.com* — Ganti dengan alamat email yang telah Anda verifikasi dengan AmazonSES. Untuk informasi selengkapnya, lihat Identitas terverifikasi. Alamat email di Amazon SES peka huruf besar/kecil. Pastikan alamat yang Anda masukkan sama persis dengan alamat yang Anda verifikasi.
   - *recipient@example.com* — Ganti dengan alamat penerima. Jika akun Anda masih berada di sandbox, Anda harus memverifikasi alamat ini sebelum menggunakannya. Untuk informasi lebih lanjut, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon). Pastikan alamat yang Anda masukkan sama persis dengan alamat yang Anda verifikasi.
   - *smtp_username* — Ganti dengan kredensi nama SMTP pengguna Anda, yang Anda peroleh dari halaman SMTPPengaturan SES konsol Amazon. Kredensial ini tidak sama dengan access key ID AWS Anda. Perhatikan bahwa kredensi nama SMTP pengguna Anda adalah rangkaian huruf dan angka 20 karakter, bukan nama yang dapat dipahami.

- *smtp_password* — Melaksanakan `fetchSMTPPasswordFromSecureStorage` untuk mengambil kata sandi.
- (Opsional) *ConfigSet* — Jika Anda ingin menggunakan set konfigurasi saat mengirim email ini, ganti nilai ini dengan nama set konfigurasi. Untuk informasi selengkapnya tentang set konfigurasi, lihat Menggunakan set konfigurasi di Amazon SES.
- (Opsional) *email-smtp.us-west-2.amazonaws.com* — Jika Anda ingin menggunakan SES SMTP titik akhir Amazon di Wilayah selain AS Barat (Oregon), ganti ini dengan SES SMTP titik akhir Amazon di Wilayah yang ingin Anda gunakan. Untuk daftar SMTP titik akhir URLs Wilayah AWS tempat Amazon SES tersedia, lihat Amazon Simple Email Service (AmazonSES) di Referensi Umum AWS.

3. amazon-ses-smtp-sampleSimpan.php.
4. Untuk menjalankan program, buka prompt perintah di direktori yang sama amazon-ses-smtp-sampledengan.php, lalu ketikphp amazon-ses-smtp-sample.php.
5. Tinjau output. Jika email berhasil dikirim, konsol menampilkan “Email dikirim!” Jika tidak, ini akan menampilkan pesan kesalahan.
6. Masuk ke klien email alamat penerima. Anda akan menemukan pesan yang Anda kirim.

Mengintegrasikan Amazon SES dengan server email yang ada

Jika saat ini Anda mengelola server email Anda sendiri, Anda dapat menggunakan SES SMTP endpoint Amazon untuk mengirim semua email keluar Anda ke Amazon. SES Tidak perlu memodifikasi klien dan aplikasi email Anda yang ada; pergantian ke Amazon SES akan transparan bagi mereka.

Beberapa agen transfer surat (MTAs) mendukung pengiriman email melalui SMTP relay. Bagian ini memberikan panduan umum tentang cara mengkonfigurasi beberapa populer MTAs untuk mengirim email menggunakan SES SMTP antarmuka Amazon.

SESSMTPEndpoint Amazon mengharuskan semua koneksi dienkripsi menggunakan Transport Layer Security (). TLS

Topik

- Mengintegrasikan Amazon SES dengan Microsoft Windows Server IIS SMTP

Mengintegrasikan Amazon SES dengan Microsoft Windows Server IIS SMTP

Anda dapat mengonfigurasi server Microsoft Windows IIS SMTP Server untuk mengirim email melalui AmazonSES. Instruksi ini ditulis menggunakan Microsoft Windows Server 2012 pada EC2 instance Amazon. Anda dapat menggunakan konfigurasi yang sama di Microsoft Windows Server 2008 dan Microsoft Windows Server 2008 R2.

> **Note**
>
> Windows Server adalah aplikasi pihak ke tiga, dan tidak dikembangkan atau didukung oleh Amazon Web Services. Prosedur di bagian ini disediakan hanya untuk tujuan informasi saja, dan dapat berubah tanpa pemberitahuan.

Untuk mengintegrasikan server Microsoft Windows IIS SMTP Server dengan Amazon SES

1. Pertama, siapkan Microsoft Windows Server 2012 menggunakan petunjuk berikut.
   a. Dari konsol EC2 manajemen Amazon, luncurkan EC2 instans Amazon Microsoft Windows Server 2012 Base baru.
   b. Hubungkan ke instans dan masuk ke instans menggunakan Remote Desktop dengan mengikuti petunjuk di Memulai Instans Amazon EC2 Windows.
   c. Luncurkan Dasbor Pengelola Server.
   d. Instal peran Server Web. Pastikan untuk menyertakan IIS6 alat Kompatibilitas Manajemen (opsi di bawah kotak centang Server Web).
   e. Instal fitur SMTPServer.
2. Selanjutnya, konfigurasikan IIS SMTP layanan menggunakan instruksi berikut.
   a. Mengembalikan ke Dasbor Pengelola Server.
   b. Dari menu Tools, pilih Internet Information Services (IIS) 6.0 Manager.
   c. Klik kanan SMTPVirtual Server #1 dan kemudian pilih Properties.
   d. Di tab Akses, di bawah Pembatasan Relay, pilih Relay.
   e. Di kotak dialog Pembatasan Relay, pilih Tambahkan.
   f. Di bawah Komputer Tunggal, masukkan 127.0.0.1 sebagai alamat IP. Anda sekarang telah memberikan akses ke server ini untuk menyampaikan email ke Amazon SES melalui IIS SMTP layanan.

Di prosedur ini, kita beranggapan bahwa email Anda dihasilkan di server ini. Jika aplikasi yang menghasilkan email berjalan di server terpisah, Anda harus memberikan akses relay untuk server tersebut. IIS SMTP

> **Note**
>
> Untuk memperluas SMTP relai ke subnet pribadi, untuk Pembatasan Relay, gunakan Komputer Tunggal 127.0.0.1 dan Grup Komputer 172.1.1.0 - 255.255.255.0 (di bagian netmask). Untuk Koneksi, gunakan Komputer Tunggal 127.0.0.1 dan Grup Komputer 172.1.1.0 - 255.255.255.0 (di bagian netmask).

3. Terakhir, konfigurasikan server untuk mengirim email melalui Amazon SES menggunakan instruksi berikut.

   a. Kembali ke kotak dialog SMTPVirtual Server #1 Properties dan kemudian pilih tab Pengiriman.

   b. Di tab Penyampaian, pilih Keamanan Outbound.

   c. Pilih Autentikasi Dasar, lalu masukkan SES SMTP kredensi Amazon Anda. Anda dapat memperoleh kredensil ini dari SES konsol Amazon menggunakan prosedur di. Memperoleh SES SMTP kredensi Amazon

   > **Important**
   >
   > SMTPKredensi Anda tidak sama dengan ID kunci AWS akses dan kunci akses rahasia Anda. Jangan mencoba menggunakan AWS kredensil Anda untuk mengautentikasi diri Anda terhadap titik akhir. SMTP Untuk informasi selengkapnya tentang jenis kredensial, lihat Tipe kredensial Amazon SES.

   d. Pastikan TLSenkripsi dipilih.

   e. Kembali ke tab Penyampaian.

   f. Pilih Koneksi Outbound.

   g. Di kotak dialog Koneksi Outbound, pastikan bahwa port adalah 25 atau 587.

   h. Pilih Lanjutan.

   i. Untuk nama host Smart, masukkan SES endpoint Amazon yang akan Anda gunakan (misalnya, email-smtp.us-west-2.amazonaws.com). Untuk daftar titik akhir URLs Wilayah

AWS tempat Amazon SES tersedia, lihat Amazon Simple Email Service (AmazonSES) di Referensi Umum AWS.

j. Mengembalikan ke Dasbor Pengelola Server.

k. Pada Dasbor Server Manager, klik kanan SMTPVirtual Server #1 dan kemudian restart layanan untuk mengambil konfigurasi baru.

l. Kirim email melalui server ini. Anda dapat memeriksa header pesan untuk mengonfirmasi bahwa itu dikirim melalui AmazonSES.

Menguji koneksi Anda ke SES SMTP antarmuka Amazon menggunakan baris perintah

Anda dapat menggunakan metode yang dijelaskan di bagian ini dari baris perintah untuk menguji koneksi Anda ke SES SMTP titik akhir Amazon, memvalidasi SMTP kredensi Anda, dan memecahkan masalah koneksi. Prosedur ini menggunakan alat dan pustaka yang termasuk dengan sistem operasi yang paling umum.

Untuk informasi tambahan tentang pemecahan masalah SMTP koneksi, lihat. Masalah SMTP Amazon SES

Prasyarat

Saat Anda terhubung ke SES SMTP antarmuka Amazon, Anda harus memberikan satu set SMTP kredensil. SMTPKredensi ini berbeda dari kredensi standar AWS Anda. Dua tipe kredensial tidak dapat dipertukarkan. Untuk informasi selengkapnya tentang mendapatkan SMTP kredensi Anda, lihat. the section called “Memperoleh SMTP kredensi”

Menguji koneksi Anda ke SES SMTP antarmuka Amazon

Anda dapat menggunakan baris perintah untuk menguji koneksi Anda ke SES SMTP antarmuka Amazon tanpa mengautentikasi atau mengirim pesan apa pun. Prosedur ini berguna untuk memecahkan masalah konektivitas dasar. Jika koneksi pengujian Anda gagal, lihatMasalah SMTP.

Bagian ini mencakup prosedur untuk menguji koneksi Anda menggunakan Open SSL (yang disertakan dengan sebagian besar distribusi Linux, macOS, dan Unix, dan juga tersedia untuk Windows) dan `Test-NetConnection` cmdlet di PowerShell (yang disertakan dengan versi Windows terbaru).

Linux, macOS, or Unix

Ada dua cara untuk terhubung ke SES SMTP antarmuka Amazon dengan OpenSSL: menggunakan eksplisit SSL melalui port 587, atau menggunakan implisit SSL melalui port 465.

Untuk terhubung ke SMTP antarmuka menggunakan eksplisit SSL

- Pada baris perintah, masukkan perintah berikut untuk terhubung ke SES SMTP server Amazon:

```
openssl s_client -crlf -quiet -starttls smtp -connect email-smtp.us-
west-2.amazonaws.com:587
```

Pada perintah sebelumnya, ganti *email-smtp.us-west-2.amazonaws.com* dengan SES SMTP titik URL akhir Amazon untuk AWS Wilayah Anda. Untuk informasi selengkapnya, lihat the section called “Wilayah”.

Jika koneksi berhasil, Anda akan menemukan output yang serupa dengan berikut ini:

```
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
```

Koneksi secara otomatis ditutup setelah tidak aktif selama sekitar 10 detik.

Atau, Anda dapat menggunakan implisit SSL untuk terhubung ke SMTP antarmuka melalui port 465.

Untuk terhubung ke SMTP antarmuka menggunakan implisit SSL

- Pada baris perintah, masukkan perintah berikut untuk terhubung ke SES SMTP server Amazon:

```
openssl s_client -crlf -quiet -connect email-smtp.us-west-2.amazonaws.com:465
```

Pada perintah sebelumnya, ganti *email-smtp.us-west-2.amazonaws.com* dengan SES SMTP titik URL akhir Amazon untuk AWS Wilayah Anda. Untuk informasi selengkapnya, lihat the section called “Wilayah”.

Jika koneksi berhasil, Anda akan menemukan output yang serupa dengan berikut ini:

```
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
```

Koneksi secara otomatis ditutup setelah tidak aktif selama sekitar 10 detik.

PowerShell

Anda dapat menggunakan Test- NetConnection cmdlet in PowerShell untuk terhubung ke server Amazon SESSMTP.

> **Note**
>
> `Test-NetConnection`Cmdlet dapat menentukan apakah komputer Anda dapat terhubung ke titik akhir Amazon SESSMTP. Namun, itu tidak menguji apakah komputer Anda dapat membuat SSL koneksi implisit atau eksplisit ke titik akhirSMTP. Untuk menguji SSL koneksi, Anda dapat menginstal Open SSL for Windows untuk mengirim email pengujian.

Untuk terhubung ke SMTP antarmuka menggunakan **`Test-NetConnection`** cmdlet

- Masuk PowerShell, masukkan perintah berikut untuk terhubung ke SES SMTP server Amazon:

```
Test-NetConnection -Port 587 -ComputerName email-smtp.us-west-2.amazonaws.com
```

Pada perintah sebelumnya, ganti *email-smtp.us-west-2.amazonaws.com* dengan SES SMTP titik URL akhir Amazon untuk AWS Wilayah Anda, dan ganti *587* dengan nomor port. Untuk informasi selengkapnya tentang titik akhir regional di AmazonSES, lihatthe section called “Wilayah”.

Jika koneksi berhasil, Anda akan menemukan output yang serupa dengan contoh berikut:

```
ComputerName     : email-smtp.us-west-2.amazonaws.com
RemoteAddress    : 198.51.100.126
RemotePort       : 587
InterfaceAlias   : Ethernet
SourceAddress    : 203.0.113.46
TcpTestSucceeded : True
```

Menggunakan baris perintah untuk mengirim email menggunakan SES SMTP antarmuka Amazon

Anda juga dapat menggunakan baris perintah untuk mengirim pesan menggunakan SES SMTP antarmuka Amazon. Prosedur ini berguna untuk menguji SMTP kredensil dan untuk menguji kemampuan penerima tertentu untuk menerima pesan yang Anda kirim dengan menggunakan Amazon. SES

Linux, macOS, or Unix

Ketika pengirim email terhubung ke SMTP server, klien mengeluarkan serangkaian permintaan standar, dan server membalas setiap permintaan dengan respons standar. Serangkaian permintaan dan tanggapan ini disebut SMTPpercakapan. Saat Anda terhubung ke SES SMTP server Amazon menggunakan OpenSSL, server mengharapkan SMTP percakapan terjadi.

Saat Anda menggunakan Open SSL untuk terhubung ke SMTP antarmuka, Anda harus menyandikan SMTP kredensi Anda menggunakan pengkodean base64. Bagian ini mencakup prosedur untuk pengodean kredensial Anda menggunakan base64.

Untuk mengirim email dari baris perintah menggunakan SMTP antarmuka

1. Masukkan yang berikut ini di baris perintah dan ganti *email-smtp.us-west-2.amazonaws.com* dengan SES SMTP titik URL akhir Amazon untuk Anda Wilayah AWS. Untuk informasi lebih lanjut, lihatthe section called “Wilayah”. :

```
#!/bin/bash

# Prompt user to provide following information
read -p "Configuration set: " CONFIGSET
read -p "Enter SMTP username: " SMTPUsername
read -p "Enter SMTP password: " SMTPPassword
read -p "Sender email address: " MAILFROM
read -p "Receiver email address: " RCPT
read -p "Email subject: " SUBJECT
read -p "Message to send: " DATA

echo

# Encode SMTP username and password using base64
EncodedSMTPUsername=$(echo -n "$SMTPUsername" | openssl enc -base64)
EncodedSMTPPassword=$(echo -n "$SMTPPassword" | openssl enc -base64)

# Construct the email
Email="EHLO example.com
AUTH LOGIN
$EncodedSMTPUsername
$EncodedSMTPPassword
MAIL FROM: $MAILFROM
RCPT TO: $RCPT
DATA
X-SES-CONFIGURATION-SET: $CONFIGSET
From: $MAILFROM
To: $RCPT
Subject: $SUBJECT

$DATA
.
QUIT"

echo "$Email" | openssl s_client -crlf -quiet -starttls smtp -connect email-
smtp.us-west-2.amazonaws.com:587
```

2. Pada prompt untuk setiap variabel, masukkan nilai Anda.
3. • Untuk mengirim menggunakan implisit SSL melalui port 465, gunakan:

```
openssl s_client -crlf -quiet -connect email-smtp.us-west-2.amazonaws.com:465
```

Jika pesan diterima oleh AmazonSES, Anda melihat output yang menyerupai contoh berikut:

```
250 Ok 01010160d7de98d8-21e57d9a-JZho-416c-bbe1-8ebaAexample-000000
```

String angka dan teks yang mengikuti `250 Ok` adalah ID pesan email.

> **Note**
>
> Koneksi ditutup secara otomatis setelah tidak aktif selama sekitar 10 detik.

PowerShell

Anda dapat menggunakan Net.Mail. SmtpClientkelas untuk mengirim email menggunakan eksplisit SSL melalui port 587.

> **Note**
>
> Kelas `Net.Mail.SmtpClient` telah secara resmi usang, dan Microsoft merekomendasikan Anda menggunakan pustaka pihak ke tiga. Kode ini hanya ditujukan untuk tujuan pengujian saja, dan tidak boleh digunakan untuk beban kerja produksi.

Untuk mengirim email melalui PowerShell penggunaan eksplisit SSL

1. Di editor teks, buat file baru. Tempel kode berikut ke file:

```
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)."

    $Sender = "sender@example.com"
    $Recipient = "recipient@example.com"

    SendEmail $Server $Port $Sender $Recipient $Subject $Body
}

SendTestEmail
```

Setelah selesai, simpan file sebagai `SendEmail.ps1`.

2. Buat perubahan berikut ke file yang Anda buat di langkah sebelumnya:

   - Ganti *sender@example.com* dengan alamat email tempat Anda ingin mengirim pesan dari.
   - Ganti *recipient@example.com* dengan alamat email yang ingin Anda kirimi pesan.
   - Ganti *email-smtp.us-west-2.amazonaws.com* dengan SES SMTP titik URL akhir Amazon untuk AWS Wilayah Anda. Untuk informasi selengkapnya, lihat Wilayah dan Amazon SES.

3. Di PowerShell, masukkan perintah berikut:

```
.\path\to\SendEmail.ps1
```

   Pada perintah sebelumnya, ganti *path\to\SendEmail.ps1* dengan jalur ke file yang Anda buat di langkah 1.

4. Saat diminta, masukkan nama SMTP pengguna dan kata sandi Anda.

Atau, Anda dapat menggunakan System.Web.Mail. SmtpMailkelas untuk mengirim email menggunakan implisit SSL melalui port 465.

> **Note**
>
> Kelas `System.Web.Mail.SmtpMail` telah secara resmi usang, dan Microsoft merekomendasikan Anda menggunakan pustaka pihak ke tiga. Kode ini hanya ditujukan untuk tujuan pengujian saja, dan tidak boleh digunakan untuk beban kerja produksi.

Untuk mengirim email melalui PowerShell penggunaan implisit SSL

1. Di editor teks, buat file baru. Tempel kode berikut ke file:

```
[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
```

Setelah selesai, simpan file sebagai `SendEmail.ps1`.

2. Buat perubahan berikut ke file yang Anda buat di langkah sebelumnya:

   - Ganti *sender@example.com* dengan alamat email tempat Anda ingin mengirim pesan dari.
   - Ganti *recipient@example.com* dengan alamat email yang ingin Anda kirimi pesan.
   - Ganti *email-smtp.us-west-2.amazonaws.com* dengan SES SMTP titik URL akhir Amazon untuk AWS Wilayah Anda. Untuk informasi selengkapnya, lihat Wilayah dan Amazon SES.

3. Di PowerShell, masukkan perintah berikut:

```
.\path\to\SendEmail.ps1
```

Pada perintah sebelumnya, ganti *path\to\SendEmail.ps1* dengan jalur ke file yang Anda buat di langkah 1.

4. Saat diminta, masukkan nama SMTP pengguna dan kata sandi Anda.

Menggunakan Amazon SES API untuk mengirim email

Untuk mengirim email produksi melalui AmazonSES, Anda dapat menggunakan antarmuka Simple Mail Transfer Protocol (SMTP) atau Amazon SESAPI. Untuk informasi selengkapnya tentang SMTP antarmuka, lihatMenggunakan SES SMTP antarmuka Amazon untuk mengirim email. Bagian ini menjelaskan cara mengirim email dengan menggunakanAPI.

Saat Anda mengirim email menggunakan Amazon SESAPI, Anda menentukan konten pesan, dan Amazon SES mengumpulkan MIME email untuk Anda. Atau, Anda dapat menyusun email sendiri sehingga Anda memiliki kendali penuh atas isi pesan. Untuk informasi selengkapnya tentangAPI, lihat APIReferensi Layanan Email Sederhana Amazon. Untuk daftar titik akhir URLs Wilayah AWS tempat Amazon SES tersedia, lihat titik akhir dan kuota Layanan Email Sederhana Amazon di. Referensi Umum AWS

Anda dapat menelepon dengan API cara berikut:

- Membuat HTTPS permintaan langsung— Ini adalah metode yang paling canggih, karena Anda harus secara manual menangani otentikasi dan penandatanganan permintaan Anda, dan kemudian secara manual membuat permintaan. Untuk informasi tentang Amazon SESAPI, lihat halaman Selamat Datang di Referensi API v2.
- Gunakan AWS SDK —AWS SDKs membuatnya mudah untuk mengakses APIs untuk beberapa AWS layanan, termasuk AmazonSES. Ketika Anda menggunakanSDK, itu menangani otentikasi, penandatanganan permintaan, logika coba lagi, penanganan kesalahan, dan fungsi tingkat rendah lainnya sehingga Anda dapat fokus pada membangun aplikasi yang menyenangkan pelanggan Anda.
- Gunakan antarmuka baris perintah- AWS Command Line InterfaceIni adalah alat baris perintah untuk AmazonSES. Kami juga menawarkan AWS Alat PowerShell untuk mereka yang membuat skrip di PowerShell lingkungan.

Terlepas dari apakah Anda mengakses Amazon SES API secara langsung atau tidak langsung melalui AWS SDK, the AWS Command Line Interface atau AWS Tools for PowerShell, Amazon SES

API menyediakan dua cara berbeda bagi Anda untuk mengirim email, tergantung pada seberapa besar kontrol yang Anda inginkan atas komposisi pesan email:

- Diformat- Amazon SES menyusun dan mengirim pesan email yang diformat dengan benar. Anda hanya perlu menyediakan alamat "Dari:" dan "Kepada:", subjek, dan badan pesan. Amazon SES mengurus sisanya. Untuk informasi selengkapnya, lihat Mengirim email yang diformat menggunakan Amazon SES API.
- Raw- Anda secara manual menulis dan mengirim pesan email, menentukan header email Anda sendiri dan jenis. MIME Jika Anda berpengalaman dalam memformat email Anda sendiri, antarmuka mentah memberi Anda kendali lebih besar atas komposisi pesan Anda. Untuk informasi lebih lanjut, lihat Mengirim email mentah menggunakan Amazon SES API v2.

Daftar Isi

- Mengirim email yang diformat menggunakan Amazon SES API
- Mengirim email mentah menggunakan Amazon SES API v2
- Menggunakan template untuk mengirim email yang dipersonalisasi dengan Amazon SES API
- Mengirim email melalui Amazon SES menggunakan AWS SDK
- Pengkodean konten yang didukung oleh Amazon SES

Mengirim email yang diformat menggunakan Amazon SES API

Anda dapat mengirim email yang diformat dengan menggunakan AWS Management Console atau dengan menelepon Amazon SES API melalui aplikasi secara langsung, atau tidak langsung melalui AWS SDK, file AWS Command Line Interface, atau. AWS Tools for Windows PowerShell

Amazon SES API menyediakan `SendEmail` tindakan, yang memungkinkan Anda menulis dan mengirim email yang diformat. `SendEmail`memerlukan alamat Dari:, Ke: alamat, subjek pesan, dan badan pesan—teks,HTML, atau keduanya. Untuk informasi selengkapnya, lihat SendEmail(APIReferensi) atau SendEmail(Referensi API v2).

> **Note**
>
> String alamat email harus 7-bitASCII. Jika Anda ingin mengirim ke atau dari alamat email yang berisi karakter Unicode di bagian domain alamat, Anda harus mengodekan domain menggunakan Punycode. Untuk informasi lebih lanjut, lihat RFC3492.

Untuk contoh cara menulis pesan yang diformat menggunakan berbagai bahasa pemrograman, lihat. Contoh kode

Untuk tips tentang cara meningkatkan kecepatan pengiriman email ketika Anda melakukan beberapa panggilan ke `SendEmail`, lihat Meningkatkan throughput dengan Amazon SES.

Mengirim email mentah menggunakan Amazon SES API v2

Anda dapat menggunakan `SendEmail` operasi Amazon SES API v2 dengan jenis konten yang ditentukan `raw` untuk mengirim pesan yang disesuaikan ke penerima menggunakan format email mentah.

Tentang bidang header email

Simple Mail Transfer Protocol (SMTP) menentukan bagaimana pesan email akan dikirim dengan mendefinisikan amplop surat dan beberapa parameternya, tetapi tidak berkaitan dengan isi pesan. Sebaliknya, Internet Message Format (RFC5322) mendefinisikan bagaimana pesan akan dibangun.

Dengan spesifikasi Format Pesan Internet, setiap pesan email terdiri dari header dan badan. Header terdiri dari metadata pesan, dan badan berisi pesan itu sendiri. Untuk informasi selengkapnya tentang header dan badan email, lihat Format email dan Amazon SES.

Menggunakan MIME

SMTPProtokol ini awalnya dirancang untuk mengirim pesan email yang hanya berisi ASCII karakter 7-bit. Spesifikasi ini membuat SMTP tidak cukup untuk pengkodean ASCII non-teks (seperti Unicode), konten biner, atau lampiran. Standar Multipurpose Internet Mail Extensions (MIME) dikembangkan untuk memungkinkan pengiriman banyak jenis konten lainnya menggunakanSMTP.

MIMEStandar bekerja dengan memecah badan pesan menjadi beberapa bagian dan kemudian menentukan apa yang harus dilakukan dengan setiap bagian. Misalnya, satu bagian dari badan pesan email mungkin berupa teks biasa, sementara yang lain mungkinHTML. Selain itu, MIME memungkinkan pesan email berisi satu atau lebih lampiran. Penerima pesan dapat melihat lampiran dari dalam klien email mereka, atau mereka dapat menyimpan lampiran.

Header dan konten pesan dipisahkan oleh baris kosong. Setiap bagian dari email dipisahkan oleh batas, string karakter yang menunjukkan awal dan akhir dari setiap bagian.

Pesan multipart dalam contoh berikut berisi teks dan HTML bagian, dan lampiran. Lampiran harus ditempatkan tepat di bawah header lampiran dan paling sering dikodekan `base64` seperti yang ditunjukkan dalam contoh ini.

```
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

SUQsRmlyc3ROYW1lLExhc3ROYW1lLENvdW50cnkKMzQ4LEpvaG4sU3RpbGVzLENhbmFkYQo5MjM4
OSxKaWUsTGl1LENoaW5hCjczNCxTaGlybGV5LFJvZHJpZ3VleixVbml0ZWQgU3RhdGVzCjI4OTMs
QW5heWEsSXllbmdhcixJbmRpYQ==

--a3f166a86b56ff6c37755292d690675717ea3cd9de81228ec2b76ed4a15d6d1a--
```

Tipe konten untuk pesan adalah `multipart/mixed`, menunjukkan bahwa pesan memiliki banyak bagian (di contoh ini, badan dan lampiran), dan klien penerima harus menangani setiap bagian secara terpisah.

Nest yang dilakukan dalam bagian badan adalah bagian kedua yang menggunakan tipe konten `multipart/alternative`. Jenis konten ini menunjukkan bahwa setiap bagian berisi versi alternatif dari konten yang sama (dalam hal ini, versi teks dan HTML versi). Jika klien email penerima dapat menampilkan HTML konten, maka itu menunjukkan HTML versi badan pesan. Jika klien email penerima tidak dapat menampilkan HTML konten, maka itu akan menampilkan versi teks biasa dari badan pesan.

Kedua versi pesan juga berisi lampiran (di kasus ini, file teks pendek yang berisi beberapa nama pelanggan).

Ketika Anda menyarangkan MIME bagian dalam bagian lain, seperti dalam contoh ini, bagian bersarang harus menggunakan `boundary` parameter yang berbeda dari `boundary` parameter di bagian induk. Batas-batas ini harus berupa string unik karakter. Untuk menentukan batas antar MIME bagian, ketik dua tanda hubung (--) diikuti oleh string batas. Di akhir MIME bagian, tempatkan dua tanda hubung di awal dan akhir string batas.

> **Note**
>
> Sebuah pesan tidak dapat memiliki lebih dari 500 MIME bagian.

MIMEPengkodean

Untuk menjaga kompatibilitas dengan sistem yang lebih lama, Amazon SES menghormati ASCII batasan 7-bit SMTP seperti yang didefinisikan dalam RFC 2821. Jika Anda ingin mengirim konten yang berisi ASCII non-karakter, Anda harus menyandikan karakter tersebut ke dalam format yang menggunakan karakter 7-bitASCII.

Alamat email

String alamat email harus 7-bitASCII. Jika Anda ingin mengirim ke atau dari alamat email yang berisi karakter Unicode di bagian domain alamat, Anda harus mengodekan domain menggunakan Punycode. Punycode tidak diizinkan di bagian lokal dari alamat email (bagian sebelum tanda @) atau dalam nama "friendly from". Jika Anda ingin menggunakan karakter Unicode dalam nama “friendly from”, Anda harus menyandikan nama “friendly from” menggunakan sintaks MIME encoded-word,

seperti yang dijelaskan dalam. Mengirim email mentah menggunakan Amazon SES API v2 Untuk informasi lebih lanjut tentang Punycode, lihat RFC3492.

> **Note**
>
> Aturan ini hanya berlaku untuk alamat email yang Anda tentukan di envelope pesan, bukan header pesan. Saat Anda menggunakan `SendEmail` operasi Amazon SES API v2, alamat yang Anda tentukan dalam `Destinations` parameter `Source` dan menentukan pengirim dan penerima amplop, masing-masing.

Header email

Untuk menyandikan header pesan, gunakan sintaks kata yang MIME dikodekan. MIMEsintaks kata yang dikodekan menggunakan format berikut:

```
=?charset?encoding?encoded-text?=
```

Nilai *`encoding`* dapat berupa `Q` atau `B`. Jika nilai pengodean adalah `Q`, maka nilai *`encoded-text`* harus menggunakan Q-encoding. Jika nilai pengodean adalah `B`, maka nilai *`encoded-text`* harus menggunakan pengodean base64.

Misalnya, jika Anda ingin menggunakan string "Як ти поживаєш?" di baris subjek email, Anda dapat menggunakan salah satu pengodean berikut:

- Pengkodean Q

```
=?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?=
```

- Pengkodean Base64

```
=?utf-8?B?0K/QuiDRgtC4INC/0L7QttC40LLQsNGU0Yg/?=
```

Untuk informasi lebih lanjut tentang Q-encoding, lihat 2047. RFC Untuk informasi lebih lanjut tentang pengkodean base64, lihat RFC 2045.

Badan pesan

Untuk mengodekan badan pesan, Anda dapat menggunakan pengodean quoted-printable atau pengodean base64. Kemudian, gunakan header `Content-Transfer-Encoding` untuk menunjukkan skema pengodean yang Anda gunakan.

Sebagai contoh, asumsikan badan pesan Anda berisi teks berikut:

१९७२ मे रे टॉमलंसिन ने पहला ई-मेल सेंदश भेजा | रे टॉमलंसिन ने ही स्रव्परथम @ चन्हि का चयन कयिा और इनही को ईमल का आवषिकारक माना जाता है

Jika Anda memilih untuk mengodekan teks ini menggunakan pengodean base64, pertama tentukan header berikut:

```
Content-Transfer-Encoding: base64
```

Kemudian, di bagian badan email, sertakan teks yang dikodekan base64:

```
4KWn4KWv4KWt4KWoIOCkruClhyDgpLDgpYcg4KSf4KWJ4KSu4KSy4KS/4KSC4KS44KSoIOCkqOCl
hyDgpKrgpLngpLLgpL4g4KSILeCkruClh+CksiDgpLjgpILgpKbgpYfgpLYg4KSt4KWH4KSc4KS+
IHwg4KSw4KWHIOCkn+ClieCkruCksuCkv+CkguCkuOCkqCDgpKjgpYcg4KS54KWAIOCkuOCksOCl
jeCkteCkquCljeCksOCkpeCkriBAIOCkmuCkv+CkqOCljeCkuSDgpJXgpL4g4KSa4KSv4KSoIOCk
leCkv+Ckr+CkviDgpJTgpLAg4KSH4KSo4KWN4KS54KWAIOCkleCliyDgpIjgpK7gpYfgpLIg4KSV
4KS+IOCkhuCkteCkv+Ckt+CljeCkleCkvuCksOCklSDgpK7gpL7gpKjgpL4g4KSc4KS+4KSk4KS+
IOCkueCliAo=
```

> **Note**
>
> Dalam beberapa kasus, Anda dapat menggunakan 8bit `Content-Transfer-Encoding` dalam pesan yang Anda kirim menggunakan AmazonSES. Namun, jika Amazon SES harus membuat perubahan apa pun pada pesan Anda (misalnya, saat Anda menggunakan pelacakan terbuka dan klik), konten yang disandikan 8-bit mungkin tidak muncul dengan benar saat masuk ke kotak masuk penerima. Untuk alasan ini, Anda harus selalu menyandikan konten yang tidak ASCII 7-bit.

Lampiran file

Untuk melampirkan file ke email, Anda harus mengodekan lampiran menggunakan pengodean base64. Lampiran biasanya ditempatkan di bagian MIME pesan khusus, yang mencakup header berikut:

- Content-Type - Jenis file lampiran. Berikut ini adalah contoh deklarasi MIME Content-Type umum:
  - File teks biasa - `Content-Type: text/plain; name="sample.txt"`
  - Dokumen Microsoft Word — `Content-Type: application/msword; name="document.docx"`
  - JPGgambar — `Content-Type: image/jpeg; name="photo.jpeg"`
- Content-Disposition - Menentukan bagaimana klien email penerima harus menangani konten. Untuk lampiran, nilai ini adalah `Content-Disposition: attachment`.
- Content-Transfer-Encoding — Skema yang digunakan untuk menyandikan lampiran. Untuk lampiran file, nilai ini hampir selalu `base64`.
- Lampiran yang dikodekan — Anda harus menyandikan lampiran yang sebenarnya dan memasukkannya ke dalam badan di bawah header lampiran seperti yang ditunjukkan pada contoh.

Amazon SES menerima jenis file yang paling umum. Untuk daftar jenis file yang SES tidak diterima Amazon, lihatJenis lampiran Amazon SES yang tidak didukung.

Mengirim email mentah menggunakan Amazon SES API v2

Amazon SES API v2 menyediakan `SendEmail` tindakan, yang memungkinkan Anda menulis dan mengirim pesan email dalam format yang Anda tentukan saat Anda menyetel jenis konten menjadi sederhana, mentah, atau templat. Untuk deskripsi selengkapnya, lihat `SendEmail`. Contoh berikut akan menentukan jenis konten `raw` untuk mengirim pesan menggunakan format email mentah.

> **Note**
>
> Untuk tips tentang cara meningkatkan kecepatan pengiriman email ketika Anda melakukan beberapa panggilan ke `SendEmail`, lihat Meningkatkan throughput dengan Amazon SES.

Badan pesan harus berisi pesan yang diformat dengan benar dan email mentah, dengan bidang header dan pengodean badan pesan yang sesuai. Meskipun pesan mentah mungkin dapat

disusun secara manual dalam aplikasi, tapi akan jauh lebih mudah untuk melakukannya dengan menggunakan pustaka surat yang ada.

Java

Contoh kode berikut menunjukkan bagaimana menggunakan JavaMailperpustakaan dan AWS SDK for Javauntuk menulis dan mengirim email mentah.

```
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();
        }
    }
}
```

Python

Contoh kode berikut ini menunjukkan cara menggunakan paket Phyton email.mim dan AWS SDK for Python (Boto) untuk menulis dan mengirim email mentah.

```
import json
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
import os

def boto3_rawemailv2():
    SENDER = "Sender <sender@example.com>"
    RECIPIENT = "recipient@example.com"
    CONFIGURATION_SET = "ConfigSet"
    AWS_REGION = "us-east-1"
    SUBJECT = "Customer service contact info"
    ATTACHMENT = "path/to/customers-to-contact.xlsx"
    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/>
    <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"
    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)
    msg.attach(att)

    #changes start from here
    strmsg = str(msg)
    body = bytes (strmsg, 'utf-8')




    client = boto3.client('sesv2')
    response = client.send_email(
    FromEmailAddress=SENDER,
    Destination={
        'ToAddresses': [RECIPIENT]
    },
    Content={
        'Raw': {
            'Data': body
        }
    }
    )
    print(response)
boto3_rawemailv2 ()
```

Menggunakan template untuk mengirim email yang dipersonalisasi dengan Amazon SES API

Anda dapat menggunakan CreateTemplateAPIoperasi untuk membuat template email. Template ini mencakup baris subjek, dan teks dan HTML bagian dari badan email. Bagian subjek dan badan juga dapat berisi nilai unik yang dipersonalisasi untuk setiap penerima.

Ada beberapa batasan dan pertimbangan lainnya saat menggunakan fitur ini:

- Anda dapat membuat hingga 20.000 template email di masing-masing Wilayah AWS.
- Setiap template dapat berukuran hingga 500 KB, termasuk teks dan HTML bagian-bagiannya.
- Anda dapat menyertakan jumlah variabel pengganti yang tidak terbatas di setiap templat.
- Anda dapat mengirim email ke hingga 50 tujuan di setiap panggilan ke operasi `SendBulkTemplatedEmail`. Tujuan mencakup daftar penerima, termasuk CC dan BCC penerima. Jumlah tujuan yang dapat Anda hubungi dalam satu panggilan API mungkin dibatasi oleh tarif pengiriman maksimum akun Anda. Untuk informasi selengkapnya, lihat Mengelola batas pengiriman Amazon SES Anda.

Bagian ini mencakup prosedur untuk membuat templat email dan untuk mengirim email yang dipersonalisasi.

> **Note**
>
> Prosedur di bagian ini menganggap bahwa Anda telah menginstal dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Bagian 1: Siapkan notifikasi peristiwa kegagalan rendering

Jika Anda mengirim email yang berisi konten personalisasi yang tidak valid, Amazon SES mungkin menerima pesan tersebut, tetapi tidak dapat mengirimkannya. Untuk alasan ini, jika Anda berencana untuk mengirim email yang dipersonalisasi, Anda harus mengonfigurasi Amazon SES untuk mengirim pemberitahuan peristiwa Kegagalan Rendering melalui AmazonSNS. Ketika Anda menerima notifikasi peristiwa Kegagalan Rendering, Anda dapat mengidentifikasi pesan yang berisi konten yang tidak valid, memperbaiki masalah, dan mengirim pesan kembali.

Prosedur di bagian ini bersifat opsional, namun sangat disarankan.

Untuk mengonfigurasi notifikasi peristiwa Kegagalan Rendering

1. Buat SNS topik Amazon. Untuk prosedur, lihat Buat Topik di Panduan Developer Amazon Simple Notification Service.
2. Berlangganan ke SNS topik Amazon. Misalnya, jika Anda ingin menerima notifikasi Kegagalan Rendering melalui email, berlangganan titik akhir email (yaitu, alamat email Anda) ke topik tersebut.

   Untuk prosedur, lihat Berlangganan Topik di Panduan Developer Amazon Simple Notification Service.
3. Selesaikan prosedur the section called “Siapkan tujuan Amazon SNS” untuk menyiapkan set konfigurasi Anda untuk mempublikasikan peristiwa Kegagalan Rendering ke SNS topik Amazon Anda.

Bagian 2: Buat templat email

Di bagian ini, Anda menggunakan CreateTemplate API operasi untuk membuat template email baru dengan atribut personalisasi.

Prosedur ini menganggap Anda telah menginstal dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Untuk membuat templat

1. Di editor teks, buat file baru. Tempel kode berikut ke file.

```
{
  "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}}."
  }
}
```

   Kode ini berisi properti berikut:

- TemplateName— Nama template. Ketika Anda mengirim email, Anda merujuk ke nama ini.
- SubjectPart— Baris subjek email. Properti ini mungkin berisi tanda pengganti. Tanda ini menggunakan format berikut: `{{tagname}}`. Ketika Anda mengirim email, Anda dapat menentukan nilai untuk `tagname` untuk setiap tujuan.

  Contoh sebelumnya mencakup dua tanda: `{{name}}` dan `{{favoriteanimal}}`.
- HtmlPart- HTML Tubuh email. Properti ini mungkin berisi tanda pengganti.
- TextPart— Tubuh teks email. Penerima yang klien emailnya tidak menampilkan HTML email melihat versi email ini. Properti ini mungkin berisi tanda pengganti.

2. Sesuaikan contoh sebelumnya sesuai dengan kebutuhan Anda, lalu simpan file sebagai `mytemplate.json`.
3. Pada baris perintah, ketik perintah berikut untuk membuat template baru menggunakan `CreateTemplate` API operasi:

```
aws ses create-template --cli-input-json file://mytemplate.json
```

Bagian 3: Kirim email yang dipersonalisasi

Setelah membuat templat email, Anda dapat menggunakannya untuk mengirim email. Ada dua API operasi yang dapat Anda gunakan untuk mengirim email menggunakan template:`SendTemplatedEmail`, dan`SendBulkTemplatedEmail`. `SendTemplatedEmail`Operasi ini berguna untuk mengirim email yang disesuaikan ke satu tujuan (kumpulan “Kepada,” “CC,” dan "BCC" penerima yang akan menerima email yang sama). `SendBulkTemplatedEmail`Operasi ini berguna untuk mengirim email unik ke beberapa tujuan dalam satu panggilan ke Amazon SESAPI. Bagian ini memberikan contoh bagaimana menggunakan AWS CLI untuk mengirim email menggunakan kedua operasi ini.

Mengirim email yang ditemplat ke satu tujuan

Anda dapat menggunakan operasi `SendTemplatedEmail` untuk mengirim email ke satu tujuan. Semua penerima di objek `Destination` akan menerima email yang sama.

Untuk mengirim email yang ditemplat ke satu tujuan

1. Di editor teks, buat file baru. Tempel kode berikut ke file.

```
{
  "Source":"Mary Major <mary.major@example.com>",
  "Template": "MyTemplate",
  "ConfigurationSetName": "ConfigSet",
  "Destination": {
    "ToAddresses": [ "alejandro.rosalez@example.com"
    ]
  },
  "TemplateData": "{ \"name\":\"Alejandro\", \"favoriteanimal\": \"alligator\" }"
}
```

Kode ini berisi properti berikut:

- Sumber – Alamat email pengirim.
- Templat – Nama templat yang akan diterapkan ke email.
- ConfigurationSetName— Nama konfigurasi yang akan digunakan saat mengirim email.

> **Note**
>
> Sebaiknya gunakan set konfigurasi yang dikonfigurasi untuk mempublikasikan peristiwa Kegagalan Rendering ke AmazonSNS. Untuk informasi selengkapnya, lihat the section called “Bagian 1: Siapkan notifikasi”.

- Tujuan – Alamat penerima. Anda dapat menyertakan beberapa alamat “Kepada,” “CC,” dan BCC "". Saat Anda menggunakan operasi `SendTemplatedEmail`, semua penerima menerima email yang sama.
- TemplateData— JSON String yang lolos yang berisi pasangan kunci-nilai. Kunci sesuai dengan variabel di templat (misalnya, `{{name}}`). Nilai-nilai menunjukkan konten yang menggantikan variabel di email.

2. Ubah nilai dalam kode di langkah sebelumnya untuk memenuhi kebutuhan Anda, lalu simpan file sebagai `myemail.json`.
3. Di baris perintah, ketik perintah berikut untuk mengirim email:

```
aws ses send-templated-email --cli-input-json file://myemail.json
```

Mengirim email yang ditemplat ke beberapa tujuan

Anda dapat menggunakan `SendBulkTemplatedEmail` operasi untuk mengirim email ke beberapa tujuan dalam satu panggilan keAPI. Amazon SES mengirimkan email unik ke penerima atau penerima di setiap `Destination` objek.

Untuk mengirim email yang ditemplat ke beberapa tujuan

1. Di editor teks, buat file baru. Tempel kode berikut ke file.

```
{
  "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\" }"
}
```

Kode ini berisi properti berikut:

- Sumber – Alamat email pengirim.
- Templat – Nama templat yang akan diterapkan ke email.
- ConfigurationSetName— Nama konfigurasi yang akan digunakan saat mengirim email.

> **Note**
>
> Sebaiknya gunakan set konfigurasi yang dikonfigurasi untuk mempublikasikan peristiwa Kegagalan Rendering ke AmazonSNS. Untuk informasi selengkapnya, lihat the section called “Bagian 1: Siapkan notifikasi”.

- Tujuan – Array yang berisi satu Tujuan atau lebih.
  - Tujuan – Alamat penerima. Anda dapat menyertakan beberapa alamat “Kepada,” “CC,” dan BCC "". Saat Anda menggunakan operasi `SendBulkTemplatedEmail`, semua penerima di dalam objek `Destination` yang sama menerima email yang sama.
  - ReplacementTemplateData— JSON Objek yang berisi pasangan kunci-nilai. Kunci sesuai dengan variabel di templat (misalnya, `{{name}}`). Nilai-nilai menunjukkan konten yang menggantikan variabel di email.
- DefaultTemplateData— JSON Objek yang berisi pasangan kunci-nilai. Kunci sesuai dengan variabel di templat (misalnya, `{{name}}`). Nilai-nilai menunjukkan konten yang menggantikan variabel di email. Objek ini berisi data fallback. Jika sebuah `Destination` objek berisi JSON objek kosong di `ReplacementTemplateData` properti, nilai-nilai dalam `DefaultTemplateData` properti yang digunakan.

2. Ubah nilai dalam kode di langkah sebelumnya untuk memenuhi kebutuhan Anda, lalu simpan file sebagai `mybulkemail.json`.
3. Di baris perintah, ketik perintah berikut untuk mengirim banyak email:

```
aws ses send-bulk-templated-email --cli-input-json file://mybulkemail.json
```

Personalisasi email lanjutan

Fitur template di Amazon SES didasarkan pada sistem template Handlebars. Anda dapat menggunakan Handlebar untuk membuat templat yang mencakup fitur lanjutan, seperti atribut yang di-nest, pengulangan array, pernyataan bersyarat dasar, dan pembuatan parsial inline. Bagian ini menyediakan contoh fitur ini.

Handlebar mencakup fitur tambahan di luar yang didokumentasikan di bagian ini. Untuk informasi lebih lanjut, lihat Bantuan Bawaan di handlebarsjs.com.

> **Note**
>
> SEStidak luput dari HTML konten saat merender HTML template untuk pesan. Ini berarti jika Anda menyertakan data yang dimasukkan pengguna, seperti dari formulir kontak, Anda harus menghindarinya di sisi klien.

Topik

- Menguraikan atribut yang di-nest
- Pengulangan melalui daftar
- Menggunakan pernyataan persyaratan dasar
- Membuat parsial inline

Menguraikan atribut yang di-nest

Handlebar mencakup dukungan untuk jalur yang di-nest sehingga mempermudah pengaturan data pelanggan yang rumit, dan kemudian data tersebut dapat dirujuk di templat email Anda.

Misalnya, Anda dapat mengatur data penerima ke dalam beberapa kategori umum. Dalam setiap kategori tersebut, Anda dapat menyertakan informasi detail. Contoh kode berikut menunjukkan contoh struktur untuk satu penerima:

```
{
  "meta":{
```

```
    "userId":"51806220607"
  },
  "contact":{
    "firstName":"Anaya",
    "lastName":"Iyengar",
    "city":"Bengaluru",
    "country":"India",
    "postalCode":"560052"
  },
  "subscription":[
    {
      "interest":"Sports"
    },
    {
      "interest":"Travel"
    },
    {
      "interest":"Cooking"
    }
  ]
}
```

Di templat email, Anda dapat merujuk ke atribut yang di-nest dengan memberikan nama atribut induk, diikuti oleh titik (.), diikuti dengan nama atribut yang Anda ingin sertakan nilainya. Misalnya, jika Anda menggunakan struktur data yang ditampilkan di contoh sebelumnya, dan Anda ingin menyertakan nama depan setiap penerima di templat email, masukkan teks berikut di templat email Anda: `Hello {{contact.firstName}}!`

Handlebar dapat mengurai jalur yang di-nest beberapa tingkat, yang berarti Anda memiliki fleksibilitas dalam penyusunan data templat.

Pengulangan melalui daftar

Fungsi bantuan `each` mengulang melalui item di array. Kode berikut adalah contoh templat email yang menggunakan fungsi bantuan `each` untuk membuat daftar terperinci dari setiap kepentingan penerima.

```
{
  "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/prefererences/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/prefererences/i.aspx?id={{meta.userId}}"
  }
}
```

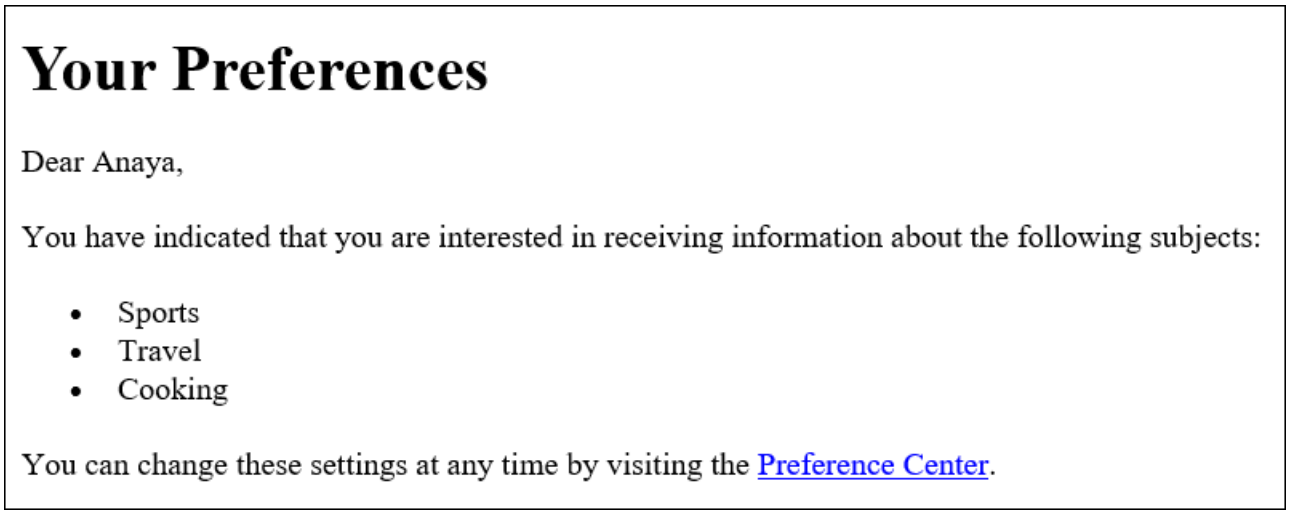
> **Important**
>
> Di contoh kode sebelumnya, nilai-nilai atribut `HtmlPart` dan `TextPart` memiliki jeda baris agar contoh lebih mudah dibaca. JSONFile untuk template Anda tidak dapat berisi jeda baris dalam nilai-nilai ini. Jika Anda menyalin dan menempelkan contoh ini ke JSON file Anda sendiri, hapus jeda baris dan spasi ekstra dari `TextPart` bagian `HtmlPart` dan sebelum melanjutkan.

Setelah Anda membuat templat, Anda dapat menggunakan operasi `SendTemplatedEmail` atau `SendBulkTemplatedEmail` untuk mengirim email ke penerima menggunakan templat ini. Selama setiap penerima memiliki setidaknya satu nilai di objek `Interests`, penerima menerima email yang berisi daftar kepentingan terperinci mereka. Contoh berikut menunjukkan JSON file yang dapat digunakan untuk mengirim email ke beberapa penerima menggunakan template sebelumnya:

```
{
  "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\":[]}"
}
```

Ketika Anda mengirim email ke penerima yang tercantum di contoh sebelumnya menggunakan operasi `SendBulkTemplatedEmail`, mereka menerima pesan yang menyerupai contoh yang ditunjukkan pada citra berikut:



Your Preferences

Dear Anaya,

You have indicated that you are interested in receiving information about the following subjects:

- Sports
- Travel
- Cooking

You can change these settings at any time by visiting the Preference Center.

Menggunakan pernyataan persyaratan dasar

Bagian ini dibangun di atas contoh yang dijelaskan di bagian sebelumnya. Contoh di bagian sebelumnya menggunakan bantuan `each` untuk mengulangi melalui daftar kepentingan. Namun, penerima yang kepentingannya tidak ditentukan menerima email yang berisi daftar kosong. Dengan menggunakan bantuan `{{if}}`, email dapat diberi format yang berbeda jika atribut tertentu hadir di data templat. Kode berikut menggunakan bantuan `{{if}}` untuk menampilkan daftar poin dari bagian sebelumnya jika array `Subscription` berisi nilai. Jika array kosong, blok teks yang berbeda ditampilkan.

```
{
  "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/prefererences/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/prefererences/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/prefererences/i.aspx?
id={{meta.userId}}.
                {{else}}
                  Please update your subscription preferences by visiting the
                  Preference Center at https://www.example.com/prefererences/i.aspx?
id={{meta.userId}}.
                {{/if}}"
  }
}
```

> **Important**
>
> Di contoh kode sebelumnya, nilai-nilai atribut `HtmlPart` dan `TextPart` memiliki jeda baris agar contoh lebih mudah dibaca. JSONFile untuk template Anda tidak dapat berisi jeda baris dalam nilai-nilai ini. Jika Anda menyalin dan menempelkan contoh ini ke JSON file Anda sendiri, hapus jeda baris dan spasi ekstra dari `TextPart` bagian `HtmlPart` dan sebelum melanjutkan.

Contoh berikut menunjukkan JSON file yang dapat digunakan untuk mengirim email ke beberapa penerima menggunakan template sebelumnya:

```
{
  "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\":[]}"
}
```

Di contoh ini, penerima yang data templatnya memiliki daftar kepentingan menerima email yang sama seperti contoh yang ditunjukkan di bagian sebelumnya. Penerima yang data templatnya tidak memiliki kepentingan apa pun, namun, menerima email yang menyerupai contoh yang ditunjukkan pada citra berikut:

Membuat parsial inline

Anda dapat menggunakan parsial inline untuk menyederhanakan templat yang mencakup string berulang. Sebagai contoh, Anda dapat membuat parsial inline yang mencakup nama depan penerima, dan, jika tersedia, nama belakang mereka dengan menambahkan kode berikut ke depan templat Anda:

```
{{#* inline \"fullName\"}}{{firstName}}{{#if lastName}} {{lastName}}{{/if}}{{/
inline}}\n
```

> **Note**
>
> Karakter baris baru (\n) diperlukan untuk memisahkan blok `{{inline}}` dari konten di templat Anda. Baris baru tidak diberikan di output akhir.

Setelah Anda membuat parsial `fullName`, Anda dapat memasukkannya di mana saja di templat Anda dengan meletakkan tanda lebih besar-daripada (>) sebelum nama parsial lalu diikuti oleh spasi, seperti di contoh berikut: `{{> fullName}}`. Parsial inline tidak ditransfer antar bagian dari email.

Misalnya, jika Anda ingin menggunakan sebagian sebaris yang sama di versi HTML dan teks email, Anda harus mendefinisikannya di bagian `HtmlPart` dan `TextPart` bagian.

Anda juga dapat menggunakan parsial inline ketika mengulang melalui array. Anda dapat menggunakan kode berikut untuk membuat templat yang menggunakan parsial inline `fullName`. Di contoh ini, parsial inline berlaku untuk kedua nama penerima dan array nama lain:

```
{
  "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

Di contoh kode sebelumnya, nilai-nilai atribut `HtmlPart` dan `TextPart` memiliki jeda baris agar contoh lebih mudah dibaca. JSONFile untuk template Anda tidak dapat berisi jeda baris dalam nilai-nilai ini. Jika Anda menyalin dan menempelkan contoh ini ke JSON file Anda sendiri, hapus jeda baris dan spasi tambahan dari bagian ini.

Mengelola templat email

Selain membuat template email, Anda juga dapat menggunakan Amazon SES API untuk memperbarui atau menghapus template yang ada, untuk daftar semua template yang ada, atau untuk melihat konten template.

Bagian ini berisi prosedur untuk menggunakan AWS CLI untuk melakukan tugas yang terkait dengan SES template Amazon.

> **Note**
>
> Prosedur di bagian ini menganggap bahwa Anda telah menginstal dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Melihat daftar templat email

Anda dapat menggunakan ListTemplatesoperasi di Amazon SES API untuk melihat daftar semua template email yang ada.

Untuk melihat daftar templat email

- Di baris perintah, masukkan perintah berikut:

```
aws ses list-templates
```

Jika ada template email yang ada di SES akun Amazon Anda di Wilayah saat ini, perintah ini mengembalikan respons yang menyerupai contoh berikut:

```
{
    "TemplatesMetadata": [
        {
            "Name": "SpecialOffers",
            "CreatedTimestamp": "2020-08-05T16:04:12.640Z"
        },
        {
            "Name": "NewsAndUpdates",
            "CreatedTimestamp": "2019-10-03T20:03:34.574Z"
        }
```

```
    ]
}
```

Jika Anda belum membuat templat apa pun, perintah tersebut mengembalikan objek `TemplatesMetadata` dengan tanpa anggota.

Melihat konten templat email tertentu

Anda dapat menggunakan GetTemplateoperasi di Amazon SES API untuk melihat konten template email tertentu.

Untuk melihat konten templat email

- Di baris perintah, masukkan perintah berikut:

```
aws ses get-template --template-name MyTemplate
```

Pada perintah sebelumnya, ganti *MyTemplate* dengan nama template yang ingin Anda lihat.

Jika nama template yang Anda berikan cocok dengan template yang ada di SES akun Amazon Anda, perintah ini menampilkan respons yang menyerupai contoh berikut:

```
{
    "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>"
    }
}
```

Jika nama template yang Anda berikan tidak cocok dengan template yang ada di SES akun Amazon Anda, perintah akan mengembalikan `TemplateDoesNotExist` kesalahan.

Menghapus templat email

Anda dapat menggunakan DeleteTemplateoperasi di Amazon SES API untuk menghapus template email tertentu.

Untuk menghapus templat email

- Di baris perintah, masukkan perintah berikut:

```
aws ses delete-template --template-name MyTemplate
```

Pada perintah sebelumnya, ganti *MyTemplate* dengan nama template yang ingin Anda hapus.

Perintah ini tidak memberikan output apa pun. Anda dapat memverifikasi bahwa template telah dihapus dengan menggunakan GetTemplateoperasi.

Memperbarui templat email

Anda dapat menggunakan UpdateTemplateoperasi di Amazon SES API untuk memperbarui template email yang ada. Misalnya, operasi ini sangat membantu jika Anda ingin mengubah baris subjek templat email, atau jika Anda perlu mengubah badan pesan itu sendiri.

Untuk memperbarui templat email

1. Gunakan perintah `GetTemplate` untuk mengambil templat yang ada dengan memasukkan perintah berikut di baris perintah:

```
aws ses get-template --template-name MyTemplate
```

Pada perintah sebelumnya, ganti *MyTemplate* dengan nama template yang ingin Anda perbarui.

Jika nama template yang Anda berikan cocok dengan template yang ada di SES akun Amazon Anda, perintah ini menampilkan respons yang menyerupai contoh berikut:

```
{
    "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. Di editor teks, buat file baru. Tempel output dari perintah sebelumnya ke dalam file.
3. Ubah templat sesuai kebutuhan. Setiap baris yang Anda hilangkan akan dihapus dari templat. Misalnya, jika Anda hanya ingin mengubah `SubjectPart` dari templat, Anda masih perlu menyertakan properti `TextPart` dan `HtmlPart`.

   Setelah selesai, simpan file sebagai `update_template.json`.
4. Di baris perintah, masukkan perintah berikut:

   ```
   aws ses update-template --cli-input-json file://path/to/update_template.json
   ```

   Pada perintah sebelumnya, ganti *`path/to/update_template.json`* dengan path ke `update_template.json` file yang Anda buat pada langkah sebelumnya.

   Jika templat berhasil diperbarui, perintah ini tidak menyediakan output apa pun. Anda dapat memverifikasi bahwa template telah diperbarui dengan menggunakan GetTemplateoperasi.

   Jika templat yang Anda tentukan tidak ada, perintah ini mengembalikan kesalahan `TemplateDoesNotExist`. Jika templat tidak berisi properti `TextPart` atau `HtmlPart` (atau keduanya), perintah ini mengembalikan kesalahan `InvalidParameterValue`.

Mengirim email melalui Amazon SES menggunakan AWS SDK

Anda dapat menggunakan file AWS SDK untuk mengirim email melalui AmazonSES. AWS SDKstersedia untuk beberapa bahasa pemrograman. Untuk informasi lebih lanjut, lihat Alat untuk Amazon Web Services.

Prasyarat

Prasyarat berikut harus diselesaikan untuk menyelesaikan salah satu contoh kode di bagian berikutnya:

- Jika Anda belum melakukannya, selesaikan tugas di Menyiapkan Amazon Simple Email Service.
- Verifikasi alamat email Anda dengan Amazon SES —Sebelum Anda dapat mengirim email dengan AmazonSES, Anda harus memverifikasi bahwa Anda memiliki alamat email pengirim. Jika akun Anda masih di SES kotak pasir Amazon, Anda juga harus memverifikasi alamat email penerima. Kami menyarankan Anda menggunakan SES konsol Amazon untuk memverifikasi alamat email. Untuk informasi selengkapnya, lihat Membuat identitas alamat email.

- Dapatkan AWS kredensialmu —Anda memerlukan ID kunci AWS akses dan kunci akses AWS rahasia untuk mengakses Amazon SES menggunakan file. SDK Anda dapat menemukan kredensial Anda dengan menggunakan halaman Kredensial Keamanan di AWS Management Console. Untuk informasi selengkapnya tentang kredensial, lihat Tipe kredensial Amazon SES.
- Buat file kredensial bersama—Agar kode sampel di bagian ini berfungsi dengan baik, Anda harus membuat file kredensial bersama. Untuk informasi lebih lanjut, lihat Membuat file kredensi bersama untuk digunakan saat mengirim email melalui Amazon SES menggunakan AWS SDK.

Contoh kode

> **Important**
>
> Dalam tutorial berikut, Anda mengirim email ke diri Anda sendiri sehingga Anda dapat memeriksa apakah Anda sudah menerimanya. Untuk eksperimen lebih lanjut atau pengujian beban, gunakan simulator SES kotak surat Amazon. Email yang Anda kirim ke simulator kotak surat tidak dihitung terhadap kuota pengiriman atau kecepatan pentalan dan aduan Anda. Untuk informasi lebih lanjut, lihat Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual.

.NET

Prosedur berikut menunjukkan kepada Anda cara mengirim email melalui Amazon SES menggunakan Visual Studio dan file AWS SDK for .NET.

Solusi ini diuji dengan menggunakan komponen berikut:

- Microsoft Visual Studio Community 2017, versi 15.4.0.
- Microsoft. NETKerangka versi 4.6.1.
- AWSSDKPaket.Core (versi 3.3.19), diinstal menggunakan. NuGet
- The AWSSDK. SimpleEmail paket (versi 3.3.6.1), diinstal menggunakan. NuGet

Sebelum Anda memulai, lakukan tugas berikut:

- Instal Visual Studio —Visual Studio tersedia di https://www.visualstudio.com/.

Untuk mengirim email menggunakan AWS SDK for .NET

1. Buat proyek baru dengan melakukan langkah-langkah berikut:

   a. Mulai Visual Studio.

   b. Di menu File, pilih Baru, Proyek.

   c. Di jendela Proyek Baru, di panel sebelah kiri, perluas Terinstal, lalu perluas Visual C#.

   d. Di panel di sebelah kanan, pilih Aplikasi Konsol (. NETKerangka kerja).

   e. Untuk Nama, ketik **`AmazonSESSample`**, lalu pilih OK.

2. Gunakan NuGet untuk menyertakan SES paket Amazon dalam solusi Anda dengan menyelesaikan langkah-langkah berikut:

   a. Di panel Solution Explorer, klik kanan project Anda, lalu pilih Manage NuGet Packages.

   b. Pada mazonSESSample tab NuGet: A, pilih Browse.

   c. Di kotak pencarian, ketik **`AWSSDK.SimpleEmail`**.

   d. Pilih AWSSDK. SimpleEmailpaket, dan kemudian pilih Install.

   e. Di jendela Pratinjau Perubahan, pilih OK.

3. Di tab Program.cs, tempel kode berikut:

```
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. Di editor kode, lakukan hal berikut:

   - Ganti *sender@example.com* dengan alamat email “Dari:”. Alamat ini harus diverifikasi. Untuk informasi selengkapnya, lihat Identitas terverifikasi.
   - Ganti *recipient@example.com* dengan alamat “To:”. Jika akun Anda masih berada di sandbox, alamat ini juga harus diverifikasi.
   - Ganti *ConfigSet* dengan nama konfigurasi yang akan digunakan saat mengirim email ini.

- Ganti *USWest2* dengan nama Wilayah AWS endpoint yang Anda gunakan untuk mengirim email menggunakan AmazonSES. Untuk daftar wilayah tempat Amazon SES tersedia, lihat Layanan Email Sederhana Amazon (AmazonSES) di Referensi Umum AWS.

Setelah selesai, simpan `Program.cs`.

5. Bangun dan jalankan aplikasi dengan menyelesaikan langkah berikut:

   a. Di menu Bangun, pilih Bangun Solusi.

   b. Di menu Debug, pilih Mulai Debugging. Jendela konsol muncul.

6. Tinjau output dari konsol tersebut. Jika email berhasil dikirim, konsol tersebut akan menampilkan "`The email was sent successfully.`"

7. Jika email berhasil dikirim, masuk ke klien email alamat penerima. Anda akan menemukan pesan yang Anda kirim.

Java

Prosedur berikut menunjukkan kepada Anda cara menggunakan Eclipse IDE untuk Pengembang Java EE dan AWS Toolkit for Eclipseuntuk membuat AWS SDK proyek dan memodifikasi kode Java untuk mengirim email melalui Amazon. SES

Sebelum Anda memulai, lakukan tugas berikut:

- Instal Eclipse—Eclipse tersedia di https://www.eclipse.org/downloads. Kode dalam tutorial ini diuji menggunakan Eclipse Neon.3 (versi 4.6.3), menjalankan Lingkungan Waktu Aktif Java versi 1.8.
- Instal AWS Toolkit for Eclipse—Instruksi untuk menambahkan instalasi Eclipse Anda tersedia di /eclipse. AWS Toolkit for Eclipse https://aws.amazon.com Kode di tutorial ini diuji menggunakan AWS Toolkit for Eclipse versi 2.3.1.

Untuk mengirim email menggunakan AWS SDK for Java

1. Buat Proyek AWS Java di Eclipse dengan melakukan langkah-langkah berikut:

   a. Mulai Eclipse.

   b. Di menu File, pilih Baru, lalu pilih Lainnya. Di jendela Baru, perluas folder AWS, lalu pilih Java Project AWS ..

c. Di kotak dialog New AWS Java Project, lakukan hal berikut:

   i. Untuk Nama proyek, ketik nama proyek.

   ii. Di bawah AWS SDK for Java Sampel, pilih JavaMail Sampel Layanan Email Sederhana Amazon.

   iii. Pilih Selesai.

2. Di Eclipse, di panel Penjelajah Paket, perluas proyek Anda.
3. Di bawah proyek Anda, perluas folder `src/main/java`, perluas folder `com.amazon.aws.samples`, lalu klik dua kali `AmazonSESSample.java`.
4. Ganti seluruh konten `AmazonSESSample.java` dengan kode berikut:

```
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. Di `AmazonSESSample.java`, ganti berikut dengan nilai-nilai Anda sendiri:

   > **Important**
   >
   > Alamat email peka huruf besar kecil. Pastikan alamatnya sama persis dengan alamat yang Anda verifikasi.

   - `SENDER@EXAMPLE.COM`—Ganti dengan alamat email "Dari" Anda. Alamat ini harus diverifikasi sebelum Anda menjalankan program ini. Untuk informasi lebih lanjut, lihat Identitas terverifikasi di Amazon SES.
   - `RECIPIENT@EXAMPLE.COM`—Ganti dengan alamat email "Kepada" Anda. Jika akun Anda masih berada di sandbox, Anda harus memverifikasi alamat ini sebelum menggunakannya. Untuk informasi selengkapnya, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).
   - (Opsional) **`us-west-2`** —Jika Anda ingin menggunakan Amazon SES di Wilayah selain AS Barat (Oregon), ganti ini dengan Wilayah yang ingin Anda gunakan. Untuk daftar Wilayah tempat Amazon SES tersedia, lihat Layanan Email Sederhana Amazon (AmazonSES) di bagian Referensi Umum AWS.

6. Simpan `AmazonSESSample.java`.
7. Untuk membangun proyek, pilih Proyek lalu pilih Bangun Proyek.

   > **Note**
   >
   > Jika opsi ini dinonaktifkan, pembangunan otomatis dapat diaktifkan; jika demikian, lewati langkah ini.

8. Untuk memulai program dan mengirim email, pilih Jalankan lalu pilih Jalankan lagi.
9. Tinjau output dari panel konsol tersebut di Eclipse. Jika email berhasil dikirim, konsol tersebut akan menampilkan "`Email sent!`", Jika tidak, pesan kesalahan akan ditampilkan.
10. Jika email berhasil dikirim, masuk ke klien email alamat penerima. Anda akan menemukan pesan yang Anda kirim.

PHP

Topik ini menunjukkan cara menggunakan AWS SDK for PHPuntuk mengirim email melalui AmazonSES.

Sebelum Anda memulai, lakukan tugas berikut:

- Instal PHP - PHP tersedia di http://php.net/downloads.php. Tutorial ini membutuhkan PHP versi 5.5 atau lebih tinggi. Setelah Anda menginstalPHP, tambahkan path ke PHP dalam variabel lingkungan Anda sehingga Anda dapat menjalankan PHP dari prompt perintah apa pun. Kode dalam tutorial ini diuji menggunakan PHP 7.2.7.
- Instal AWS SDK for PHP versi 3 —Untuk petunjuk pengunduhan dan penginstalan, lihat AWS SDK for PHP dokumentasi. Kode dalam tutorial ini diuji menggunakan versi 3.64.13 dari file. SDK

Untuk mengirim email melalui Amazon SES menggunakan AWS SDK for PHP

1. Di editor teks, buat file bernama `amazon-ses-sample.php`. Tempel kode berikut:

```
<?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. Di `amazon-ses-sample.php`, ganti berikut dengan nilai-nilai Anda sendiri:

   - **`path_to_sdk_inclusion`**—Ganti dengan jalur yang diperlukan untuk memasukkan AWS SDK for PHP dalam program. Untuk informasi lebih lanjut, lihat dokumentasi AWS SDK for PHP.
   - **`sender@example.com`**—Ganti dengan alamat email yang telah Anda verifikasi dengan AmazonSES. Untuk informasi selengkapnya, lihat Identitas terverifikasi. Alamat email di Amazon SES peka huruf besar/kecil. Pastikan alamat yang Anda masukkan sama persis dengan alamat yang Anda verifikasi.
   - **`recipient1@example.com`**,**`recipient2@example.com`**—Ganti dengan alamat penerima Anda. Jika akun Anda masih berada di sandbox, alamat penerima Anda juga harus diverifikasi. Untuk informasi lebih lanjut, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon). Pastikan alamat yang Anda masukkan sama persis dengan alamat yang Anda verifikasi.
   - **`ConfigSet`** (Opsional)—Jika Anda ingin menggunakan set konfigurasi saat mengirim email ini, ganti nilai dengan nama set konfigurasi. Untuk informasi selengkapnya tentang set konfigurasi, lihat Menggunakan set konfigurasi di Amazon SES.
   - (Opsional) **`us-west-2`** —Jika Anda ingin menggunakan Amazon SES di Wilayah selain AS Barat (Oregon), ganti ini dengan Wilayah yang ingin Anda gunakan. Untuk daftar Wilayah tempat Amazon SES tersedia, lihat Layanan Email Sederhana Amazon (AmazonSES) di bagian Referensi Umum AWS.

3. Simpan `amazon-ses-sample.php`.
4. Untuk menjalankan program, buka prompt perintah di direktori yang sama seperti `amazon-ses-sample.php`, lalu ketik perintah berikut:

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

5. Tinjau output. Jika email berhasil dikirim, konsol tersebut akan menampilkan "`Email sent!`", Jika tidak, pesan kesalahan akan ditampilkan.

> **Note**
>
> Jika Anda menemukan kesalahan “c URL error 60: SSL certificate problem” saat menjalankan program, unduh bundel CA terbaru seperti yang dijelaskan dalam AWS SDK for PHP dokumentasi. Kemudian, di `amazon-ses-sample.php`, tambahkan baris berikut ini ke array `SesClient::factory`, ganti `path_of_certs` dengan jalur ke paket CA yang Anda unduh, lalu kembali jalankan program.
>
> ```
> 'http' => [
>    'verify' => 'path_of_certs\ca-bundle.crt'
> ]
> ```

6. Masuk ke klien email alamat penerima. Anda akan menemukan pesan yang Anda kirim.

Ruby

Topik ini menunjukkan cara menggunakan AWS SDK for Rubyuntuk mengirim email melalui AmazonSES.

Sebelum Anda memulai, lakukan tugas berikut:

- Instal Ruby—Ruby tersedia di https://www.ruby-lang.org/en/downloads/. Kode di tutorial ini diuji menggunakan Ruby 1.9.3. Setelah Anda meginstal Ruby, tambahkan jalur ke Ruby di variabel lingkungan Anda, sehingga Anda dapat menjalankan Ruby dari prompt perintah.
- Instal AWS SDK for Ruby —Untuk petunjuk pengunduhan dan penginstalan, lihat Menginstal AWS SDK for Ruby di Panduan AWS SDK for Ruby Pengembang. Kode sampel di tutorial ini diuji menggunakan AWS SDK for Ruby versi 2.9.36.
- Buat file kredensial bersama—Agar kode sampel di bagian ini berfungsi dengan baik, Anda harus membuat file kredensial bersama. Untuk informasi selengkapnya, lihat Membuat file kredensi bersama untuk digunakan saat mengirim email melalui Amazon SES menggunakan AWS SDK.

Untuk mengirim email melalui Amazon SES menggunakan AWS SDK for Ruby

1. Di editor teks, buat file bernama `amazon-ses-sample.rb`. Tempel kode berikut ke file:

```
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
```

2. Di `amazon-ses-sample.rb`, ganti berikut dengan nilai-nilai Anda sendiri:

   - **`sender@example.com`**—Ganti dengan alamat email yang telah Anda verifikasi dengan AmazonSES. Untuk informasi selengkapnya, lihat Identitas terverifikasi. Alamat email di Amazon SES peka huruf besar/kecil. Pastikan alamat yang Anda masukkan sama persis dengan alamat yang Anda verifikasi.

- **`recipient@example.com`**—Ganti dengan alamat penerima. Jika akun Anda masih berada di sandbox, Anda harus memverifikasi alamat ini sebelum menggunakannya. Untuk informasi lebih lanjut, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon). Pastikan alamat yang Anda masukkan sama persis dengan alamat yang Anda verifikasi.
- (Opsional) **`us-west-2`** —Jika Anda ingin menggunakan Amazon SES di Wilayah selain AS Barat (Oregon), ganti ini dengan Wilayah yang ingin Anda gunakan. Untuk daftar Wilayah tempat Amazon SES tersedia, lihat Layanan Email Sederhana Amazon (AmazonSES) di bagian Referensi Umum AWS.

3. Simpan `amazon-ses-sample.rb`.
4. Untuk menjalankan program, buka prompt perintah di direktori yang sama seperti `amazon-ses-sample.rb`, dan ketik ruby amazon-ses-sample.rb
5. Tinjau output. Jika email berhasil dikirim, konsol tersebut akan menampilkan "`Email sent!`", Jika tidak, pesan kesalahan akan ditampilkan.
6. Masuk ke klien email alamat penerima. Anda akan menemukan pesan yang Anda kirim.

Python

Topik ini menunjukkan cara menggunakan AWS SDK for Python (Boto)untuk mengirim email melalui AmazonSES.

Sebelum Anda memulai, lakukan tugas berikut:

- Verifikasi alamat email Anda dengan Amazon SES —Sebelum Anda dapat mengirim email dengan AmazonSES, Anda harus memverifikasi bahwa Anda memiliki alamat email pengirim. Jika akun Anda masih di SES kotak pasir Amazon, Anda juga harus memverifikasi alamat email penerima. Kami menyarankan Anda menggunakan SES konsol Amazon untuk memverifikasi alamat email. Untuk informasi selengkapnya, lihat Membuat identitas alamat email.
- Dapatkan AWS kredensialmu —Anda memerlukan ID kunci AWS akses dan kunci akses AWS rahasia untuk mengakses Amazon SES menggunakan file. SDK Anda dapat menemukan kredensial Anda dengan menggunakan halaman AWS Management ConsoleKredensial Keamanan. Untuk informasi selengkapnya tentang jenis kredensial, lihat Tipe kredensial Amazon SES.
- Instal Python —Python tersedia di thon.org/downloads/. https://www.py Kode di tutorial ini diuji menggunakan Python 2.7.6 dan Python 3.6.1. Setelah Anda menginstal Phyton, tambahkan jalur ke Phyton di variabel lingkungan Anda, sehingga Anda dapat menjalankan Phyton dari prompt perintah.

- Instal AWS SDK for Python (Boto) —Untuk petunjuk unduhan dan penginstalan, lihat AWS SDK for Python (Boto) dokumentasi. Kode sampel dalam tutorial ini diuji menggunakan versi 1.4.4 SDK untuk Python.

Untuk mengirim email melalui Amazon SES menggunakan SDK untuk Python

1. Di editor teks, buat file bernama `amazon-ses-sample.py`. Tempel kode berikut ke file:

```
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. Di `amazon-ses-sample.py`, ganti berikut dengan nilai-nilai Anda sendiri:

   - **`sender@example.com`**—Ganti dengan alamat email yang telah Anda verifikasi dengan AmazonSES. Untuk informasi selengkapnya, lihat Identitas terverifikasi. Alamat email di Amazon SES peka huruf besar/kecil. Pastikan alamat yang Anda masukkan sama persis dengan alamat yang Anda verifikasi.
   - **`recipient@example.com`**—Ganti dengan alamat penerima. Jika akun Anda masih berada di sandbox, Anda harus memverifikasi alamat ini sebelum menggunakannya. Untuk informasi lebih lanjut, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon). Pastikan alamat yang Anda masukkan sama persis dengan alamat yang Anda verifikasi.
   - (Opsional) **`us-west-2`** —Jika Anda ingin menggunakan Amazon SES di Wilayah selain AS Barat (Oregon), ganti ini dengan Wilayah yang ingin Anda gunakan. Untuk daftar Wilayah tempat Amazon SES tersedia, lihat Layanan Email Sederhana Amazon (AmazonSES) di bagian Referensi Umum AWS.

3. Simpan `amazon-ses-sample.py`.
4. Untuk menjalankan program, buka prompt perintah di direktori yang sama seperti `amazon-ses-sample.py`, lalu ketik python amazon-ses-sample.py.
5. Tinjau output. Jika email berhasil dikirim, konsol tersebut akan menampilkan "`Email sent!`", Jika tidak, pesan kesalahan akan ditampilkan.
6. Masuk ke klien email alamat penerima. Anda akan menemukan pesan yang Anda kirim.

Membuat file kredensi bersama untuk digunakan saat mengirim email melalui Amazon SES menggunakan AWS SDK

Prosedur berikut ini menunjukkan cara membuat file kredensial bersama di direktori beranda Anda. Agar kode SDK sampel berfungsi dengan baik, Anda harus membuat file ini.

1. Di editor teks, buat file baru. Di file, tempel kode berikut:

```
[default]
```

```
aws_access_key_id = YOUR_AWS_ACCESS_KEY_ID
aws_secret_access_key = YOUR_AWS_SECRET_ACCESS_KEY
```

2. Dalam file teks yang baru saja Anda buat, ganti YOUR_AWS_ACCESS_KEY dengan ID kunci AWS akses unik Anda, dan ganti YOUR_AWS_SECRET_ACCESS_KEY dengan kunci akses AWS rahasia unik Anda.
3. Simpan file tersebut. Tabel berikut menunjukkan lokasi dan nama file yang benar untuk sistem operasi Anda.

Jika Anda menggunakan...	Simpan file sebagai...
Windows	`C:\Users\<yourUserName>\.aws\credentials`
Linux, macOS, atau Unix	`~/.aws/credentials`

> **Important**
>
> Jangan sertakan ekstensi file saat menyimpan file kredensial.

Pengkodean konten yang didukung oleh Amazon SES

Berikut ini disediakan untuk referensi.

Amazon SES mendukung pengkodean konten berikut:

- `deflate`
- `gzip`
- `identity`

Amazon SES juga mendukung format header Accept-Encoding berikut, sesuai dengan spesifikasi 7231: RFC

- `Accept-Encoding:deflate,gzip`
- `Accept-Encoding:`
- `Accept-Encoding:*`

- `Accept-Encoding:deflate;q=0.5,gzip;q=1.0`
- `Accept-Encoding:gzip;q=1.0,identity;q=0.5,*;q=0`

Amazon SES dan protokol keamanan

Topik ini menjelaskan protokol keamanan yang dapat Anda gunakan saat terhubung ke AmazonSES, dan saat Amazon SES mengirimkan email ke penerima.

Pengirim email ke Amazon SES

Protokol keamanan yang Anda gunakan untuk terhubung ke Amazon SES tergantung pada apakah Anda menggunakan SES SMTP antarmuka Amazon SES API atau Amazon, seperti yang dijelaskan selanjutnya.

HTTPS

Jika Anda menggunakan Amazon SES API (baik secara langsung atau melalui AWS SDK), maka semua komunikasi dienkripsi TLS melalui titik akhir Amazon SESHTTPS. SESHTTPSEndpoint Amazon mendukung TLS 1.2 dan TLS 1.3.

SMTPantarmuka

Jika Anda mengakses Amazon SES melalui SMTP antarmuka, Anda harus mengenkripsi koneksi menggunakan Transport Layer Security ()TLS. Perhatikan bahwa sering TLS disebut dengan nama protokol pendahulunya, Secure Sockets Layer (SSL).

Amazon SES mendukung dua mekanisme untuk membuat koneksi TLS -terenkripsi: STARTTLS dan Wrapper. TLS

- STARTTLS— STARTTLS adalah sarana untuk meningkatkan koneksi yang tidak terenkripsi ke koneksi terenkripsi. Ada versi STARTTLS untuk berbagai protokol; SMTP versi didefinisikan dalam RFC 3207. Untuk STARTTLS koneksi, Amazon SES mendukung TLS 1.2 dan TLS 1.3.
- TLSTLSWrapper (juga dikenal sebagai SMTPS atau Handshake Protocol) adalah sarana untuk memulai koneksi terenkripsi tanpa terlebih dahulu membuat koneksi yang tidak terenkripsi. Dengan TLS Wrapper, SES SMTP endpoint Amazon tidak melakukan TLS negosiasi: itu adalah tanggung jawab klien untuk terhubung ke titik akhir menggunakanTLS, dan untuk terus menggunakan TLS untuk seluruh percakapan. TLSWrapper adalah protokol yang lebih lama, tetapi banyak klien masih mendukungnya. Untuk koneksi TLS Wrapper, Amazon SES mendukung TLS 1.2 dan TLS 1.3.

Untuk informasi tentang menghubungkan ke SES SMTP antarmuka Amazon menggunakan metode ini, lihatMenghubungkan ke titik SES SMTP akhir Amazon.

Amazon SES ke penerima

Sementara TLS 1.3 adalah metode pengiriman default kami, SES dapat mengirimkan email ke server email menggunakan versi sebelumnyaTLS.

Secara default, Amazon SES menggunakan oportunistik TLS. Ini berarti bahwa Amazon SES selalu mencoba untuk membuat koneksi aman ke server email penerima. Jika Amazon tidak SES dapat membuat koneksi aman, Amazon mengirimkan pesan yang tidak terenkripsi.

Anda dapat mengubah perilaku ini dengan menggunakan set konfigurasi. Gunakan PutConfigurationSetDeliveryOptionsAPIoperasi untuk mengatur `TlsPolicy` properti untuk konfigurasi yang disetel ke`Require`. Anda dapat menggunakan AWS CLI untuk membuat perubahan ini.

Untuk mengonfigurasi Amazon SES agar memerlukan TLS koneksi untuk set konfigurasi

- Di baris perintah, masukkan perintah berikut:

  ```
  aws sesv2 put-configuration-set-delivery-options --configuration-set-
  name MyConfigurationSet --tls-policy REQUIRE
  ```

  Pada contoh sebelumnya, ganti *`MyConfigurationSet`* dengan nama set konfigurasi Anda.

  Saat Anda mengirim email menggunakan set konfigurasi ini, Amazon SES hanya mengirim pesan ke server email penerima jika dapat membuat koneksi aman. Jika Amazon tidak SES dapat membuat koneksi aman ke server email penerima, itu akan menghapus pesan.

nd-to-end Enkripsi E

Anda dapat menggunakan Amazon SES untuk mengirim pesan yang dienkripsi menggunakan MIME S/ atau. PGP Pesan yang menggunakan protokol ini dienkripsi oleh pengirim. Konten mereka hanya dapat dilihat oleh penerima yang memiliki kunci pribadi yang diperlukan untuk mendekripsi pesan.

Amazon SES mendukung MIME jenis berikut, yang dapat Anda gunakan untuk mengirim email S/ MIME terenkripsi:

- `application/pkcs7-mime`

- `application/pkcs7-signature`
- `application/x-pkcs7-mime`
- `application/x-pkcs7-signature`

Amazon SES juga mendukung MIME jenis berikut, yang dapat Anda gunakan untuk mengirim email yang PGP dienkripsi:

- `application/pgp-encrypted`
- `application/pgp-keys`
- `application/pgp-signature`

Bidang SES header Amazon

Amazon SES dapat menerima semua header email yang mengikuti format yang dijelaskan dalam RFC822.

Bidang berikut tidak dapat muncul lebih dari sekali di bagian header pesan:

- `Accept-Language`
- `acceptLanguage`
- `Archived-At`
- `Auto-Submitted`
- `Bounces-to`
- `Comments`
- `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
- Date
- 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
- List-Unsubscribe-Post
- Message-Context
- Message-ID
- MIME-Version
- Organization

- `Original-From`
- `Original-Message-ID`
- `Original-Recipient`
- `Original-Subject`
- `Precedence`
- `Priority`
- `References`
- `Reply-To`
- `Return-Path`
- `Return-Receipt-To`
- `Sender`
- `Solicitation`
- `Sensitivity`
- `Subject`
- `Thread-Index`
- `Thread-Topic`
- `User-Agent`
- `VBR-Info`

Pertimbangan

- Bidang `acceptLanguage` ini tidak standar. Jika memungkinkan, Anda sebaiknya menggunakan header `Accept-Language` sebagai gantinya.
- Jika Anda menentukan `Date` header, Amazon akan SES menggantinya dengan stempel waktu yang sesuai dengan tanggal dan waktu di UTC zona waktu saat Amazon SES menerima pesan tersebut.
- Jika Anda memberikan `Message-ID` header, Amazon SES mengganti header dengan nilainya sendiri.
- Jika Anda menentukan `Return-Path` header, Amazon SES mengirimkan pemberitahuan pantulan dan keluhan ke alamat yang Anda tentukan. Namun, pesan yang diterima penerima Anda berisi nilai yang berbeda pada header `Return-Path`.

- Jika Anda menggunakan `SendEmail` operasi Amazon SES API v2 dengan konten Sederhana atau Templated, atau menggunakan `SendBulkEmail` operasi, Anda tidak dapat menyetel konten header khusus untuk header yang disetel olehSES; oleh karena itu, header berikut tidak diizinkan sebagai header khusus:
  - `BCC`, `CC`, `Content-Disposition`, `Content-Type`, `Date`, `From`, `Message-ID`, `MIME-Version`, `Reply-To`, `Return-Path`, `Subject`, `To`

Jenis lampiran Amazon SES yang tidak didukung

Anda dapat mengirim pesan dengan lampiran melalui Amazon SES dengan menggunakan standar Multipurpose Internet Mail Extensions (MIME). Amazon SES menerima semua jenis lampiran file kecuali lampiran dengan ekstensi file dalam daftar berikut.

.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

Beberapa ISPs memiliki batasan lebih lanjut (seperti pembatasan terkait lampiran yang diarsipkan), jadi kami sarankan untuk menguji pengiriman email Anda melalui major ISPs sebelum Anda mengirim email produksi Anda.

Penerimaan email dengan Amazon SES

Selain menggunakan Amazon SES untuk mengelola pengiriman email Anda, Anda juga dapat mengonfigurasi SES untuk menerima email atas nama satu atau beberapa domain Anda. Sebagai penerima email, SES menangani operasi penerimaan surat yang mendasarinya, seperti berkomunikasi dengan server email lain, memindai spam dan virus, memblokir email dari sumber yang tidak tepercaya (alamat pada daftar blokir baik Spamhaus atauSES), dan menerima email untuk penerima di domain Anda.

Tingkat pemrosesan pada email yang Anda terima ditentukan oleh instruksi khusus yang Anda tentukan. Instruksi ini ada dalam dua bentuk:

- Aturan penerimaan (kontrol berbasis penerima) memberikan granularitas terbaik kontrol atas email yang masuk. Aturan penerimaan dapat melakukan pemrosesan lanjutan seperti mengirimkan email masuk ke bucket Amazon S3, mempublikasikannya ke topik Amazon, mengirimkannya ke SNS WorkMail Amazon, atau secara otomatis mengirim pesan bouncing saat pesan ke alamat email tertentu, dan banyak lagi.
- Pemfilteran alamat IP (Kontrol berbasis IP) memberikan tingkat kontrol yang luas dan mudah diatur. Filter ini mengizinkan Anda untuk secara eksplisit memblokir atau mengizinkan semua pesan dari alamat IP tertentu atau rentang alamat IP.

Untuk mulai belajar tentang menerima email, menyiapkannya, dan menerapkannya menggunakan aturan penerimaan atau pemfilteran alamat IP, bacalah Konsep penerimaan email & kasus penggunaan terlebih dahulu guna mendapatkan gambaran umum tentang cara kerjanya dan cara berbeda yang dapat Anda gunakan. Selanjutnya, Menyiapkan penerimaan email akan memandu Anda melalui prasyarat penyiapan penerimaan email. Kemudian, Panduan konsol penerimaan email akan memandu Anda melalui wizard yang digunakan untuk mengonfigurasi aturan penerimaan dan pemfilteran alamat IP.

> **Note**
>
> Penerimaan email hanya dapat digunakan jika akun Anda berada di Wilayah AWS tempat yang SES mendukung penerimaan email. Tabel titik akhir Penerimaan Email dalam Referensi Umum AWS daftar semua Wilayah AWS tempat SES mendukung penerimaan email.

Topik di bagian ini:

- Konsep penerimaan email dan kasus penggunaan Amazon SES
- Menyiapkan penerimaan email Amazon SES
- Panduan konsol penerimaan email Amazon SES
- Melihat metrik untuk penerimaan email Amazon SES

Konsep penerimaan email dan kasus penggunaan Amazon SES

Ketika Anda menggunakan Amazon SES sebagai penerima email Anda, Anda memberitahu yang harus dilakukan layanan dengan surat Anda. Metode utama, aturan penerimaan, memberi Anda kontrol yang sangat baik atas penerimaan email Anda dengan memanfaatkan kontrol berbasis penerima untuk menentukan set tindakan yang harus diambil berdasarkan penerima. Untuk metode lain, filter alamat IP, menyediakan tingkat kontrol berbasis IP yang luas untuk memblokir atau mengizinkan surat berdasarkan asal alamat IP atau rentang alamat.

Kedua metode ini dijelaskan di bagian ini bersama dengan gambaran umum tentang cara proses Amazon SES menerima email, dan kasus penggunaan untuk membantu Anda mempertimbangkan cara Anda ingin menerima, filter, dan memproses email Anda saat mengatur aturan dan filter.

Topik di bagian ini:

- Kontrol berbasis penerima menggunakan aturan penerimaan
- Kontrol berbasis IP menggunakan filter alamat IP
- Proses penerimaan email
- Kasus penggunaan dan pembatasan untuk penerimaan email Amazon SES
- Otentikasi penerima email dan pemindaian malware

Kontrol berbasis penerima menggunakan aturan penerimaan

Cara utama mengontrol surat Anda yang masuk adalah dengan menentukan cara surat ditangani melalui daftar tindakan yang diperintahkan untuk setiap identitas identitas domain terverifikasi Anda yang mencakup domain, sub-domain, atau alamat email - perhatikan bahwa alamat email harus termasuk dalam salah satu identitas domain terverifikasi Anda. Tindakan ini ditentukan dan diperintahkan dalam aturan penerimaan yang Anda buat dalam set aturan.

Sebagai opsi, Anda juga dapat menambahkan syarat penerima sebagai cara untuk menentukan bahwa tindakan hanya diambil jika penerima kepada surat yang masuk cocok ditujukan dengan

identitas penerima yang ditentukan pada syarat. Misalnya, jika Anda memiliki example.com, Anda dapat menentukan surat tersebut untuk user@example.com harus pentalan, dan semua surat lain untuk example.com dan subdomainnya harus dikirimkan.

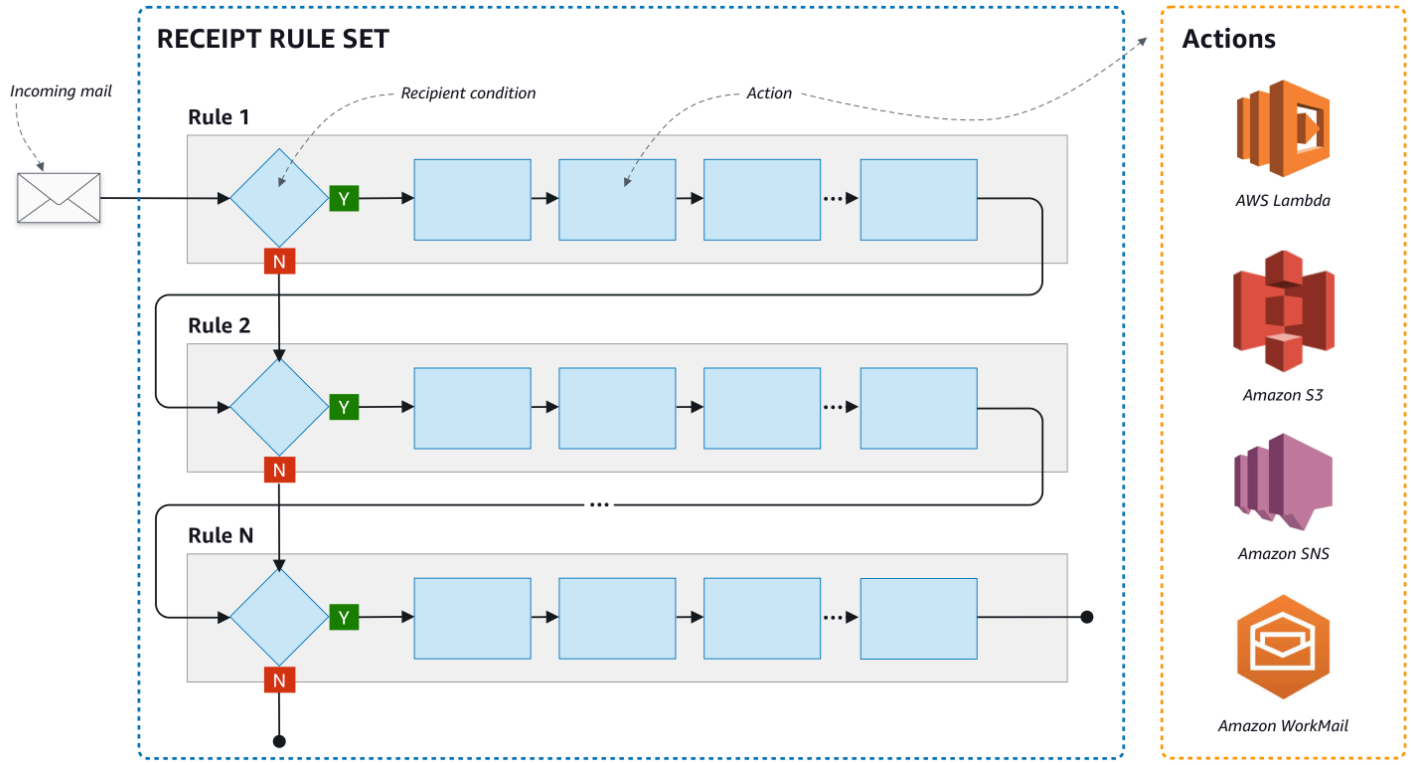
Sebaliknya, jika Anda tidak menambahkan persyaratan penerima apa pun, tindakan akan diterapkan pada semua hal - semua alamat email, domain, dan sub-domain milik domain yang terverifikasi. Tindakan berikut tersedia untuk diterapkan pada aturan penerimaan Anda:

- Tambahkan tindakan header—Tambahkan header ke email yang diterima. Anda biasanya menggunakan tindakan ini hanya dalam kombinasi dengan tindakan lain.
- Kembalikan tindakan respons pentalan— Blok email dengan mengembalikan respons pentalan ke pengirim dan, secara opsional, memberi tahu Anda melalui Amazon SNS.
- Aktifkan tindakan fungsi AWS Lambda—Memanggil kode Anda melalui fungsi Lambda dan, secara opsional, memberitahu Anda melalui Amazon SNS.
- Kirim ke tindakan bucket S3—Mengirimkan surat ke bucket Amazon S3 dan, secara opsional, memberitahu Anda melalui Amazon SNS.
- Publikasikan ke tindakan topik Amazon SNS—Memublikasikan email lengkap untuk topik Amazon SNS.

> **Note**
>
> Tindakan SNS mencakup salinan lengkap konten email dalam notifikasi Amazon SNS. Opsi notifikasi Amazon SNS lain yang disebutkan di sini hanya memberitahu Anda tentang pengiriman email; yang berisi informasi tentang email, bukan konten email itu sendiri.

- Hentikan tindakan set aturan—Mengakhiri evaluasi set aturan penerimaan dan, secara opsional, memberitahu Anda melalui Amazon SNS.
- Integrasikan dengan WorkMail tindakan Amazon— Menangani surat dengan AmazonWorkMail. Anda biasanya tidak akan menggunakan tindakan ini secara langsung karena Amazon WorkMail menangani pengaturannya.

Aturan penerimaan dikelompokkan bersama ke set aturan. Jika Anda tidak memiliki set aturan yang ada, Anda harus membuat set aturan terlebih dahulu sebelum mulai membuat aturan penerimaan. Anda dapat menentukan beberapa set aturan untuk akun AWS, namun hanya satu set aturan yang aktif setiap saat. Gambar berikut menunjukkan cara aturan penerimaan, set aturan, dan tindakan berhubungan satu sama lain.



Kontrol berbasis IP menggunakan filter alamat IP

Anda dapat mengontrol alur surat dengan mengatur filter alamat IP. Filter alamat IP merupakan opsional dan memungkinkan Anda untuk menentukan apakah akan menerima atau memblokir surat yang berasal dari alamat IP atau kisaran alamat IP. Filter alamat IP Anda dapat mencakup daftar blokir (Alamat IP dari tempat Anda ingin memblokir surat masuk) dan izinkan daftar (Alamat IP dari tempat Anda ingin selalu menerima surat).

Filter alamat IP berguna untuk memblokir spam. Amazon SES mempertahankan daftar blokirnya sendiri dari alamat IP yang dikenal untuk mengirim spam termasuk yang terdaftar di Spamhaus. Namun, Anda dapat memilih untuk menerima surat dari alamat IP tersebut dengan menambahkannya ke daftar izin Anda. Karena tidak ada log yang menunjukkan alamat IP mana yang diblokir, pengirim yang diblokir perlu memberi tahu Anda. Ini juga merupakan kesempatan yang baik untuk membantu pengirim menentukan apakah alamat IP mereka ada di daftar blok, seperti Spamhaus, dan merekomendasikan mereka meminta agar tidak terdaftar. Melakukan hal itu akan bermanfaat bagi Anda dan pengirim karena Anda tidak perlu memelihara filter alamat IP untuk mereka dan mereka akan meningkatkan pengiriman email mereka.

> **Note**
>
> - Terlepas dari konfigurasi filter alamat IP Anda, Amazon EC2 akan memblokir lalu lintas keluar pada port 25 (pengiriman email) kecuali diizinkan terdaftar. Lihat artikel AWS Re: posting ini untuk informasi lebih lanjut.
> - Jika Anda hanya ingin menerima surat dari daftar alamat IP terbatas yang diketahui, maka atur daftar blokir yang berisi `0.0.0.0/0`, dan atur daftar izinkan yang berisi alamat IP yang Anda percayai. Konfigurasi ini memblokir semua alamat IP secara default, dan hanya mengizinkan surat dari alamat IP yang Anda tentukan secara eksplisit.

Proses penerimaan email

Ketika Amazon SES menerima email untuk domain Anda, kejadian berikut terjadi:

1. Amazon SES pertama melihat alamat IP dari pengirim. Amazon SES mengizinkan surat untuk meneruskan tahap ini kecuali:
   - Alamat IP ada di daftar blokir Anda.
   - Alamat IP ada di daftar blokir Amazon SES, tapi tidak ada di daftar izin Anda.
2. Amazon SES memeriksa set aturan aktif Anda untuk menentukan salah satu aturan penerimaan Anda yang berisi syarat penerima:
   - Jika ada syarat penerima dan itu cocok dengan salah satu penerima email yang masuk, Amazon SES menerima email. Sebaliknya, jika tidak ada kecocokan, Amazon SES memblokir email.
   - Jika aturan penerimaan tidak berisi syarat penerima, Amazon SES menerima surat - semua tindakan aturan akan diterapkan untuk semua identitas terverifikasi yang Anda miliki.
3. Amazon SES mengautentikasi email dan memindai kontennya untuk spam dan malware:
   - Alamat IP host jarak jauh yang mengirimkan email ke Amazon SES diperiksa terhadap kebijakan SPF yang ditentukan di bawah domain MAIL FROM yang digunakan selama transaksi SMTP.
   - Tanda tangan DKIM yang ada di bagian header email dicentang.
   - Jika pemindaian konten diaktifkan, konten email dipindai untuk spam dan malware.
   - Otentikasi email dan hasil pemindaian konten tersedia untuk Anda selama evaluasi aturan penerimaan.

   Lihat Otentikasi email dan deteksi malware untuk informasi selengkapnya.

4. Untuk email yang diterima Amazon SES, semua aturan penerimaan dalam set aturan aktif Anda diterapkan sesuai urutan yang telah Anda tetapkan; dan dalam setiap aturan penerimaan, tindakan dijalankan sesuai urutan yang telah Anda tetapkan.

Kasus penggunaan dan pembatasan untuk penerimaan email Amazon SES

Bagian ini melampaui beberapa pertimbangan umum dan kasus penggunaan untuk penerimaan email Amazon SES. Disajikan dalam format tanya jawab, pertanyaan dan fakta yang sering diajukan untuk membantu menentukan bahwa itu akan bermanfaat atau tidak untuk menggunakan Amazon SES untuk menerima dan mengelola email atas nama satu atau lebih domain terverifikasi yang Anda miliki.

Ketersediaan Wilayah

Apakah Amazon SES mendukung penerimaan email di Wilayah Anda?

Amazon SES hanya mendukung penerimaan email di Wilayah AWS tertentu. Untuk daftar lengkap Wilayah yang mendukung penerimaan email, lihat titik akhir dan kuota Amazon Simple Email Service dalam. Referensi Umum AWS

Klien email berbasis POP atau IMAP

Dapatkah Microsoft Outlook digunakan untuk menerima email masuk?

Amazon SES tidak menyertakan server POP atau IMAP untuk menerima email masuk. Ini berarti Anda tidak dapat menggunakan klien email seperti Microsoft Outlook untuk menerima email masuk. Jika Anda memerlukan solusi yang dapat mengirim dan menerima email dengan menggunakan klien email, pertimbangkan untuk menggunakan klien email, pertimbangkan untuk menggunakan klien email, pertimbangkan untuk menggunakan klien email, pertimbangkan untuk menggunakan klien email, pertimbangkan untuk menggunakan klien email, WorkMail pertimbangkan untuk

Menggunakan layanan AWS lainnya

Sudahkan Anda mengatur izin yang sesuai?

Jika Anda ingin email Anda dikirim ke bucket S3, dipublikasikan ke topik Amazon SNS yang tidak Anda miliki, memicu fungsi Lambda, atau menggunakan kunci yang dikelola pelanggan, Anda perlu memberikan Amazon SES izin untuk mengakses sumber daya tersebut. Untuk memberikan Amazon

SES akses, Anda membuat kebijakan pada sumber daya dari konsol atau API untuk layanan AWS tersebut. Untuk informasi selengkapnya Memberikan izin.

Konten email

Bagaimana Anda ingin Amazon SES menyampaikan konten email kepada Anda?

Amazon SES dapat memberikan konten email dalam dua cara: itu dapat menyimpan email dalam bucket S3 yang Anda tentukan, atau dapat mengirimkan Anda notifikasi Amazon SNS yang berisi salinan email. Amazon SES mengirimi Anda email mentah, email yang dimodifikasi dalam format Multipurpose Internet Mail Extensions (MIME). Untuk informasi selengkapnya tentang format MIME, lihat RFC 2045.

Seberapa besar email yang akan Anda terima?

Jika Anda menyimpan email dalam bucket S3, ukuran email maksimum (termasuk header) adalah 40 MB. Jika Anda menerima email Anda melalui notifikasi Amazon SNS, ukuran email maksimum (termasuk header) adalah 150 KB.

Bagaimana Anda ingin memicu pemrosesan surat Anda?

Setelah surat Anda dikirim, Anda akan menginginkan untuk memprosesnya dengan kode Anda sendiri. Misalnya, aplikasi Anda mungkin mengonversi basis email yang dikodekan 64 ke dalam format yang dapat ditampilkan lalu membuatnya tersedia untuk pengguna akhir melalui klien email. Ada beberapa cara untuk memulai prosesnya:

- Jika email Anda dikirim ke Amazon S3, aplikasi Anda dapat mendengarkan notifikasi Amazon SNS yang dihasilkan oleh tindakan S3, ekstrak ID pesan email dari notifikasi, lalu gunakan ID pesan untuk mengambil email dari Amazon S3.

  Atau, Anda dapat memasukkan pemrosesan email ke aturan penerimaan Anda dengan menulis fungsi Lambda. Dalam kasus ini, aturan penerimaan Anda harus terlebih dahulu menulis email ke Amazon S3, lalu memicu fungsi Lambda. tindakan Lambda dapat dijalankan secara sinkron atau asinkron dari dalam aturan penerimaan Anda, tergantung pada fungsi Lambda yang perlu mengembalikan hasil yang mempengaruhi cara tindakan lain dijalankan. Kami merekomendasikan agar Anda menggunakan eksekusi asinkron kecuali sinkron benar-benar diperlukan untuk kasus penggunaan Anda. Untuk informasi selengkapnya tentang AWS Lambda, lihat Panduan Developer AWS Lambda.
- Jika email Anda dikirim melalui notifikasi Amazon SNS dengan menggunakan tindakan SNS, aplikasi Anda dapat mendengarkan notifikasi Amazon SNS, lalu ekstrak pesan email dari notifikasi.

Apakah Anda ingin email dienkripsi?

Amazon SES terintegrasi dengan AWS Key Management Service (AWS KMS) untuk mengenkripsi surat secara opsional yang ditulis ke bucket S3 Anda. Amazon SES menggunakan enkripsi di sisi klien untuk mengenkripsi surat Anda sebelum menulisnya ke Amazon S3. Ini berarti Anda harus mendekripsi konten di sisi Anda setelah mengambil surat dari Amazon S3. AWS SDK for Java dan AWS SDK for Ruby memberikan klien yang dapat menangani dekripsi untuk Anda. Amazon SES dapat mengenkripsi email untuk Anda hanya jika Anda memilih untuk email Anda yang akan dikirim ke bucket S3.

Email tidak diinginkan

Pada titik mana dalam proses penerimaan email Anda ingin memblokir surat yang tidak diinginkan?

Saat pengirim mencoba mengirim email ke penerima, server email pengirim akan bertukar urutan perintah dengan server penerima. Urutan ini disebut percakapan SMTP.

Anda dapat memblokir email masuk pada dua titik dalam proses penerimaan email: selama percakapan SMTP, dan setelah percakapan SMTP. Anda menggunakan filter alamat IP untuk memblokir pesan selama percakapan SMTP, dan aturan penerimaan untuk memblokir email setelah percakapan SMTP.

Anda dapat menggunakan filter alamat IP untuk memblokir email yang berasal dari alamat IP tertentu. Manfaat menggunakan filter alamat IP untuk memblokir surat yang tidak diinginkan adalah kami tidak mengenakan biaya untuk pesan yang diblokir selama percakapan SMTP. Kelemahan menggunakan filter alamat IP adalah mereka memblokir email dari alamat IP yang Anda tentukan tanpa melakukan analisis apa pun pada konten sebenarnya dari pesan. Untuk informasi selengkapnya tentang filter alamat IP, lihat Buat panduan konsol filter alamat IP.

Anda dapat menggunakan aturan penerimaan untuk mengirim notifikasi pentalan ke pengirim email berdasarkan alamat (atau domain, atau subdomain) tempat pesan tersebut dikirim. Manfaat menggunakan aturan penerimaan adalah Anda dapat melakukan analisis tambahan pada pesan masuk sebelum mengirim notifikasi pentalan ke pengirim. Misalnya, Anda dapat menggunakan AWS Lambda untuk mengirim notifikasi pentalan hanya jika pesan gagal mengautentikasi DKIM atau diidentifikasi sebagai spam. Kelemahan menggunakan aturan penerimaan adalah, karena aturan penerimaan diproses setelah percakapan SMTP, kami menagih Anda untuk setiap pesan yang Anda terima. Anda mungkin juga akan dikenakan biaya jika menggunakan Lambda untuk menganalisis konten pesan masuk. Untuk informasi selengkapnya tentang aturan penerimaan, lihat Membuat aturan penerimaan pada panduan konsol. Untuk informasi selengkapnya tentang menggunakan Lambda untuk menganalisis email masuk, lihat Contoh fungsi Lambda.

Pengaliran surat

Bagaimana Anda ingin membagi pengaliran surat Anda?

Domain Anda kemungkinan besar menerima kelas surat yang berbeda. Misalnya, beberapa surat domain Anda, seperti email ke user@example.com, mungkin ditujukan untuk kotak masuk pribadi. Surat lainnya, seperti email ke unsubscribe@example.com, mungkin lebih baik diarahkan ke sistem otomatis sebagai gantinya. Anda dapat menggunakan aturan penerimaan untuk membagi surat masuk sehingga surat tersebut dapat diproses secara berbeda. Untuk informasi tentang cara mengatur aturan penerimaan, lihat Membuat aturan penerimaan.

Otentikasi penerima email dan pemindaian malware

Amazon SES mengautentikasi setiap email yang diterima dan secara opsional memindai konten email untuk spam dan malware. SES tidak mengambil tindakan apa pun pada email yang diterima berdasarkan hasil otentikasi email atau pemindaian konten; namun, hasil operasi ini diberikan kepada Anda sebagai atribut yang dapat Anda gunakan dalam tindakan aturan penerimaan SES seperti pemberitahuan Amazon SNS atau sebagai header dalam pesan yang dikirimkan ke Amazon S3.

Otentikasi email

Amazon SES mengautentikasi setiap email yang diterima menggunakan SPF, DKIM, dan DMARC. Hasil dari setiap mekanisme autentikasi disediakan dalam notifikasi Amazon SNS yang dikirim SES sebagai bagian dari evaluasi aturan dalam aturan penerimaan aktif yang ditetapkan. Selain itu, jika Anda memilih untuk menerima salinan email di Amazon S3, hasil otentikasi email ditangkap di `Authentication-Results` header yang ditambahkan SES ke bagian header email:

```
Authentication-Results: example.com;
spf=pass (spfCheck: 10.0.0.1 is permitted by domain of example.com) client-ip=10.0.0.1;
 envelope-from=example@example.com; helo=10.0.0.1;
dkim=pass header.i=example.com;
dkim=permerror header.i=some-example.com;
dmarc=pass header.from=example@example.com;
```

`Authentication-Results`Header dijelaskan dalam RFC 8601

Pemindaian konten email untuk deteksi spam dan malware

Amazon SES memindai konten email yang diterima untuk malware bergantung pada nilai atribut ScanEnabled(API) atau pemindaian spam dan virus (konsol) dari aturan penerimaan yang

cocok dengan email. Secara default SES scan menerima konten email untuk malware. Untuk menonaktifkan pemindaian konten untuk email yang diterima yang cocok dengan aturan penerimaan tertentu, Anda harus menetapkan tanda terima aturan ke false jika menggunakan API, atau menghapus kotak centang Spam dan pemindaian virus jika menggunakan konsol. ScanEnabled Jika aturan penerimaan yang cocok dengan email diaktifkan pemindaian, hasil pemindaian konten disediakan dalam notifikasi Amazon SNS yang dikirim SES sebagai bagian dari evaluasi aturan dalam aturan penerimaan aktif yang ditetapkan. Selain itu, jika Anda memilih untuk menerima salinan email di Amazon S3, hasil pemindaian konten ditangkap di `X-SES-Spam-Verdict` dan `X-SES-Virus-Verdict` header yang ditambahkan SES ke bagian header email.

```
X-SES-Spam-Verdict: PASS
X-SES-Virus-Verdict: FAIL
```

Nilai yang mungkin untuk header di atas tercantum dalam:

- spam
- virus

Sekarang setelah Anda memahami konsep penerimaan email, cara kerjanya, dan kasus penggunaannya, Anda dapat memulai dengan membuka Menyiapkan penerimaan email.

Menyiapkan penerimaan email Amazon SES

Bagian ini menjelaskan prasyarat yang diperlukan sebelum Anda dapat mulai mengonfigurasi Amazon SES untuk menerima email Anda. Sangat penting bahwa Anda telah membaca Konsep penerimaan email & kasus penggunaan untuk memahami konsep bagaimana Amazon SES bekerja dan mempertimbangkan cara Anda ingin menerima, memfilter, dan memproses email Anda.

Sebelum Anda dapat mengonfigurasi penerimaan email dengan membuat set aturan, aturan penerimaan, dan filter alamat IP, Anda harus terlebih dahulu menyelesaikan prasyarat penyiapan berikut:

- Verifikasi domain Anda dengan Amazon SES dengan menerbitkan catatan DNS untuk membuktikan bahwa Anda memilikinya.
- Izinkan Amazon SES menerima email untuk domain Anda dengan menerbitkan catatan MX.
- Berikan izin Amazon SES untuk mengakses sumber daya AWS lainnya guna melaksanakan tindakan aturan penerimaan.

Saat membuat dan memverifikasi identitas domain, Anda menerbitkan catatan ke pengaturan DNS untuk menyelesaikan proses verifikasi, namun hal ini tidak cukup untuk menggunakan penerimaan email. Khusus untuk penerimaan email, catatan MX juga diperlukan untuk menerbitkan catatan MX untuk menentukan domain email kustom. Catatan ini digunakan dalam pengaturan DNS domain Anda untuk mengizinkan SES menerima email untuk domain Anda. Pemberian izin diperlukan karena tindakan yang Anda pilih dalam aturan penerimaan tidak akan berfungsi kecuali jika Amazon SES memiliki izin untuk menggunakan layanan AWS masing-masing yang diperlukan untuk tindakan tersebut.

Tiga prasyarat yang diperlukan untuk menggunakan penerimaan email ini dijelaskan dalam topik berikut:

- Memverifikasi domain Anda untuk menerima email Amazon SES
- Menerbitkan catatan MX untuk penerimaan email Amazon SES
- Memberikan izin ke Amazon SES untuk menerima email

Memverifikasi domain Anda untuk menerima email Amazon SES

Seperti halnya domain yang ingin Anda gunakan untuk mengirim atau menerima email dengan Amazon SES, Anda harus terlebih dahulu membuktikan bahwa Anda memilikinya. Prosedur verifikasi mencakup memulai verifikasi domain dengan SES dan kemudian menerbitkan data DNS, baik CNAME atau TXT, ke penyedia DNS Anda tergantung pada metode verifikasi yang Anda gunakan.

Melalui konsol, Anda dapat memverifikasi domain Anda dengan baikEasy DKIMatauBawa DKIM Anda Sendiri (BYODKIM)dan dengan mudah menyalin catatan DNS mereka untuk dipublikasikan ke penyedia DNS Anda - bagaimana melakukannya dijelaskan dalamMembuat identitas domain. Opsional, Anda dapat menggunakan salah satu SES`VerifyDomainDkim`atau`VerifyDomainIdentity`API.

Anda dapat dengan mudah mengonfirmasi bahwa domain atau alamat email Anda diverifikasi dengan melihat statusnya diIdentitas terverifikasitabel di konsol SES atau dengan menggunakan SES`GetIdentityVerificationAttributes`atau`GetEmailIdentity`API.

Menerbitkan catatan MX untuk penerimaan email Amazon SES

Catatan penukar surat (catatan MX) adalah konfigurasi yang menentukan server surat mana yang dapat menerima email yang dikirim ke domain Anda.

Agar Amazon SES mengelola email masuk, Anda perlu menambahkan catatan MX ke konfigurasi DNS domain Anda. Catatan MX yang Anda buat merujuk ke titik akhir yang menerima email untuk Wilayah AWS tempat Anda menggunakan Amazon SES. Sebagai contoh, titik akhir untuk Wilayah US West (Oregon) adalah inbound-smtp.us-west-2.amazonaws.com. Untuk daftar lengkap titik akhir, lihat SESwilayah dan titik akhir.

> **Note**
>
> Titik akhir yang menerima email di Amazon SES bukan server email IMAP atau POP3. Anda tidak dapat menggunakan URL ini sebagai server email masuk di klien email.
> Jika Anda memerlukan solusi yang dapat mengirim dan menerima email dengan menggunakan klien email, pertimbangkan untuk menggunakan Amazon WorkMail.

Prosedur berikut mencakup langkah-langkah umum untuk membuat catatan MX. Prosedur khusus untuk membuat catatan MX bergantung pada penyedia DNS atau hosting Anda. Lihat dokumentasi penyedia untuk informasi tentang penambahan catatan MX ke konfigurasi DNS untuk domain Anda.

> **Note**
>
> Untuk menyelesaikan prosedur berikut, Anda harus dapat mengubah catatan DNS untuk domain Anda. Jika Anda tidak dapat mengakses data DNS untuk domain, atau merasa tidak nyaman melakukannya, kontak administrator sistem untuk mendapatkan bantuan.

Menambahkan catatan MX ke konfigurasi DNS untuk domain Anda

1. Masuk ke konsol manajemen untuk penyedia DNS Anda.
2. Buat catatan MX baru.
3. Untuk Nama data MX, masukkan domain Anda. Misalnya, jika Anda ingin Amazon SES mengelola email yang dikirim ke domain example.com, masukkan perintah berikut:

   ```
   example.com
   ```

> **Note**
>
> Beberapa penyedia DNS mengacu pada kolom Nama sebagai Host, Domain, atau Domain Surat.

4. Untuk Tipe, pilih MX.

> **Note**
>
> Beberapa penyedia DNS mengacu pada kolom Tipe sebagai Tipe Catatan atau nama yang sama.

5. Untuk Nilai, masukkan perintah berikut:

```
10 inbound-smtp.region.amazonaws.com
```

Dalam contoh sebelumnya, ganti *regionte* yang menerima email untuk AWS Wilayah yang Anda gunakan dengan Amazon SES. Sebagai contoh, jika Anda menggunakan Wilayah US East (N. Virginia), ganti *wilayah* dengan `us-east-1`. Untuk daftar lengkap dari titik akhir yang menerima email, lihat SESwilayah dan titik akhir.

> **Note**
>
> Konsol manajemen dari beberapa penyedia DNS termasuk kolom terpisah untuk Nilai catatan dan Prioritas catatan. Jika hal ini terjadi pada penyedia DNS Anda, masukkan `10` untuk nilai Prioritas, dan masukkan URL titik akhir surat masuk untuk Nilai.

Petunjuk untuk membuat catatan MX untuk berbagai penyedia

Prosedur untuk membuat catatan MX untuk domain bergantung pada penyedia DNS yang Anda gunakan. Bagian ini mencakup tautan ke dokumentasi untuk beberapa penyedia DNS umum. Daftar ini bukan daftar lengkap penyedia. Jika penyedia Anda tidak tercantum di bawah ini, Anda mungkin masih dapat menggunakannya dengan Amazon SES. Penyertaan dalam daftar ini bukan merupakan dukungan atau rekomendasi dari produk atau layanan perusahaan apa pun.

Nama Penyedia DNS/Hosting	Tautan Dokumentasi
Amazon Route 53	Membuat Catatan Menggunakan Konsol Amazon Route 53
GoDaddy	Tambahkan catatan MX (tautan eksternal)
DreamHost	Bagaimana cara mengubah catatan MX saya? (tautan eksternal)
Cloudflare	Mengatur catatan email (tautan eksternal)
HostGator	Mengubah catatan MX - Windows (tautan eksternal)
Namecheap	Bagaimana cara mengatur catatan MX yang diperlukan untuk layanan surat? (tautan eksternal)
Names.co.uk	Mengubah pengaturan DNS domain (tautan eksternal)
Wix	Menambahkan atau Memperbarui Catatan MX di Akun Wix Anda (tautan eksternal)

Memberikan izin ke Amazon SES untuk menerima email

Beberapa tugas yang dapat Anda lakukan saat menerima emailSES, seperti mengirim email ke bucket Amazon Simple Storage Service (Amazon S3) atau memanggil fungsi, memerlukan AWS Lambda izin khusus. Bagian ini mencakup kebijakan contoh untuk beberapa kasus penggunaan umum.

Topik di bagian ini:

- Menyiapkan izin IAM peran untuk aksi bucket Deliver to S3
- Berikan SES izin untuk menulis ke bucket S3
- Berikan SES izin untuk menggunakan AWS KMS kunci Anda
- Berikan SES izin untuk memanggil fungsi AWS Lambda

- Berikan SES izin untuk mempublikasikan ke SNS topik Amazon yang termasuk dalam AWS akun lain

Menyiapkan izin IAM peran untuk aksi bucket Deliver to S3

Poin-poin berikut berlaku untuk IAM peran ini:

- Itu hanya bisa digunakan untukMengirimkan ke tindakan bucket S3.
- Ini harus digunakan jika ingin menulis ke bucket S3 yang ada di wilayah yang SES the section called “Penerimaan email” tidak tersedia.

Jika ingin menulis ke bucket S3, Anda dapat memberikan IAM peran dengan izin untuk mengakses sumber daya yang relevan untuk file. Mengirimkan ke tindakan bucket S3 Anda juga perlu memberikan SES izin untuk mengambil peran itu untuk melakukan tindakan melalui kebijakan IAM kepercayaan seperti yang dijelaskan di bagian berikutnya.

Kebijakan izin ini harus ditempelkan ke editor kebijakan sebaris peran—lihat Mengirimkan ke tindakan bucket S3 dan ikuti langkah-langkah yang diberikan dalam item IAM peran. IAM (Contoh berikut juga menyertakan izin opsional jika Anda ingin menggunakan pemberitahuan SNS topik, atau kunci yang dikelola pelanggan dalam tindakan S3.)

```
{
    "Version": "2012-10-17",
    "Statement": [
        // Required: allows SES to write in the bucket
        {
            "Sid": "S3Access",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/*"
        },
        // Optional: use if an SNS topic is used in the S3 action
        {
            "Sid": "SNSAccess",
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "arn:aws:sns:region:111122223333:my-topic"
        },
        // Optional: use if a customer managed key is used in the S3 action
        {
```

```
            "Sid": "KMSAccess",
            "Effect": "Allow",
            "Action": "kms:GenerateDataKey*",
            "Resource": "arn:aws:kms:region::111122223333:key/key-id"
        }
    ]
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *my-bucket* dengan nama bucket S3 yang ingin Anda tulis.
- Ganti *region* dengan Wilayah AWS tempat Anda membuat aturan tanda terima.
- Ganti *111122223333* dengan ID AWS akun Anda.
- Ganti *my-topic* dengan nama SNS topik yang ingin Anda publikasikan notifikasi.
- Ganti *key-id* dengan ID KMS kunci Anda.

Kebijakan kepercayaan untuk peran tindakan IAM S3

Kebijakan kepercayaan berikut harus ditambahkan ke dalam Hubungan kepercayaan IAM peran untuk memungkinkan SES untuk mengambil peran itu.

> **Note**
>
> Penambahan manual kebijakan kepercayaan ini hanya diperlukan jika Anda tidak membuat IAM peran dari SES konsol menggunakan langkah-langkah yang diberikan dalam item IAMperan Mengirimkan ke tindakan bucket S3 alur kerja. Saat Anda membuat IAM peran dari konsol, kebijakan kepercayaan ini secara otomatis dibuat dan diterapkan ke peran yang membuat langkah ini tidak diperlukan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSESAssume",
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
```

```
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                  "AWS:SourceAccount":"111122223333",
                  "AWS:SourceArn": "arn:aws:ses:region:111122223333:receipt-rule-
set/rule_set_name:receipt-rule/receipt_rule_name"
                }
            }
        }
    ]
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *region* dengan Wilayah AWS tempat Anda membuat aturan tanda terima.
- Ganti *111122223333* dengan ID AWS akun Anda.
- Ganti *rule_set_name* dengan nama set aturan yang berisi aturan tanda terima yang berisi tindakan bucket pengiriman ke Amazon S3.
- Ganti *receipt_rule_name* dengan nama aturan tanda terima yang berisi aksi bucket pengiriman ke Amazon S3.

Berikan SES izin untuk menulis ke bucket S3

Jika Anda menerapkan kebijakan berikut ke bucket S3, kebijakan tersebut memberikan SES izin untuk menulis ke bucket tersebut selama ada di wilayah di mana penerimaan SES Email tersedia—jika Anda ingin menulis ke bucket di luar wilayah penerima Email, lihat. Menyiapkan izin IAM peran untuk aksi bucket Deliver to S3 Untuk informasi selengkapnya tentang membuat aturan penerimaan yang mentransfer email masuk ke Amazon S3, lihat Mengirimkan ke tindakan bucket S3.

Untuk informasi selengkapnya tentang melampirkan kebijakan ke bucket S3, lihat Menggunakan Kebijakan Bucket dan Kebijakan Pengguna di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

```
{
  "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:SourceAccount":"111122223333",
          "AWS:SourceArn": "arn:aws:ses:region:111122223333:receipt-rule-
set/rule_set_name:receipt-rule/receipt_rule_name"
        }
      }
    }
  ]
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *myBucket* dengan nama bucket S3 yang ingin Anda tulis.
- Ganti *region* dengan AWS Wilayah tempat Anda membuat aturan tanda terima.
- Ganti *111122223333* dengan ID AWS akun Anda.
- Ganti *rule_set_name* dengan nama set aturan yang berisi aturan tanda terima yang berisi tindakan bucket pengiriman ke Amazon S3.
- Ganti *receipt_rule_name* dengan nama aturan tanda terima yang berisi aksi bucket pengiriman ke Amazon S3.

Berikan SES izin untuk menggunakan AWS KMS kunci Anda

SESUntuk mengenkripsi email Anda, itu harus memiliki izin untuk menggunakan AWS KMS kunci yang Anda tentukan saat Anda mengatur aturan tanda terima Anda. Anda dapat menggunakan KMS kunci default (aws/ses) di akun Anda, atau menggunakan kunci terkelola pelanggan yang Anda buat. Jika Anda menggunakan KMS kunci default, Anda tidak perlu melakukan langkah tambahan untuk memberikan SES izin untuk menggunakannya. Jika Anda menggunakan kunci yang dikelola pelanggan, Anda harus memberikan SES izin untuk menggunakannya dengan menambahkan pernyataan ke kebijakan kunci.

Gunakan pernyataan kebijakan berikut sebagai kebijakan utama SES untuk mengizinkan penggunaan kunci yang dikelola pelanggan Anda saat menerima email di domain Anda.

```
{
  "Sid": "AllowSESToEncryptMessagesBelongingToThisAccount",
  "Effect": "Allow",
  "Principal": {
    "Service":"ses.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition":{
        "StringEquals":{
          "AWS:SourceAccount":"111122223333",
          "AWS:SourceArn": "arn:aws:ses:region:111122223333:receipt-rule-
set/rule_set_name:receipt-rule/receipt_rule_name"
        }
      }
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *region* dengan AWS Wilayah tempat Anda membuat aturan tanda terima.
- Ganti *111122223333* dengan ID AWS akun Anda.
- Ganti *rule_set_name* dengan nama set aturan yang berisi aturan tanda terima yang Anda kaitkan dengan penerimaan email.
- Ganti *receipt_rule_name* dengan nama aturan tanda terima yang Anda kaitkan dengan penerimaan email.

Jika Anda menggunakannya AWS KMS untuk mengirim pesan terenkripsi ke bucket S3 dengan enkripsi sisi server diaktifkan, Anda perlu menambahkan tindakan kebijakan,. `"kms:Decrypt"` Menggunakan contoh sebelumnya, menambahkan tindakan ini ke kebijakan Anda akan muncul sebagai berikut:

```
{
  "Sid": "AllowSESToEncryptMessagesBelongingToThisAccount",
  "Effect": "Allow",
  "Principal": {
    "Service":"ses.amazonaws.com"
  },
  "Action": [
```

```
    "kms:Decrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition":{
        "StringEquals":{
          "AWS:SourceAccount":"111122223333",
          "AWS:SourceArn": "arn:aws:ses:region:111122223333:receipt-rule-
set/rule_set_name:receipt-rule/receipt_rule_name"
        }
      }
}
```

Untuk informasi selengkapnya tentang melampirkan kebijakan ke AWS KMS kunci, lihat Menggunakan Kebijakan Utama AWS KMS di Panduan AWS Key Management Service Pengembang.

Berikan SES izin untuk memanggil fungsi AWS Lambda

SESUntuk mengaktifkan memanggil AWS Lambda fungsi, Anda dapat memilih fungsi saat membuat aturan tanda terima di SES konsol. Ketika Anda melakukannya, SES secara otomatis menambahkan izin yang diperlukan ke fungsi.

Atau, Anda dapat menggunakan `AddPermission` operasi di AWS Lambda API untuk melampirkan kebijakan ke fungsi. Panggilan berikut untuk `AddPermission` API memberikan SES izin untuk menjalankan fungsi Lambda Anda. Untuk informasi selengkapnya tentang melampirkan kebijakan fungsi Lambda, lihat Izin AWS Lambda dalam Panduan Developer AWS Lambda .

```
{
  "Action": "lambda:InvokeFunction",
  "Principal": "ses.amazonaws.com",
  "SourceAccount": "111122223333",
  "SourceArn": "arn:aws:ses:region:111122223333:receipt-rule-set/rule_set_name:receipt-
rule/receipt_rule_name"
  "StatementId": "GiveSESPermissionToInvokeFunction"
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *region* dengan AWS Wilayah tempat Anda membuat aturan tanda terima.
- Ganti *111122223333* dengan ID AWS akun Anda.

- Ganti *rule_set_name* dengan nama set aturan yang berisi aturan tanda terima tempat Anda membuat fungsi Lambda.
- Ganti *receipt_rule_name* dengan nama aturan tanda terima yang berisi fungsi Lambda Anda.

Berikan SES izin untuk mempublikasikan ke SNS topik Amazon yang termasuk dalam AWS akun lain

Untuk mempublikasikan pemberitahuan ke topik di AWS akun terpisah, Anda harus melampirkan kebijakan ke SNS topik Amazon. SNSTopik harus berada di Wilayah yang sama dengan domain dan aturan tanda terima yang ditetapkan.

Kebijakan berikut memberikan SES izin untuk mempublikasikan ke SNS topik Amazon di AWS akun terpisah.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Principal":{
        "Service":"ses.amazonaws.com"
      },
      "Action":"SNS:Publish",
      "Resource":"arn:aws:sns:topic_region:sns_topic_account_id:topic_name",
      "Condition":{
        "StringEquals":{
          "AWS:SourceAccount":"aws_account_id",
          "AWS:SourceArn": "arn:aws:ses:receipt_region:aws_account_id:receipt-rule-
set/rule_set_name:receipt-rule/receipt_rule_name"
        }
      }
    }
  ]
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *topic_region* dengan Wilayah AWS SNS topik Amazon dibuat.
- Ganti *sns_topic_account_id* dengan ID AWS akun yang memiliki SNS topik Amazon.
- Ganti *topic_name* dengan nama SNS topik Amazon yang ingin Anda publikasikan notifikasi.

- Ganti *aws_account_id* dengan ID AWS akun yang dikonfigurasi untuk menerima email.
- Ganti *receipt_region* dengan Wilayah AWS tempat Anda membuat aturan tanda terima.
- Ganti *rule_set_name* dengan nama set aturan yang berisi aturan tanda terima tempat Anda membuat tindakan SNS topik publikasi ke Amazon.
- Ganti *receipt_rule_name* dengan nama aturan tanda terima yang berisi aksi SNS topik publikasi ke Amazon.

Jika SNS topik Amazon Anda menggunakan AWS KMS enkripsi sisi server, Anda harus menambahkan izin ke kebijakan utama. AWS KMS Anda dapat menambahkan izin dengan melampirkan kebijakan berikut ke kebijakan AWS KMS utama:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSESToUseKMSKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

Panduan konsol penerimaan email Amazon SES

Bagian ini menjelaskan tentang wizard konsol penerima email yang digunakan untuk mengonfigurasikan aturan penerimaan dan Filter alamat IP untuk mengelola penerimaan email Anda. Sebelum menggunakan wizard konsol tersebut, penting bagi Anda untuk membaca Konsep penerimaan email & kasus penggunaan untuk memahami konsep cara kerja penerimaan email dan Menyiapkan penerimaan email untuk memastikan Anda telah menyelesaikan prasyarat penyiapan.

Wizard konsol tersebut digunakan untuk mengonfigurasi aturan tanda terima dan filter alamat IP yang dijelaskan pada contoh berikut:

- Membuat aturan penerimaan pada panduan konsol
- Buat panduan konsol filter alamat IP

Membuat aturan penerimaan pada panduan konsol

Bagian ini akan memandu Anda membuat dan mendefinisikan aturan tanda terima menggunakan SES konsol Amazon. Poin-poin penting untuk memahami cara kerja aturan penerimaan adalah:

- Set aturan berisi set aturan penerimaan yang berurutan; Aturan penerimaan berisi set tindakan yang berurutan.
- Aturan tanda terima memberi tahu Amazon SES cara menangani email masuk dengan menjalankan daftar tindakan yang diurutkan yang Anda tentukan.
- Daftar tindakan yang diurutkan ini secara opsional dapat dibuat bergantung pada pencocokan pertama kondisi penerima; jika tidak ditentukan, tindakan akan diterapkan ke semua identitas milik domain terverifikasi Anda.
- Aturan penerimaan dibuat dan ditentukan dalam kontainer yang disebut set aturan - sementara Anda dapat membuat beberapa set aturan, hanya satu yang dapat aktif pada satu waktu.
- Aturan penerimaan dalam set aturan aktif dijalankan sesuai urutan yang Anda tentukan.
- Sebelum Anda membuat aturan penerimaan, Anda harus membuat Set aturan untuk menyimpan aturan tersebut.

Secara opsional, Anda dapat menggunakan aturan `CreateReceiptRuleSet` API untuk membuat tanda terima kosong, seperti yang dijelaskan dalam APIReferensi Layanan Email Sederhana Amazon. Kemudian, Anda dapat menggunakan SES konsol Amazon atau `CreateReceiptRule` API untuk menambahkan aturan tanda terima ke dalamnya.

Sebelum melanjutkan panduan, pastikan Anda telah memenuhi semua prasyarat yang diperlukan untuk menggunakan penerimaan email berbasis penerima. Juga

Prasyarat

Prasyarat berikut harus dipenuhi sebelum melanjutkan penyiapan kontrol email berbasis penerima menggunakan aturan penerimaan:

1. Pastikan titik akhir Anda berada di Wilayah AWS tempat Amazon SES mendukung penerimaan email. Tabel titik akhir Penerimaan Email dalam Referensi Umum AWS daftar titik akhir penerima email untuk semua Wilayah AWS tempat SES mendukung penerimaan email.
2. Anda harus terlebih dahulu membuat dan memverifikasi identitas domain di AmazonSES.
3. Selanjutnya, Anda perlu menentukan server email mana yang dapat menerima email untuk domain Anda dengan menerbitkan catatan MX ke DNS pengaturan domain Anda. (Catatan MX harus merujuk ke SES titik akhir Amazon yang menerima email untuk AWS Wilayah tempat Anda menggunakan AmazonSES.)
4. Terakhir, Anda perlu memberi SES izin Amazon untuk mengakses AWS sumber daya lain untuk menjalankan tindakan aturan penerimaan.

Membuat set aturan dan aturan penerimaan

Panduan ini terlebih dahulu dimulai dengan membuat set aturan untuk menyimpan aturan Anda dan dilanjutkan ke wizard Buat aturan untuk membuat, menentukan, dan mengurutkan aturan penerimaan Anda. Wizard berisi empat layar untuk menentukan pengaturan aturan, menambahkan syarat penerima, menambahkan tindakan, dan meninjau semua pengaturan Anda.

Untuk membuat set aturan dan aturan penerimaan menggunakan konsol tersebut.

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di Konfigurasi, pilih Penerimaan email.

> **Note**
>
> Penerimaan email tidak akan terlihat di panel navigasi kiri SES konsol jika akun Anda berada di Wilayah AWS tempat yang SES tidak mendukung penerimaan email. Lihat item pertama yang tercantum dithe section called “Prasyarat”.

3. Di bawah tab Set aturan tanda terima di panel Penerima email, pilih Buat set aturan.
4. Masukkan nama yang unik untuk set aturan Anda dan pilih Buat set aturan.
5. Pilih Buat aturan dan pilihan ini akan membuka wizard Buat aturan.
6. Pada halaman Tentukan pengaturan aturan, di Detail aturan penerimaan, masukkan Nama aturan.

7. Untuk Status, hanya kosongkan kotak centang Diaktifkan jika Anda tidak SES ingin Amazon menjalankan aturan ini setelah pembuatan; jika tidak, biarkan opsi ini dipilih.
8. (Opsional) Di bawah Opsi keamanan dan perlindungan, untuk Transport Layer Security (TLS), pilih Diperlukan jika Anda SES ingin Amazon menolak pesan masuk yang tidak dikirim melalui sambungan aman.
9. (Opsional) Untuk pemindaian Spam dan virus, pilih Diaktifkan jika Anda SES ingin Amazon memindai pesan masuk untuk spam dan virus.
10. Untuk melanjutkan ke langkah berikutnya, pilih Selanjutnya.
11. (Opsional) Pada Halaman Tambah syarat penerima, untuk menentukan satu kondisi penerima atau lebih gunakan prosedur berikut ini. Untuk setiap aturan penerimaan Anda dapat memiliki maksimal 100 syarat penerima.

    a. Di Syarat penerima, pilih Menambahkan kondisi penerima baru untuk menentukan alamat email penerima atau domain yang ingin Anda terapkan aturan penerimaan. Alamat pengguna@contoh.com pada tabel berikut menunjukkan cara menentukan kondisi penerima.

Jika Anda ingin...	Tentukan penerima berikut...	Catatan
Mencocokkan alamat email tertentu.	user@example.com	Juga cocok dengan variasi alamat yang berisi label (seperti pengguna +123@contoh.com dan pengguna+xyz@conto h.com). Namun, jika Anda menentukan alamat yang berisi label, hanya alamat tersebut yang cocok.
Cocokkan semua alamat dalam domain, tetapi bukan alamat dalam subdomain nya.	example.com	
Cocokkan semua alamat dalam subdomain tertentu,	subdomain.example.com	

Jika Anda ingin...	Tentukan penerima berikut...	Catatan
tetapi bukan alamat dalam domain induk.		
Cocokkan semua alamat dalam semua subdomain, tetapi bukan alamat dalam domain induk.	.example.com	Perhatikan tanda titik (.) sebelum nama domain.
Cocokkan semua alamat dalam domain, dan semua alamat dalam semua subdomainnya.	example.com .example.com	Buat dua penerima terpisah: satu dengan nama domain, dan satu dengan tanda titik diikuti dengan nama domain.
Cocokkan semua penerima di semua domain yang terverifikasi	[Tidak ada]	Biarkan bidang penerima kosong.

Important

Jika beberapa SES akun Amazon menerima email pada domain umum (misalnya, jika beberapa tim di perusahaan yang sama masing-masing memiliki SES akun Amazon terpisah), Amazon SES memproses semua aturan tanda terima yang cocok secara bersamaan untuk masing-masing akun tersebut. Perilaku ini dapat mengakibatkan situasi ketika satu akun menghasilkan pentalan, sementara akun lain menerima email.

Sebaiknya Anda berkoordinasi dengan tim lain di organisasi Anda yang menggunakan Amazon SES untuk memastikan bahwa setiap akun menggunakan aturan tanda terima unik, dan aturan tersebut tidak tumpang tindih. Dalam situasi ini, hal terbaik adalah mengonfigurasi aturan penerimaan Anda dengan hanya menggunakan alamat email atau subdomain yang unik untuk grup atau tim Anda.

b. Ulangi langkah ini untuk setiap syarat penerima yang ingin Anda tambahkan. Setelah Anda selesai menambahkan syarat penerima, pilih Selanjutnya.

12. Pada halaman Tambahkan tindakan, untuk menambahkan satu tindakan ke aturan penerimaan atau lebih gunakan prosedur berikut.

    a. Buka menu Tambah aksi baru, dan kemudian pilih salah satu dari tipe tindakan berikut:

    - Tambahkan header - Tindakan ini menambahkan header kustom ke email yang diterima.
    - Mengembalikan respons pentalan - Tindakan ini menolak email yang diterima dengan mengembalikan respons pentalan ke pengirim.
    - Panggil fungsi Lambda- Tindakan ini memanggil kode Anda melalui fungsi AWS Lambda.
    - Mengirimkan ke bucket S3 - Tindakan ini menyimpan email yang diterima dalam bucket Amazon Simple Storage Service (S3).
    - Terbitkan ke topik Amazon SNS- Tindakan ini menerbitkan email lengkap ke topik Amazon Simple Notification Service (SNS).
    - Hentikan set aturan - Tindakan ini mengakhiri evaluasi dari set aturan penerimaan.
    - Integrasi dengan Amazon WorkMail- Tindakan ini terintegrasi dengan Amazon WorkMail.

    Untuk informasi lebih lanjut tentang setiap tindakan ini, lihat Pilihan tindakan.

    b. Ulangi langkah ini untuk setiap tindakan yang ingin Anda tetapkan. Jika Anda memiliki beberapa tindakan yang ditetapkan, Anda dapat menyusun ulang tindakan tersebut dengan menggunakan panah atas/bawah dalam kontainer tindakan. Pilih Selanjutnya untuk melanjutkan ke halaman Tinjau.

13. Pada halaman Tinjau, tinjau pengaturan dan tindakan aturan. Jika Anda perlu melakukan perubahan, pilih opsi Edit, atau gunakan bagian navigasi di sisi kiri halaman untuk pergi secara langsung menuju langkah yang berisi konten yang ingin Anda edit. Secara opsional Anda dapat membuat perubahan pada urutan tindakan yang tercantum dalam tabel Tindakan dari halaman Tinjau dengan menggunakan panah atas/bawah di kolom Urutkan Ulang.
14. Saat Anda siap untuk melanjutkan, pilih Buat aturan.
15. Pada halaman konfirmasi untuk set aturan, pilih Set as active jika Anda ingin segera menerapkan aturan yang ditetapkan.

Modifikasi aturan setelah pembuatan

Setelah membuat set aturan, Anda dapat mengedit kedua set aturan serta aturan penerimaan yang disimpannya. Tidak hanya dapat diedit, tetapi ada juga opsi untuk menduplikasi aturan yang ditetapkan atau aturannya sehingga yang baru dapat dibuat dengan cepat. Daftar berikut menunjukkan modifikasi yang tersedia untuk set aturan dan aturan penerimaan:

- Set aturan terdaftar dengan namanya, status dan tanggal pembuatannya. Pilihan modifikasi untuk set aturan adalah:
  - Tombol alihkan Tetapkan sebagai aktif/tidak aktif akan beralih di antara pengaturan status.
  - Tombol Duplikasi akan menyalin set aturan. Anda akan diminta untuk memberikan nama yang unik.
  - Tombol Hapus akan menghapus set aturan. Anda akan diminta untuk mengonfirmasi tindakan yang tidak dapat dipulihkan ini.
- Aturan penerimaan tercantum dengan nama, status, keamanan, dan urutan mereka. Pilihan modifikasi untuk aturan penerimaan adalah:
  - Panah naik/bawah untuk menyusun ulang eksekusi aturan dalam set aturan.
  - Tombol Duplikasi akan membuat salinan aturan yang dipilih. Anda akan diminta untuk memberikan nama yang unik.
  - Tombol Edit akan membuka aturan yang dipilih sehingga salah satu parameternya seperti pengaturan aturan, kondisi penerima, dan tindakan dapat diedit.
  - Tombol Hapus akan menghapus aturan yang dipilih. Anda akan diminta untuk mengonfirmasi tindakan yang tidak dapat dipulihkan ini.
  - Tombol Buat aturan akan mengizinkan Anda untuk membuat dan menambahkan aturan baru untuk set aturan saat ini.

Pilihan tindakan

Setiap aturan penerimaan untuk penerimaan email Amazon SES berisi daftar tindakan yang berurutan. Bagian ini menjelaskan opsi spesifik untuk setiap tipe tindakan.

Tipe tindakan adalah sebagai berikut:

- Tambahkan tindakan header
- Mengembalikan tindakan respons pentalan
- Panggil tindakan Fungsi Lambda

- Mengirimkan ke tindakan bucket S3
- Terbitkan ke topik tindakan Amazon SNS
- Hentikan tindakan set aturan
- Integrasi dengan Amazon WorkMail aksi

Tambahkan tindakan header

Tindakan Tambah Header menambahkan header kustom ke email yang diterima. Anda biasanya menggunakan tindakan ini hanya jika dikombinasikan dengan tindakan lain. Tindakan ini memiliki opsi berikut.

- Nama header—Nama header yang akan ditambahkan. Nama header harus berkisar antara 1 hingga 50 karakter, inklusif, dan hanya terdiri dari karakter alfanumerik (a-z, A-Z, 0-9) dan tanda hubung.
- Nilai header—Nilai header yang akan ditambahkan. Nilai header harus kurang dari 2048 karakter, dan tidak boleh berisi karakter baris baru ("\r" atau "\n").

Mengembalikan tindakan respons pentalan

Tindakan Pentalan menolak email dengan mengembalikan respons pentalan ke pengirim dan memberitahu Anda melalui Amazon SNS secara opsional. Tindakan ini memiliki opsi berikut.

- Kode Balasan SMTP— Kode balasan SMTP, seperti yang ditentukan oleh RFC 5321.
- Kode Status SMTP—Kode status disempurnakan SMTP, seperti yang ditentukan oleh RFC 3463.
- Pesan—Teks yang dapat dibaca manusia untuk disertakan dalam email pentalan.
- Pengirim Balasan—Alamat email dari pengirim email yang terpental. Ini adalah alamat tempat mengirim email pentalan. Alamat tersebut harus diverifikasi dengan Amazon SES.
- Topik SNS—Nama atau ARN dari topik Amazon SNS yang secara opsional akan memberi tahu ketika email pentalan dikirim. Contoh topik Amazon SNS adalaharn:aws:sns:us-east-1:123456789012:mytopic. Anda juga dapat membuat topik Amazon SNS saat mengatur tindakan dengan memilih Buat Topik SNS. Untuk informasi lebih lanjut tentang topik Amazon SNS, lihat Panduan Developer Amazon Simple Notification Service.

> **Note**
>
> Topik Amazon SNS yang Anda pilih harus samaAWSWilayah sebagai titik akhir Amazon SES yang Anda gunakan untuk menerima email.

Anda dapat mengetikkan nilai Anda sendiri untuk bidang ini, atau Anda dapat memilih templat yang mengisi bidang Kode Balasan SMTP, Kode Status SMTP, dan Pesan dengan nilai berdasarkan alasan pentalan. Templat berikut ini tersedia:

- Kotak Surat Tidak Ada— Kode Balasan SMTP = 550, SMTP Status Code = 5.1.1
- Pesan Terlalu Besar— Kode Balasan SMTP = 552, Kode Status SMTP = 5.3.4
- Kotak Surat Penuh— Kode Balasan SMTP = 552, Kode Status SMTP = 5.2.2
- Konten Pesan Ditolak— Kode Balasan SMTP = 500, Kode Status SMTP = 5.6.1
- Kegagalan yang Tidak Diketahui— Kode Balasan SMTP = 554, Kode Status SMTP = 5.0.0
- Kegagalan Sementara— Kode Balasan SMTP = 450, Kode Status SMTP = 4.0.0

Untuk kode pentalan tambahan yang mungkin Anda gunakan dengan mengetikkan nilai kustom di bidang, lihat RFC 3463.

Panggil tindakan Fungsi Lambda

Tindakan Lambda memanggil kode Anda melalui fungsi Lambda dan memberitahu Anda melalui Amazon SNS secara opsional. Tindakan ini memiliki opsi dan persyaratan berikut.

Opsi

- Fungsi Lambda—ARN fungsi Lambda. Contoh dari fungsi Lambda ARN adalaharn:aws:lambda:us-east-1:account-id:function:myfunction.
- Tipe pemanggilan—Tipe pemanggilan fungsi Lambda. Jenis penangananRequestResponseberarti bahwa pelaksanaan hasil fungsi dalam respon langsung. Jenis penangananPeristiwaberarti bahwa fungsi ini dipanggil asynchronously. Sebaiknya Anda menggunakanPeristiwaJenis penanganan kecuali eksekusi sinkron diperlukan untuk kasus penggunaan Anda.

  Ada batas waktu 30 detik pada pemanggilan RequestResponse.

Untuk informasi selengkapnya, lihatMemanggil fungsi LambdadiAWS LambdaPanduan Pengembang.

- Topik SNS—Nama atau ARN dari topik Amazon SNS untuk memberi tahu ketika fungsi Lambda yang ditentukan dipicu. Contoh topik Amazon SNS ARN adalaharn:aws:sns:us-east-1:123456789012:mytopic. Untuk informasi lebih lanjut, lihat Membuat topik Amazon SNS dalam Panduan Developer Amazon Simple Notification Service.

Persyaratan

- Fungsi Lambda yang Anda pilih harus samaAWSWilayah sebagai titik akhir Amazon SES yang Anda gunakan untuk menerima email.
- Topik Amazon SNS yang Anda pilih harus dalam hal yang samaAWSWilayah sebagai titik akhir Amazon SES yang Anda gunakan untuk menerima email.

Menulis fungsi Lambda Anda

Untuk memproses email Anda, fungsi Lambda Anda dapat dipanggil secara tidak sinkron (yaitu, menggunakan tipe pemanggilan `Event`). Objek peristiwa yang diteruskan ke fungsi Lambda Anda akan berisi metadata yang berkaitan dengan peristiwa email masuk. Anda juga dapat menggunakan metadata untuk mengakses konten pesan dari bucket Amazon S3 Anda.

Jika Anda benar-benar ingin mengontrol arus email, fungsi Lambda Anda harus dipanggil secara sinkron (yaitu, menggunakan tipe pemanggilan `RequestResponse`) dan fungsi Lambda Anda harus memanggil metode `callback` dengan dua argumen: argumen pertama adalah `null`, dan argumen kedua adalah properti `disposition` yang diatur ke `STOP_RULE`, `STOP_RULE_SET`, atau `CONTINUE`. Jika argumen kedua adalah `null` atau tidak memiliki properti `disposition` yang valid, arus email akan berlanjut dan tindakan serta aturan diproses lebih lanjut, yang sama dengan `CONTINUE`.

Misalnya, Anda dapat menghentikan set aturan penerimaan dengan menulis baris berikut di akhir kode fungsi Lambda Anda:

```
callback( null, { "disposition" : "STOP_RULE_SET" });
```

Untuk sampel kode AWS Lambda, lihat Contoh fungsi Lambda. Untuk contoh kasus penggunaan tingkat tinggi, lihat Contoh kasus penggunaan.

Format input

Amazon SES meneruskan informasi ke fungsi Lambda dalam format JSON. Objek tingkat atas berisi array `Records`, yang diisi dengan properti `eventSource`, `eventVersion`, dan `ses`. Objek `ses` berisi objek `receipt` dan `mail`, dalam format yang sama persis seperti di notifikasi Amazon SNS yang telah dijelaskan di Isi notifikasi.

Data yang diteruskan Amazon SES ke Lambda mencakup metadata tentang pesan, serta beberapa header email. Namun, data tersebut tidak berisi isi pesan.

Berikut ini adalah tampilan tingkat tinggi dari struktur input yang diberikan Amazon SES ke fungsi Lambda.

```
{
   "Records": [
      {
        "eventSource": "aws:ses",
        "eventVersion": "1.0",
        "ses": {
           "receipt": {
               <same contents as SNS notification>
            },
           "mail": {
               <same contents as SNS notification>
           }
         }
     }
   ]
}
```

Nilai pengembalian

Fungsi Lambda Anda dapat mengendalikan arus surat dengan mengembalikan salah satu nilai berikut:

- `STOP_RULE`—Saat ini tidak ada tindakan lebih lanjut dalam aturan penerimaan yang akan diproses, tetapi aturan penerimaan dapat diproses lebih lanjut.
- `STOP_RULE_SET`—Tidak ada tindakan atau aturan penerimaan akan diproses lebih lanjut.
- `CONTINUE` atau nilai tidak valid lainnya— Hal ini berarti tindakan dan aturan penerimaan dapat diproses lebih lanjut.

Topik berikut membahas contoh peristiwa email masuk, contoh kasus penggunaan tingkat tinggi, dan contoh kode AWS Lambda:

- Contoh kasus penggunaan
- Contoh fungsi Lambda

Contoh kasus penggunaan

Contoh berikut menguraikan beberapa aturan yang mungkin Anda atur untuk menggunakan hasil fungsi Lambda untuk mengontrol aliran surat. Untuk tujuan demonstrasi, banyak dari contoh ini menggunakan tindakan S3 sebagai hasilnya.

Kasus penggunaan 1: Menghapus spam di semua domain

Contoh ini menunjukkan aturan global yang menghapus spam di semua domain Anda. Aturan 2 dan 3 disertakan untuk menunjukkan bahwa Anda dapat menerapkan aturan khusus domain setelah spam dihapus di semua domain.

Aturan 1

Daftar penerima:Kosong. Oleh karena itu, aturan ini akan berlaku untuk semua penerima di semua domain terverifikasi Anda.

Tindakan

1. Tindakan Lambda (sinkron) yang mengembalikan `STOP_RULE_SET` jika email tersebut adalah spam. Jika tidak, ia mengembalikan `CONTINUE`. Lihat contoh fungsi Lambda untuk menghapus spam di Contoh fungsi Lambda.

Aturan 2

Daftar penerima: example1.com

Tindakan

1. Setiap tindakan.

Aturan 3

Daftar penerima: contoh2.com

Tindakan

1. Setiap tindakan.

Kasus penggunaan 2: Memantulkan spam di semua domain

Contoh ini menunjukkan aturan global yang mementalkan spam di semua domain Anda. Aturan 2 dan 3 disertakan untuk menunjukkan bahwa Anda dapat menerapkan aturan khusus domain setelah spam dipentalkan di semua domain.

Aturan 1

Daftar penerima:Kosong. Oleh karena itu, aturan ini akan berlaku untuk semua penerima di semua domain terverifikasi Anda.

Tindakan

1. Tindakan Lambda (sinkron) yang mengembalikan `CONTINUE` jika email tersebut adalah spam. Jika tidak, ia mengembalikan `STOP_RULE`.
2. Tindakan pentalan ("500 5.6.1. Isi pesan ditolak").
3. Hentikan tindakan.

Aturan 2

Daftar penerima: contoh1.com

Tindakan

1. Setiap tindakan

Aturan 3

Daftar penerima: contoh2.com

Tindakan

1. Setiap tindakan

Kasus penggunaan 3: Menerapkan aturan yang paling spesifik

Contoh ini menunjukkan bagaimana Anda dapat menggunakan tindakan Berhenti untuk mencegah email diproses oleh beberapa aturan. Pada contoh ini, Anda memiliki satu aturan untuk alamat tertentu, dan aturan lain untuk semua alamat email di domain tersebut. Dengan menggunakan tindakan Berhenti, pesan yang sesuai dengan aturan untuk alamat email tertentu tidak diproses oleh aturan umum yang berlaku untuk domain.

Aturan 1

Daftar penerima: pengguna@contoh.com

Tindakan

1. Tindakan Lambda (asinkron).
2. Hentikan tindakan.

Aturan 2

Daftar penerima: contoh.com

Tindakan

1. Setiap tindakan.

Kasus penggunaan 4: Mencatat peristiwa email ke CloudWatch

Contoh ini menunjukkan cara menyimpan log audit semua email melalui sistem Anda sebelum menyimpan email tersebut ke Amazon SES.

Aturan 1

Daftar penerima: contoh.com

Tindakan

1. Tindakan Lambda (tidak sinkron) yang menulis objek peristiwa ke log CloudWatch. Contoh fungsi Lambda dalam log Contoh fungsi Lambda ke CloudWatch.
2. Tindakan S3.

Kasus penggunaan 5: Menghapus email yang gagal DKIM

Contoh ini menunjukkan bagaimana Anda dapat menyimpan semua email masuk ke bucket Amazon S3, tetapi hanya mengirimkan email yang masuk ke alamat email tertentu, dan lolos DKIM, ke aplikasi email otomatis Anda.

Aturan 1

Daftar penerima: contoh.com

Tindakan

1. Tindakan S3.
2. Tindakan Lambda (sinkron) yang mengembalikan `STOP_RULE_SET` jika pesan menggagalkan DKIM. Jika tidak, ia mengembalikan `CONTINUE`.

Aturan 2

Daftar penerima: support@contoh.com

Tindakan

1. Tindakan Lambda (asinkron) yang memicu aplikasi otomatis.

Kasus penggunaan 6: Menyaring email berdasarkan baris subjek

Contoh ini menunjukkan bagaimana Anda dapat menghapus semua email masuk domain yang berisi kata "diskon" di baris subjek, lalu memproses email yang ditujukan untuk sistem otomatis satu arah, dan memproses email yang ditujukan ke semua penerima lain di domain dengan cara berbeda.

Aturan 1

Daftar penerima: contoh.com

Tindakan

1. Tindakan Lambda (sinkron) yang mengembalikan `STOP_RULE_SET` jika baris subjek berisi kata "diskon". Jika tidak, ia mengembalikan `CONTINUE`.

Aturan 2

Daftar penerima: support@contoh.com

Tindakan

1. Tindakan S3 dengan bucket 1.
2. Tindakan Lambda (tidak sinkron) yang memicu aplikasi otomatis.
3. Hentikan tindakan.

Aturan 3

Daftar penerima: contoh.com

Tindakan

1. Tindakan S3 dengan bucket 2.
2. Tindakan Lambda (tidak sinkron) yang memproses email untuk domain lainnya.

Contoh fungsi Lambda

Topik ini berisi contoh fungsi Lambda yang mengendalikan aliran email.

Contoh 1: Jatuhkan spam

Contoh ini menghentikan pemrosesan pesan yang memiliki setidaknya satu indikator spam.

```
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);
    }
};
```

Contoh 2: Lanjutkan jika header tertentu ditemukan

Contoh ini terus memproses aturan saat ini hanya jika email berisi nilai header tertentu.

```
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'});
};
```

Contoh 3: Mengambil email dari Amazon S3

Contoh ini mendapatkan email mentah dari Amazon S3 dan memprosesnya.

> **Note**
>
> Anda harus terlebih dahulu menulis email ke Amazon S3 menggunakan Tindakan S3.

```
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);
            }
        });
};
```

Contoh 4: Pentalkan pesan yang menggagalkan autentikasi DMARC

Contoh ini mengirimkan pesan pentalan jika email yang masuk gagal autentikasi DMARC.

> **Note**
>
> Saat menggunakan contoh ini, atur nilai variabel lingkungan `emailDomain` menjadi domain penerima email Anda.

```
'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();
    }
};
```

Mengirimkan ke tindakan bucket S3

Tindakan bucket Deliver to S3 mengirimkan email ke bucket S3 dan secara opsional dapat memberi tahu Anda dan banyak lagi. SNS Tindakan ini memiliki opsi berikut.

- Bucket S3 — Nama bucket S3 untuk menyimpan email yang diterima. Anda juga dapat membuat bucket S3 baru saat menyiapkan action dengan memilih Create S3 Bucket. Amazon SES memberi Anda email mentah yang tidak dimodifikasi, yang biasanya dalam format Multipurpose Internet Mail Extensions (MIME). Untuk informasi lebih lanjut tentang MIME format, lihat RFC2045.

  Important

  - Bucket Amazon S3 harus ada di wilayah yang tersedia; jika tidak, Anda harus menggunakan opsi IAM peran yang dijelaskan di bawah ini. SES the section called “Penerimaan email”
  - Saat Anda menyimpan email ke bucket S3, ukuran email maksimum default (termasuk header) adalah 40 MB.
  - SEStidak mendukung aturan penerimaan yang mengunggah ke bucket S3 yang diaktifkan dengan kunci objek yang dikonfigurasi dengan periode retensi default.
  - Jika menerapkan enkripsi pada bucket S3 Anda dengan menentukan KMS kunci Anda sendiri, pastikan untuk menggunakan KMS kunci yang sepenuhnya memenuhi syaratARN, dan bukan alias KMS kunci; menggunakan alias dapat mengakibatkan data

dienkripsi dengan KMS kunci milik pemohon, dan bukan administrator bucket. Lihat Menggunakan enkripsi untuk operasi lintas akun.

- Object key prefix — Sebuah awalan nama kunci opsional untuk digunakan dalam bucket S3. Awalan nama kunci memungkinkan Anda mengatur bucket S3 Anda dalam struktur folder. Misalnya, jika Anda menggunakan Email sebagai key prefix Object, email Anda akan muncul di bucket S3 di folder bernama Email.
- Enkripsi pesan — Opsi untuk mengenkripsi pesan email yang diterima sebelum mengirimkannya ke bucket S3 Anda.
- KMSkunci enkripsi — (Tersedia jika enkripsi Pesan dipilih.) AWS KMS Kunci yang SES harus digunakan untuk mengenkripsi email Anda sebelum menyimpannya ke ember S3. Anda dapat menggunakan KMS kunci default atau kunci terkelola pelanggan yang Anda buatKMS.

> **Note**
>
> KMSKunci yang Anda pilih harus berada di AWS wilayah yang sama dengan SES titik akhir yang Anda gunakan untuk menerima email.

  - Untuk menggunakan KMS kunci default, pilih aws/ses saat Anda mengatur aturan tanda terima di SES konsol. Jika Anda menggunakan SESAPI, Anda dapat menentukan KMS kunci default dengan memberikan ARN dalam bentuk`arn:aws:kms:REGION:AWSACCOUNTID:alias/aws/ses`. Misalnya, jika ID AWS akun Anda adalah 123456789012 dan Anda ingin menggunakan kunci default KMS di wilayah us-east-1, kunci defaultnya adalah. ARN KMS `arn:aws:kms:us-east-1:123456789012:alias/aws/ses` Jika Anda menggunakan KMS kunci default, Anda tidak perlu melakukan langkah tambahan untuk memberikan SES izin untuk menggunakan kunci tersebut.
  - Untuk menggunakan kunci terkelola pelanggan yang Anda buatKMS, berikan KMS kunci tersebut dan pastikan Anda menambahkan pernyataan ke kebijakan kunci Anda untuk memberikan SES izin untuk menggunakannya. ARN Untuk informasi lebih lanjut tentang memberi izin, lihat Memberikan izin ke Amazon SES untuk menerima email.

  Untuk informasi selengkapnya tentang menggunakan KMS withSES, lihat Panduan AWS Key Management Service Pengembang. Jika Anda tidak menentukan KMS kunci di konsol atauAPI, tidak SES akan mengenkripsi email Anda.

> **Important**
>
> Email Anda dienkripsi dengan SES menggunakan klien enkripsi S3 sebelum email dikirimkan ke S3 untuk penyimpanan. Itu tidak dienkripsi menggunakan enkripsi sisi server S3. Ini berarti Anda harus menggunakan klien enkripsi S3 untuk mendekripsi email setelah mengambilnya dari S3, karena layanan tidak memiliki akses untuk menggunakan kunci Anda untuk dekripsi. KMS Klien enkripsi ini tersedia di AWS SDK for Java dan AWS SDK for Ruby. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon Simple Storage Service.

- IAMrole — IAM Peran yang digunakan SES untuk mengakses sumber daya dalam aksi Deliver to S3 (bucketSNS, topic, KMS dan key Amazon S3). Jika tidak disediakan, Anda harus secara eksplisit memberikan izin untuk mengakses setiap sumber daya secara SES individual—lihat. Memberikan izin ke Amazon SES untuk menerima email

  Jika ingin menulis ke bucket S3 yang ada di wilayah di mana Penerimaan SES email tidak tersedia, Anda harus menggunakan IAM peran yang memiliki kebijakan izin tulis ke S3 sebagai kebijakan peran sebaris. Anda dapat menerapkan kebijakan izin untuk tindakan ini langsung dari konsol:

  1. Pilih Buat peran baru di bidang IAMperan dan masukkan nama diikuti oleh Buat peran. (Kebijakan IAM kepercayaan untuk peran ini akan dibuat secara otomatis di latar belakang.)
  2. Karena kebijakan IAM kepercayaan dibuat secara otomatis, Anda hanya perlu menambahkan kebijakan izin tindakan ke peran—pilih Lihat peran di bawah bidang IAMperan untuk membuka konsol. IAM
  3. Di bawah tab Izin, pilih Tambahkan izin dan pilih Buat kebijakan sebaris.
  4. Pada halaman Tentukan izin, pilih JSONdi Editor kebijakan.
  5. Salin dan tempel kebijakan izin dari IAMizin peran untuk tindakan S3 ke editor Kebijakan dan ganti data dalam teks merah dengan milik Anda. (Pastikan untuk menghapus kode contoh apa pun di editor.)
  6. Pilih Berikutnya.
  7. Tinjau dan buat kebijakan izin Anda untuk IAM peran tersebut dengan memilih Buat kebijakan.
  8. Pilih tab browser Anda di mana Anda memiliki halaman SES Create rule — Add actions terbuka dan lanjutkan dengan langkah-langkah yang tersisa untuk membuat aturan.
- SNStopik — Nama atau ARN SNS topik Amazon untuk memberi tahu saat email disimpan ke bucket S3. Contoh SNS topik adalah ARN arn:aws:sns:us-east - 1:123456789012:. MyTopic Anda

juga dapat membuat SNS topik saat menyiapkan tindakan dengan memilih Buat SNS Topik. Untuk informasi selengkapnya tentang SNS topik, lihat Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

> **Note**
>
> - SNSTopik yang Anda pilih harus berada di AWS wilayah yang sama dengan SES titik akhir yang Anda gunakan untuk menerima email.
> - Hanya gunakan enkripsi KMS kunci terkelola pelanggan dengan SNS topik yang Anda kaitkan dengan aturan SES tanda terima karena Anda akan diminta untuk mengedit kebijakan KMS kunci SES agar memungkinkan publikasiSNS. Ini berbeda dengan kebijakan KMS kunci AWS terkelola yang tidak dapat diedit oleh desain.

Terbitkan ke topik tindakan Amazon SNS

Tindakan SNS mempublikasikan email menggunakan notifikasi Amazon SNS. Notifikasi menyertakan konten email yang lengkap. Tindakan ini memiliki opsi berikut.

- Topik SNS—Nama atau ARN dari topik Amazon SNS yang akan digunakan untuk mempublikasikan email. Notifikasi Amazon SNS akan berisi salinan email yang mentah dan tidak dimodifikasi, yang umumnya dalam format Multipurpose Internet Mail Extensions (MIME). Untuk informasi lebih lanjut tentang format MIME, lihat RFC 2045.

> **Important**
>
> Jika Anda memilih untuk menerima email Anda melalui notifikasi Amazon SNS, ukuran maksimum email (termasuk header) adalah 150 KB. Email yang lebih besar akan terpental. Jika Anda menginginkan email yang lebih besar dari ini, simpan email ke bucket Amazon S3.

Contoh topik Amazon SNS ARN adalah arn:aws:sns:us-east - 1:123456789012:. MyTopic Anda juga dapat membuat topik Amazon SNS saat mengatur tindakan dengan memilih Buat Topik SNS. Untuk informasi lebih lanjut tentang topik Amazon SNS, lihat Panduan Developer Amazon Simple Notification Service.

> **Note**
>
> - Topik Amazon SNS yang Anda pilih harus berada di AWS wilayah yang sama dengan titik akhir Amazon SES yang Anda gunakan untuk menerima email.
> - Hanya gunakan enkripsi kunci KMS yang dikelola pelanggan dengan topik SNS yang Anda kaitkan dengan aturan tanda terima SES karena Anda akan diminta untuk mengedit kebijakan kunci KMS untuk memungkinkan SES mempublikasikan ke SNS. Hal ini berbeda dengan kebijakan kunci KMS AWS terkelola yang tidak dapat diedit oleh desain.

- Pengodean—Pengodean yang digunakan untuk email dalam notifikasi Amazon SNS. UTF-8 lebih mudah untuk digunakan, namun mungkin tidak mempertahankan semua karakter khusus ketika pesan dikodekan dengan format pengodean yang berbeda. Base64 mempertahankan semua karakter khusus. Untuk informasi tentang UTF-8 dan Base64, lihat RFC 3629 dan RFC 4648, secara berurutan.

Ketika Anda menerima email, Amazon SES mengeksekusi aturan dalam set aturan penerimaan yang aktif. Anda dapat mengonfigurasi aturan penerimaan untuk mengirimkan notifikasi menggunakan Amazon SNS. Aturan tanda penerimaan dapat mengirimkan dua tipe notifikasi yang berbeda:

- Notifikasi yang dikirim dari tindakan SNS — Saat Anda menambahkan tindakan SNS untuk aturan penerimaan, SNS mengirimkan informasi tentang email serta konten email tersebut. Jika pesan berukuran 150KB atau lebih kecil, tipe notifikasi ini juga mencakup email dengan badan MIME yang lengkap.
- Pemberitahuan yang dikirim dari jenis tindakan lain — Saat menambahkan jenis tindakan lain (termasuk Bounce, Lambda, Set Aturan Hentikan, WorkMailatau tindakan) ke aturan tanda terima, Anda dapat menentukan topik Amazon SNS secara opsional. Jika Anda melakukannya, Anda akan menerima notifikasi saat tindakan ini dilakukan. Notifikasi ini berisi informasi tentang email, tetapi tidak berisi konten email.

Topik berikut menjelaskan isi notifikasi ini dan memberikan contoh pada setiap tipe notifikasi:

- Isi notifikasi untuk menerima email Amazon SES
- Contoh pemberitahuan untuk penerimaan SES email Amazon

Isi notifikasi untuk menerima email Amazon SES

Semua Simple Notification Service (Amazon SNS) untuk penerimaan. JavaScript

Misalnya pemberitahuan, lihatContoh notifikasi.

Daftar Isi

- Objek JSON tingkat atas
- Objek penerimaan
  - objek tindakan
  - Objek dkimVerdict
  - Objek dmarcVerdict
  - Objek spamVerdict
  - Objek spfVerdict
  - Objek virusVerdict
- objek surat
  - Objek commonHeaders

Objek JSON tingkat atas

Objek JSON tingkat atas berisi beberapa bidang berikut.

Nama Bidang	Deskripsi
`notificationType`	Tipe notifikasi. Untuk tipe notifikasi ini, nilainya selalu `Received`.
`receipt`	Objek yang berisi informasi tentang pengiriman email.
`mail`	Objek yang berisi informasi tentang email yang terkait dengan notifikasi.
`content`	String yang berisi email mentah dan tidak dimodifikasi, yang umumnya dalam format Multipurpose Internet Mail Extensions (MIME).

Nama Bidang	Deskripsi
	Untuk informasi lebih lanjut tentang format MIME, lihat RFC 2045. **Note** Bidang ini hanya ada jika notifikasi dipicu oleh tindakan SNS. Notifikasi yang dipicu oleh semua tindakan lain tidak menahan bidang ini.

Objek penerimaan

Objek `receipt` berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`action`	Objek yang merangkum informasi tentang tindakan yang dieksekusi. Untuk daftar peluang nilai, lihat objek tindakan.
`dkimVerdict`	Objek yang menunjukkan apakah pemeriksa an DomainKeys Identified Mail (DKIM) telah dilewati. Untuk daftar peluang nilai, lihat Objek dkimVerdict.
`dmarcPolicy`	Mengindikasikan pengaturan autentikasi pesan, pelaporan & kesesuaian berbasis Domain (DMARC) untuk domain pengiriman. Bidang ini hanya muncul jika pesan gagal diautentikasi DMARC. Peluang nilai untuk kolom ini meliputi: • `none`: Pemilik domain pengiriman meminta bahwa tidak ada tindakan tertentu yang

Nama Bidang	Deskripsi
	dilakukan pada pesan yang gagal diautenti kasikan DMARC. • `quarantine` : Pemilik domain pengiriman meminta bahwa pesan yang gagal diautenti kasikan DMARC dianggap mencurigakan oleh penerima. • `reject`: Pemilik domain pengiriman meminta pesan yang gagal diautentikasikan DMARC untuk ditolak.
`dmarcVerdict`	Objek yang mengindikasikan apakah pengecekan pada Pesan Autentikasi, Pelaporan & kesesuaian berbasis Domain (DMARC) sudah berlalu. Untuk daftar peluang nilai, lihat Objek dmarcVerdict.
`processingTimeMillis`	String yang menentukan tanda titik, dalam milidetik, pada saat Amazon SES menerima pesan ketika Amazon SES tersebut memicu tindakan.
`recipients`	Penerima (khususnya, envelope dengan alamat RCPT TO) yang cocok dengan aktivasi Aturan penerimaan. Alamat yang tercantum di sini mungkin berbeda dari yang tercantum oleh bidang `destination` di bidang the section called “objek surat”.
`spamVerdict`	Objek yang mengidikasikan apakah pesan adalah spam. Untuk daftar peluang nilai, lihat Objek spamVerdict.
`spfVerdict`	Objek yang mengindikasikan apakah pemeriksaan Pengirim Kebijakan Kerangka Kerja (SPF) sudah berlalu. Untuk daftar peluang nilai, lihat Objek spfVerdict.

Nama Bidang	Deskripsi
`timestamp`	String yang menentukan tanggal dan waktu yang memenuhi syarat pada saat tindakan dipicu, di format ISO 8601.
VirusPutusan	Objek yang menunjukkan apakah pesan berisi virus. Untuk daftar peluang nilai, lihat Objek virusVerdict.

objek tindakan

Objek `action` berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`type`	String yang mengindikasikan tipe tindakan yang dieksekusi. Peluang nilai adalah `S3`, `SNS`, `Bounce`, `Lambda`, `Stop`, dan `WorkMail`.
`topicArn`	String yang berisi Amazon Resource Name (ARN) dari topik Amazon SNS yang dipublika sikan oleh notifikasi.
`bucketName`	String yang berisi nama bucket Amazon S3 yang diterbitkan pesan. Hadir hanya untuk tipe tindakan S3.
`objectKey`	String dengan nama yang secara unik mengidentifikasi email dalam bucket Amazon S3. Ini sama dengan `messageId` di the section called “objek surat”. Hadir hanya untuk tipe tindakan S3.
`smtpReplyCode`	String yang berisi kode balasan SMTP, seperti yang didefinisikan oleh RFC 5321. Hadir hanya untuk tipe tindakan pentalan.

Nama Bidang	Deskripsi
`statusCode`	String dengan kode status yang ditingkat kan SMTP, seperti yang didefinisikan oleh RFC 3463. Hadir hanya untuk tipe tindakan pentalan.
`message`	String yang berisi teks yang dapat dibaca manusia untuk dimasukkan dalam pesan pentalan. Hadir hanya untuk tipe tindakan pentalan.
`sender`	String yang berisi alamat email dari pengirim email yang memantul. Ini adalah alamat dari mana pesan pentalan dikirim. Hadir hanya untuk tipe tindakan pentalan.
`functionArn`	String yang berisi ARN fungsi Lambda yang dipicu. Hadir hanya untuk tipe tindakan Lambda.
`invocationType`	String yang berisi tipe invokasi fungsi Lambda. Kemungkinan nilai adalah `RequestRe sponse` dan `Event`. Hadir hanya untuk tipe tindakan Lambda.
`organizationArn`	String yang berisi ARN WorkMail organisas i Amazon. Hadir hanya untuk tipe WorkMail tindakan.

Objek dkimVerdict

Objek `dkimVerdict` berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`status`	String yang berisi DKIM verdict. Kemungkinan nilai adalah:

Nama Bidang	Deskripsi
	• PASS: Pesan melewati autentikasi DKIM. • FAIL: Pesan gagal dalam autentikasi DKIM. • GRAY: Pesan tidak ditandatangani DKIM atau domain dari domain dan tanda tangan DKIM tidak cocok. • PROCESSING_FAILED : Terdapat masalah yang mencegah Amazon SES untuk memeriksa tanda tangan DKIM. Misalnya, Kueri DNS gagal atau header tanda tangan DKIM tidak diformat dengan benar.

Objek dmarcVerdict

Objek `dmarcVerdict` berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`status`	String yang berisi DMARC verdict. Kemungkin an nilai adalah: • PASS: Pesan melewati autentikasi DMARC. • FAIL: Pesan gagal dalam autentikasi DMARC. • GRAY: Setidaknya satu dari SPF atau DKIM, tetapi domain pengiriman tidak memiliki kebijakan DMARC atau menggunak anp=none kebijakan. • PROCESSING_FAILED : Terdapat masalah yang mencegah Amazon SES memberikan DMARC verdict.

Objek spamVerdict

Objek `spamVerdict` berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`status`	String yang berisi hasil pemindaian spam. Kemungkinan nilai adalah: • `PASS`: Pemindaian spam menentukan bahwa pesan tidak mungkin berisi spam. • `FAIL`: Pemindaian spam menentukan bahwa pesan kemungkinan berisi spam. • `GRAY`: Amazon SES memindai email tetapi tidak bisa menentukan dengan jelas apakah itu spam. • `PROCESSING_FAILED` : Amazon SES tidak dapat memindai email. Sebagai contoh, email bukan pesan MIME yang valid.

Objek spfVerdict

Objek `spfVerdict` berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`status`	String yang berisi SPF verdict. Kemungkinan nilai adalah: • `PASS`: Pesan melewati autentikasi SPF. • `FAIL`: Pesan gagal dalam autentikasi SPF. • `GRAY`: Hasil SPF adalah`none`,`softfail`, atau`neutral`. • `PROCESSING_FAILED` : Terdapat masalah yang mencegah Amazon SES memeriksa catatan SPF. Sebagai contoh, kueri DNS mengalami kegagalan.

Objek virusVerdict

Objek `virusVerdict` berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`status`	String yang berisi hasil pemindaian virus. Kemungkinan nilai adalah: • `PASS`: Pesan tidak berisi virus. • `FAIL`: Pesan berisi virus. • `GRAY`: Amazon SES memindai email tetapi tidak bisa menentukan dengan jelas apakah pesan berisi virus. • `PROCESSING_FAILED` : Amazon SES tidak dapat memindai isi email. Sebagai contoh, email bukan pesan MIME yang valid.

objek surat

Objek `mail` berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`destination`	Daftar lengkap untuk semua penerima alamat (termasuk penerima Kepada: dan CC:) dari header email MIME yang masuk.
`messageId`	String yang berisi ID unik yang ditetapkan ke email oleh Amazon SES. Jika email dikirim ke Amazon S3, ID pesan juga merupakan kunci objek Amazon S3 yang digunakan untuk menulis pesan ke bucket Amazon S3 Anda.
`source`	String yang berisi alamat email (khususnya, alamat envelope MAIL FROM) tempat email dikirimkan.

Nama Bidang	Deskripsi
`timestamp`	String yang berisi waktu saat email diterima, dalam format ISO8601.
`headers`	header Amazon SES dan header khusus Anda. Setiap header memiliki bidang berikut:`name` dan`value`.
`commonHeaders`	header umum untuk semua email. Setiap header memiliki bidang berikut:`name` dan`value`.
`headersTruncated`	Menentukan apakah header dipotong dalam notifikasi, yang terjadi jika header lebih besar dari 10 KB. Kemungkinan nilai adalah `true` dan `false`.

Objek commonHeaders

Objek `commonHeaders` dapat memiliki bidang yang ditunjukkan dalam tabel berikut. Bidang yang ada dalam objek ini bervariasi tergantung pada bidang yang ada dalam email masuk.

Nama Bidang	Deskripsi
`messageId`	ID dari pesan asli.
`date`	Tanggal dan waktu ketika Amazon SES menerima pesan.
`to`	Toheader.
`cc`	CCheader.
`bcc`	BCCheader.
`from`	Fromheader.
`sender`	Senderheader.

Nama Bidang	Deskripsi
`returnPath`	Return-Pathheader.
`replyTo`	Reply-Toheader.
`subject`	Subjectheader.

Contoh pemberitahuan untuk penerimaan SES email Amazon

Bagian ini mencakup contoh tipe notifikasi berikut:

- Pemberitahuan yang dikirim sebagai hasil dari suatu SNS tindakan.
- Notifikasi yang dikirim sebagai hasil dari tipe tindakan lain(sebuah pemberitahuan peringatan).

Pemberitahuan suatu SNS tindakan

Bagian ini berisi contoh pemberitahuan SNS tindakan. Tidak seperti pemberitahuan peringatan yang ditampilkan sebelumnya, ini mencakup `content` bagian yang berisi email, yang biasanya dalam format Multipurpose Internet Mail Extensions (MIME).

```
{
  "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-BEC9-87CBCF2211C9@example.com",
    "messageId":"d6iitobk75ur44p8kdnnp7g2n800",
    "destination":[
      "recipient@example.com"
    ],
    "headersTruncated":false,
    "headers":[
      {
        "name":"Return-Path",

 "value":"<0000014fbe1c09cf-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=ug7nbtf4gccmlpwj322ax3p6ow6yfsug; 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+zPR0DW1kp7oJZakrzupr6LBe6sUuvqpBkig56UzUwc29rFbJF
 hlX3Ov7DeYVNoN38stqwsF8ivcajXpQsXRC1cW9z8x875J041rClAjV7EGbLmudVpPX
 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\ts=ug7nbtf4gccmlpwj322ax3p6ow6yfsug; d=amazonses.com; t=1442003552;\r\n
\th=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Date:Message-
ID:Feedback-ID;\r\n\tbh=DWr3IOmYWoXCA9ARqGC/UaODfghffiwFNRIb2Mckyt4=;\r\n
\tb=p4ukUDSFqhqiub+zPR0DW1kp7oJZakrzupr6LBe6sUuvqpBkig56UzUwc29rFbJF\r\n
\thlX3Ov7DeYVNoN38stqwsF8ivcajXpQsXRC1cW9z8x875J041rClAjV7EGbLmudVpPX\r\n
\t4hHst1XPyX5wmgdHIhmUuh8oZKpVqGi6bHGzzf7g=\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"
}
```

Notifikasi pemberitahuan

Bagian ini berisi contoh SNS pemberitahuan Amazon yang dapat dipicu oleh tindakan S3. Pemberitahuan yang dipicu oleh tindakan Lambda, tindakan pantulan, tindakan berhenti, dan WorkMail tindakan serupa. Meskipun notifikasi berisi informasi tentang email, notifikasi tersebut tidak berisi isi email itu sendiri.

```
{
   "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":
 "0000014fbe1c09cf-7cb9f704-7531-4e53-89a1-5fa9744f5eb6-000000@amazonses.com",
      "messageId": "d6iitobk75ur44p8kdnnp7g2n800",
      "destination": [
        "recipient@example.com"
      ],
      "headersTruncated": false,
      "headers": [
        {
          "name": "Return-Path",
          "value":
 "<0000014fbe1c09cf-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=ug7nbtf4gccmlpwj322ax3p6ow6yfsug; 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+zPR0DW1kp7oJZakrzupr6LBe6sUuvqpBkig56UzUwc29rFbJF
 hlX3Ov7DeYVNoN38stqwsF8ivcajXpQsXRC1cW9z8x875J041rClAjV7EGbLmudVpPX
 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"
      }
    }
  }
```

Hentikan tindakan set aturan

Tindakan Stop mengakhiri evaluasi set aturan penerimaan dan secara opsional memberi tahu Anda melalui Amazon SNS. Tindakan ini memiliki opsi berikut.

- Topik SNS—Nama atau ARN dari topik Amazon SNS yang akan memberi tahu ketika tindakan Berhenti dilakukan. Contoh ARN topik Amazon SNS adalaharn:aws:sns:us-east-1:123456789012:mytopic. Anda juga dapat membuat topik Amazon SNS saat mengatur tindakan dengan memilih Buat Topik SNS. Untuk informasi lebih lanjut tentang topik Amazon SNS, lihat Panduan Developer Amazon Simple Notification Service.

  > **Note**
  >
  > Topik Amazon SNS yang Anda pilih harus samaAWSWilayah sebagai titik akhir Amazon SES yang Anda gunakan untuk menerima email.

Integrasi dengan Amazon WorkMail aksi

TheWorkMailtindakan terintegrasi dengan Amazon WorkMail. Jika Amazon WorkMail melakukan semua pemrosesan email Anda, Umumnya Anda tidak akan menggunakan tindakan ini secara langsung karena Amazon WorkMail menangani pengaturannya. Tindakan ini memiliki opsi berikut.

- Organisasi ARN—ARN Amazon WorkMailorganisasi. Amazon WorkMail ARN organisasi dalam bentuk`arn:aws:workmail:`*`region`*`:`*`account_ID`*`:organization/`*`organization_ID`*, dimana:

- `region`adalah wilayah di mana Anda menggunakan Amazon SES dan Amazon SES dan Amazon WorkMail. (Anda harus menggunakan mereka dari Wilayah yang sama.) Contohnya adalah us-east-1.
- `account_ID` adalah akun ID AWS. Anda dapat menemukan ID akun AWS Anda pada halaman Akun dari halaman Konsol Manajemen AWS.
- `organization_ID`adalah pengenal unik yang Amazon WorkMail menghasilkan saat Anda membuat organisasi. Anda dapat menemukan ID organisasi di Amazon WorkMail konsol di halaman Pengaturan Organisasi organisasi Anda.

Contoh Amazon yang lengkap WorkMail Organisasi ARN adalaharn:aws:workmail:us-east-1:123456789012: organisasi/m-68755160c4e29a2b2f8fb58f359d7. Untuk informasi tentang Amazon WorkMail organisasi, lihatAmazon WorkMail Panduan Administrator.

- SNS Topik-Nama atau ARN dari topik Amazon SNS untuk memberi tahu saat Amazon SNS untuk memberi tahu saat Amazon WorkMail tindakan diambil. Contoh dari Amazon SNS topik ARN adalaharn:aws:SNS: us-timur- 1:123456789012:MyTopic. Anda juga dapat membuat topik Amazon SNS saat mengatur tindakan dengan memilih Buat Topik SNS. Untuk informasi lebih lanjut tentang topik Amazon SNS, lihat Panduan Developer Amazon Simple Notification Service.

> **Note**
>
> Topik Amazon SNS yang Anda pilih harus samaAWSWilayah sebagai titik akhir Amazon SES yang Anda gunakan untuk menerima email.

> **Note**
>
> Amazon SES hanya mendukung WorkMail tindakan di daerah di mana WorkMail tersedia. LihatAmazon WorkMail titik akhir dan kuotadiReferensi Umum AWS.

Buat panduan konsol filter alamat IP

Bagian ini akan memandu Anda melalui pengaturan filter alamat IP menggunakan konsol Amazon SES. Penyaringan alamat IP memungkinkan Anda untuk memberikan tingkat kendali yang luas. Filter IP ini memungkinkan Anda untuk secara eksplisit memblokir atau mengizinkan semua pesan dari alamat IP tertentu atau rentang alamat IP.

Secara opsional, Anda dapat menggunakan API `CreateReceiptFilter` untuk membuat filter alamat IP seperti yang dijelaskan dalam Referensi Amazon Simple Email Service.

> **Note**
>
> Jika Anda hanya ingin menerima surat dari daftar alamat IP terbatas yang diketahui, maka atur daftar blokir yang berisi `0.0.0.0/0`, dan atur daftar izinkan yang berisi alamat IP yang Anda percayai. Konfigurasi ini memblokir semua alamat IP secara default, dan hanya mengizinkan surat dari alamat IP yang Anda tentukan secara eksplisit.

Prasyarat

Prasyarat berikut harus dipenuhi sebelum melanjutkan pengaturan pengendalian email berbasis penerima menggunakan filter alamat IP:

1. Anda harus terlebih dahulu membuat dan memverifikasi identitas domain di Amazon SES.
2. Selanjutnya, Anda perlu menentukan server email mana yang dapat menerima email untuk domain Anda dengan menerbitkan catatan MX ke pengaturan DNS domain Anda. (Catatan MX harus mengacu pada titik akhir Amazon SES yang menerima email untuk wilayah AWS tempat Anda menggunakan Amazon SES.)

Buat filter alamat IP

Untuk membuat filter alamat IP menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di bagian panel navigasi kiri, pilih Penerimaan email.
3. Pilih tab Filter alamat IP.
4. Pilih Buat Filter.
5. Masukkan nama unik untuk filter Anda - riwayat bidang akan mengindikasikan persyaratan sintaks. (Nama harus diawali dan diakhiri dengan huruf atau angka dan berisi kurang dari 64 alfanumerik, tanda hubung (-), garis bawah (_), dan karakter titik (.))
6. Masukkan alamat IP atau rentang alamat IP - riwayat bidang akan memberikan contoh yang ditentukan dalam sintaksis Classless Inter-Domain Routing (CIDR). (Contoh alamat IP

tunggal adalah 10.0.0.1. Contoh dari rentang alamat IP adalah 10.0.0.1/24. Untuk informasi selengkapnya tentang notasi CIDR, lihat RFC 2317.)

7. Pilih Tipe kebijakan dengan memilih salah satu Blok atau Izinkan tombol radio.
8. Pilih Buat filter.
9. Jika Anda ingin menambahkan filter IP lain, pilih Buat filter dan ulangi langkah-langkah sebelumnya untuk setiap filter tambahan yang ingin Anda tambahkan.
10. Jika Anda ingin menghapus filter alamat IP, pilih filter tersebut dan pilih tombol kotak centang Hapus.

Melihat metrik untuk penerimaan email Amazon SES

Jika Anda telah mengaktifkan penerimaan email di Amazon SES dan Anda telah membuat aturan tanda terima untuk email Anda, Anda dapat melihat metrik untuk set aturan tanda terima dan aturan tersebut menggunakan Amazon CloudWatch.

Di CloudWatch konsol, Anda akan menemukan metrik di bawah Metrik > Semua metrik > SES > Metrik Set Aturan Tanda Terima dan Metrik Aturan Penerimaan.

> **Note**
>
> Metrik Set Aturan Tanda Terima dan Metrik Aturan Penerimaan tidak akan muncul di bawah SES jika Anda belum:
>
> - mengaktifkan penerimaan email
> - membuat aturan tanda terima apa pun
> - menerima surat apa pun yang sesuai dengan aturan Anda.

Metrik pesan berikut tersedia:

- Menerima pesan

Cakupan	Metrik	Deskripsi	Dimensi
Metrik Set Aturan	Diterima	SES berhasil menerima pesan yang memiliki setidaknya satu aturan yang berlaku. Metrik ini hanya dapat memiliki nilai1.	RuleSetName

Cakupan	Metrik	Deskripsi	Dimensi
Tanda Terima			
Metrik Aturan Tanda Terima	Diterima	SES berhasil menerima pesan dan akan mencoba memproses aturan yang diterapkan. Metrik ini hanya dapat memiliki nilai1.	RuleName

- Penerbitan pesan

Cakupan	Metrik	Deskripsi	Dimensi
Metrik Set Aturan Tanda Terima	PublishSuccess	SES berhasil mengeksekusi semua aturan yang berlaku dalam set aturan.	RuleSetName
Metrik Aturan Tanda Terima	PublishSuccess	SES berhasil mengeksekusi aturan yang berlaku untuk pesan penerima.	RuleName
Metrik Set Aturan Tanda Terima	PublishFailure	SES mengalami kesalahan ketika mencoba mengeksekusi aturan dalam set aturan, eksekusi akan dicoba lagi.	RuleSetName
Metrik Aturan Tanda Terima	PublishFailure	SES mengalami kesalahan ketika mencoba mengeksekusi tindakan dalam suatu aturan —tergantung pada kesalahannya, eksekusi dapat dicoba ulang.	RuleName
Metrik Set Aturan Tanda Terima	PublishExpired	SES tidak akan lagi mencoba mengeksekusi aturan karena aturan tersebut tidak berhasil dalam waktu 36 jam, atau mengalami kesalahan yang tidak dapat diambil kembali.	RuleSetName

Cakupan	Metrik	Deskripsi	Dimensi
Metrik Aturan Tanda Terima	PublishExpired	SES tidak akan lagi mencoba untuk mengeksekusi tindakan aturan karena mereka tidak berhasil dalam waktu 36 jam.	RuleName

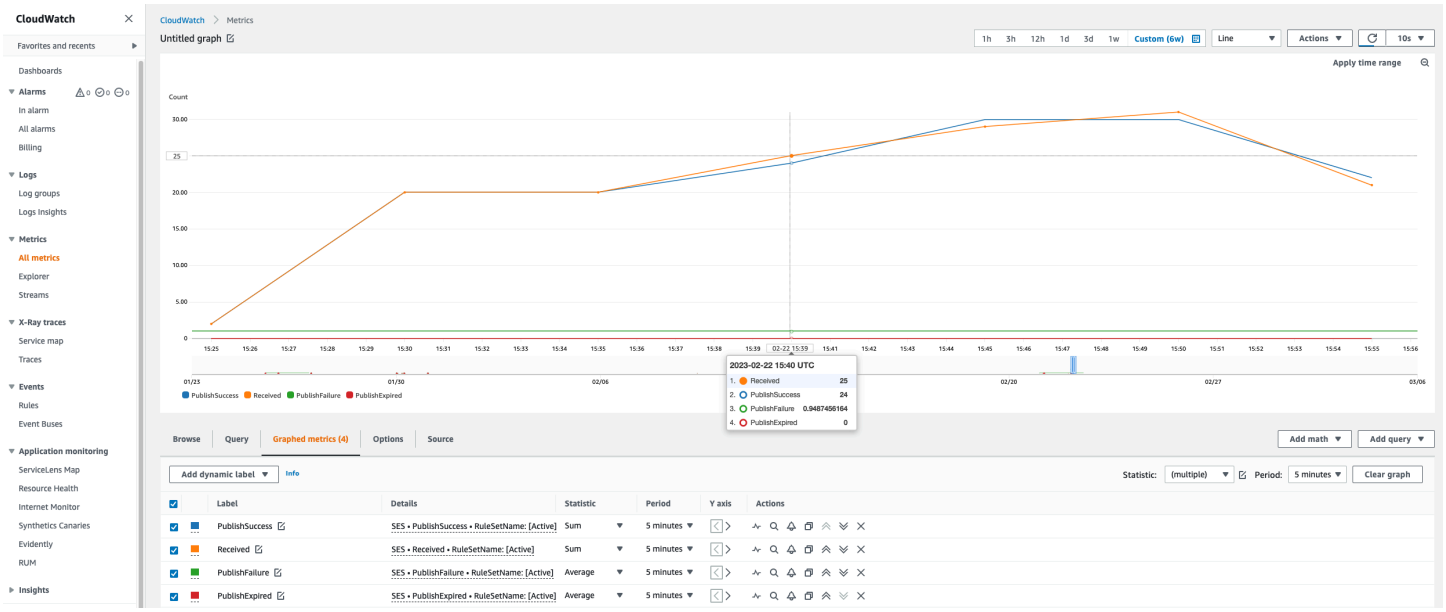
Note

- Dalam tabel sebelumnya, istilah berlaku berarti bahwa pengirim tidak diblokir oleh Filter IP atau berada di daftar blokir internal SES, dan aturan memiliki kondisi penerima yang cocok dan kebijakan TLS yang cocok.
- Kesalahan kegagalan publikasi dapat terjadi, misalnya, jika Anda menghapus atau mencabut izin ke bucket Amazon S3, topik Amazon SNS, atau fungsi Lambda yang tindakan di salah satu aturan tanda terima dikonfigurasi untuk digunakan.
- Karena hanya satu set aturan yang dapat aktif pada satu waktu, SES menerbitkan metrik agregat yang ditampilkan sebagai RuleSetName: [Aktif] untuk semua kumpulan aturan yang aktif untuk rentang waktu yang Anda pilih. CloudWatch Ini memiliki keuntungan membiarkan Anda dengan bebas mengubah set aturan tanpa perubahan apa pun pada pengaturan Anda yang mengkhawatirkan.

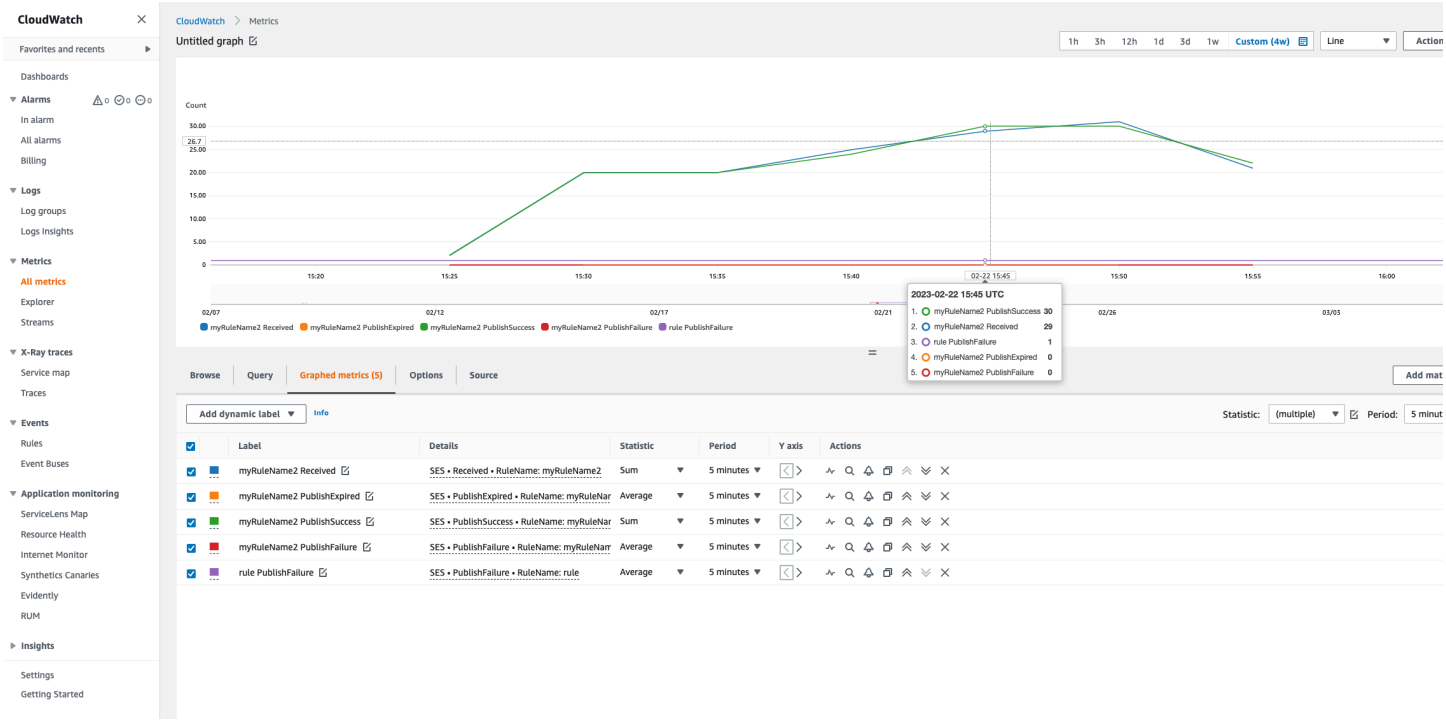
Important

Perubahan yang Anda buat untuk memperbaiki set aturan penerimaan hanya akan berlaku untuk email yang diterima Amazon SES setelah pembaruan. Email selalu dievaluasi terhadap set aturan penerimaan yang ada pada saat email diterima.

Metrik untuk set aturan tanda terima SES ditampilkan di CloudWatch konsol.



Metrik untuk aturan tanda terima SES ditampilkan di CloudWatch konsol.



Identitas terverifikasi di Amazon SES

Di Amazon SES, identitas terverifikasi adalah domain atau alamat email yang Anda gunakan untuk mengirim atau menerima email. Sebelum Anda dapat mengirim email menggunakan Amazon SES, Anda harus membuat dan memverifikasi setiap identitas yang akan Anda gunakan sebagai alamat "Dari", "Sumber", "Pengirim", atau "Jalur Kembali". Memverifikasi identitas dengan Amazon SES menegaskan bahwa Anda memilikinya dan membantu mencegah penggunaan yang tidak sah.

Jika akun Anda masih dalam kotak pasir Amazon SES, Anda juga perlu memverifikasi alamat email apa pun yang Anda rencanakan untuk mengirim email, kecuali jika Anda mengirim untuk menguji kotak masuk yang disediakan oleh simulator kotak surat Amazon SES. Untuk informasi selengkapnya, lihat the section called "Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual".

Anda dapat membuat identitas dengan menggunakan konsol Amazon SES atau API Amazon SES. Proses verifikasi identitas bergantung pada jenis identitas yang Anda pilih untuk dibuat.

> **Tip**
>
> Jika Anda pengguna pertama kali SES, Anda dapat menggunakan panduan Memulai untuk membuat dan memverifikasi identitas pertama Anda (alamat email atau domain).

Konten

- Membuat dan memverifikasi identitas di Amazon SES
- Mengelola identitas di Amazon SES
- Mengonfigurasi identitas di Amazon SES
- Mengirim email di Amazon SES di Amazon SES di Amazon SES di Amazon SES di Amazon SES di Amazon SES

Membuat dan memverifikasi identitas di Amazon SES

Di Amazon SES, Anda dapat membuat identitas di tingkat domain atau Anda dapat membuat identitas alamat email. Tipe identitas ini tidak saling eksklusif. Dalam kebanyakan kasus, membuat identitas domain menghilangkan kebutuhan untuk membuat dan memverifikasi identitas alamat

email individual, kecuali jika Anda ingin menerapkan konfigurasi khusus ke alamat email tertentu. Apakah Anda membuat domain dan menggunakan alamat email berdasarkan domain, atau membuat alamat email individual, ada manfaat untuk kedua pendekatan tersebut. Metode mana yang Anda pilih tergantung pada kebutuhan spesifik Anda seperti yang dibahas di bawah ini.

Membuat dan memverifikasi identitas alamat email adalah cara tercepat untuk memulai di SES, tetapi ada manfaat untuk memverifikasi identitas di tingkat domain. Ketika Anda memverifikasi identitas alamat email, hanya alamat email yang dapat digunakan untuk mengirim email, tetapi ketika Anda memverifikasi identitas domain, Anda dapat mengirim email dari subdomain atau alamat email dari domain terverifikasi tanpa harus memverifikasi masing-masing satu per satu. Misalnya, jika Anda membuat dan memverifikasi identitas domain yang disebut example.com, Anda tidak perlu membuat identitas subdomain terpisah untuk.example.com, a.b.example.com, atau memisahkan identitas alamat email untuk user@example.com, user@a.example.com, dan sebagainya.

Namun, perlu diingat bahwa identitas alamat email yang menggunakan verifikasi yang diwariskan dari domainnya terbatas pada pengiriman email langsung. Jika Anda ingin melakukan pengiriman lebih lanjut, Anda juga harus memverifikasinya secara eksplisit sebagai identitas alamat email. Pengiriman lanjutan mencakup penggunaan alamat email dengan set konfigurasi, otorisasi kebijakan untuk pengiriman delegasi, dan konfigurasi yang mengganti setelan domain.

Untuk membantu memperjelas warisan verifikasi dan kemampuan pengiriman email yang dibahas di atas, tabel berikut mengkategorikan setiap kombinasi verifikasi domain/alamat email dan mencantumkan warisan, tingkat pengiriman, dan status tampilan untuk masing-masing:

	Hanya domain yang diverifikasi	Hanya alamat email yang diverifikasi	Kedua domain & alamat email diverifikasi
Tingkat warisan	Subdomain dan alamat email mewarisi verifikasi dari domain induk.	Alamat email diverifikasi secara eksplisit.	• Subdomain mewarisi verifikasi dari domain induk. • Alamat email diverifikasi secara eksplisit.
Tingkat pengiriman	Alamat email terbatas pada pengiriman email langsung.	Alamat email dapat digunakan dalam pengiriman lanjutan*.	Alamat email dapat digunakan dalam pengiriman lanjutan*.

	Hanya domain yang diverifikasi	Hanya alamat email yang diverifikasi	Kedua domain & alamat email diverifikasi
Status yang ditampilkan	Status konsol/API: • Domain/Subdomain = Diverifikasi • Alamat email = Tidak diverifikasi.	Status konsol/API: • Alamat email = Terverifikasi	Status konsol/API: • Domain/Subdomain = Diverifikasi • Alamat email = Terverifikasi.

* Pengiriman lanjutan termasuk menggunakan alamat email dengan set konfigurasi, otorisasi kebijakan untuk pengiriman delegasi, dan konfigurasi yang mengesampingkan pengaturan domain.

Untuk mengirim email dari domain atau alamat email yang sama di lebih dari satuWilayah AWS, Anda harus membuat dan memverifikasi identitas terpisah untuk setiap Wilayah. Anda dapat memverifikasi sebanyak 10.000 identitas di setiap Wilayah.

Saat Anda membuat dan memverifikasi identitas domain dan alamat email, pertimbangkan hal berikut:

- Anda dapat mengirim email dari subdomain atau alamat email apa pun dari domain yang diverifikasi tanpa harus memverifikasi masing-masing secara individu. Misalnya, jika Anda membuat dan memverifikasi identitas untuk example.com, Anda tidak perlu membuat identitas terpisah untuk a.example.com, a.b.example.com, user@example.com, user@a.example.com, dan sebagainya.
- Seperti yang ditentukan di RFC 1034, setiap label DNS dapat memiliki hingga 63 karakter dan panjang seluruh nama domain tidak boleh melebihi total 255 karakter.
- Jika Anda memverifikasi domain, subdomain, atau alamat email yang berbagi domain root, pengaturan identitas (seperti pemberitahuan umpan balik) berlaku pada tingkat paling terperinci yang Anda verifikasi.
  - Pengaturan identitas alamat email terverifikasi membatalkan pengaturan identitas domain terverifikasi.
  - Pengaturan identitas subdomain terverifikasi membatalkan pengaturan identitas domain terverifikasi, dengan pengaturan subdomain tingkat lebih rendah membatalkan pengaturan subdomain tingkat yang lebih tinggi.

Sebagai contoh, asumsikan bahwa Anda memverifikasi user@a.b.example.com, a.b.example.com, b.example.com, dan example.com. Ini adalah pengaturan identitas terverifikasi yang akan digunakan dalam skenario berikut:

- Email yang dikirim dari user@example.com (alamat email yang tidak diverifikasi secara khusus) akan menggunakan pengaturan untuk example.com.
- Email yang dikirim dari user@a.b.example.com (alamat email yang diverifikasi secara khusus) akan menggunakan pengaturan untuk user@a.b.example.com.
- Email yang dikirim dari user@b.example.com (alamat email yang tidak diverifikasi secara khusus) akan menggunakan pengaturan untuk b.example.com.

- Anda dapat menambahkan label ke alamat email yang terverifikasi tanpa melakukan langkah verifikasi tambahan. Untuk menambahkan label ke alamat email, tambahkan tanda tambah (+) antara nama akun dan tanda "at" (@), diikuti dengan label teks. Misalnya, jika Anda sudah memverifikasi sender@example.com, Anda dapat menggunakan sender+myLabel@example.com sebagai alamat "Dari" atau "Jalur Kembali" untuk email Anda. Anda dapat menggunakan fitur ini untuk menerapkan Variable Envelope Return Path (VERP). Kemudian Anda dapat menggunakan VERP untuk mendeteksi dan menghapus alamat email yang tidak terkirim dari milis Anda.
- Nama domain peka huruf besar dan kecil. Jika Anda memverifikasi example.com, Anda juga dapat mengirim dari EXAMPLE.com.
- Alamat email peka huruf besar/kecil. Jika Anda memverifikasi sender@EXAMPLE.com, Anda tidak dapat mengirim email dari sender@example.com kecuali Anda memverifikasi sender@example.com juga.
- Di setiap Wilayah AWS, Anda dapat memverifikasi hingga 10.000 identitas (domain dan alamat email, dengan kombinasi apa pun).

> **Tip**
>
> Jika Anda pengguna pertama kali SES, Anda dapat menggunakan panduan Memulai untuk membuat dan memverifikasi identitas pertama Anda (alamat email atau domain).

Daftar Isi

- Membuat identitas domain
- Memverifikasi identitas domain DKIM dengan penyedia DNS Anda
- Membuat identitas alamat email

- Memverifikasi identitas alamat email
- Membuat dan memverifikasi identitas dan menetapkan konfigurasi default yang ditetapkan pada saat yang sama
- Menggunakan templat email verifikasi kustom

Membuat identitas domain

Bagian dari pembuatan identitas domain adalah mengonfigurasi verifikasi berbasis DKIM-nya. DomainKeysIdentified Mail (DKIM) adalah metode otentikasi email yang digunakan Amazon SES untuk memverifikasi kepemilikan domain, dan menerima server email digunakan untuk memvalidasi keaslian email. Anda dapat memilih untuk mengkonfigurasi DKIM dengan menggunakan Easy DKIM atau Bring Your Own DKIM (BYODKIM), dan tergantung pada pilihan Anda, Anda harus mengonfigurasi panjang kunci penandatanganan kunci pribadi sebagai berikut:

- Easy DKIM - terima default Amazon SES sebesar 2048 bit, atau ganti dengan memilih 1024 bit.
- BYODKIM - panjang kunci pribadi harus minimal 1024 bit dan hingga 2048-bit.

Lihat the section called “Panjang kunci penandatanganan DKIM” untuk mempelajari lebih lanjut tentang panjang kunci penandatanganan DKIM dan cara mengubahnya.

Prosedur berikut menunjukkan cara membuat identitas domain menggunakan konsol Amazon SES.

- Jika Anda sudah membuat domain dan hanya perlu memverifikasinya, lewati ke prosedur the section called “Memverifikasi identitas domain” di halaman ini.

Membuat identitas domain

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas Terverifikasi.
3. Pilih Buat identitas.
4. Di bawah Detail identitas, pilih domain sebagai tipe identitas yang ingin Anda buat. Anda harus memiliki akses ke pengaturan DNS domain untuk menyelesaikan proses verifikasi domain.
5. Masukkan nama domain atau subdomain di bidang Domain.

> **Tip**
>
> Jika domain Anda adalah www.example.com, masukkan example.com sebagai domain Anda. Jangan sertakan bagian "www." karena proses verifikasi domain tidak akan berhasil jika Anda melakukannya.

6. (Opsional) Jika Anda ingin Menetapkan set konfigurasi default, pilih kotak centang.

   1. Untuk set konfigurasi Default, pilih set konfigurasi yang ada yang ingin Anda tetapkan ke identitas Anda. Jika Anda belum membuat rangkaian konfigurasi apa pun, lihat Set konfigurasi.

      > **Note**
      >
      > Amazon SES memilih default ke konfigurasi yang ditetapkan saat tidak ada set lain yang ditentukan pada saat pengiriman. Jika set konfigurasi ditentukan, Amazon SES menerapkan set yang ditentukan sebagai pengganti set default.

7. (Opsional) Jika Anda ingin Menggunakan email kustom DARI domain, pilih kotak centang dan selesaikan langkah-langkah berikut. Untuk informasi selengkapnya, lihat the section called “Menggunakan domain MAIL FROM kustom”.

   1. Untuk domain MAIL FROM, masukkan subdomain yang ingin Anda gunakan sebagai domain MAIL FROM. Ini harus merupakan subdomain dari identitas domain yang Anda verifikasi. Domain MAIL FROM seharusnya bukan domain tempat Anda mengirim email.
   2. Untuk Perilaku pada kegagalan MX, tunjukkan tindakan yang harus dilakukan Amazon SES jika tidak dapat menemukan catatan MX yang diperlukan pada saat pengiriman. Pilih salah satu opsi berikut:
      - Gunakan domain MAIL FROM default – Jika catatan MX domain MAIL FROM kustom tidak diatur dengan benar, Amazon SES akan menggunakan subdomain dari amazonses.com. Subdomain bervariasi berdasarkan Wilayah AWS tempat Anda menggunakan Amazon SES.
      - Tolak pesan - Jika data MX MAIL FROM domain kustom tidak diatur dengan benar, Amazon SES akan mengembalikan kesalahan. `MailFromDomainNotVerified` Jika Anda memilih opsi ini, maka email yang akan dikirim dari domain ini akan ditolak secara otomatis.

3. Untuk Publikasikan catatan DNS ke Route53, jika domain Anda di-host melalui Amazon Route 53, Anda memiliki opsi untuk membiarkan SES menerbitkan data TXT dan MX terkait pada saat pembuatan dengan membiarkan Enabled dicentang. Jika Anda lebih suka mempublikasikan catatan ini nanti, kosongkan kotak centang Diaktifkan. (Anda dapat kembali di lain waktu untuk mempublikasikan catatan ke Route 53 dengan mengedit identitas - lihatthe section called “Mengedit identitas menggunakan konsol”.)

8. (Opsional) Untuk mengonfigurasi verifikasi berbasis DKIM yang disesuaikan di luar pengaturan default SES yang menggunakan Easy DKIM dengan panjang nyanyian 2048 bit, di bawah Memverifikasi domain Anda, perluas pengaturan DKIM Lanjutan dan pilih jenis DKIM yang ingin Anda konfigurasikan:

   a. DKIM mudah:

      i. Di bidang Jenis identitas, pilih Easy DKIM.

      ii. Di bidang panjang kunci penandatanganan DKIM, pilih RSA_2048_BIT atau RSA_1024_BIT.

      iii. Untuk Publikasikan catatan DNS ke Route53, jika domain Anda di-host melalui Amazon Route 53, Anda memiliki opsi untuk membiarkan SES menerbitkan catatan CNAME terkait pada saat pembuatan dengan membiarkan Enabled dicentang. Jika Anda lebih suka mempublikasikan catatan ini nanti, kosongkan kotak centang Diaktifkan. (Anda dapat kembali di lain waktu untuk mempublikasikan catatan ke Route 53 dengan mengedit identitas - lihatthe section called “Mengedit identitas menggunakan konsol”.)

   b. Menyediakan token otentikasi DKIM (BYODKIM):

      i. Pastikan Anda telah membuat key pair public-private dan telah menambahkan kunci publik ke penyedia host DNS Anda. Untuk informasi selengkapnya, lihat the section called “BYODKIM - Bawa DKIM Anda Sendiri”.

      ii. Di bidang Jenis identitas, pilih Berikan token otentikasi DKIM (BYODKIM).

      iii. Untuk kunci Privat, tempel kunci pribadi yang Anda buat dari key pair publik-pribadi Anda. Kunci pribadi harus menggunakan setidaknya enkripsi RSA 1024-bit dan hingga 2048-bit, dan harus dikodekan menggunakan pengkodean base64 (PEM).

      > **Note**
      >
      > Anda harus menghapus baris pertama dan terakhir (`-----BEGIN PRIVATE KEY-----`dan`-----END PRIVATE KEY-----`, masing-masing) dari kunci

pribadi yang dihasilkan. Selain itu, Anda harus menghapus jeda baris di kunci pribadi yang dihasilkan. Nilai yang dihasilkan adalah string karakter tanpa spasi atau jeda baris.

iv. Untuk nama Pemilih, masukkan nama pemilih yang akan ditentukan dalam pengaturan DNS domain Anda.

9. Pastikan kotak Diaktifkan dicentang di bidang tanda tangan DKIM.
10. (Opsional) Tambahkan satu atau beberapa Tag ke identitas domain Anda dengan menyertakan kunci tag dan nilai opsional untuk kunci:
    1. Pilih Tambahkan tanda baru dan masukkan Kunci. Anda dapat menambahkan Nilai untuk tanda.
    2. Ulangi agar tag tambahan tidak melebihi 50, atau pilih Hapus untuk menghapus tag.
11. Pilih Buat identitas.

Sekarang setelah Anda membuat dan mengonfigurasi identitas domain Anda dengan DKIM, Anda harus menyelesaikan proses verifikasi dengan penyedia DNS Anda - lanjutkan ke the section called “Memverifikasi identitas domain” dan ikuti prosedur otentikasi DNS untuk jenis DKIM yang Anda konfigurasikan identitas Anda.

Memverifikasi identitas domain DKIM dengan penyedia DNS Anda

Setelah Anda membuat identitas domain yang dikonfigurasi dengan DKIM, Anda harus menyelesaikan proses verifikasi dengan penyedia DNS Anda dengan mengikuti prosedur otentikasi masing-masing untuk jenis DKIM yang Anda pilih.

Jika Anda belum membuat identitas domain, lihatthe section called “Membuat identitas domain”.

> **Note**
>
> Memverifikasi identitas domain memerlukan akses ke pengaturan DNS domain. Perubahan pada pengaturan ini dapat memakan waktu hingga 72 jam untuk disebarkan.

Untuk memverifikasi identitas domain DKIM dengan penyedia DNS Anda

1. Dari tabel Loaded identities, pilih domain yang ingin Anda verifikasi.

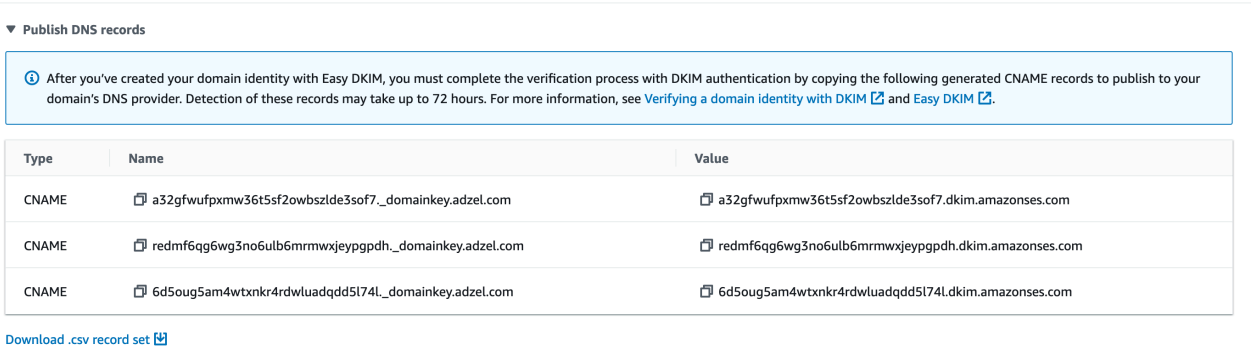
2. Pada tab Otentikasi halaman detail identitas, perluas Publikasikan catatan DNS.
3. Bergantung pada ragam DKIM yang Anda gunakan untuk mengonfigurasi domain Anda, Easy DKIM atau BYODKIM, ikuti instruksi masing-masing:

   Easy DKIM

   Untuk memverifikasi domain yang dikonfigurasi dengan Easy DKIM

   1. Dari tabel Publikasikan catatan DNS, salin tiga catatan CNAME yang muncul di bagian ini untuk dipublikasikan (ditambahkan) ke penyedia DNS Anda. Sebagai alternatif, Anda dapat memilih Unduh set catatan .csv untuk menyimpan salinan catatan ke komputer Anda.

      Gambar berikut menunjukkan contoh catatan CNAME untuk dipublikasikan ke penyedia DNS Anda.

   2. Tambahkan catatan CNAME ke pengaturan DNS domain Anda masing-masing dari penyedia host DNS Anda:

      - Semua penyedia host DNS (tidak termasuk Route 53) — Masuk ke DNS atau penyedia hosting web domain Anda, lalu tambahkan catatan CNAME yang berisi nilai yang Anda salin atau simpan sebelumnya. Penyedia yang berbeda memiliki prosedur yang berbeda untuk memperbarui catatan DNS. Lihat tabel penyedia DNS/hosting mengikuti prosedur ini.

        > **Note**
        >
        > Sejumlah kecil penyedia DNS tidak mengizinkan Anda untuk menyertakan garis bawah (_) di nama catatan. Namun, garis bawah di nama catatan DKIM diperlukan. Jika penyedia DNS Anda tidak mengizinkan Anda untuk

memasukkan garis bawah di nama catatan, kontak tim dukungan pelanggan penyedia untuk mendapatkan bantuan.

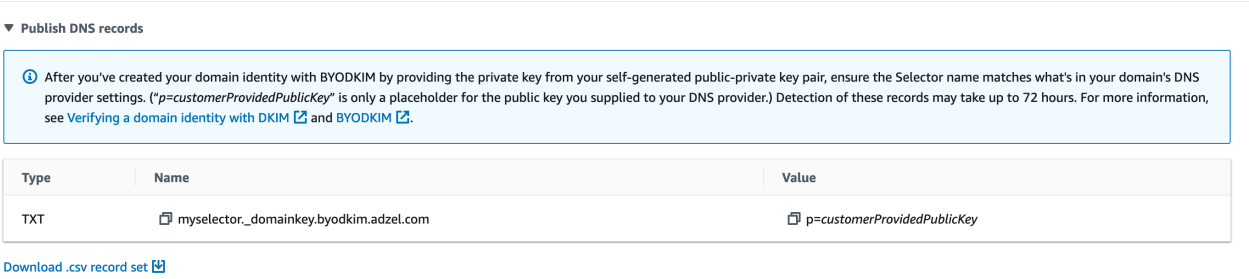
- Route 53 sebagai penyedia host DNS Anda — Jika Anda menggunakan Route 53 pada akun yang sama dengan yang Anda gunakan saat mengirim email menggunakan SES, dan domain terdaftar, SES secara otomatis memperbarui pengaturan DNS untuk domain Anda jika Anda mengaktifkan SES untuk mempublikasikannya pada saat pembuatan. Jika tidak, Anda dapat dengan mudah mempublikasikannya ke Route 53 dengan klik tombol setelah pembuatan - lihatthe section called “Mengedit identitas menggunakan konsol”. Jika pengaturan DNS Anda tidak diperbarui secara otomatis, atau Anda ingin menambahkan catatan CNAME ke Route 53 yang tidak berada di akun yang sama yang Anda gunakan saat mengirim email menggunakan SES, selesaikan prosedur dalam Mengedit catatan.
- Jika Anda tidak yakin siapa penyedia DNS Anda – Minta administrator sistem Anda untuk informasi selengkapnya.

BYODKIM

Untuk memverifikasi domain yang dikonfigurasi dengan BYODKIM

1. Untuk rekap, saat Anda membuat domain dengan BYODKIM, atau mengonfigurasi domain yang ada dengan BYODKIM, Anda menambahkan kunci pribadi (dari key pair publik-pribadi yang dibuat sendiri) dan awalan nama pemilih ke dalam bidang masing-masing di halaman Pengaturan DKIM Tingkat Lanjut konsol SES. Sekarang Anda harus menyelesaikan proses verifikasi dengan memperbarui catatan berikut untuk penyedia host DNS Anda.
2. Dari tabel Publikasikan catatan DNS, salin catatan nama pemilih yang muncul di kolom Nama yang akan dipublikasikan (ditambahkan) ke penyedia DNS Anda. Atau, Anda dapat memilih Unduh set catatan.csv untuk menyimpan salinannya ke komputer Anda.

   Gambar berikut menunjukkan contoh catatan nama pemilih untuk dipublikasikan ke penyedia DNS Anda.



3. Masuk ke DNS atau penyedia hosting web domain Anda, lalu tambahkan catatan nama pemilih yang Anda salin atau simpan sebelumnya. Penyedia yang berbeda memiliki prosedur yang berbeda untuk memperbarui catatan DNS. Lihat tabel penyedia DNS/hosting mengikuti prosedur ini.

   > **Note**
   >
   > Sejumlah kecil penyedia DNS tidak mengizinkan Anda untuk menyertakan garis bawah (_) di nama catatan. Namun, garis bawah di nama catatan DKIM diperlukan. Jika penyedia DNS Anda tidak mengizinkan Anda untuk memasukkan garis bawah di nama catatan, kontak tim dukungan pelanggan penyedia untuk mendapatkan bantuan.

4. Jika Anda belum melakukannya, pastikan untuk menambahkan kunci publik dari key pair public-private yang dibuat sendiri ke DNS atau penyedia hosting web domain Anda.

   Perhatikan bahwa dalam tabel Publikasikan catatan DNS, catatan kunci publik yang muncul di kolom Nilai hanya menampilkan, “p = customerProvidedPublicKunci”, sebagai pengganti untuk nilai kunci publik yang Anda simpan ke komputer atau diberikan ke penyedia DNS Anda.

   > **Note**
   >
   > Ketika Anda mempublikasikan (menambahkan) kunci publik Anda ke penyedia DNS Anda, itu harus diformat sebagai berikut:
   >
   > - Anda harus menghapus baris pertama dan terakhir (`-----BEGIN PUBLIC KEY-----` dan `-----END PUBLIC KEY-----`, secara berturut-turut) dari kunci publik yang dihasilkan. Selain itu, Anda harus menghapus jeda baris di kunci publik yang dihasilkan. Nilai yang dihasilkan adalah string karakter tanpa spasi atau jeda baris.

- Anda harus menyertakan p= awalan seperti yang ditunjukkan pada kolom Nilai dalam tabel Publikasikan catatan DNS.

4. Diperlukan waktu hingga 72 jam agar perubahan pada pengaturan DNS menyebar. Segera setelah Amazon SES mendeteksi semua catatan DKIM yang diperlukan dalam pengaturan DNS domain Anda, proses verifikasi selesai. Konfigurasi DKIM domain Anda muncul sebagai Berhasil dan Identitas status muncul sebagai Terverifikasi.
5. Jika ingin mengonfigurasi dan memverifikasi domain MAIL FROM kustom, ikuti prosedurnya diMengonfigurasi domain MAIL FROM kustom Anda.

Tabel berikut mencakup tautan ke dokumentasi untuk beberapa penyedia DNS yang banyak digunakan. Daftar ini tidak lengkap dan tidak menandakan dukungan; demikian juga, jika penyedia DNS Anda tidak terdaftar, itu tidak berarti Anda tidak dapat menggunakan domain dengan Amazon SES.

Penyedia DNS/Hosting	Tautan dokumentasi
GoDaddy	Tambahkan catatan CNAME (tautan eksternal)
DreamHost	Bagaimana cara menambahkan catatan DNS kustom? (tautan eksternal)
Cloudflare	Mengelola catatan DNS di Cloudflare (tautan eksternal)
HostGator	Kelola Rekaman DNS HostGator dengan/eNom (tautan eksternal)
Namecheap	Bagaimana cara menambahkan catatan TXT/SPF/DKIM/DMARC untuk domain saya? (tautan eksternal)
Names.co.uk	Mengubah Pengaturan DNS domain (tautan eksternal)
Wix	Menambahkan atau Memperbarui Catatan CNAME di Akun Wix (tautan eksternal)

Memecahkan masalah verifikasi domain

Jika Anda menyelesaikan langkah-langkah di atas, namun domain Anda tidak terverifikasi setelah 72 jam, periksa langkah berikut:

- Pastikan bahwa Anda memasukkan nilai untuk data DNS pada kolom yang benar. Beberapa penyedia DNS mengacu pada kolom Nama/Host sebagai Host atau Nama Host. Selain itu, beberapa penyedia layanan mengacu pada kolom Nilai catatan sebagai Poin ke atau Hasil.
- Pastikan bahwa penyedia Anda tidak secara otomatis menambahkan nama domain Anda ke nilai Nama/Host yang Anda masukkan dalam data DNS. Beberapa penyedia menambahkan nama domain tanpa menunjukkan bahwa mereka telah melakukannya. Jika penyedia Anda menambahkan nama domain Anda ke nilai Nama/Host, hapus nama domain dari akhir nilai. Anda juga dapat mencoba menambahkan periode ke akhir nilai dalam catatan DNS. Periode ini menunjukkan kepada penyedia bahwa nama domain sepenuhnya memenuhi syarat.
- Karakter garis bawah (_) diperlukan dalam nilai Nama/host dari setiap catatan DNS. Jika penyedia Anda tidak mengizinkan garis bawah di nama catatan DNS, hubungi tim dukungan pelanggan penyedia untuk mendapatkan bantuan tambahan.
- Catatan validasi yang harus ditambahkan ke pengaturan DNS domain berbeda untuk setiap Wilayah AWS. Jika Anda ingin menggunakan domain untuk mengirim email dari beberapa Wilayah AWS, Anda harus membuat dan memverifikasi identitas domain secara terpisah untuk masing-masing Wilayah tersebut.

Membuat identitas alamat email

Selesaikan prosedur berikut untuk membuat identitas alamat email dengan menggunakan konsol Amazon SES.

Untuk membuat identitas alamat email (konsol)

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas Terverifikasi.
3. Pilih Buat identitas.
4. Di bawah Detail identitas, pilih domain sebagai tipe identitas yang ingin Anda buat.
5. Untuk Alamat email, masukkan alamat email yang ingin Anda gunakan. Alamat email harus berupa alamat yang dapat menerima email dan dapat Anda akses.

6. (Opsional) Jika Anda ingin Menetapkan set konfigurasi default, pilih kotak centang.

   1. Untuk set konfigurasi Default, pilih set konfigurasi yang ada yang ingin Anda tetapkan ke identitas Anda. Jika Anda belum membuat rangkaian konfigurasi apa pun, lihat Set konfigurasi.

      > **Note**
      >
      > Amazon SES memilih default ke konfigurasi yang ditetapkan saat tidak ada set lain yang ditentukan pada saat pengiriman. Jika set konfigurasi ditentukan, Amazon SES menerapkan set yang ditentukan sebagai pengganti set default.

7. (Opsional) Tambahkan satu atau beberapa Tag ke identitas domain Anda dengan menyertakan kunci tag dan nilai opsional untuk kunci:

   1. Pilih Tambahkan tanda baru dan masukkan Kunci. Anda dapat menambahkan Nilai untuk tanda.
   2. Ulangi agar tag tambahan tidak melebihi 50, atau pilih Hapus untuk menghapus tag.

8. Untuk membuat identitas alamat email Anda, pilih Buat identitas. Setelah dibuat, Anda akan menerima email verifikasi dalam waktu lima menit. Langkah selanjutnya adalah memverifikasi alamat email Anda dengan mengikuti prosedur verifikasi di bagian berikutnya.

   > **Note**
   >
   > Anda dapat menyesuaikan pesan yang dikirim ke alamat email yang Anda coba verifikasi. Untuk informasi selengkapnya, lihat the section called “Menggunakan templat email verifikasi kustom”.

Sekarang setelah Anda membuat identitas alamat email Anda, Anda harus menyelesaikan proses verifikasi - lanjutkan kethe section called “Memverifikasi identitas alamat email”.

Memverifikasi identitas alamat email

Setelah Anda membuat identitas alamat email Anda, Anda harus menyelesaikan proses verifikasi.

Jika Anda belum membuat identitas alamat email, lihatthe section called “Membuat identitas alamat email”.

Untuk memverifikasi identitas alamat email

1. Periksa kotak masuk alamat email yang digunakan untuk membuat identitas Anda dan cari email dari no-reply-aws @amazon .com.
2. Buka email dan klik tautannya untuk menyelesaikan proses verifikasi alamat email tersebut. Setelah selesai, Status identitas berubah menjadi Terverifikasi.

Memecahkan masalah verifikasi alamat email

Jika Anda tidak menerima email verifikasi dalam waktu lima menit sejak pembuatan identitas, maka coba langkah-langkah pemecahan masalah berikut:

- Pastikan bahwa Anda memasukkan alamat email dengan benar.
- Pastikan bahwa alamat email yang Anda coba verifikasi dapat menerima email. Anda dapat menguji ini dengan menggunakan alamat email lain untuk mengirim email percobaan ke alamat yang ingin Anda verifikasi.
- Periksa folder email sampah Anda.
- Tautan di email verifikasi kedaluwarsa setelah 24 jam. Untuk mengirim email verifikasi baru, pilih Kirim Ulang di bagian atas halaman detail identitas.

Membuat dan memverifikasi identitas dan menetapkan konfigurasi default yang ditetapkan pada saat yang sama

Anda dapat menggunakan CreateEmailIdentityoperasi di Amazon SES API v2 untuk membuat identitas email baru dan mengatur konfigurasi defaultnya pada saat yang bersamaan.

> **Note**
>
> Sebelum Anda menyelesaikan prosedur di bagian ini, Anda harus memasang dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Command Line Interface.

Untuk mengatur konfigurasi default yang ditetapkan menggunakan AWS CLI

- Pada baris perintah, masukkan perintah berikut untuk menggunakan CreateEmailIdentityoperasi.

```
aws sesv2 create-email-identity --email-identity ADDRESS-OR-DOMAIN --configuration-set-
name CONFIG-SET
```

Di perintah sebelumnya, ganti *ADDRESS-OR-DOMAIN* dengan identitas email yang ingin Anda verifikasi. Ganti *CONFIG-SET* dengan nama set konfigurasi yang ingin Anda atur sebagai set konfigurasi default untuk identitas tersebut.

Jika perintah berhasil dijalankan, perintah keluar tanpa memberikan output apa pun.

Untuk memverifikasi alamat email Anda

1. Periksa kotak masuk untuk alamat email yang Anda verifikasi. Anda akan menerima pesan dengan baris subjek berikut: “Amazon Web Services - Permintaan Verifikasi Alamat Email di wilayah *RegionName*,” di mana *RegionName*nama tempat Anda mencoba memverifikasi alamat email tersebut. Wilayah AWS

   Buka pesan, dan kemudian klik tautan di dalamnya.

   > **Note**
   >
   > Tautan di pesan verifikasi kedaluwarsa 24 jam setelah pesan dikirim. Jika 24 jam telah berlalu sejak Anda menerima email verifikasi, ulangi langkah 1–5 untuk menerima email verifikasi dengan tautan yang valid.

2. Di konsol Amazon SES, di bawah Manajemen Identitas, pilih Alamat Email. Di daftar alamat email, cari alamat email yang Anda verifikasi. Jika alamat email telah terverifikasi, nilai di kolom Status adalah "terverifikasi".

Untuk memverifikasi domain Anda

Jika Anda memasukkan nama domain untuk `--email-identity` parameter dalam prosedur baris perintah di atas, lihat Memverifikasi identitas domain untuk informasi selengkapnya.

Menggunakan templat email verifikasi kustom

Ketika Anda mencoba untuk memverifikasi alamat email, Amazon SES mengirimkan email ke alamat yang menyerupai contoh yang ditunjukkan pada citra berikut.

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=AKIADQKE4EXAMPLE&Context=10987654321&Identity.IdentityName=recipient%40example.com&Identity.IdentityType=EmailAddress&Namespace=Bacon&Operation=ConfirmVerification&Signature=TJDufFhYYK1fSHCSBq4cjbodBQq%2FnyyZgzjqZ%2BXsDYEXAMPLE&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2017-12-06T19%3A53%3A12.311Z

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 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.

Beberapa pelanggan Amazon SES membangun aplikasi (seperti rangkaian pemasaran email atau sistem tiket) yang mengirim email melalui Amazon SES atas nama pelanggan mereka sendiri. Untuk pengguna akhir aplikasi ini, proses verifikasi email dapat membuat bingung: email verifikasi menggunakan merek Amazon SES, bukan merek aplikasi, dan pengguna akhir tersebut tidak pernah mendaftar untuk menggunakan Amazon SES secara langsung.

Jika kasus penggunaan Amazon SES Anda mengharuskan pelanggan Anda untuk memverifikasi alamat email mereka untuk digunakan dengan Amazon SES, Anda dapat membuat email verifikasi yang disesuaikan. Email yang disesuaikan ini membantu mengurangi kebingungan pelanggan dan meningkatkan laju saat pelanggan Anda menyelesaikan proses pendaftaran.

> **Note**
>
> Untuk menggunakan fitur ini, akun Amazon SES Anda harus keluar dari sandbox. Untuk informasi lebih lanjut, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).

Topik di bagian ini:

- Membuat templat email verifikasi kustom
- Mengedit templat email verifikasi kustom
- Mengirim email verifikasi menggunakan templat kustom
- Pertanyaan yang sering diajukan email verifikasi kustom

Membuat templat email verifikasi kustom

Untuk membuat email verifikasi khusus, gunakan operasi API `CreateCustomVerificationEmailTemplate`. Operasi ini membutuhkan input berikut:

Atribut	Deskripsi
`TemplateName`	Nama templat. Nama yang Anda tentukan harus unik.
`FromEmailAddress`	Alamat email tempat email verifikasi dikirim. Alamat atau domain yang Anda tentukan harus diverifikasi untuk digunakan dengan akun Amazon SES Anda. **Note** Atribut `FromEmailAddress` tidak mendukung nama tampilan (juga dikenal sebagai nama "ramah dari").
`TemplateSubject`	Baris subjek untuk email verifikasi.
`TemplateContent`	Badan email. Badan email dapat berisi HTML, dengan pembatasan tertentu. Untuk informasi lebih lanjut, lihat Pertanyaan yang sering diajukan email verifikasi kustom.
`SuccessRedirectionURL`	URL yang dikirim pengguna, jika alamat email mereka berhasil diverifikasi.
`FailureRedirectionURL`	URL yang dikirim pengguna, jika alamat email mereka gagal diverifikasi.

Anda dapat menggunakan SDK AWS atau AWS CLI untuk membuat templat email verifikasi kustom dengan operasi `CreateCustomVerificationEmailTemplate`. Untuk mempelajari selengkapnya tentang SDK AWS, lihat Alat untuk Amazon Web Services. Untuk informasi selengkapnya tentang AWS CLI, lihat Antarmuka Baris Perintah AWS.

Bagian berikut mencakup prosedur untuk membuat email verifikasi kustom menggunakan AWS CLI. Prosedur di bagian ini menganggap Anda telah meneginstal dan mengonfigurasi AWS CLI. Untuk

informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan Pengguna AWS Command Line Interface.

> **Note**
>
> Untuk menyelesaikan prosedur di bagian ini, Anda harus menggunakan AWS CLI versi 1.14.6 atau yang lebih baru. Untuk hasil terbaik, tingkatkan AWS CLI ke versi terbaru. Untuk informasi selengkapnya tentang memperbarui AWS CLI, lihat Menginstal AWS Command Line Interface di Panduan Pengguna AWS Command Line Interface.

1. Di editor teks, buat file baru. Tempelkan konten berikut ke editor:

```
{
  "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**
>
> Untuk memudahkan dalam membaca pada contoh sebelumnya, atribut `TemplateContent` berisi jeda baris. Jika Anda menempelkan contoh sebelumnya ke file teks Anda, hapus jeda baris sebelum melanjutkan.

Ganti nilai `TemplateName`, `FromEmailAddress`, `TemplateSubject`, `TemplateContent`, `SuccessRedirectionURL`, dan `FailureRedirectionURL` dengan nilai-nilai milik Anda sendiri.

> **Note**
>
> Alamat email yang Anda tentukan untuk parameter `FromEmailAddress` harus diverifikasi, atau harus berupa alamat pada domain terverifikasi. Untuk informasi lebih lanjut, lihat Identitas terverifikasi di Amazon SES.

Setelah selesai, simpan file sebagai `customverificationemail.json`.

2. Di baris perintah, ketik perintah berikut untuk membuat templat email verifikasi kustom:

```
aws sesv2 create-custom-verification-email-template --cli-input-json file://
customverificationemail.json
```

3. (Opsional) Anda dapat mengonfirmasi bahwa templat dibuat dengan mengetik perintah berikut:

```
aws sesv2 list-custom-verification-email-templates
```

Mengedit templat email verifikasi kustom

Anda dapat mengedit templat email verifikasi kustom menggunakan operasi `UpdateCustomVerificationEmailTemplate`. Operasi ini menerima input yang sama sebagai operasi `CreateCustomVerificationEmailTemplate` (yaitu atribut `TemplateName`, `FromEmailAddress`, `TemplateSubject`, `TemplateContent`, `SuccessRedirectionURL`, dan `FailureRedirectionURL`). Namun, dengan operasi `UpdateCustomVerificationEmailTemplate`, tidak ada atribut tersebut diperlukan. Ketika Anda meneruskan nilai untuk `TemplateName` yang sama dengan nama templat email verifikasi kustom yang ada, atribut yang Anda tentukan menimpa atribut yang awalnya ada di templat.

Mengirim email verifikasi menggunakan templat kustom

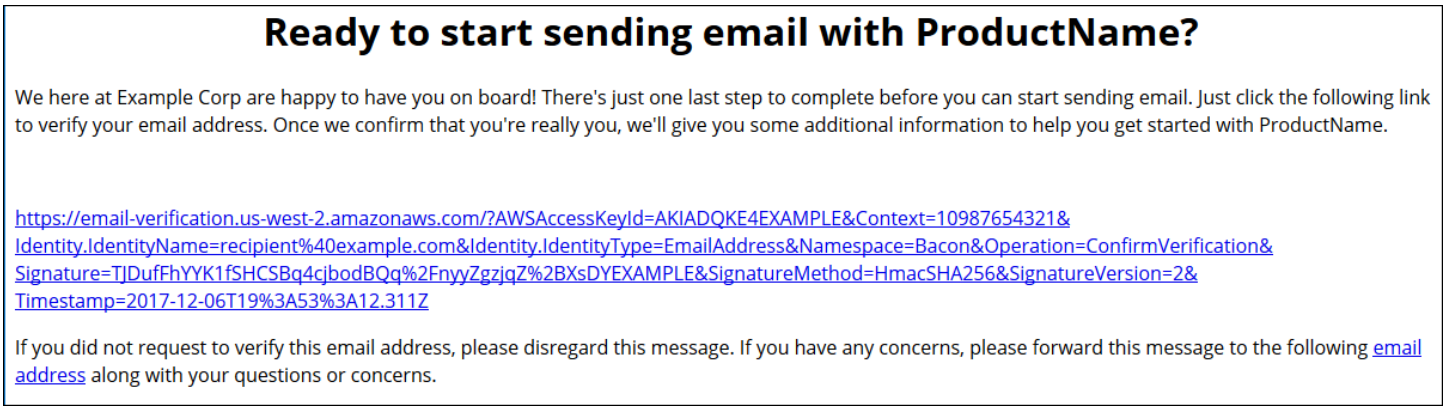
Setelah Anda membuat setidaknya satu template email verifikasi kustom, Anda dapat mengirimkannya ke pelanggan Anda dengan memanggil operasi SendCustomVerificationEmailAPI.

Anda dapat memanggil operasi `SendCustomVerificationEmail` dengan menggunakan salah satu dari SDK AWS atau AWS CLI. Operasi `SendCustomVerificationEmail` membutuhkan input berikut:

Atribut	Deskripsi
`EmailAddress`	Alamat email yang sedang diverifikasi.
`TemplateName`	Nama templat email verifikasi kustom yang dikirim ke alamat email yang sedang diverifikasi.
`ConfigurationSetName`	(Opsional) Nama set konfigurasi untuk digunakan saat mengirim email verifikasi.

Misalnya, anggap pelanggan Anda mendaftar untuk layanan Anda menggunakan formulir di aplikasi Anda. Ketika pelanggan melengkapi formulir dan mengirimkannya, aplikasi Anda memanggil operasi `SendCustomVerificationEmail`, meneruskan alamat email pelanggan dan nama templat yang ingin Anda gunakan.

Pelanggan Anda menerima email yang menggunakan templat email yang disesuaikan yang Anda buat. Amazon SES secara otomatis menambahkan tautan unik ke penerima, dan juga sangkalan singkat. Citra berikut menunjukkan contoh email verifikasi yang menggunakan templat yang dibuat di Membuat templat email verifikasi kustom.


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%2FnyyZgzjqZ%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.

Pertanyaan yang sering diajukan email verifikasi kustom

Bagian ini berisi jawaban atas pertanyaan yang sering diajukan tentang fitur templat email verifikasi kustom.

Q1. Berapa banyak templat email verifikasi kustom yang dapat saya buat?

Anda dapat membuat hingga 50 templat email verifikasi khusus per akun Amazon SES.

Q2. Bagaimana email verifikasi kustom ditampilkan kepada penerima?

Email verifikasi kustom mencakup konten yang Anda tentukan saat membuat templat, diikuti dengan tautan yang harus diklik penerima untuk memverifikasi alamat email mereka.

P3. Dapatkah saya melihat pratinjau email verifikasi kustom?

Untuk melihat pratinjau email verifikasi kustom, gunakan operasi `SendCustomVerificationEmail` untuk mengirim email verifikasi ke alamat yang Anda miliki. Jika Anda tidak mengeklik tautan verifikasi, Amazon SES tidak membuat identitas baru. Jika Anda mengeklik tautan verifikasi, Anda dapat menghapus identitas yang baru dibuat menggunakan operasi `DeleteIdentity`.

T4. Dapatkah saya menyertakan citra di templat email verifikasi kustom saya?

Anda dapat melampirkan citra di HTML untuk templat Anda dengan menggunakan pengodean base64. Ketika Anda melampirkan citra dengan cara ini, Amazon SES secara otomatis mengubahnya menjadi lampiran. Anda dapat mengodekan citra di baris perintah dengan memberikan salah satu dari perintah berikut:

Linux, macOS, or Unix

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

Windows

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

Ganti *`imagefile.png`* dengan nama file yang ingin Anda kodekan. Di kedua perintah di atas, citra yang dikodekan base64 disimpan ke `output.txt`.

Anda dapat melampirkan citra yang dikodekan base64 dengan menyertakan hal berikut di HTML untuk templat: `<img src="`*`base64EncodedImage`*`"/>`

Dalam contoh sebelumnya, ganti *`png`* dengan tipe file citra yang dikodekan (seperti jpg atau gif), dan ganti *`base64EncodedImage`* dengan citra yang dikodekan base64 (yaitu isi `output.txt` dari salah satu perintah sebelumnya).

T5. Apakah ada batasan untuk konten yang dapat saya sertakan di templat email verifikasi kustom?

Ukuran templat email verifikasi kustom tidak boleh melebihi 10 MB. Selain itu, templat email verifikasi kustom yang berisi HTML hanya dapat menggunakan tanda dan atribut yang tercantum di tabel berikut.

Tanda HTML	Atribut yang diizinkan
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

Tanda HTML	Atribut yang diizinkan
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
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

Tanda HTML	Atribut yang diizinkan
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

Tanda HTML	Atribut yang diizinkan
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
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

Tanda HTML	Atribut yang diizinkan
`var`	`class, id, style, title`

> **Note**
>
> Templat email verifikasi kustom tidak boleh menyertakan tanda komentar.

T6. Berapa banyak alamat email terverifikasi yang muat di akun saya?

Akun Amazon SES Anda dapat memiliki hingga 10.000 identitas terverifikasi di setiap Wilayah AWS. Di Amazon SES, identitas mencakup domain dan alamat email terverifikasi.

T7. Dapatkah saya membuat templat email verifikasi kustom menggunakan konsol Amazon SES?

Saat ini, konsol hanya memungkinkan untuk membuat, mengedit, dan menghapus email verifikasi kustom menggunakan API Amazon SES.

T8. Dapatkah saya melacak peristiwa pembukaan dan pengeklikan yang terjadi ketika pelanggan menerima email verifikasi kustom?

Email verifikasi kustom tidak dapat menyertakan pelacakan peristiwa pembukaan atau pengeklikan.

T9. Dapatkah email verifikasi kustom menyertakan header kustom?

Email verifikasi kustom tidak dapat menyertakan header kustom.

T10. Dapatkah saya menghapus teks yang muncul di bagian bawah email verifikasi kustom?

Teks berikut secara otomatis ditambahkan ke akhir setiap email verifikasi kustom dan tidak dapat dihapus:

Jika Anda tidak meminta untuk memverifikasi alamat email ini, harap abaikan pesan ini.

T11. Apakah email verifikasi kustom ditandatangani DKIM?

Agar email verifikasi ditandatangani DKIM, alamat email yang Anda tentukan di atribut `FromEmailAddress` ketika Anda membuat templat email verifikasi harus dikonfigurasi untuk menghasilkan tanda tangan DKIM. Untuk informasi selengkapnya tentang mengatur DKIM untuk domain dan alamat email, lihat the section called “Mengautentikasi Email dengan DKIM”.

T12. Mengapa operasi API templat email verifikasi kustom tidak muncul di SDK atau CLI?

Jika Anda tidak dapat menggunakan operasi templat email verifikasi kustom di SDK atau AWS CLI, Anda mungkin menggunakan versi lama dari SDK atau CLI. Operasi templat email verifikasi kustom tersedia di SDK dan CLI berikut:

- AWS Command Line Interface versi 1.14.6 atau yang lebih baru
- AWS SDK for .NET versi 3.3.205.0 atau yang lebih baru
- AWS SDK for C++ versi 1.3.20170531.19 atau yang lebih baru
- AWS SDK for Go versi 1.12.43 atau yang lebih baru
- AWS SDK for Java versi 1.11.245 atau yang lebih baru
- AWS SDK for JavaScript versi 2.166.0 atau yang lebih baru
- AWS SDK for PHP versi 3.45.2 atau yang lebih baru
- AWS SDK for Python (Boto) versi 1.5.1 atau yang lebih baru
- Gem `aws-sdk-ses` versi 1.5.0 atau yang lebih baru di AWS SDK for Ruby

T13. Mengapa saya menerima kesalahan **`ProductionAccessNotGranted`** ketika mengirim email verifikasi kustom?

Kesalahan `ProductionAccessNotGranted` menunjukkan bahwa akun Anda masih berada di sandbox Amazon SES. Anda hanya dapat mengirim email verifikasi kustom jika akun Anda telah dihapus dari sandbox. Untuk informasi lebih lanjut, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).

Mengelola identitas di Amazon SES

Di konsol Amazon SES, Anda dapat melihat identitas yang dibuat untuk masing-masing identitas Wilayah AWS, membuka identitas untuk melihat dan mengedit pengaturan detailnya, mengaitkan set konfigurasi default, atau menghapus satu atau beberapa identitas.

> **Note**
>
> Prosedur yang direferensikan dalam bagian ini hanya berlaku untuk identitas yang dipilih. Wilayah AWS Untuk mengelola identitas yang dibuat di lebih dari satu Wilayah, ulangi prosedur untuk masing-masing Wilayah AWS.

Konten

- Melihat identitas menggunakan konsol SES
- Menghapus identitas menggunakan konsol SES
- Mengedit identitas menggunakan konsol SES
- Mengedit identitas untuk menggunakan set konfigurasi default menggunakan SES API
- Ambil set konfigurasi default yang digunakan oleh identitas menggunakan SES API
- Ganti set konfigurasi default saat ini yang digunakan oleh identitas menggunakan SES API

Melihat identitas menggunakan konsol SES

Anda dapat menggunakan konsol Amazon SES untuk melihat identitas domain dan alamat email yang diverifikasi atau sedang menunggu verifikasi. Anda juga dapat melihat identifikasi yang tidak berhasil diverifikasi.

Untuk melihat identitas domain dan alamat email

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di konsol, gunakan pemilih Wilayah untuk memilih daftar identitas yang ingin Anda lihat. Wilayah AWS

   > **Note**
   >
   > Prosedur ini hanya menampilkan daftar identitas untuk Wilayah AWS yang di pilih.

3. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi. Tabel identitas yang dimuat menampilkan identitas domain dan alamat email. Kolom Status menampilkan apakah identitas telah diverifikasi, sedang menunggu verifikasi, atau telah gagal dalam proses verifikasi - definisi semua nilai status yang mungkin adalah sebagai berikut:

   - Terverifikasi - identitas Anda berhasil diverifikasi untuk dikirim dalam SES.
   - Kegagalan — SES tidak dapat memverifikasi identitas Anda. Jika itu adalah domain, itu berarti SES tidak dapat mendeteksi catatan DNS dalam waktu 72 jam. Jika itu adalah alamat email, itu berarti email verifikasi yang dikirim ke alamat email tidak diakui dalam waktu 24 jam.
   - Tertunda — SES masih mencoba memverifikasi identitas.

- Kegagalan Sementara — untuk domain yang diverifikasi sebelumnya, SES akan secara berkala memeriksa catatan DNS yang diperlukan untuk verifikasi. Jika pada titik tertentu, SES tidak dapat mendeteksi catatan, status akan berubah menjadi Kegagalan Sementara. SES akan memeriksa ulang data DNS selama 72 jam, dan jika tidak dapat mendeteksi catatan, status domain akan berubah menjadi Kegagalan. Jika dapat mendeteksi rekaman, status domain akan berubah menjadi Verified.
- Belum dimulai — Anda belum memulai proses verifikasi.

4. Untuk mengurutkan identitas berdasarkan status verifikasi, pilih di kolom Status.
5. Untuk melihat halaman detail identitas, pilih identitas yang ingin Anda lihat.

Menghapus identitas menggunakan konsol SES

Anda dapat menggunakan konsol Amazon SES untuk menghapus domain atau identitas alamat email dari akun Anda di yang dipilih Wilayah AWS.

Untuk menghapus identitas domain atau alamat email

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di konsol, gunakan pemilih Wilayah untuk memilih Wilayah AWS dari mana Anda ingin menghapus satu atau beberapa identitas.
3. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.

   Tabel Identitas yang dimuat menampilkan daftar identitas domain dan alamat email.
4. Di kolom Identitas, pilih identitas yang ingin Anda hapus. Anda dapat menghapus beberapa identitas dengan mencentang kotak di bagian samping setiap identitas yang ingin Anda hapus.
5. Pilih Hapus.

Mengedit identitas menggunakan konsol SES

Anda dapat menggunakan konsol Amazon SES untuk mengedit domain atau identitas alamat email di akun Anda di yang dipilih Wilayah AWS.

Untuk mengedit identitas domain atau alamat email

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di konsol, gunakan pemilih Wilayah untuk memilih Wilayah AWS dari mana Anda ingin mengedit satu atau beberapa identitas.
3. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.

   Tabel Identitas yang dimuat menampilkan daftar identitas domain dan alamat email.
4. Di kolom Identitas, pilih identitas yang ingin Anda edit (dengan mengklik langsung pada nama identitas sebagai lawan memilih kotak centang).
5. Pada halaman detail identitas, pilih tab yang berisi kategori yang ingin Anda edit.
6. Di salah satu wadah kategoris tab yang dipilih, pilih tombol Edit dari atribut yang ingin Anda edit, buat perubahan, lalu pilih Simpan perubahan.

   a. Jika Anda ingin mengedit atribut di bawah tab Autentikasi dan identitas domain Anda di-host di Amazon Route 53, dan Anda belum menerbitkan catatan DNS-nya, akan ada tombol Publikasikan catatan DNS ke Route53 (di sebelah tombol Edit) di salah satu atau kedua wadah domain DomainKeys Identified Mail (DKIM) atau Custom MAIL FROM.

   > **Note**
   >
   > Tab Autentikasi hanya ada ketika akun Anda memiliki domain terverifikasi atau alamat email yang menggunakan domain terverifikasi di akun Anda.

   b. Anda dapat mempublikasikan catatan DNS langsung dari tombol Publikasikan catatan DNS ke Route53 - cukup klik, spanduk konfirmasi akan ditampilkan, dan tombol Publikasikan catatan DNS ke Route53 tidak akan lagi terlihat untuk wadah masing-masing.
7. Ulangi langkah 5 & 6 untuk setiap atribut identitas yang ingin Anda edit.

Mengedit identitas untuk menggunakan set konfigurasi default menggunakan SES API

Anda dapat menggunakan PutEmailIdentityConfigurationSetAttributesoperasi untuk menambah atau menghapus set konfigurasi default dari identitas email yang ada.

> **Note**
>
> Sebelum Anda menyelesaikan prosedur di bagian ini, Anda harus memasang dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Command Line Interface.

Untuk menambahkan set konfigurasi default menggunakan AWS CLI

- Pada baris perintah, masukkan perintah berikut untuk menggunakan PutEmailIdentityConfigurationSetAttributesoperasi.

```
aws sesv2 put-email-identity-configuration-set-attributes --email-identity ADDRESS-OR-DOMAIN --configuration-set-name CONFIG-SET
```

Di perintah sebelumnya, ganti *ADDRESS-OR-DOMAIN* dengan identitas email yang ingin Anda verifikasi. Ganti *CONFIG-SET* dengan nama set konfigurasi yang ingin Anda tetapkan sebagai set konfigurasi default identitas.

Jika perintah berhasil dijalankan, perintah keluar tanpa memberikan output apa pun.

Untuk menghapus set konfigurasi default menggunakan AWS CLI

- Pada baris perintah, masukkan perintah berikut untuk menggunakan PutEmailIdentityConfigurationSetAttributesoperasi.

```
aws sesv2 put-email-identity-configuration-set-attributes --email-identity ADDRESS-OR-DOMAIN
```

Di perintah sebelumnya, ganti *ADDRESS-OR-DOMAIN* dengan identitas email yang ingin Anda verifikasi.

Jika perintah berhasil dijalankan, perintah keluar tanpa memberikan output apa pun.

Ambil set konfigurasi default yang digunakan oleh identitas menggunakan SES API

Anda dapat menggunakan GetEmailIdentityoperasi untuk mengembalikan konfigurasi default yang ditetapkan untuk identitas email, jika berlaku.

> **Note**
>
> Sebelum Anda menyelesaikan prosedur di bagian ini, Anda harus memasang dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Command Line Interface.

Untuk mengembalikan set konfigurasi default menggunakan AWS CLI

- Pada baris perintah, masukkan perintah berikut untuk menggunakan GetEmailIdentityoperasi.

```
aws sesv2 get-email-identity --email-identity ADDRESS-OR-DOMAIN
```

Di perintah sebelumnya, ganti *ADDRESS-OR-DOMAIN* dengan identitas email yang Anda ingin ketahui set konfigurasi defaultnya, jika ada.

Jika perintah berhasil dijalankan, perintah menyediakan objek JSON dengan detail identitas email.

Ganti set konfigurasi default saat ini yang digunakan oleh identitas menggunakan SES API

Anda dapat menggunakan SendEmailoperasi untuk mengirim email dengan set konfigurasi yang berbeda. Jika Anda melakukannya, set konfigurasi yang Anda tentukan menimpa set konfigurasi default untuk identitas.

> **Note**
>
> Sebelum Anda menyelesaikan prosedur di bagian ini, Anda harus memasang dan mengonfigurasi AWS CLI. Untuk informasi selengkapnya, lihat Panduan Pengguna AWS Command Line Interface.

Untuk mengganti set konfigurasi default menggunakan AWS CLI

- Pada baris perintah, masukkan perintah berikut untuk menggunakan SendEmailoperasi.

```
aws sesv2 send-email --destination file://DESTINATION-JSON --content file://CONTENT-
JSON --from-email-address ADDRESS-OR-DOMAIN --configuration-set-name CONFIG-SET
```

Di perintah sebelumnya, ganti *DESTINATION-JSON* dengan file JSON tujuan Anda, *CONTENT-JSON* dengan file JSON konten Anda, *ADDRESS-OR-DOMAIN* dengan alamat email FROM Anda, dan *CONFIG-SET* dengan nama set konfigurasi yang ingin Anda gunakan sebagai pengganti set konfigurasi default untuk identitas.

Jika perintah berhasil dijalankan, perintah membuat output `MessageId`.

Mengonfigurasi identitas di Amazon SES

Amazon Simple Email Service (Amazon SES) menggunakan Protokol Transfer Surat Sederhana (SMTP) untuk mengirim email. Karena SMTP tidak menyediakan autentikasi apa pun dengan sendirinya, pelaku spam dapat mengirim pesan email yang mengklaim berasal dari orang lain, sambil menyembunyikan asal yang sebenarnya. Dengan memalsukan header email dan spoofing alamat IP sumber, spammer dapat menyesatkan penerima agar percaya bahwa pesan email yang diterima asli.

Sebagian besar ISP yang meneruskan lalu lintas email mengambil langkah-langkah untuk mengevaluasi jika email sah. Salah satu langkah yang diambil ISP adalah untuk menentukan apakah email diautentikasi. Autentikasi mengharuskan pengirim untuk memverifikasi bahwa mereka adalah pemilik akun yang mengirim. Dalam beberapa kasus, ISP menolak untuk meneruskan email yang tidak diautentikasi. Untuk memastikan kemampuan pengiriman yang optimal, sebaiknya Anda mengautentikasi email Anda.

Bagian berikut menjelaskan dua mekanisme autentikasi penggunaan ISP—Kerangka Kebijakan Pengirim (SPF) dan DomainKeys Identified Mail (DKIM) —dan memberikan petunjuk cara menggunakan standar ini dengan Amazon SES.

- Untuk mempelajari tentang SPF, yang menyediakan cara untuk melacak pesan email kembali ke sistem yang mengirim, lihat Mengautentikasi Email dengan SPF di Amazon SES.
- Untuk mempelajari tentang DKIM, standar yang memungkinkan Anda menandatangani pesan email untuk menunjukkan ISP bahwa pesan Anda sah dan belum diubah saat transit, lihat Mengautentikasi Email dengan DKIM di Amazon SES.

- Untuk mempelajari cara mematuhi Autentikasi Pesan, Pelaporan, dan Kesesuaian berbasis Domain (DMARC), yang bergantung pada SPF dan DKIM, lihat Mematuhi protokol DMARC otentikasi di Amazon SES.

Metode autentikasi email

Amazon Simple Email Service (Amazon SES) menggunakan Protokol Transfer Surat Sederhana (SMTP) untuk mengirim email. Karena SMTP tidak menyediakan autentikasi dengan sendirinya, spammer dapat mengirim pesan email yang mengklaim berasal dari orang lain, sambil menyembunyikan asal yang sebenarnya. Dengan memalsukan header email dan spoofing alamat IP sumber, spammer dapat menyesatkan penerima agar percaya bahwa pesan email yang diterima asli.

Sebagian besar ISP yang meneruskan lalu lintas email mengambil langkah-langkah untuk mengevaluasi jika email sah. Salah satu langkah yang diambil ISP adalah untuk menentukan jika email diautentikasi. Autentikasi mengharuskan pengirim untuk memverifikasi bahwa mereka adalah pemilik akun yang mengirim. Dalam beberapa kasus, ISP menolak untuk meneruskan email yang tidak diautentikasi. Untuk memastikan kemampuan pengiriman yang optimal, sebaiknya Anda mengautentikasi email Anda.

Konten

- Mengautentikasi Email dengan DKIM di Amazon SES
- Mengautentikasi Email dengan SPF di Amazon SES
- Menggunakan domain MAIL FROM kustom
- Mematuhi protokol DMARC otentikasi di Amazon SES
- Mengkonfigurasi BIMI di Amazon SES

Mengautentikasi Email dengan DKIM di Amazon SES

DomainKeysIdentified Mail (DKIM) adalah standar keamanan email yang dirancang untuk memastikan bahwa email yang mengklaim berasal dari domain tertentu memang diotorisasi oleh pemilik domain tersebut. Ini menggunakan kriptografi kunci publik untuk menandatangani email dengan kunci pribadi. Server penerima kemudian dapat menggunakan kunci publik yang dipublikasikan ke DNS domain untuk memverifikasi bahwa bagian email belum diubah selama transit.

Tanda tangan DKIM adalah opsional. Anda dapat memutuskan untuk menandatangani email menggunakan tanda tangan DKIM untuk meningkatkan kemampuan pengiriman dengan penyedia

email yang sesuai dengan DKIM. Amazon SES menyediakan tiga pilihan untuk menandatangani pesan Anda menggunakan tanda tangan DKIM:

- Easy DKIM: SES menghasilkan pasangan kunci public-private dan secara otomatis menambahkan tanda tangan DKIM ke setiap pesan yang Anda kirim dari identitas itu, lihat. Easy DKIM di Amazon SES
- BYODKIM (Bawa DKIM Anda Sendiri): Anda memberikan pasangan kunci publik-privat Anda sendiri dan SES menambahkan tanda tangan DKIM ke setiap pesan yang Anda kirim dari identitas itu, lihat. Berikan token autentikasi DKIM Anda (BYODKIM) di Amazon SES
- Tambahkan tanda tangan DKIM secara manual: Anda menambahkan tanda tangan DKIM Anda sendiri ke email yang Anda kirim menggunakan `SendRawEmail` API, lihat. Penandatanganan Manual DKIM di Amazon SES

Panjang kunci penandatanganan DKIM

Karena banyak penyedia DNS sekarang sepenuhnya mendukung enkripsi RSA bit DKIM 2048, Amazon SES juga mendukung DKIM 2048 untuk memungkinkan otentikasi email yang lebih aman dan karenanya menggunakannya sebagai panjang kunci default saat Anda mengonfigurasi Easy DKIM baik dari API atau konsol. Tombol 2048 bit dapat diatur dan digunakan dalam Bring Your Own DKIM (BYODKIM) juga, di mana panjang kunci penandatanganan Anda harus setidaknya 1024 bit dan tidak lebih dari 2048 bit.

Demi keamanan serta pengiriman email Anda, ketika dikonfigurasi dengan Easy DKIM, Anda memiliki pilihan untuk menggunakan panjang kunci 1024 dan 2048 bit bersama dengan fleksibilitas membalik kembali ke 1024 jika ada masalah yang disebabkan oleh penyedia DNS yang masih tidak mendukung 2048. Ketika Anda membuat identitas baru, itu akan dibuat dengan DKIM 2048 secara default kecuali Anda menentukan 1024.

Untuk mempertahankan pengiriman email transit, ada batasan pada frekuensi di mana Anda dapat mengubah panjang kunci DKIM Anda. Pembatasan meliputi:

- Tidak dapat beralih ke panjang kunci yang sama seperti yang sudah dikonfigurasi.
- Tidak dapat beralih ke panjang kunci yang berbeda lebih dari sekali dalam periode 24 jam (kecuali itu downgrade pertama ke 1024 dalam periode itu).

Ketika email Anda sedang transit, DNS menggunakan kunci publik Anda untuk mengautentikasi email Anda; oleh karena itu, jika Anda mengubah kunci terlalu cepat atau sering, DNS mungkin tidak

dapat DKIM mengotentikasi email Anda karena kunci sebelumnya mungkin sudah tidak valid, dengan demikian, pembatasan ini melindungi terhadap hal itu.

Pertimbangan DKIM

Saat Anda menggunakan DKIM untuk mengautentikasi email Anda, aturan berikut berlaku:

- Anda hanya perlu menyiapkan DKIM untuk domain yang Anda gunakan di alamat “Dari”. Anda tidak perlu menyiapkan DKIM untuk domain yang Anda gunakan di alamat “Return-Path” atau “Reply-to”.
- Amazon SES tersedia di beberapa Wilayah AWS. Jika Anda menggunakan lebih dari satu AWS Wilayah untuk mengirim email, Anda harus menyelesaikan proses penyiapan DKIM di masing-masing Wilayah tersebut untuk memastikan bahwa semua email Anda ditandatangani DKIM.
- Karena properti DKIM diwariskan dari domain induk, saat Anda memverifikasi domain dengan autentikasi DKIM:
  - Otentikasi DKIM juga akan berlaku untuk semua subdomain domain tersebut.
    - Pengaturan DKIM untuk subdomain dapat menimpa pengaturan untuk domain induk dengan menonaktifkan warisan jika Anda tidak ingin subdomain menggunakan otentikasi DKIM, serta kemampuan untuk mengaktifkan kembali nanti.
  - Otentikasi DKIM juga akan berlaku untuk semua email yang dikirim dari identitas email yang mereferensikan domain terverifikasi DKIM di alamatnya.
    - Pengaturan DKIM untuk alamat email dapat menimpa pengaturan untuk subdomain (jika ada) dan domain induk dengan menonaktifkan warisan jika Anda ingin mengirim email tanpa otentikasi DKIM, serta kemampuan untuk mengaktifkan kembali nanti.

Memahami properti penandatanganan DKIM yang diwariskan

Penting untuk dipahami terlebih dahulu bahwa identitas alamat email mewarisi properti penandatanganan DKIM dari domain induknya jika domain tersebut dikonfigurasi dengan DKIM, terlepas dari apakah Easy DKIM atau BYODKIM digunakan. Oleh karena itu, menonaktifkan atau mengaktifkan penandatanganan DKIM pada identitas alamat email, berlaku, mengesampingkan properti penandatanganan DKIM domain berdasarkan fakta-fakta utama berikut:

- Jika Anda sudah menyiapkan DKIM untuk domain tempat alamat email milik, Anda tidak perlu mengaktifkan penandatanganan DKIM untuk identitas alamat email juga.

  - Saat Anda menyiapkan DKIM untuk domain, Amazon SES secara otomatis mengautentikasi setiap email dari setiap alamat pada domain tersebut melalui properti DKIM yang diwariskan dari domain induk.
- Pengaturan DKIM untuk identitas alamat email tertentu secara otomatis menimpa pengaturan domain induk atau subdomain (jika ada) alamat milik.

Karena properti penandatanganan DKIM identitas alamat email diwarisi dari domain induk, jika Anda berencana mengganti properti ini, Anda harus mengingat aturan hierarkis menimpa seperti yang dijelaskan dalam tabel di bawah ini.

Domain induk tidak mengaktifkan penandatanganan DKIM	Domain induk telah mengaktifkan penandatanganan DKIM
Anda tidak dapat mengaktifkan penandatanganan DKIM pada identitas alamat email.	Anda dapat menonaktifkan penandatanganan DKIM pada identitas alamat email.
	Anda dapat mengaktifkan kembali penandatanganan DKIM pada identitas alamat email.

Umumnya tidak pernah disarankan untuk menonaktifkan penandatanganan DKIM Anda karena berisiko menodai reputasi pengirim Anda, dan meningkatkan risiko agar surat terkirim Anda masuk ke folder sampah atau spam atau memalsukan domain Anda.

Namun, ada kemampuan untuk mengganti properti penandatanganan DKIM yang diwarisi domain pada identitas alamat email untuk kasus penggunaan tertentu atau keputusan bisnis terpencil yang mungkin Anda miliki untuk menonaktifkan penandatanganan DKIM secara permanen atau sementara, atau untuk mengaktifkannya kembali di lain waktu. Lihat the section called “Mengesampingkan DKIM penandatanganan pada alamat email”.

Easy DKIM di Amazon SES

Ketika Anda mengatur Easy DKIM untuk identitas domain, Amazon SES secara otomatis menambahkan kunci DKIM 2048-bit ke setiap email yang Anda kirim dari identitas tersebut. Anda dapat mengonfigurasi Easy DKIM dengan menggunakan konsol Amazon SES, atau menggunakan API.

> **Note**
>
> Untuk menyiapkan Easy DKIM, Anda harus mengubah pengaturan DNS untuk domain Anda. Jika Anda menggunakan Route 53 sebagai penyedia DNS, Amazon SES dapat secara otomatis membuat catatan yang sesuai untuk Anda. Jika Anda menggunakan penyedia DNS lain, lihat dokumentasi penyedia Anda untuk mempelajari selengkapnya tentang mengubah pengaturan DNS untuk domain Anda.

> **Warning**
>
> Jika saat ini Anda mengaktifkan BYODKIM dan beralih ke Easy DKIM, ketahuilah bahwa Amazon SES tidak akan menggunakan BYODKIM untuk menandatangani email Anda saat Easy DKIM sedang diatur dan status DKIM Anda dalam keadaan tertunda. Antara saat Anda membuat panggilan untuk mengaktifkan Easy DKIM (baik melalui API atau konsol) dan saat SES dapat mengkonfirmasi konfigurasi DNS Anda, email Anda dapat dikirim oleh SES tanpa tanda tangan DKIM. Oleh karena itu, disarankan untuk menggunakan langkah perantara untuk bermigrasi dari satu metode penandatanganan DKIM ke yang lain (misalnya, menggunakan subdomain domain Anda dengan BYODKIM diaktifkan dan kemudian menghapusnya setelah verifikasi Easy DKIM telah berlalu), atau melakukan aktivitas ini selama downtime aplikasi Anda, jika ada.

Menyiapkan Easy DKIM untuk identitas domain terverifikasi

Prosedur di bagian ini disederhanakan untuk hanya menunjukkan langkah-langkah yang diperlukan untuk mengkonfigurasi Easy DKIM pada identitas domain yang telah Anda buat. Jika Anda belum membuat identitas domain atau ingin melihat semua opsi yang tersedia untuk menyesuaikan identitas domain, seperti menggunakan set konfigurasi default, domain MAIL FROM kustom, dan tag, lihatthe section called “Membuat identitas domain”.

Bagian dari pembuatan identitas domain Easy DKIM adalah mengonfigurasi verifikasi berbasis DKIM di mana Anda akan memiliki pilihan untuk menerima default Amazon SES sebesar 2048 bit, atau mengganti default dengan memilih 1024 bit. Lihatthe section called “Panjang kunci penandatanganan DKIM”untuk mempelajari lebih lanjut tentang panjang kunci penandatanganan DKIM dan cara mengubahnya.

Untuk menyiapkan Easy DKIM untuk domain

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Di daftar identitas, pilih identitas di mana identitasTipe IdentitasadalahDomain.

   > **Note**
   >
   > Jika Anda perlu membuat atau memverifikasi domain, lihatMembuat identitas domain.

4. Di bawahAutentikasitab, diDomainKeys Identified Mail (DKIM)wadah, pilihedit.
5. DiPengaturan lanjutan DKIMwadah, pilihEasy DKIMtombol diTipe IdentitasBidang.
6. DiPanjang kuenal DKIMbidang, pilih salah satuRSA_2048_BITatauRSA_1024_BIT.
7. DiTanda tangan DKIMlapangan, periksaDiaktifkankotak.
8. Pilih Save changes (Simpan perubahan).
9. Sekarang setelah Anda mengonfigurasi identitas domain dengan Easy DKIM, Anda harus menyelesaikan proses verifikasi dengan penyedia DNS Anda - lanjutkan kethe section called “Memverifikasi identitas domain”dan ikuti prosedur otentikasi DNS untuk Easy DKIM.

Ubah panjang kunci penandatanganan Easy DKIM untuk identitas

Prosedur di bagian ini menunjukkan bagaimana Anda dapat dengan mudah mengubah bit Easy DKIM yang diperlukan untuk algoritma penandatanganan. Sementara panjang penandatanganan 2048 bit selalu disukai untuk keamanan yang disempurnakan yang disediakannya, mungkin ada situasi yang mengharuskan Anda untuk menggunakan panjang 1024 bit, seperti harus menggunakan penyedia DNS yang hanya mendukung DKIM 1024.

Untuk menjaga pengiriman dalam email transit, ada pembatasan pada frekuensi di mana Anda dapat mengubah atau membalik panjang tombol DKIM Anda.

Ketika email Anda sedang transit, DNS menggunakan kunci publik Anda untuk mengautentikasi email Anda; oleh karena itu, jika Anda mengubah kunci terlalu cepat atau sering, DNS mungkin tidak dapat DKIM mengotentikasi email Anda karena kunci sebelumnya mungkin sudah dibatalkan, dengan demikian, pembatasan berikut melindungi terhadap hal tersebut:

- Anda tidak dapat beralih ke panjang tombol yang sama seperti yang sudah dikonfigurasi.

- Anda tidak dapat beralih ke panjang tombol yang berbeda lebih dari sekali dalam periode 24 jam (kecuali jika downgrade pertama menjadi 1024 pada periode tersebut).

Dalam menggunakan prosedur berikut untuk mengubah panjang kunci Anda, jika Anda melanggar salah satu pembatasan ini, konsol akan mengembalikan spanduk kesalahan yang menyatakan bahwamasukan yang Anda berikan tidak validbersama dengan alasan mengapa itu tidak valid.

Untuk mengubah DKIM penandatanganan kunci panjang bit

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Di daftar identitas, pilih identitas yang ingin Anda ubah untuk penandatanganan DKIM.
4. Di bawahAutentikasitab, diDomainKeys Identified Mail (DKIM)wadah, pilihedit.
5. DiPengaturan lanjutan DKIMwadah, pilih salah satuRSA_2048_BITatauRSA_1024_BITdi dalamPanjang kuenal DKIMBidang.
6. Pilih Simpan perubahan.

Berikan token autentikasi DKIM Anda (BYODKIM) di Amazon SES

Sebagai alternatif untuk menggunakan Easy DKIM, Anda dapat mengonfigurasi autentikasi DKIM dengan menggunakan pasangan kunci publik-privat Anda sendiri. Proses ini dikenal sebagai Bring Your Own DKIM (BYODKIM).

Dengan BYODKIM, Anda dapat menggunakan satu catatan DNS untuk mengonfigurasi autentikasi DKIM untuk domain Anda, dibandingkan dengan Easy DKIM yang mengharuskan Anda memublikasikan tiga catatan DNS terpisah. Selain itu, BYODKIM memungkinkan Anda memutar kunci DKIM untuk domain sesering yang Anda inginkan.

Topik di bagian ini:

- Langkah 1: Buat pasangan kunci
- Langkah 2: Tambahkan kunci publik dan kunci publik ke konfigurasi domain penyedia DNS Anda
- Langkah 3: Mengonfigurasi dan memverifikasi domain untuk menggunakan BYODKIM

> **Warning**
>
> Jika saat ini Anda mengaktifkan Easy DKIM dan sedang beralih ke BYODKIM, ketahuilah bahwa Amazon SES tidak akan menggunakan Easy DKIM untuk menandatangani email Anda saat BYODKIM sedang diatur dan status DKIM Anda dalam keadaan tertunda. Antara saat Anda melakukan panggilan untuk mengaktifkan BYODKIM (baik melalui API atau konsol) dan saat SES dapat mengonfirmasi konfigurasi DNS Anda, email Anda dapat dikirim oleh SES tanpa tanda tangan DKIM. Oleh karena itu, disarankan untuk menggunakan langkah perantara untuk bermigrasi dari satu metode penandatanganan DKIM ke yang lain (misalnya, menggunakan subdomain domain Anda dengan Easy DKIM diaktifkan dan kemudian menghapusnya setelah verifikasi BYODKIM berlalu), atau melakukan aktivitas ini selama waktu henti aplikasi Anda, jika ada.

Langkah 1: Buat pasangan kunci

Untuk menggunakan fitur Bring Your Own DKIM, Anda harus membuat key pair RSA terlebih dahulu.

Kunci pribadi yang Anda hasilkan harus dalam format PKCS #1 atau PKCS #8, harus menggunakan setidaknya enkripsi RSA 1024-bit dan hingga 2048-bit, dan dikodekan menggunakan pengkodean base64 (PEM). Lihat the section called “Panjang kunci penandatanganan DKIM” untuk mempelajari lebih lanjut tentang panjang kunci penandatanganan DKIM dan cara mengubahnya.

> **Note**
>
> Anda dapat menggunakan aplikasi dan alat pihak ketiga untuk menghasilkan pasangan kunci RSA selama kunci pribadi dibuat dengan setidaknya enkripsi RSA 1024-bit dan hingga 2048-bit, dan dikodekan menggunakan pengkodean base64 (PEM).

Di prosedur berikut, kode contoh yang menggunakan `openssl genrsa` perintah yang merupakan bawaan untuk sebagian besar sistem operasi Linux, macOS, atau Unix untuk membuat key pair akan secara otomatis menggunakan pengkodean base64 (PEM) untuk membuat pasangan kunci secara otomatis akan menggunakan pengkodean base64 (PEM).

Untuk membuat pasangan kunci dari baris perintah Linux, macOS, atau Unix

1. Pada baris perintah, masukkan perintah berikut untuk menghasilkan kunci privat menggantikan *nnnnn* dengan panjang bit minimal 1024 dan hingga 2048:

```
openssl genrsa -f4 -out private.key nnnn
```

2. Pada baris perintah, masukkan perintah berikut untuk menghasilkan kunci publik:

```
openssl rsa -in private.key -outform PEM -pubout -out public.key
```

Langkah 2: Tambahkan kunci publik dan kunci publik ke konfigurasi domain penyedia DNS Anda

Setelah membuat pasangan kunci, Anda harus menambahkan kunci publik sebagai catatan TXT ke konfigurasi DNS untuk domain Anda.

Untuk menambahkan kunci publik ke konfigurasi DNS untuk domain Anda

1. Masuk ke konsol manajemen untuk penyedia DNS atau hosting Anda.
2. Tambahkan catatan teks ke konfigurasi DNS untuk domain Anda. Catatan harus menggunakan format berikut:

Nama	Tipe	Nilai
selector._domainkey.*example.com*	TXT	p= *yourPublicKey*

Pada contoh sebelumnya, lakukan perubahan berikut:

- Ganti *selector* dengan nama unik yang mengidentifikasi kunci.

> **Note**
>
> Sejumlah kecil penyedia DNS tidak mengizinkan Anda untuk menyertakan garis bawah (_) di nama catatan. Namun, garis bawah di nama catatan DKIM diperlukan. Jika penyedia DNS Anda tidak mengizinkan Anda untuk memasukkan garis bawah di nama catatan, kontak tim dukungan pelanggan penyedia untuk mendapatkan bantuan.

- Ganti *example.com* dengan domain Anda.
- Ganti *yourPublicKey*dengan kunci publik yang Anda buat sebelumnya dan sertakan p= awalan seperti yang ditunjukkan pada kolom Nilai di atas.

> **Note**
>
> Ketika Anda mempublikasikan (menambahkan) kunci publik Anda ke penyedia DNS Anda, itu harus diformat sebagai berikut:
>
> - Anda harus menghapus baris pertama dan terakhir (`-----BEGIN PUBLIC KEY-----` dan `-----END PUBLIC KEY-----`, secara berturut-turut) dari kunci publik yang dihasilkan. Selain itu, Anda harus menghapus jeda baris di kunci publik yang dihasilkan. Nilai yang dihasilkan adalah string karakter tanpa spasi atau jeda baris.
> - Anda harus menyertakan p= awalan seperti yang ditunjukkan pada kolom Nilai pada tabel di atas.

Penyedia yang berbeda memiliki prosedur yang berbeda untuk memperbarui catatan DNS. Tabel berikut menyertakan tautan ke dokumentasi untuk beberapa penyedia DNS yang banyak digunakan. Daftar ini tidak lengkap dan tidak menandakan dukungan; demikian juga, jika penyedia DNS Anda tidak terdaftar, itu tidak berarti Anda tidak dapat menggunakan domain dengan Amazon SES.

Penyedia DNS/Hosting	Tautan dokumentasi
Amazon Route 53	Mengedit Catatan di Panduan Developer Amazon Route 53
GoDaddy	Tambahkan catatan TXT (tautan eksternal)
DreamHost	Bagaimana cara menambahkan catatan DNS kustom? (tautan eksternal)
Cloudflare	Mengelola catatan DNS di Cloudflare (tautan eksternal)
HostGator	Kelola Catatan DNS HostGator dengan/eNom (tautan eksternal)

Penyedia DNS/Hosting	Tautan dokumentasi
Namecheap	Bagaimana cara menambahkan catatan TXT/SPF/DKIM/DMARC untuk domain saya? (tautan eksternal)
Names.co.uk	Mengubah Pengaturan DNS domain (tautan eksternal)
Wix	Menambahkan atau Memperbarui Catatan TXT di Akun Wix (tautan eksternal)

Langkah 3: Mengonfigurasi dan memverifikasi domain untuk menggunakan BYODKIM

Anda dapat menyiapkan BYODKIM untuk kedua domain baru (yaitu, domain yang saat ini tidak Anda gunakan untuk mengirim email melalui Amazon SES) dan domain yang ada (yaitu, domain yang telah Anda siapkan untuk digunakan dengan Amazon SES) dengan menggunakan konsol atau. AWS CLI Sebelum Anda menggunakan AWS CLI prosedur di bagian ini, Anda harus menginstal dan mengonfigurasiAWS CLI. Untuk informasi selengkapnya, lihat Panduan AWS Command Line Interface Pengguna..

Opsi 1: Membuat identitas domain baru yang menggunakan BYODKIM

Bagian ini berisi prosedur untuk membuat identitas domain baru yang menggunakan BYODKIM. Identitas domain baru adalah domain yang belum Anda siapkan untuk mengirim email menggunakan Amazon SES sebelumnya.

Jika Anda ingin mengonfigurasi domain yang ada untuk menggunakan BYODKIM, selesaikan prosedur di Opsi 2: Mengonfigurasi identitas domain yang ada sebagai gantinya.

Untuk membuat identitas menggunakan BYODKIM dari konsol

- Ikuti prosedur diMembuat identitas domain, dan ketika Anda sampai ke Langkah 8, ikuti petunjuk khusus BYODKIM.

Untuk membuat identitas menggunakan BYODKIM dari AWS CLI

Untuk mengonfigurasi domain baru, gunakan `CreateEmailIdentity` operasi di API Amazon SES.

1. Di editor teks, tempel kode berikut:

```
{
    "EmailIdentity":"example.com",
    "DkimSigningAttributes":{
        "DomainSigningPrivateKey":"privateKey",
        "DomainSigningSelector":"selector"
    }
}
```

Di contoh sebelumnya, lakukan perubahan berikut:

- Ganti *example.com* dengan domain yang ingin Anda buat.
- Ganti *privateKey* dengan kunci privat Anda.

> **Note**
>
> Anda harus menghapus baris pertama dan terakhir (`-----BEGIN PRIVATE KEY-----`dan`-----END PRIVATE KEY-----`, secara berturut-turut) dari kunci privat yang dihasilkan. Selain itu, Anda harus menghapus jeda baris di kunci privat yang dihasilkan. Nilai yang dihasilkan adalah string karakter tanpa spasi atau jeda baris.

- Ganti *selector* dengan selector unik yang Anda tentukan saat membuat catatan TXT di konfigurasi DNS untuk domain Anda.

Setelah selesai, simpan file sebagai `create-identity.json`.

2. Di baris perintah, masukkan perintah berikut:

```
aws sesv2 create-email-identity --cli-input-json file://path/to/create-identity.json
```

Di perintah sebelumnya, ganti *path/to/create-identity.json* dengan jalur lengkap ke file yang Anda buat di langkah sebelumnya.

Opsi 2: Mengonfigurasi identitas domain yang ada

Bagian ini berisi prosedur untuk memperbarui identitas domain yang ada untuk menggunakan BYODKIM. Identitas domain yang ada adalah domain yang sebelumnya sudah Anda siapkan untuk mengirim email menggunakan Amazon SES.

Untuk memperbarui identitas domain menggunakan BYODKIM dari konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Dalam daftar identitas, pilih identitas di mana jenis Identitas adalah Domain.

> **Note**
>
> Jika Anda perlu membuat atau memverifikasi domain, lihatMembuat identitas domain.

4. Di bawah tab Otentikasi, di panel DomainKeysIdentified Mail (DKIM), pilih Edit.
5. Di panel Pengaturan DKIM Lanjutan, pilih tombol Berikan token otentikasi DKIM (BYODKIM) di bidang Jenis identitas.
6. Untuk Kunci privat, tempelkan kunci privat yang Anda buat sebelumnya.

> **Note**
>
> Anda harus menghapus baris pertama dan terakhir (`-----BEGIN PRIVATE KEY-----`dan`-----END PRIVATE KEY-----`, secara berturut-turut) dari kunci privat yang dihasilkan. Selain itu, Anda harus menghapus jeda baris di kunci privat yang dihasilkan. Nilai yang dihasilkan adalah string karakter tanpa spasi atau jeda baris.

7. Untuk Nama pemilih, masukkan nama pemilih yang Anda tentukan di pengaturan DNS domain Anda.
8. Di bidang tanda tangan DKIM, centang kotak Diaktifkan.
9. Pilih Save changes (Simpan perubahan).

Untuk memperbarui identitas domain menggunakan BYODKIM dari AWS CLI

Untuk mengonfigurasi domain yang ada, gunakan `PutEmailIdentityDkimSigningAttributes` operasi di API Amazon SES.

1. Di editor teks, tempel kode berikut:

```
{
    "SigningAttributes":{
        "DomainSigningPrivateKey":"privateKey",
        "DomainSigningSelector":"selector"
    },
    "SigningAttributesOrigin":"EXTERNAL"
}
```

   Di contoh sebelumnya, lakukan perubahan berikut:

   - Ganti *`privateKey`* dengan kunci privat Anda.

     > **Note**
     >
     > Anda harus menghapus baris pertama dan terakhir (`-----BEGIN PRIVATE KEY-----`dan`-----END PRIVATE KEY-----`, secara berturut-turut) dari kunci privat yang dihasilkan. Selain itu, Anda harus menghapus jeda baris di kunci privat yang dihasilkan. Nilai yang dihasilkan adalah string karakter tanpa spasi atau jeda baris.

   - Ganti *`selector`* dengan selector unik yang Anda tentukan saat membuat catatan TXT di konfigurasi DNS untuk domain Anda.

   Setelah selesai, simpan file sebagai `update-identity.json`.

2. Di baris perintah, masukkan perintah berikut:

```
aws sesv2 put-email-identity-dkim-signing-attributes --email-identity example.com
 --cli-input-json file://path/to/update-identity.json
```

   Di perintah sebelumnya, lakukan perubahan berikut:

- Ganti *path/to/update-identity.json* dengan jalur lengkap ke file yang Anda buat di langkah sebelumnya.
- Ganti *example.com* dengan domain yang ingin Anda perbarui.

Memverifikasi status DKIM untuk domain yang menggunakan BYODKIM

Untuk memverifikasi status DKIM domain dari konsol

Setelah Anda mengonfigurasi domain menggunakan BYODKIM, Anda dapat menggunakan konsol SES untuk memverifikasi bahwa DKIM dikonfigurasi dengan benar.

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Di daftar identitas, pilih identitas yang status DKIM yang ingin Anda verifikasi.
4. Hal ini dapat memakan waktu hingga 72 jam agar perubahan pada pengaturan DNS diterapkan. Segera setelah Amazon SES mendeteksi semua catatan DKIM yang diperlukan dalam pengaturan DNS domain Anda, proses verifikasi selesai. Jika semuanya telah dikonfigurasi dengan benar, bidang konfigurasi DKIM domain Anda akan menampilkan Sukses di panel Surat DomainKeys Teridentifikasi (DKIM), dan bidang status Identitas ditampilkan Terverifikasi di panel Ringkasan.

Untuk memverifikasi status DKIM domain menggunakan AWS CLI

Setelah Anda mengonfigurasi domain menggunakan BYODKIM, Anda dapat menggunakan GetEmailIdentity operasi untuk memverifikasi bahwa DKIM dikonfigurasi dengan benar.

- Di baris perintah, masukkan perintah berikut:

```
aws sesv2 get-email-identity --email-identity example.com
```

Di perintah sebelumnya, ganti *example.com* dengan domain Anda.

Perintah ini mengembalikan sebuah objek JSON yang berisi bagian yang menyerupai contoh berikut.

```
{
```

```
    ...
    "DkimAttributes": {
        "SigningAttributesOrigin": "EXTERNAL",
        "SigningEnabled": true,
        "Status": "SUCCESS",
        "Tokens": [ ]
    },
    ...
}
```

BYODKIM dikonfigurasi dengan benar untuk domain jika semua hal berikut betul:

- Nilai dari properti `SigningAttributesOrigin` adalah `EXTERNAL`.
- Nilai dari `SigningEnabled` adalah `true`.
- Nilai dari `Status` adalah `SUCCESS`.

Mengelola Mudah DKIM dan BYODKIM

Anda dapat mengelola DKIM pengaturan untuk identitas Anda yang diautentikasi dengan Easy DKIM atau BYODKIM dengan menggunakan konsol SES Amazon berbasis web, atau dengan menggunakan Amazon. SES API Anda dapat menggunakan salah satu dari metode ini untuk mendapatkan DKIM catatan identitas, atau untuk mengaktifkan atau menonaktifkan DKIM penandatanganan identitas.

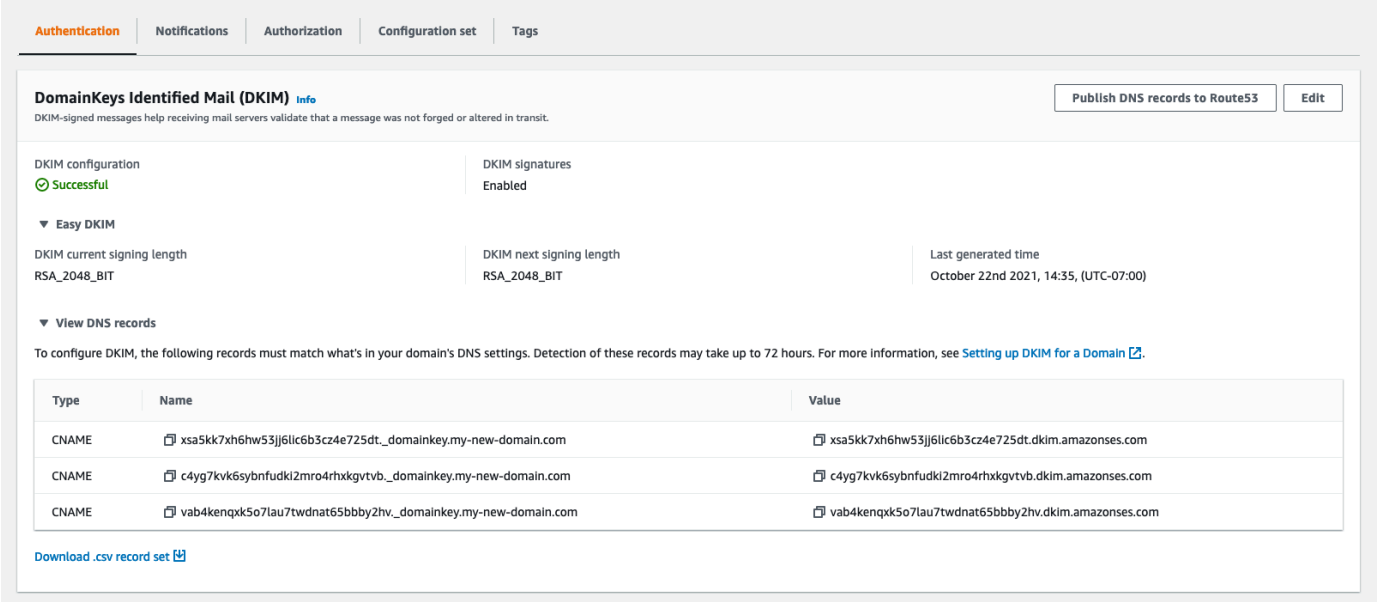
Memperoleh DKIM Catatan untuk identitas

Anda dapat memperoleh DKIM catatan untuk domain atau alamat email Anda kapan saja dengan menggunakan SES konsol Amazon.

Untuk mendapatkan DKIM catatan untuk identitas dengan menggunakan konsol

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Dalam daftar identitas, pilih identitas yang ingin Anda dapatkan DKIM catatannya.
4. Pada tab Otentikasi halaman detail identitas, perluas Tampilan DNS catatan.
5. Salin ketiga CNAME catatan jika Anda menggunakan EasyDKIM, atau TXT catatan jika Anda menggunakanBYODKIM, yang muncul di bagian ini. Sebagai alternatif, Anda dapat memilih Unduh set catatan .csv untuk menyimpan salinan catatan ke komputer Anda.

Gambar berikut menunjukkan contoh bagian DNSCatatan Tampilan yang diperluas yang mengungkapkan CNAME catatan yang terkait dengan MudahDKIM.



Anda juga dapat memperoleh DKIM catatan untuk identitas dengan menggunakan Amazon SESAPI. Metode umum untuk berinteraksi dengan API adalah dengan menggunakan. AWS CLI

Untuk mendapatkan DKIM catatan untuk identitas dengan menggunakan AWS CLI

1. Di baris perintah, ketik perintah berikut:

```
aws ses get-identity-dkim-attributes --identities "example.com"
```

   Pada contoh sebelumnya, ganti *example.com* dengan identitas yang ingin Anda dapatkan DKIM catatannya. Anda dapat menentukan alamat email atau domain.

2. Output dari perintah ini berisi bagian `DkimTokens`, seperti yang ditunjukkan pada contoh berikut:

```
{
    "DkimAttributes": {
        "example.com": {
            "DkimEnabled": true,
            "DkimVerificationStatus": "Success",
            "DkimTokens": [
                "hirjd4exampled5477y22yd23ettobi",
                "v3rnz522czcl46quexamplek3efo5o6x",
```

```
                "y4examplexbhyhnsjcmtvzotfvqjmdqoj"
            ]
        }
    }
}
```

Anda dapat menggunakan token untuk membuat CNAME catatan yang Anda tambahkan ke DNS pengaturan untuk domain Anda. Untuk membuat CNAME catatan, gunakan template berikut:

```
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
```

Ganti setiap instance dari *token1* dengan token pertama dalam daftar yang Anda terima saat menjalankan `get-identity-dkim-attributes` perintah, ganti semua instance *token2* dengan token kedua dalam daftar, dan ganti semua contoh *token3* dengan token ketiga dalam daftar.

Misalnya, menerapkan templat ini ke token yang ditampilkan di contoh sebelumnya menghasilkan catatan berikut:

```
hirjd4exampled5477y22yd23ettobi._domainkey.example.com CNAME
 hirjd4exampled5477y22yd23ettobi.dkim.amazonses.com
v3rnz522czcl46quexamplek3efo5o6x._domainkey.example.com CNAME
 v3rnz522czcl46quexamplek3efo5o6x.dkim.amazonses.com
y4examplexbhyhnsjcmtvzotfvqjmdqoj._domainkey.example.com CNAME
 y4examplexbhyhnsjcmtvzotfvqjmdqoj.dkim.amazonses.com
```

> **Note**
>
> Tidak semua Wilayah AWS menggunakan SES DKIM domain default, `dkim.amazonses.com` —untuk melihat apakah wilayah Anda menggunakan DKIM domain spesifik wilayah, periksa tabel DKIM domain di. Referensi Umum AWS

Menonaktifkan Mudah DKIM untuk identitas

Anda dapat dengan cepat menonaktifkan DKIM otentikasi untuk identitas dengan menggunakan SES konsol Amazon.

DKIMUntuk menonaktifkan identitas

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Dalam daftar identitas, pilih identitas yang ingin Anda nonaktifkanDKIM.
4. Di bawah tab Autentikasi, dalam wadah DomainKeysIdentified Mail (DKIM), pilih Edit.
5. Di DKIMPengaturan lanjutan, kosongkan kotak Diaktifkan di bidang DKIMtanda tangan.

Anda juga dapat DKIM menonaktifkan identitas dengan menggunakan Amazon SESAPI. Metode umum untuk berinteraksi dengan API adalah dengan menggunakan. AWS CLI

DKIMUntuk menonaktifkan identitas dengan menggunakan AWS CLI

- Di baris perintah, ketik perintah berikut:

  ```
  aws ses set-identity-dkim-enabled --identity example.com --no-dkim-enabled
  ```

  Pada contoh sebelumnya, ganti *example.com* dengan identitas yang ingin Anda nonaktifkanDKIM. Anda dapat menentukan alamat email atau domain.

Mengaktifkan Mudah DKIM untuk sebuah identitas

Jika sebelumnya Anda DKIM menonaktifkan identitas, Anda dapat mengaktifkannya lagi dengan menggunakan SES konsol Amazon.

DKIMUntuk mengaktifkan identitas

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Dalam daftar identitas, pilih identitas yang ingin Anda aktifkanDKIM.

4. Di bawah tab Autentikasi, dalam wadah DomainKeysIdentified Mail (DKIM), pilih Edit.
5. Di DKIMPengaturan lanjutan, centang kotak Diaktifkan di bidang DKIMtanda tangan.

Anda juga dapat mengaktifkan DKIM identitas dengan menggunakan Amazon SESAPI. Metode umum untuk berinteraksi dengan API adalah dengan menggunakan. AWS CLI

DKIMUntuk mengaktifkan identitas dengan menggunakan AWS CLI

- Di baris perintah, ketik perintah berikut:

  ```
  aws ses set-identity-dkim-enabled --identity example.com --dkim-enabled
  ```

  Pada contoh sebelumnya, ganti *example.com* dengan identitas yang ingin Anda aktifkanDKIM. Anda dapat menentukan alamat email atau domain.

Mengganti DKIM penandatanganan warisan pada identitas alamat email

Di bagian ini, Anda akan mempelajari cara mengganti (menonaktifkan atau mengaktifkan) properti DKIM penandatanganan yang diwarisi dari domain induk pada identitas alamat email tertentu yang telah Anda verifikasi dengan Amazon. SES Anda hanya dapat melakukan ini untuk identitas alamat email milik domain yang sudah Anda miliki karena DNS pengaturan dikonfigurasi pada tingkat domain.

> **Important**
>
> Anda tidak dapat menonaktifkan/mengaktifkan DKIM penandatanganan identitas alamat email...
>
> - pada domain yang tidak Anda miliki. Misalnya, Anda tidak dapat beralih DKIM penandatanganan untuk alamat gmail.com atau hotmail.com,
> - pada domain yang Anda miliki, tetapi belum diverifikasi di AmazonSES,
> - pada domain yang Anda miliki, tetapi belum mengaktifkan DKIM penandatanganan pada domain.

Bagian ini berisi topik berikut:

- Memahami properti DKIM penandatanganan yang diwariskan

- Mengesampingkan DKIM penandatanganan identitas alamat email (konsol)
- Mengesampingkan DKIM penandatanganan identitas alamat email ()AWS CLI

Memahami properti DKIM penandatanganan yang diwariskan

Penting untuk dipahami terlebih dahulu bahwa identitas alamat email mewarisi properti DKIM penandatanganannya dari domain induknya jika domain tersebut dikonfigurasiDKIM, terlepas dari apakah Easy DKIM atau BYODKIM digunakan. Oleh karena itu, menonaktifkan atau mengaktifkan DKIM penandatanganan pada identitas alamat email, berlaku, mengesampingkan properti DKIM penandatanganan domain berdasarkan fakta-fakta kunci ini:

- Jika Anda sudah menyiapkan DKIM domain yang menjadi milik alamat email, Anda tidak perlu mengaktifkan DKIM penandatanganan identitas alamat email juga.
  - Saat Anda menyiapkan DKIM domain, Amazon SES secara otomatis mengautentikasi setiap email dari setiap alamat pada domain tersebut melalui DKIM properti yang diwariskan dari domain induk.
- DKIMpengaturan untuk identitas alamat email tertentu secara otomatis mengganti pengaturan domain induk atau subdomain (jika ada) yang menjadi milik alamat tersebut.

Karena properti DKIM penandatanganan identitas alamat email diwarisi dari domain induk, jika Anda berencana mengganti properti ini, Anda harus mengingat aturan hierarkis penggantian seperti yang dijelaskan dalam tabel di bawah ini.

Domain induk tidak mengaktifkan DKIM penandatanganan	Domain induk telah mengaktifkan DKIM penandatanganan
Anda tidak dapat mengaktifkan DKIM penandatanganan pada identitas alamat email.	Anda dapat menonaktifkan DKIM penandata nganan pada identitas alamat email.
	Anda dapat mengaktifkan kembali DKIM penandatanganan pada identitas alamat email.

Umumnya tidak pernah disarankan untuk menonaktifkan DKIM penandatanganan Anda karena berisiko menodai reputasi pengirim Anda, dan itu meningkatkan risiko email terkirim Anda masuk ke folder sampah atau spam atau domain Anda dipalsukan.

Namun, ada kemampuan untuk mengganti properti DKIM penandatanganan yang diwarisi domain pada identitas alamat email untuk kasus penggunaan tertentu atau keputusan bisnis luar yang mungkin harus Anda nonaktifkan DKIM penandatanganan secara permanen atau sementara, atau untuk mengaktifkannya kembali di lain waktu.

Mengesampingkan DKIM penandatanganan identitas alamat email (konsol)

Prosedur SES konsol berikut menjelaskan cara mengganti (menonaktifkan atau mengaktifkan) properti DKIM penandatanganan yang diwarisi dari domain induk pada identitas alamat email tertentu yang telah Anda verifikasi dengan Amazon. SES

Untuk menonaktifkan/mengaktifkan DKIM penandatanganan identitas alamat email menggunakan konsol

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Dalam daftar identitas, pilih identitas di mana tipe Identitas adalah Alamat email dan milik salah satu domain terverifikasi Anda.
4. Di bawah tab Autentikasi, dalam wadah DomainKeys Identified Mail (DKIM), pilih Edit.

> **Note**
>
> Tab Otentikasi hanya ada jika identitas alamat email yang dipilih milik domain yang telah diverifikasi olehSES. Jika Anda belum memverifikasi domain Anda, lihatMembuat identitas domain.

5. Di bawah DKIMPengaturan lanjutan, di bidang DKIMtanda tangan, kosongkan kotak centang Diaktifkan untuk menonaktifkan DKIM penandatanganan, atau pilih untuk mengaktifkan kembali DKIM penandatanganan (jika telah diganti sebelumnya).
6. Pilih Simpan perubahan.

Mengesampingkan DKIM penandatanganan identitas alamat email ()AWS CLI

Contoh berikut menggunakan SES API perintah dan parameter AWS CLI with a yang akan mengganti (menonaktifkan atau mengaktifkan) properti DKIM penandatanganan yang diwariskan dari domain induk pada identitas alamat email tertentu yang telah Anda verifikasi. SES

Untuk menonaktifkan/mengaktifkan DKIM penandatanganan identitas alamat email menggunakan AWS CLI

- Dengan asumsi Anda memiliki domain example.com, dan Anda ingin menonaktifkan DKIM penandatanganan untuk salah satu alamat email domain, pada baris perintah, ketikkan perintah berikut:

```
aws sesv2 put-email-identity-dkim-attributes --email-identity marketing@example.com
 --no-signing-enabled
```

  a. Ganti *marketing@example.com* dengan identitas alamat email yang ingin Anda nonaktifkan DKIM penandatanganan.

  b. `--no-signing-enabled`dakan menonaktifkan DKIM penandatanganan. Untuk mengaktifkan kembali DKIM penandatanganan, gunakan`--signing-enabled`.

Penandatanganan Manual DKIM di Amazon SES

Sebagai alternatif untuk menggunakan Easy DKIM, Anda dapat menambahkan tanda tangan DKIM secara manual ke pesan Anda, dan kemudian mengirim pesan tersebut menggunakan Amazon SES. Jika Anda memilih untuk menandatangani pesan secara manual, Anda harus membuat tanda tangan DKIM terlebih dahulu. Setelah Anda membuat pesan dan tanda tangan DKIM, Anda dapat menggunakan SendRawEmail API untuk mengirimkannya.

Jika Anda memutuskan untuk menandatangani email secara manual, pertimbangkan faktor-faktor berikut:

- Setiap pesan yang Anda kirim menggunakan Amazon SES berisi header DKIM yang merujuk domain penandatangananamazonses.com(yaitu, berisi string berikut: `d=amazonses.com`). Oleh karena itu, jika Anda menandatangani pesan secara manual, pesan Anda akan disertakanduaHeader DKIM: satu untuk domain Anda, dan satu yang dibuat otomatis oleh Amazon SESamazonses.com.
- Amazon SES tidak memvalidasi tanda tangan DKIM yang Anda tambahkan secara manual ke pesan Anda. Jika ada kesalahan dengan tanda tangan DKIM di pesan, pesan tersebut mungkin ditolak oleh penyedia email.
- Ketika Anda menandatangani pesan, Anda harus menggunakan panjang bit setidaknya 1024 bit.
- Jangan menandatangani bidang berikut: ID Pesan, Tanggal, Jalur Kembali, Memantul Ke.

> **Note**
>
> Jika Anda menggunakan klien email untuk mengirim email menggunakan antarmuka SMTP Amazon SES, klien Anda mungkin secara otomatis melakukan penandatanganan DKIM pesan Anda. Beberapa klien mungkin menandatangani beberapa bidang ini. Untuk informasi tentang bidang yang ditandatangani secara default, lihat dokumentasi untuk klien email Anda.

Mengautentikasi Email dengan SPF di Amazon SES

Kerangka Kerja Kebijakan Pengirim (SPF) adalah standar validasi email yang dirancang untuk mencegah pemalsuan email. Pemilik domain menggunakan SPF untuk memberi tahu penyedia email server mana yang diizinkan untuk mengirim email dari domain mereka. SPF ditentukan di RFC 7208.

Pesan yang Anda kirim melalui Amazon SES secara otomatis menggunakan subdomain `amazonses.com` sebagai domain MAIL FROM default. Otentikasi SPF berhasil memvalidasi pesan-pesan ini karena domain MAIL FROM default cocok dengan aplikasi yang mengirim email—dalam hal ini, SES. Oleh karena itu, di SES, SPF secara implisit disiapkan untuk Anda.

Namun, jika Anda tidak ingin menggunakan domain SES MAIL FROM default, dan lebih suka menggunakan subdomain dari domain yang Anda miliki, ini disebut dalam SES sebagai menggunakan domain MAIL FROM kustom. Untuk melakukan ini, Anda harus mempublikasikan catatan SPF Anda sendiri untuk domain MAIL FROM kustom Anda. Selain itu, SES juga mengharuskan Anda untuk menyiapkan data MX sehingga domain MAIL FROM kustom Anda dapat menerima pemberitahuan pentalan dan keluhan yang dikirimkan oleh penyedia email kepada Anda.

Pelajari cara mengatur otentikasi SPF

Instruksi diberikan untuk mengonfigurasi domain Anda dengan SPF dan cara mempublikasikan catatan MX dan SPF (tipe TXT) di. the section called “Menggunakan domain MAIL FROM kustom”

Menggunakan domain MAIL FROM kustom

Ketika email dikirim, ada dua alamat yang menunjukkan sumbernya: alamat Dari yang ditampilkan ke penerima pesan, dan alamat MAIL FROM yang menunjukkan asal pesan tersebut. Alamat MAIL FROM terkadang disebut pengirim envelope, envelope dari, alamat pentalan, atau alamat Jalur Kembali. Server mail menggunakan alamat MAIL FROM untuk mengembalikan pesan pentalan dan

notifikasi kesalahan lainnya. Alamat MAIL FROM biasanya hanya dapat dilihat oleh penerima jika mereka melihat kode sumber untuk pesan tersebut.

Amazon SES menetapkan domain MAIL FROM untuk pesan yang Anda kirim ke nilai default kecuali Anda menentukan domain (kustom) Anda sendiri. Bagian ini membahas manfaat menyiapkan domain MAIL FROM kustom, dan mencakup prosedur pengaturan.

Mengapa menggunakan domain MAIL FROM kustom?

Pesan yang Anda kirim melalui Amazon SES secara otomatis menggunakan subdomain `amazonses.com` sebagai domain MAIL FROM default. Otentikasi Kerangka Kebijakan Pengirim (SPF) berhasil memvalidasi pesan-pesan ini karena domain MAIL FROM default cocok dengan aplikasi yang mengirim email—dalam hal ini, SES.

Jika Anda tidak ingin menggunakan domain SES MAIL FROM default, dan lebih suka menggunakan subdomain dari domain yang Anda miliki, ini disebut dalam SES sebagai menggunakan domain MAIL FROM kustom. Untuk melakukan ini, Anda harus mempublikasikan catatan SPF Anda sendiri untuk domain MAIL FROM kustom Anda. Selain itu, SES juga mengharuskan Anda untuk menyiapkan catatan MX sehingga domain Anda dapat menerima pemberitahuan pentalan dan keluhan yang dikirimkan oleh penyedia email kepada Anda.

Dengan menggunakan domain MAIL FROM kustom, Anda memiliki fleksibilitas untuk menggunakan SPF, DKIM, atau keduanya untuk mencapai validasi Autentikasi Pesan, Pelaporan, dan Kesesuaian (DMARC) berbasis Domain. DMARC memungkinkan domain pengirim untuk menunjukkan bahwa email yang dikirim dari domain dilindungi oleh satu atau beberapa sistem autentikasi. Ada dua cara untuk mencapai validasi DMARC: dan. the section called “Mematuhi melalui DMARC SPF” the section called “Mematuhi melalui DMARC DKIM”

Memilih domain MAIL FROM kustom

Berikut ini, istilah MAIL FROM domain selalu mengacu pada subdomain dari domain yang Anda miliki - subdomain ini yang Anda gunakan untuk domain MAIL FROM kustom Anda tidak boleh digunakan untuk hal lain dan memenuhi persyaratan berikut:

- Domain MAIL FROM harus menjadi subdomain dari domain induk dari identitas terverifikasi (alamat email atau domain).
- Domain MAIL FROM seharusnya tidak menjadi subdomain yang juga Anda gunakan untuk mengirim email.
- Domain MAIL FROM tidak boleh menjadi subdomain yang Anda gunakan untuk menerima email.

Menggunakan SPF dengan domain MAIL FROM kustom Anda

Kerangka Kerja Kebijakan Pengirim (SPF) adalah standar validasi email yang dirancang untuk mencegah pemalsuan email. Anda dapat mengonfigurasi domain MAIL FROM kustom Anda dengan SPF untuk memberi tahu penyedia email server mana yang diizinkan mengirim email dari domain MAIL FROM kustom Anda. SPF ditentukan di RFC 7208.

Untuk menyiapkan SPF, Anda menerbitkan catatan TXT ke konfigurasi DNS untuk domain MAIL FROM kustom Anda. Catatan ini berisi daftar server yang Anda otorisasi untuk mengirim email menggunakan domain MAIL FROM kustom Anda. Ketika penyedia email menerima pesan dari domain MAIL FROM kustom Anda, ia memeriksa catatan DNS untuk domain tersebut untuk memastikan bahwa email tersebut dikirim dari server resmi.

Jika Anda ingin menggunakan data SPF ini sebagai cara untuk mematuhi DMARC, domain di alamat Dari harus cocok dengan domain MAIL FROM. Lihat the section called “Mematuhi melalui DMARC SPF”.

Bagian selanjutnyathe section called “Mengonfigurasi domain MAIL FROM kustom Anda”, menjelaskan cara mengatur SPF untuk domain MAIL FROM kustom Anda.

Mengonfigurasi domain MAIL FROM kustom Anda

Proses penyiapan domain MAIL FROM kustom mengharuskan Anda untuk menambahkan catatan ke konfigurasi DNS untuk domain. SES mengharuskan Anda untuk mempublikasikan catatan MX sehingga domain Anda dapat menerima pemberitahuan pentalan dan keluhan yang dikirimkan oleh penyedia email kepada Anda. Anda juga harus mempublikasikan catatan SPF (tipe TXT) untuk membuktikan bahwa Amazon SES berwenang untuk mengirim email dari domain Anda.

Anda dapat mengatur domain MAIL FROM kustom untuk seluruh domain atau subdomain, serta untuk alamat email individual. Prosedur berikut menunjukkan cara menggunakan konsol Amazon SES untuk mengonfigurasi domain MAIL FROM kustom. Anda juga dapat mengonfigurasi domain MAIL FROM kustom menggunakan operasi API SetIdentityMailFromDomain.

Menyiapkan domain MAIL FROM kustom untuk domain terverifikasi

Prosedur ini menunjukkan kepada Anda cara mengonfigurasi domain MAIL FROM kustom untuk seluruh domain atau subdomain sehingga semua pesan yang dikirim dari alamat pada domain tersebut akan menggunakan domain MAIL FROM kustom ini.

Untuk mengonfigurasi domain terverifikasi untuk menggunakan domain MAIL FROM kustom yang ditentukan

1. Buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, di bawah Konfigurasi, pilih Identitas.
3. Dalam daftar identitas, pilih identitas yang ingin Anda konfigurasikan di mana tipe Identitas adalah Domain dan Status Terverifikasi.

   - Jika Status Belum Diverifikasi, selesaikan prosedur di Memverifikasi identitas domain DKIM dengan penyedia DNS Anda untuk memverifikasi domain alamat email.

4. Di bagian bawah layar di panel Custom MAIL FROM domain, pilih Edit.
5. Di panel Detail umum, lakukan hal berikut:

   a. Pilih kotak centang Gunakan email kustom DARI domain.

   b. Untuk domain MAIL FROM, masukkan subdomain yang ingin Anda gunakan sebagai domain MAIL FROM.

   c. Untuk Perilaku pada kegagalan MX, pilih salah satu opsi berikut:

      - Gunakan domain MAIL FROM default — Jika data MX domain MAIL FROM kustom tidak diatur dengan benar, Amazon SES menggunakan subdomain. `amazonses.com` Subdomain bervariasi berdasarkan tempat Wilayah AWS Anda menggunakan Amazon SES.
      - Tolak pesan – Jika catatan MX domain MAIL FROM kustom tidak disiapkan dengan benar, Amazon SES akan mengembalikan kesalahan `MailFromDomainNotVerified`. Email yang Anda coba kirimkan dari domain ini akan ditolak secara otomatis.

   d. Pilih Simpan perubahan - Anda akan dikembalikan ke layar sebelumnya.

6. Publikasikan catatan MX dan SPF (tipe TXT) ke server DNS domain MAIL FROM kustom:

   Di panel domain KUSTOM MAIL FROM, tabel Publikasikan catatan DNS sekarang menampilkan data MX dan SPF (tipe TXT) yang harus Anda publikasikan (tambahkan) ke konfigurasi DNS domain Anda. Catatan ini menggunakan format yang ditunjukkan di tabel berikut.

Nama	Tipe	Nilai
subdomain .*domain*.*com*	MX	10 feedback-smtp.*wilayah*.amazonses.com
subdomain .*domain*.*com*	TXT	"v=spf1 include:amazonses.com ~all"

Dalam catatan sebelumnya,

- *subdomain*. *domain*. *com* akan diisi dengan subdomain MAIL FROM Anda
- *wilayah* akan diisi dengan nama Wilayah AWS tempat Anda ingin memverifikasi domain MAIL FROM (sepertius-west-2,, atau us-east-1eu-west-1, dll.)
- Angka 10 yang tercantum bersama dengan nilai MX adalah urutan preferensi untuk server email dan harus dimasukkan ke dalam bidang nilai terpisah seperti yang ditentukan oleh GUI penyedia DNS Anda
- Nilai catatan TXT SPF harus menyertakan tanda kutip

Dari tabel Publikasikan catatan DNS, salin catatan MX dan SPF (tipe TXT) dengan memilih ikon salin di samping setiap nilai dan tempelkan ke bidang yang sesuai di GUI penyedia DNS Anda. Sebagai alternatif, Anda dapat memilih Unduh set catatan .csv untuk menyimpan salinan catatan ke komputer Anda.

> **Important**
>
> Agar berhasil menyiapkan domain MAIL FROM kustom dengan Amazon SES, Anda harus memublikasikan persis satu catatan MX ke server DNS domain MAIL FROM Anda. Jika domain MAIL FROM memiliki beberapa catatan MX, penyiapan MAIL FROM kustom dengan Amazon SES akan gagal.

Jika Route 53 menyediakan layanan DNS untuk domain MAIL FROM Anda, dan Anda masuk ke akun yang sama dengan AWS Management Console yang Anda gunakan untuk Route 53, lalu

pilih Publikasikan Catatan Menggunakan Route 53. Catatan DNS secara otomatis diterapkan ke konfigurasi DNS domain Anda.

Jika Anda menggunakan penyedia DNS yang berbeda, Anda harus memublikasikan catatan DNS ke server DNS domain MAIL FROM secara manual. Prosedur untuk menambahkan catatan DNS ke server DNS domain bervariasi berdasarkan layanan hosting web atau penyedia DNS Anda.

Prosedur untuk memublikasikan catatan DNS untuk domain Anda bergantung pada penyedia DNS yang digunakan. Tabel berikut mencakup tautan ke dokumentasi untuk beberapa penyedia DNS yang banyak digunakan. Daftar ini tidak lengkap dan tidak menandakan dukungan; demikian juga, jika penyedia DNS Anda tidak terdaftar, itu tidak berarti mereka tidak mendukung konfigurasi domain MAIL FROM.

Nama penyedia DNS/Hosting	Tautan dokumentasi
GoDaddy	• MX: Tambahkan catatan MX (tautan eksternal) • TXT: Tambahkan catatan TXT (tautan eksternal)
DreamHost	• MX: Bagaimana cara mengubah catatan MX saya? (tautan eksternal) • TXT: Bagaimana cara menambahkan catatan DNS kustom? (tautan eksternal)
Cloudflare	• MX: Bagaimana cara menambahkan atau mengedit email atau catatan MX? (tautan eksternal) • TXT: Mengelola catatan DNS di Cloudflare (tautan eksternal)
HostGator	• MX: Siapkan MX Records (tautan eksternal) • TXT: Kelola Catatan DNS HostGator dengan/eNom (tautan eksternal)

Nama penyedia DNS/Hosting	Tautan dokumentasi
Namecheap	• MX: Bagaimana cara menyiapkan catatan MX yang diperlukan untuk layanan email? (tautan eksternal) • TXT: Bagaimana cara menambahkan catatan TXT/SPF/DKIM/DMARC untuk domain saya? (tautan eksternal)
Names.co.uk	• MX: Mengubah pengaturan DNS domain (tautan eksternal) • TXT: Mengubah Pengaturan DNS domain (tautan eksternal)
Wix	• MX: Menambahkan atau Memperbarui Catatan MX di Akun Wix (tautan eksternal) • TXT: Menambahkan atau Memperbarui Catatan TXT di Akun Wix (tautan eksternal)

Ketika Amazon SES mendeteksi bahwa catatan ada di tempatnya, Anda menerima email yang menginformasikan bahwa domain MAIL FROM kustom Anda berhasil disiapkan. Tergantung pada penyedia DNS Anda, mungkin ada penundaan hingga 72 jam sebelum Amazon SES mendeteksi catatan MX.

Menyiapkan domain MAIL FROM kustom untuk alamat email terverifikasi

Anda juga dapat menyiapkan domain MAIL FROM kustom untuk alamat email tertentu. Untuk menyiapkan domain MAIL FROM kustom untuk alamat email, Anda harus mengubah catatan DNS untuk domain yang terkait dengan alamat email.

> **Note**
>
> Anda tidak dapat menyiapkan domain MAIL FROM kustom untuk alamat di domain yang tidak Anda miliki (misalnya, Anda tidak dapat membuat domain MAIL FROM kustom untuk

alamat di domain `gmail.com`, karena Anda tidak dapat menambahkan catatan DNS yang diperlukan ke domain tersebut).

Untuk mengonfigurasi alamat email terverifikasi untuk menggunakan domain MAIL FROM tertentu

1. Buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, di bawah Konfigurasi, pilih Identitas.
3. Dalam daftar identitas, pilih identitas yang ingin Anda konfigurasikan di mana Jenis identitas adalah Alamat email dan Status Terverifikasi.
   - Jika Status Belum Diverifikasi, selesaikan prosedur di Memverifikasi identitas alamat email untuk memverifikasi domain alamat email.
4. Di bawah tab MAIL FROM Domain, pilih Edit di panel Custom MAIL FROM domain.
5. Di panel Detail umum, lakukan hal berikut:
   a. Pilih kotak centang Gunakan email kustom DARI domain.
   b. Untuk domain MAIL FROM, masukkan subdomain yang ingin Anda gunakan sebagai domain MAIL FROM.
   c. Untuk Perilaku pada kegagalan MX, pilih salah satu opsi berikut:
      - Gunakan domain MAIL FROM default — Jika data MX domain MAIL FROM kustom tidak diatur dengan benar, Amazon SES menggunakan subdomain. `amazonses.com` Subdomain bervariasi berdasarkan tempat Wilayah AWS Anda menggunakan Amazon SES.
      - Tolak pesan – Jika catatan MX domain MAIL FROM kustom tidak disiapkan dengan benar, Amazon SES akan mengembalikan kesalahan `MailFromDomainNotVerified`. Email yang Anda coba kirimkan dari alamat email ini akan ditolak secara otomatis.
   d. Pilih Simpan perubahan - Anda akan dikembalikan ke layar sebelumnya.
6. Publikasikan catatan MX dan SPF (tipe TXT) ke server DNS domain MAIL FROM kustom:

   Di panel domain KUSTOM MAIL FROM, tabel Publikasikan catatan DNS sekarang menampilkan data MX dan SPF (tipe TXT) yang harus Anda publikasikan (tambahkan) ke konfigurasi DNS domain Anda. Catatan ini menggunakan format yang ditunjukkan di tabel berikut.

Nama	Tipe	Nilai
subdomain .*domain*.*com*	MX	10 feedback-smtp.*wilayah*.amazonses.com
subdomain .*domain*.*com*	TXT	"v=spf1 include:amazonses.com ~all"

Dalam catatan sebelumnya,

- *subdomain*. *domain*. *com* akan diisi dengan subdomain MAIL FROM Anda
- *wilayah* akan diisi dengan nama Wilayah AWS tempat Anda ingin memverifikasi domain MAIL FROM (sepertius-west-2,, atau us-east-1eu-west-1, dll.)
- Angka 10 yang tercantum bersama dengan nilai MX adalah urutan preferensi untuk server email dan harus dimasukkan ke dalam bidang nilai terpisah seperti yang ditentukan oleh GUI penyedia DNS Anda
- Nilai catatan TXT SPF harus menyertakan tanda kutip

Dari tabel Publikasikan catatan DNS, salin catatan MX dan SPF (tipe TXT) dengan memilih ikon salin di samping setiap nilai dan tempelkan ke bidang yang sesuai di GUI penyedia DNS Anda. Sebagai alternatif, Anda dapat memilih Unduh set catatan .csv untuk menyimpan salinan catatan ke komputer Anda.

> **Important**
>
> Agar berhasil menyiapkan domain MAIL FROM kustom dengan Amazon SES, Anda harus memublikasikan persis satu catatan MX ke server DNS domain MAIL FROM Anda. Jika domain MAIL FROM memiliki beberapa catatan MX, penyiapan MAIL FROM kustom dengan Amazon SES akan gagal.

Jika Route 53 menyediakan layanan DNS untuk domain MAIL FROM Anda, dan Anda masuk ke akun yang sama dengan AWS Management Console yang Anda gunakan untuk Route 53, lalu

pilih Publikasikan Catatan Menggunakan Route 53. Catatan DNS secara otomatis diterapkan ke konfigurasi DNS domain Anda.

Jika Anda menggunakan penyedia DNS yang berbeda, Anda harus memublikasikan catatan DNS ke server DNS domain MAIL FROM secara manual. Prosedur untuk menambahkan catatan DNS ke server DNS domain bervariasi berdasarkan layanan hosting web atau penyedia DNS Anda.

Prosedur untuk memublikasikan catatan DNS untuk domain Anda bergantung pada penyedia DNS yang digunakan. Tabel berikut mencakup tautan ke dokumentasi untuk beberapa penyedia DNS yang banyak digunakan. Daftar ini tidak lengkap dan tidak menandakan dukungan; demikian juga, jika penyedia DNS Anda tidak terdaftar, itu tidak berarti mereka tidak mendukung konfigurasi domain MAIL FROM.

Nama penyedia DNS/Hosting	Tautan dokumentasi
GoDaddy	• MX: Tambahkan catatan MX (tautan eksternal) • TXT: Tambahkan catatan TXT (tautan eksternal)
DreamHost	• MX: Bagaimana cara mengubah catatan MX saya? (tautan eksternal) • TXT: Bagaimana cara menambahkan catatan DNS kustom? (tautan eksternal)
Cloudflare	• MX: Bagaimana cara menambahkan atau mengedit email atau catatan MX? (tautan eksternal) • TXT: Mengelola catatan DNS di Cloudflare (tautan eksternal)
HostGator	• MX: Mengubah catatan MX - Windows (tautan eksternal) • TXT: Kelola Catatan DNS HostGator dengan/eNom (tautan eksternal)

Nama penyedia DNS/Hosting	Tautan dokumentasi
Namecheap	• MX: Bagaimana cara menyiapkan catatan MX yang diperlukan untuk layanan email? (tautan eksternal) • TXT: Bagaimana cara menambahkan catatan TXT/SPF/DKIM/DMARC untuk domain saya? (tautan eksternal)
Names.co.uk	• MX: Mengubah pengaturan DNS domain (tautan eksternal) • TXT: Mengubah Pengaturan DNS domain (tautan eksternal)
Wix	• MX: Menambahkan atau Memperbarui Catatan MX di Akun Wix (tautan eksternal) • TXT: Menambahkan atau Memperbarui Catatan TXT di Akun Wix (tautan eksternal)

Ketika Amazon SES mendeteksi bahwa catatan ada di tempatnya, Anda menerima email yang menginformasikan bahwa domain MAIL FROM kustom Anda berhasil disiapkan. Tergantung pada penyedia DNS Anda, mungkin ada penundaan hingga 72 jam sebelum Amazon SES mendeteksi catatan MX.

EMAIL kustom DARI status pengaturan domain dengan Amazon SES

Setelah Anda mengonfigurasi identitas untuk menggunakan domain MAIL FROM kustom, status penyiapan adalah "tertunda" sementara Amazon SES mencoba untuk mendeteksi catatan MX yang diperlukan di pengaturan DNS Anda. Status kemudian berubah tergantung jika Amazon SES mendeteksi catatan MX. Tabel berikut menjelaskan perilaku pengiriman email, dan tindakan Amazon SES terkait dengan masing-masing status. Setiap kali status berubah, Amazon SES mengirimkan pemberitahuan ke alamat email yang terkait dengan Anda Akun AWS.

Status	Perilaku pengiriman email	Tindakan Amazon SES
Tertunda	Menggunakan pengaturan fallback MAIL FROM kustom	Amazon SES mencoba untuk mendeteksi catatan MX yang diperlukan selama 72 jam. Jika tidak berhasil, status berubah menjadi "Gagal".
Berhasil	Menggunakan domain MAIL FROM kustom	Amazon SES terus memeriksa bahwa catatan MX yang diperlukan ada di tempatnya.
Temporary Failure	Menggunakan pengaturan fallback MAIL FROM kustom	Amazon SES mencoba untuk mendeteksi catatan MX yang diperlukan selama 72 jam. Jika tidak berhasil, status berubah menjadi "Gagal"; jika berhasil, status berubah menjadi "Berhasil".
Gagal	Menggunakan pengaturan fallback MAIL FROM kustom	Amazon SES tidak lagi

Status	Perilaku pengiriman email	Tindakan Amazon SES
		mencoba untuk mendeteksi catatan MX yang diperlukan. Untuk menggunakan domain MAIL FROM kustom, Anda harus memulai ulang proses pengaturan di Mengonfigurasi domain MAIL FROM kustom Anda.

Mematuhi protokol DMARC otentikasi di Amazon SES

Domain-based Message Authentication, Reporting and Conformance (DMARC) adalah protokol otentikasi email yang menggunakan Sender Policy Framework (SPF) dan DomainKeys Identified Mail (DKIM) untuk mendeteksi spoofing dan phishing email. Untuk mematuhiDMARC, pesan harus diautentikasi melalui salah satu SPF atauDKIM, tetapi idealnya, ketika keduanya digunakanDMARC, Anda akan memastikan tingkat perlindungan tertinggi yang mungkin untuk pengiriman email Anda.

Mari kita tinjau secara singkat mana yang masing-masing lakukan dan bagaimana DMARC mengikat mereka semua:

- SPF— Mengidentifikasi server email mana yang diizinkan untuk mengirim email atas nama MAIL FROM domain kustom Anda melalui DNS TXT catatan yang digunakan olehDNS. Sistem surat penerima merujuk ke SPF TXT catatan untuk menentukan apakah pesan dari domain kustom Anda berasal dari server pesan resmi. Pada dasarnya, SPF dirancang untuk membantu mencegah spoofing, tetapi ada teknik spoofing yang SPF rentan dalam praktik dan inilah mengapa Anda juga perlu menggunakannya DKIM bersama. DMARC

- DKIM— Menambahkan tanda tangan digital ke pesan keluar Anda di header email. Sistem email penerima dapat menggunakan tanda tangan digital ini untuk membantu memverifikasi apakah email masuk ditandatangani oleh kunci yang dimiliki oleh domain. Namun, ketika sistem email penerima meneruskan pesan, amplop pesan diubah dengan cara yang membatalkan SPF otentikasi. Karena tanda tangan digital tetap dengan pesan email karena itu adalah bagian dari header email, DKIM berfungsi bahkan ketika pesan telah diteruskan antara server email (selama konten pesan belum diubah).
- DMARC— Memastikan bahwa ada penyelarasan domain dengan setidaknya satu dari SPF danDKIM. Menggunakan SPF dan DKIM sendirian tidak melakukan apa pun untuk memastikan bahwa alamat Dari diautentikasi (ini adalah alamat email yang dilihat penerima Anda di klien email mereka). SPFhanya memeriksa domain yang ditentukan dalam MAIL FROM alamat (tidak dilihat oleh penerima Anda). DKIMhanya memeriksa domain yang ditentukan dalam DKIM tanda tangan (juga, tidak dilihat oleh penerima Anda). DMARCmengatasi dua masalah ini dengan mengharuskan penyelarasan domain menjadi benar pada salah satu SPF atauDKIM:
  - SPFUntuk meneruskan DMARC penyelarasan domain di alamat Dari harus cocok dengan domain di MAIL FROM alamat (juga disebut sebagai Return-Path dan Envelope-from address). Ini jarang dimungkinkan dengan surat yang diteruskan karena akan dilucuti atau ketika mengirim email melalui penyedia email massal pihak ketiga karena Return-Path (MAILFROM) digunakan untuk pantulan dan keluhan yang dilacak oleh penyedia (SES) menggunakan alamat yang mereka miliki.
  - DKIMUntuk meneruskan DMARC penyelarasan, domain yang ditentukan dalam DKIM tanda tangan harus cocok dengan domain di alamat Dari. Jika Anda menggunakan pengirim atau layanan pihak ketiga yang mengirim email atas nama Anda, hal ini dapat dilakukan dengan memastikan pengirim pihak ketiga dikonfigurasi dengan benar untuk DKIM penandatanganan dan Anda telah menambahkan DNS catatan yang sesuai dalam domain Anda. Menerima server email kemudian akan dapat memverifikasi email yang dikirim kepada mereka oleh pihak ketiga Anda seolah-olah itu adalah email yang dikirim oleh seseorang yang berwenang untuk menggunakan alamat dalam domain.

Menyatukan semuanya dengan DMARC

Pemeriksaan DMARC keselarasan yang kita bahas di atas menunjukkan bagaimana SPFDKIM,, dan DMARC semua bekerja sama untuk meningkatkan kepercayaan domain Anda dan pengiriman email Anda ke kotak masuk. DMARCmenyelesaikan ini dengan memastikan bahwa alamat Dari, dilihat oleh penerima, diautentikasi oleh salah satu atauSPF: DKIM

- Sebuah pesan lewat DMARC jika salah satu atau kedua yang dijelaskan SPF atau DKIM cek lulus.
- Pesan gagal DMARC jika kedua yang dijelaskan SPF atau DKIM pemeriksaan gagal.

Oleh karena itu, DKIM keduanya SPF dan diperlukan DMARC untuk memiliki kesempatan terbaik untuk mencapai otentikasi untuk email yang Anda kirim, dan dengan memanfaatkan ketiganya, Anda akan membantu memastikan Anda memiliki domain pengiriman yang sepenuhnya dilindungi.

DMARCjuga memungkinkan Anda untuk menginstruksikan server email bagaimana menangani email ketika mereka gagal DMARC otentikasi melalui kebijakan yang Anda tetapkan. Ini akan dijelaskan di bagian berikut,the section called “Menyiapkan DMARC kebijakan di domain Anda”, yang berisi informasi tentang cara mengonfigurasi SES domain Anda sehingga email yang Anda kirim mematuhi protokol DMARC otentikasi melalui keduanya SPF dan. DKIM

Menyiapkan DMARC kebijakan di domain Anda

Untuk mengaturDMARC, Anda harus mengubah DNS pengaturan untuk domain Anda. DNSPengaturan untuk domain Anda harus menyertakan TXT catatan yang menentukan DMARC pengaturan domain. Prosedur untuk menambahkan TXT catatan ke DNS konfigurasi Anda tergantung pada penyedia hosting yang DNS Anda gunakan. Jika Anda menggunakan Route 53, lihat Bekerja dengan Catatan di Panduan Developer Amazon Route 53. Jika Anda menggunakan penyedia lain, lihat dokumentasi DNS konfigurasi untuk penyedia Anda.

Nama TXT catatan yang Anda buat harus`_dmarc.`*`example.com`*, di *`example.com`* mana domain Anda. Nilai TXT rekaman berisi DMARC kebijakan yang berlaku untuk domain Anda. Berikut ini adalah contoh TXT catatan yang berisi DMARC kebijakan:

Nama	Tipe	Nilai
`_dmarc.example.com`	`TXT`	`"v=DMARC1;p=quarantine;rua=mailto:my_dmarc_report@example.com"`

Dalam contoh DMARC kebijakan sebelumnya, kebijakan ini memberi tahu penyedia email untuk melakukan hal berikut:

- Untuk setiap pesan yang gagal otentikasi, kirim ke folder Spam seperti yang ditentukan oleh parameter kebijakan,`p=quarantine`. Pilihan lain termasuk tidak melakukan apa-apa dengan menggunakan`p=none`, atau menolak pesan langsung dengan menggunakan. `p=reject`
  - Bagian selanjutnya membahas bagaimana dan kapan menggunakan ketiga pengaturan kebijakan ini — menggunakan yang salah pada waktu yang salah dapat menyebabkan email Anda tidak terkirim, lihatthe section called “Implementasi DMARC”.
- Kirim laporan tentang semua email yang gagal otentikasi dalam intisari (yaitu, laporan yang mengumpulkan data untuk jangka waktu tertentu, daripada mengirim laporan individual untuk setiap peristiwa) sebagaimana ditentukan oleh parameter pelaporan, `rua=mailto:my_dmarc_report@example.com` (rua singkatan dari Pelaporan URI untuk laporan agregat). Penyedia email biasanya mengirimkan laporan gabungan ini satu kali per hari, meskipun kebijakan ini berbeda antara satu penyedia dengan penyedia lainnya.

Untuk mempelajari lebih lanjut tentang mengonfigurasi DMARC domain Anda, lihat Ikhtisar di DMARC situs web.

Untuk spesifikasi lengkap DMARC sistem, lihat Internet Engineering Task Force (IETF) DMARC Draft.

Praktik terbaik untuk mengimplementasikan DMARC

Yang terbaik adalah menerapkan penegakan DMARC kebijakan Anda secara bertahap dan bertahap sehingga tidak mengganggu sisa alur email Anda. Buat dan implementasikan rencana peluncuran yang mengikuti langkah-langkah ini. Lakukan setiap langkah ini terlebih dahulu dengan masing-masing sub-domain Anda, dan terakhir dengan domain tingkat atas di organisasi Anda sebelum melanjutkan ke langkah berikutnya.

1. Pantau dampak implementasi DMARC (p=none).
   - Mulailah dengan catatan mode pemantauan sederhana untuk sub-domain atau domain yang meminta organisasi penerima email mengirimi Anda statistik tentang pesan yang mereka lihat menggunakan domain tersebut. Rekaman mode pemantauan adalah DMARC TXT rekaman yang kebijakannya disetel ke none. `p=none`
   - Laporan yang dihasilkan DMARC akan memberikan nomor dan sumber pesan yang lulus pemeriksaan ini, dibandingkan yang tidak. Anda dapat dengan mudah melihat berapa banyak lalu lintas sah Anda atau tidak tercakup oleh mereka. Anda akan melihat tanda-tanda penerusan, karena pesan yang diteruskan akan gagal SPF dan DKIM jika konten diubah. Anda

juga akan mulai melihat berapa banyak pesan penipuan yang dikirim, dan dari mana mereka dikirim.

- Tujuan dari langkah ini adalah untuk mempelajari email apa yang akan terpengaruh ketika Anda menerapkan salah satu dari dua langkah berikutnya, dan agar pengirim pihak ketiga atau pengirim yang berwenang mendapatkan DKIM kebijakan SPF atau kebijakannya ke dalam keselarasan.
- Terbaik untuk domain yang ada.

2. Minta agar sistem surat eksternal mengkarantina surat yang gagal DMARC (p=karantina).
   - Ketika Anda yakin bahwa semua atau sebagian besar lalu lintas sah Anda mengirimkan domain yang selaras dengan salah satu SPF atauDKIM, dan Anda memahami dampak penerapannyaDMARC, Anda dapat menerapkan kebijakan karantina. Kebijakan karantina adalah DMARC TXT catatan yang memiliki kebijakan yang ditetapkan untuk karantinap=`quarantine`. Dengan melakukan ini, Anda meminta DMARC penerima untuk memasukkan pesan dari domain Anda yang gagal DMARC ke folder spam lokal yang setara, bukan kotak masuk pelanggan Anda.
   - Terbaik untuk mentransisikan domain yang telah menganalisis DMARC laporan selama Langkah 1.
3. Meminta agar sistem surat eksternal tidak menerima pesan yang gagal DMARC (p=reject).
   - Menerapkan kebijakan penolakan biasanya merupakan langkah terakhir. Kebijakan penolakan adalah DMARC TXT catatan yang kebijakannya ditetapkan untuk ditolakp=`reject`. Ketika Anda melakukan ini, Anda meminta DMARC penerima untuk tidak menerima pesan yang gagal DMARC dicek—ini berarti mereka bahkan tidak akan dikarantina ke folder spam atau sampah, tetapi akan langsung ditolak.
   - Saat menggunakan kebijakan penolakan, Anda akan tahu persis pesan mana yang gagal dalam DMARC kebijakan karena penolakan akan menghasilkan pantulan. SMTP Dengan karantina, data agregat memberikan informasi tentang persentase email yang lewat atau gagalSPF,, DKIM dan pemeriksaan. DMARC
   - Terbaik untuk domain baru atau domain yang sudah ada yang telah melalui dua langkah sebelumnya.

Mematuhi melalui DMARC SPF

Agar email dapat dipatuhi DMARC berdasarkanSPF, kedua kondisi berikut harus dipenuhi:

- Pesan harus melewati SPF pemeriksaan berdasarkan memiliki catatan SPF (tipeTXT) valid yang harus Anda publikasikan ke DNS konfigurasi MAIL FROM domain kustom Anda.
- Domain di alamat Dari header email harus sejajar (cocok) dengan domain, atau subdomain dari, yang ditentukan dalam alamat. MAIL FROM Untuk mencapai SPF keselarasan denganSES, kebijakan domain tidak boleh menentukan DMARC SPF kebijakan ketat (aspf=s).

Untuk memenuhi persyaratan ini, selesaikan langkah berikut:

- Siapkan MAIL FROM domain kustom dengan menyelesaikan prosedur dithe section called "Menggunakan domain MAIL FROM kustom".
- Pastikan domain pengiriman Anda menggunakan kebijakan santai untukSPF. Jika Anda belum mengubah penyelarasan kebijakan domain Anda, ia menggunakan kebijakan santai secara default seperti halnyaSES.

> **Note**
>
> Anda dapat menentukan DMARC perataan domain Anda SPF dengan mengetikkan perintah berikut di baris perintah, menggantinya *example.com* dengan domain Anda:
>
> ```
> dig TXT _dmarc.example.com
> ```
>
> Di output perintah ini, di bawah Jawaban nonotoritatif, cari catatan yang diawali dengan `v=DMARC1`. Jika catatan ini menyertakan string`aspf=r`, atau jika `aspf` string tidak ada sama sekali, maka domain Anda menggunakan perataan santai untukSPF. Jika catatan menyertakan string`aspf=s`, maka domain Anda menggunakan perataan ketat untukSPF. Administrator sistem Anda harus menghapus tag ini dari DMARC TXT catatan dalam DNS konfigurasi domain Anda.
> Atau, Anda dapat menggunakan alat DMARC pencarian berbasis web, seperti DMARCInspector dari situs web dmarcian atau alat Alat DMARC Periksa dari situs MxToolBox web, untuk menentukan penyelarasan kebijakan domain Anda. SPF

Mematuhi melalui DMARC DKIM

Agar email dapat dipatuhi DMARC berdasarkanDKIM, kedua kondisi berikut harus dipenuhi:

- Pesan harus memiliki DKIM tanda tangan yang valid dan melewati DKIM cek.

- Domain yang ditentukan dalam DKIM tanda tangan harus sejajar (cocok) dengan domain di alamat Dari. Jika DMARC kebijakan domain menentukan perataan ketat untukDKIM, domain ini harus sama persis (SESmenggunakan DKIM kebijakan ketat secara default).

Untuk memenuhi persyaratan ini, selesaikan langkah berikut:

- Siapkan Mudah DKIM dengan menyelesaikan prosedur dithe section called “Easy DKIM”. Saat Anda menggunakan EasyDKIM, Amazon SES secara otomatis menandatangani email Anda.

> **Note**
>
> Daripada menggunakan EasyDKIM, Anda juga dapat menandatangani pesan secara manual. Namun, berhati-hatilah jika Anda memilih untuk melakukannya, karena Amazon SES tidak memvalidasi DKIM tanda tangan yang Anda buat. Untuk alasan ini, kami sangat menyarankan menggunakan EasyDKIM.

- Pastikan domain yang ditentukan dalam DKIM tanda tangan disejajarkan dengan domain di alamat Dari. Atau, jika mengirim dari subdomain domain di alamat Dari, pastikan DMARC kebijakan Anda disetel ke penyelarasan santai.

> **Note**
>
> Anda dapat menentukan DMARC perataan domain Anda DKIM dengan mengetikkan perintah berikut di baris perintah, menggantinya *example.com* dengan domain Anda:
>
> ```
> dig TXT _dmarc.example.com
> ```
>
> Di output perintah ini, di bawah Jawaban nonotoritatif, cari catatan yang diawali dengan `v=DMARC1`. Jika catatan ini menyertakan string`adkim=r`, atau jika `adkim` string tidak ada sama sekali, maka domain Anda menggunakan perataan santai untukDKIM. Jika catatan menyertakan string`adkim=s`, maka domain Anda menggunakan perataan ketat untukDKIM. Administrator sistem Anda harus menghapus tag ini dari DMARC TXT catatan dalam DNS konfigurasi domain Anda.
> Atau, Anda dapat menggunakan alat DMARC pencarian berbasis web, seperti DMARCInspector dari situs web dmarcian atau alat Alat DMARC Periksa dari situs MxToolBox web, untuk menentukan penyelarasan kebijakan domain Anda. DKIM

Mengkonfigurasi BIMI di Amazon SES

Indikator Merek untuk Identifikasi Pesan (BIMI) adalah spesifikasi email yang memungkinkan kotak masuk email menampilkan logo merek di samping pesan email terotentikasi merek dalam mendukung klien email.

BIMI adalah spesifikasi email yang terhubung langsung ke otentikasi, tetapi ini bukan protokol otentikasi email mandiri karena memerlukan semua email Anda untuk mematuhi otentikasi DMARC.

Sementara BIMI membutuhkan DMARC, DMARC mengharuskan domain Anda untuk memiliki data SPF atau DKIM agar selaras, tetapi sebaiknya sertakan catatan SPF dan DKIM untuk keamanan tambahan, dan karena beberapa penyedia layanan email (ESP) memerlukan keduanya saat menggunakan BIMI. Bagian berikut membahas langkah-langkah untuk mengimplementasikan BIMI di Amazon SES.

Menyiapkan BIMI di SES

Anda dapat mengkonfigurasi BIMI untuk domain email yang Anda miliki—dalam SES yang disebut sebagai domain MAIL FROM kustom. Setelah dikonfigurasi, semua pesan yang Anda kirim dari domain tersebut akan menampilkan logo tersebut akan menampilkan logo Anda di email klien Anda di email yang mendukung BIMI Anda.

Mengaktifkan email Anda untuk menampilkan logo BIMI memerlukan beberapa prasyarat untuk berada di tempat dalam SES — dalam prosedur berikut, prasyarat ini digeneralisasikan dan akan merujuk bagian khusus yang mencakup topik ini secara rinci. Langkah-langkah khusus untuk BIMI dan apa yang diperlukan untuk mengkonfigurasinya di SES akan dirinci di sini.

Mengkonfigurasi BIMI di domain kustom PESAN FROM kustom

1. Anda harus memiliki domain MAIL FROM kustom yang dikonfigurasi dalam SES dengan data SPF (tipe TXT) dan MX yang diterbitkan untuk domain tersebut. Jika Anda tidak memiliki domain pesan PESAN PESAN kustom kustom, atau ingin membuat yang baru untuk logo BIMI Anda, silakan lihatthe section called “Menggunakan domain MAIL FROM kustom”.
2. Konfigurasikan domain Anda dengan Easy DKIM. Lihat the section called “Easy DKIM”.
3. Konfigurasikan domain Anda dengan DMARC dengan menerbitkan data TXT dengan penyedia DNS Anda dengan spesifikasi kebijakan penegakan berikut yang diperlukan untuk BIMI:

Nama	Tipe	Nilai
`_dmarc.example.com`	`TXT`	`v=DMARC1;p=quarantine;pct=100;rua=mailto:dmarcreports@example.com`
		`v=DMARC1;p=reject;rua=mailto:dmarcreports@example.com`

Dalam contoh kebijakan DMARC sebelumnya sebagaimana diperlukan untuk BIMI:

- *`example.com`*harus diganti dengan nama domain atau subdomain Anda.
- p=Nilainya bisa berupa:
  - karantina dengan nilai pct diatur ke 100 seperti yang ditunjukkan, atau
  - menolak seperti yang ditunjukkan.
- Jika Anda mengirim dari subdomain, BIMI mengharuskan domain induk juga harus memiliki kebijakan penegakan ini. Subdomain akan termasuk dalam kebijakan domain induk. Namun, jika Anda menambahkan data DMARC untuk subdomain Anda selain apa yang diposting untuk domain induk, subdomain Anda juga harus memiliki kebijakan penegakan yang sama agar memenuhi syarat untuk BIMI.
- Jika Anda belum pernah menyiapkan kebijakan DMARC untuk domain Anda, lihatthe section called “Mengautentikasi Email dengan DMARC” memastikan bahwa Anda hanya menggunakan nilai kebijakan DMARC khusus untuk BIMI seperti yang ditunjukkan.

4. Buat logo BIMI Anda sebagai`.svg` file Scalable Vector Graphics (SVG) — profil SVG spesifik yang dibutuhkan oleh BIMI didefinisikan sebagai SVG Portable/Secure (SVG P/S). Agar logo Anda ditampilkan di klien email itu harus sesuai persis dengan spesifikasi ini. Lihat panduan BIMI Group untuk membuat file logo SVG dan alat konversi SVG yang direkomendasikan.
5. (Opsional) Dapatkan Sertifikat Tanda Terverifikasi (VMC). Beberapa ESP, seperti Gmail dan Apple, memerlukan VMC untuk memberikan bukti bahwa Anda memiliki merek dagang dan konten logo BIMI Anda. Meskipun ini bukan persyaratan untuk menerapkan BIMI pada domain Anda, logo BIMI Anda tidak akan ditampilkan di klien email jika ESP Anda mengirim email untuk menegakkan kepatuhan VMC. Lihat referensi Grup BIMI kepada otoritas sertifikat yang berpartisipasi untuk mendapatkan VMC untuk logo Anda.

6. Host file SVG logo BIMI Anda di server Anda memiliki akses untuk membuatnya dapat diakses publik melalui HTTPS. Misalnya, Anda dapat mengunggahnya ke bucket Amazon S3.
7. Buat dan publikasikan data DNS BIMI yang menyertakan URL ke logo Anda. Ketika ESP yang mendukung BIMI memeriksa data DMARC Anda, itu juga akan mencari catatan BIMI yang berisi URL untuk `.svg` file logo Anda, dan jika dikonfigurasi, URL untuk `.pem` file VMC Anda. Jika catatan cocok, mereka akan menampilkan logo BIMI Anda.

   Konfigurasikan domain Anda dengan BIMI dengan menerbitkan data TXT dengan penyedia DNS Anda dengan nilai berikut seperti yang ditampilkan—pengiriman dari domain diwakili dalam contoh pertama; pengiriman dari subdomain diwakili dalam contoh kedua:

Nama	Tipe	Nilai
`default._bimi.example.com`	TXT	`v=BIMI1;l=https://myhostingserver.com/images/logo.svg;a=https://myhostingserver.com/certificate/vmc_2023-01-01.pem`
`default._bimi.marketing.example.com`		

   Dalam contoh catatan BIMI sebelumnya:

   - Nilai nama harus secara harfiah menentukan`default._bimi.` sebagai subdomain*`example.com`* atau*`marketing.example.com`* yang harus diganti dengan nama domain atau subdomain Anda.
   - `v`=Nilainya adalah versi catatan BIMI.
   - `l`=Nilainya adalah logo yang mewakili URL yang menunjuk ke `.svg` file gambar Anda.
   - `a`=Nilai adalah otoritas yang mewakili URL yang menunjuk ke `.pem` file sertifikat Anda.

   Anda dapat memvalidasi data BIMI Anda dengan alat seperti BIMI Inspector Grup BIMI.

Langkah terakhir dalam proses ini adalah memiliki pola pengiriman reguler ke ESP yang mendukung penempatan logo BIMI. Domain Anda harus memiliki irama pengiriman reguler dan harus memiliki reputasi yang baik dengan ESP yang Anda kirimkan. Penempatan logo BIMI mungkin membutuhkan waktu untuk mengisi ke ESP di mana Anda tidak memiliki reputasi yang mapan atau mengirim irama.

Anda dapat menemukan lebih banyak informasi dan sumber daya yang berkaitan dengan BIMI melalui organisasi Grup BIMI.

Menyiapkan pemberitahuan acara untuk Amazon SES

Dalam rangka mengirim email menggunakan Amazon SES, Anda harus memiliki sistem di tempat untuk mengelola pentalan dan aduan. Amazon SES dapat memberi tahu Anda tentang peristiwa pentalan atau aduan dalam tiga cara: dengan mengirimkan email notifikasi, dengan memberitahukan topik Amazon SNS, atau dengan memublikasikan peristiwa pengiriman. Bagian ini berisi informasi tentang penyiapan Amazon SES untuk mengirim jenis notifikasi tertentu melalui email atau dengan memberi tahu topik Amazon SNS. Untuk informasi selengkapnya tentang publikasi peristiwa pengiriman, lihat Pantau pengiriman email menggunakan penerbitan SES acara Amazon.

Anda dapat menyiapkan notifikasi menggunakan konsol Amazon SES atau API Amazon SES.

Topik

- Pertimbangan penting
- Menerima notifikasi Amazon SES melalui email
- Menerima notifikasi Amazon SES menggunakan Amazon SNS

Pertimbangan penting

Ada beberapa poin penting yang perlu dipertimbangkan ketika Anda mengatur Amazon SES untuk mengirim notifikasi:

- Notifikasi email dan Amazon SNS berlaku untuk identitas individu (alamat email atau domain terverifikasi yang Anda gunakan untuk mengirim email). Ketika Anda mengaktifkan notifikasi untuk identitas, Amazon SES hanya mengirimkan notifikasi untuk email yang dikirim dari identitas tersebut, dan hanya di Wilayah AWS tempat Anda mengonfigurasi notifikasi.
- Anda harus mengaktifkan satu metode penerimaan notifikasi pentalan atau aduan. Anda dapat mengirim notifikasi ke domain atau alamat email yang menghasilkan pentalan atau aduan, atau ke topik Amazon SNS. Anda juga dapat menggunakan penerbitan acara untuk mengirim pemberitahuan tentang beberapa jenis acara (termasuk pantulan, keluhan, pengiriman, dan lainnya) ke topik Amazon SNS atau aliran Firehose.

  Jika Anda tidak mengatur salah satu metode notifikasi penerimaan pentalan atau aduan ini, Amazon SES secara otomatis meneruskan notifikasi pentalan dan aduan ke alamat Jalur Kembali (atau alamat Sumber, jika Anda tidak menentukan alamat Jalur Kembali) di email yang

mengakibatkan peristiwa pentalan atau aduan, bahkan jika Anda menonaktifkan penerusan umpan balik email.

Jika Anda menonaktifkan penerusan umpan balik email dan mengaktifkan publikasi peristiwa, Anda harus menerapkan set konfigurasi yang berisi aturan publikasi peristiwa untuk semua email yang Anda kirim. Dalam situasi ini, jika Anda tidak menggunakan set konfigurasi, Amazon SES secara otomatis meneruskan notifikasi pentalan dan aduan ke Jalur Kembali atau alamat Sumber di email yang mengakibatkan peristiwa pentalan atau aduan.

- Jika Anda menyiapkan Amazon SES untuk mengirim peristiwa pentalan dan aduan menggunakan lebih dari satu metode (seperti dengan mengirim notifikasi email dan dengan menggunakan peristiwa pengiriman), Anda mungkin menerima lebih dari satu notifikasi untuk peristiwa yang sama.

Menerima notifikasi Amazon SES melalui email

Amazon SES dapat mengirimkan email ketika Anda menerima pentalan dan aduan dengan menggunakan proses yang disebut penerusan umpan balik email.

Dalam rangka mengirim email menggunakan Amazon SES, Anda harus mengonfigurasinya untuk mengirim notifikasi pentalan dan aduan dengan menggunakan salah satu metode berikut:

- Dengan mengaktifkan penerusan umpan balik email. Prosedur untuk menyiapkan tipe notifikasi ini disertakan di bagian ini.
- Dengan mengirimkan notifikasi ke topik Amazon SNS. Untuk informasi lebih lanjut, lihat Menerima notifikasi Amazon SES menggunakan Amazon SNS.
- Dengan memublikasikan notifikasi peristiwa. Untuk informasi lebih lanjut, lihat Pantau pengiriman email menggunakan penerbitan SES acara Amazon.

> **Important**
>
> Untuk beberapa poin penting tentang notifikasi, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.

Topik

- Mengaktifkan penerusan umpan balik email

- Menonaktifkan penerusan umpan balik email
- Tujuan penerusan umpan balik email

Mengaktifkan penerusan umpan balik email

Penerusan umpan balik email diaktifkan secara default. Jika sebelumnya Anda menonaktifkannya, Anda dapat mengaktifkannya dengan mengikuti prosedur di bagian ini.

Untuk mengaktifkan pentalan dan penerusan aduan melalui email menggunakan konsol Amazon SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Dalam daftar alamat email atau domain terverifikasi, pilih alamat email atau domain yang ingin Anda konfigurasikan notifikasi pentalan dan aduannya.
4. Di panel detail, perluas bagian Notifikasi.
5. Pilih Edit Konfigurasi.
6. Di bawah Penerusan Umpan Balik Email, pilih Diaktifkan.

> **Note**
>
> Perubahan yang Anda buat di halaman ini mungkin memerlukan beberapa menit untuk diterapkan.

Anda juga dapat mengaktifkan pemberitahuan pentalan dan keluhan melalui email dengan menggunakan operasi SetIdentityFeedbackForwardingEnabledAPI.

Menonaktifkan penerusan umpan balik email

Jika Anda menyiapkan metode yang berbeda dalam memberikan notifikasi pentalan dan aduan, Anda dapat menonaktifkan penerusan umpan balik email sehingga Anda tidak menerima banyak notifikasi ketika terjadi peristiwa pentalan atau aduan.

Untuk menonaktifkan penerusan pentalan dan aduan melalui email menggunakan konsol Amazon SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Dalam daftar alamat email atau domain terverifikasi, pilih alamat email atau domain yang ingin Anda konfigurasikan notifikasi pentalan dan aduannya.
4. Di panel detail, perluas bagian Notifikasi.
5. Pilih Edit Konfigurasi.
6. Di bawah Penerusan Umpan Balik Email, pilih Dinonaktifkan.

> **Note**
>
> Anda harus mengonfigurasi salah satu metode penerimaan notifikasi pentalan dan aduan untuk mengirim email melalui Amazon SES. Jika menonaktifkan penerusan umpan balik email, Anda harus mengaktifkan notifikasi yang dikirim oleh Amazon SNS, atau mempublikasikan peristiwa pentalan dan keluhan ke topik Amazon SNS atau aliran Firehose dengan menggunakan penerbitan acara. Jika Anda menggunakan publikasi peristiwa, Anda juga harus menerapkan set konfigurasi yang berisi aturan publikasi peristiwa ke setiap email yang Anda kirim. Jika Anda tidak menyiapkan metode penerimaan notifikasi pentalan dan aduan, Amazon SES secara otomatis meneruskan notifikasi umpan balik melalui email ke alamat di bidang Jalur Kembali (atau bidang Sumber, jika Anda tidak menentukan alamat Jalur Kembali) dari pesan yang mengakibatkan peristiwa pentalan atau aduan. Dalam situasi ini, Amazon SES meneruskan notifikasi pentalan dan aduan bahkan jika Anda menonaktifkan notifikasi umpan balik email.

7. Untuk menyimpan konfigurasi notifikasi Anda, pilih Simpan Config.

> **Note**
>
> Perubahan yang Anda buat di halaman ini mungkin perlu beberapa menit untuk diterapkan.

Anda juga dapat menonaktifkan pemberitahuan pentalan dan keluhan melalui email dengan menggunakan operasi SetIdentityFeedbackForwardingEnabledAPI.

Tujuan penerusan umpan balik email

Ketika Anda menerima notifikasi melalui email, Amazon SES menulis ulang header `From` dan mengirimkan notifikasi kepada Anda. Alamat tempat Amazon SES meneruskan notifikasi tergantung pada cara Anda mengirim pesan asli.

Jika Anda menggunakan antarmuka SMTP untuk mengirim pesan, maka notifikasi dikirimkan sesuai dengan aturan berikut:.

- Jika Anda menentukan `Return-Path` header di `SMTP DATA` bagian tersebut, maka notifikasi masuk ke alamat itu.
- Jika tidak, pemberitahuan pergi ke alamat yang Anda tentukan saat Anda mengeluarkan perintah MAIL FROM.

Jika Anda menggunakan operasi API `SendEmail` untuk mengirim pesan, maka notifikasi dikirimkan sesuai dengan aturan berikut:

- Jika Anda menentukan parameter `ReturnPath` opsional dalam panggilan Anda ke API `SendEmail`, maka notifikasi masuk ke alamat tersebut.
- Jika tidak, notifikasi masuk ke alamat yang ditentukan dalam parameter `Source` dari `SendEmail` yang diperlukan.

Jika Anda menggunakan operasi API `SendRawEmail` untuk mengirim pesan, maka notifikasi dikirimkan sesuai dengan aturan berikut:

- Jika Anda menentukan `Return-Path` header dalam pesan mentah, maka notifikasi masuk ke alamat itu.
- Jika tidak, jika Anda menentukan `Source` parameter dalam panggilan Anda ke `SendRawEmail` API, maka notifikasi akan masuk ke alamat tersebut.
- Sebaliknya, notifikasi masuk ke alamat di header `From` dari pesan mentah.

> **Note**
>
> Ketika Anda menentukan alamat `Return-Path` di email, Anda menerima notifikasi di alamat tersebut. Namun, versi pesan yang diterima oleh penerima berisi header `Return-Path` yang menyertakan alamat email anonim (seperti a0b1c2d3e4f5a6b7-c8d9e0f1-a2b3-c4d5-e6f7-a8b9c0d1e2f3-000000@amazonses.com). Anonimisasi ini terjadi terlepas dari cara Anda mengirim email.

Menerima notifikasi Amazon SES menggunakan Amazon SNS

Anda dapat mengonfigurasi Amazon SES untuk memberi tahu topik Amazon SNS ketika Anda menerima pentalan atau aduan, atau ketika email dikirim. Notifikasi Amazon SNS berada dalam format JavaScript Object Notation (JSON), yang memungkinkan Anda untuk memprosesnya secara terprogram.

Dalam rangka mengirim email menggunakan Amazon SES, Anda harus mengonfigurasinya untuk mengirim notifikasi pentalan dan aduan dengan menggunakan salah satu metode berikut:

- Dengan mengirimkan notifikasi ke topik Amazon SNS. Prosedur untuk menyiapkan tipe notifikasi ini disertakan di bagian ini.
- Dengan mengaktifkan penerusan umpan balik email. Untuk informasi lebih lanjut, lihat Menerima notifikasi Amazon SES melalui email.
- Dengan memublikasikan notifikasi peristiwa. Untuk informasi lebih lanjut, lihat Pantau pengiriman email menggunakan penerbitan SES acara Amazon.

> **Important**
>
> Lihat Menyiapkan pemberitahuan acara untuk Amazon SES untuk informasi penting tentang notifikasi.

Topik

- Mengonfigurasi notifikasi Amazon SNS untuk Amazon SES
- Isi notifikasi Amazon SNS untuk Amazon SES
- Contoh notifikasi Amazon SNS untuk Amazon SES

Mengonfigurasi notifikasi Amazon SNS untuk Amazon SES

Amazon SES dapat memberi tahu Anda tentang pentalan, aduan, dan pengiriman Anda melalui Amazon Simple Notification Service (Amazon SNS).

Anda dapat mengonfigurasi notifikasi di konsol Amazon SES, atau dengan menggunakan API Amazon SES.

Topik di bagian ini:

- Prasyarat
- Mengonfigurasi notifikasi menggunakan konsol Amazon SES
- Mengonfigurasi notifikasi menggunakan API Amazon SES
- Pemecahan masalah notifikasi umpan balik

Prasyarat

Selesaikan langkah-langkah berikut sebelum Anda menyiapkan notifikasi Amazon SNS di Amazon SES:

1. Buat topik di Amazon SNS. Untuk informasi lebih lanjut, lihat Buat Topik di Panduan Developer Amazon Simple Notification Service.

   > **Important**
   >
   > Saat Anda membuat topik menggunakan Amazon SNS, untuk Jenis, pilih saja Standar. (SES tidak mendukung topik tipe FIFO.)

   Apakah Anda membuat topik SNS baru atau memilih yang sudah ada, Anda perlu memberikan akses ke SES untuk mempublikasikan pemberitahuan ke topik tersebut.

   Untuk memberikan izin Amazon SES untuk mempublikasikan pemberitahuan ke topik, pada layar Edit topik di konsol SNS, perluas kebijakan Access dan di editor JSON, tambahkan kebijakan izin berikut:

```
{
  "Version": "2012-10-17",
  "Id": "notification-policy",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ses.amazonaws.com"
      },
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:topic_region:111122223333:topic_name",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "111122223333",
          "AWS:SourceArn":
 "arn:aws:ses:topic_region:111122223333:identity/identity_name"
        }
      }
    }
  ]
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *topic_region dengan AWS Wilayah* tempat Anda membuat topik SNS.
- Ganti *111122223333* dengan ID akun AWS Anda.
- Ganti *topic_name dengan nama* topik SNS Anda.
- Ganti *identity_name* dengan identitas terverifikasi (alamat email atau domain) yang Anda berlangganan ke topik SNS.

2. Berlangganan setidaknya satu titik akhir ke topik. Misalnya, jika Anda ingin menerima notifikasi melalui pesan teks, berlangganlah titik akhir SMS (yaitu nomor ponsel) ke topik tersebut. Untuk menerima notifikasi melalui email, berlangganlah titik akhir email (alamat email) ke topik tersebut.

   Untuk informasi lebih lanjut, lihat Memulai di Panduan Developer Amazon Simple Notification Service.

3. (Opsional) Jika topik Amazon SNS Anda menggunakan AWS Key Management Service (AWS KMS) untuk enkripsi sisi server, Anda harus menambahkan izin ke kebijakan utama. AWS KMS Anda dapat menambahkan izin dengan melampirkan kebijakan berikut ke kebijakan AWS KMS utama:

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "AllowSESToUseKMSKey",
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

Mengonfigurasi notifikasi menggunakan konsol Amazon SES

Untuk mengonfigurasi notifikasi menggunakan konsol Amazon SES

1. Buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Dalam wadah Identitas, pilih identitas terverifikasi yang ingin Anda terima pemberitahuan umpan balik ketika pesan yang dikirim dari identitas ini menghasilkan bouncing, keluhan, atau pengiriman.

   > **Important**
   >
   > Pengaturan notifikasi domain terverifikasi berlaku untuk semua email yang dikirim dari alamat email di domain tersebut kecuali untuk alamat email yang juga terverifikasi.

4. Di layar detail identitas terverifikasi yang Anda pilih, pilih tab Pemberitahuan dan pilih Edit di wadah Pemberitahuan umpan balik.
5. Perluas kotak daftar topik SNS dari setiap jenis umpan balik yang ingin Anda terima notifikasi, dan pilih topik SNS yang Anda miliki, Tanpa topik SNS, atau topik SNS yang tidak Anda miliki.

   - Jika Anda memilih topik SNS yang tidak Anda miliki, bidang ARN topik SNS akan disajikan di mana Anda harus memasukkan topik SNS ARN yang dibagikan kepada Anda oleh pengirim delegasi Anda. (Hanya pengirim delegasi Anda yang akan mendapatkan notifikasi

ini karena mereka memiliki topik SNS. Untuk mempelajari lebih lanjut tentang pengiriman delegasi, lihatGambaran umum otorisasi pengiriman.)

> **Important**
>
> Topik Amazon SNS yang Anda gunakan untuk pemberitahuan pentalan, keluhan, dan pengiriman harus sama Wilayah AWS dengan yang Anda gunakan Amazon SES. Selain itu, Anda harus berlangganan satu atau beberapa titik akhir ke topik tersebut untuk menerima notifikasi. Misalnya, jika Anda ingin notifikasi dikirimkan ke suatu alamat email, Anda harus berlangganan titik akhir email ke topik tersebut. Untuk informasi lebih lanjut, lihat Memulai di Panduan Developer Amazon Simple Notification Service.

6. (Opsional) Jika Anda ingin pemberitahuan topik Anda menyertakan header dari email asli, centang Sertakan header email asli kotak tepat di bawah nama topik SNS dari setiap jenis umpan balik. Opsi ini hanya tersedia jika Anda telah menetapkan topik Amazon SNS untuk tipe notifikasi terkait. Untuk informasi tentang isi header email asli, lihat objek `mail` di Isi notifikasi.
7. Pilih Simpan perubahan. Perubahan yang Anda buat pada pengaturan notifikasi Anda mungkin memerlukan beberapa menit untuk diterapkan.
8. (Opsional) Jika Anda memilih notifikasi topik Amazon SNS untuk pantulan dan keluhan, Anda dapat menonaktifkan notifikasi email sepenuhnya sehingga Anda tidak menerima pemberitahuan ganda melalui email dan notifikasi SNS. Untuk menonaktifkan pemberitahuan email untuk bouncing dan keluhan, di bawah tab Notifikasi pada layar detail identitas terverifikasi, dalam wadah Penerusan Umpan Balik Email, pilih Edit, hapus centang pada kotak Diaktifkan, dan pilih Simpan perubahan.

Setelah mengonfigurasi pengaturan, Anda akan mulai menerima notifikasi pentalan, aduan, dan pengiriman ke topik Amazon SNS Anda. Notifikasi ini dalam format JavaScript Object Notation (JSON) dan mengikuti struktur yang dijelaskan dalam. Isi notifikasi

Anda akan dikenakan tarif Amazon SNS standar untuk notifikasi pentalan, aduan, dan pengiriman. Untuk informasi lebih lanjut, lihat halaman harga Amazon SNS.

> **Note**
>
> Jika upaya untuk memublikasikan ke topik Amazon SNS Anda gagal karena topik telah dihapus atau Anda Akun AWS tidak lagi memiliki izin untuk mempublikasikannya, Amazon SES menghapus konfigurasi untuk topik tersebut jika telah dikonfigurasi untuk pantulan atau

keluhan (bukan pengiriman - untuk pemberitahuan pengiriman, SES tidak akan menghapus setelan konfigurasi topik SNS). Selain itu, Amazon SES kembali mengaktifkan notifikasi email pentalan dan aduan untuk identitas, dan Anda menerima notfikasi dari perubahan tersebut melalui email. Jika beberapa identitas dikonfigurasi untuk menggunakan topik tersebut, konfigurasi topik untuk setiap identitas berubah ketika masing-masing identitas mengalami kegagalan untuk memublikasikan ke topik.

Mengonfigurasi notifikasi menggunakan API Amazon SES

Anda juga dapat mengonfigurasi notifikasi pentalan, aduan, dan pengiriman dengan menggunakan API Amazon SES. Gunakan operasi berikut untuk mengonfigurasi notifikasi:

- SetIdentityNotificationTopic
- SetIdentityFeedbackForwardingEnabled
- GetIdentityNotificationAttributes
- SetIdentityHeadersInNotificationsEnabled

Anda dapat menggunakan tindakan API ini untuk menulis aplikasi front-end yang disesuaikan untuk notifikasi. Untuk deskripsi lengkap mengenai tindakan API terkait notifikasi, lihat Referensi API Amazon Simple Email Service.

Pemecahan masalah notifikasi umpan balik

Tidak menerima notifikasi

Jika Anda tidak menerima notifikasi, pastikan bahwa Anda berlangganan titik akhir ke topik yang dikirimkan notifikasi. Ketika Anda berlangganan titik akhir email ke topik, Anda menerima email yang meminta Anda untuk mengonfirmasi langganan Anda. Anda harus mengonfirmasi langganan sebelum mulai menerima notifikasi email. Untuk informasi lebih lanjut, lihat Memulai di Panduan Developer Amazon Simple Notification Service.

Kesalahan **`InvalidParameterValue`** saat memilih topik

Jika Anda menerima kesalahan yang menyatakan bahwa terjadi kesalahan `InvalidParameterValue`, periksa topik Amazon SNS untuk melihat jika dienkripsi menggunakan AWS KMS. Jika ya, Anda harus mengubah kebijakan untuk AWS KMS kunci tersebut. Lihat Prasyarat untuk kebijakan sampel.

Isi notifikasi Amazon SNS untuk Amazon SES

Notifikasi pentalan, aduan, dan pengiriman diterbitkan ke topik Amazon Simple Notification Service (Amazon SNS) dalam format JavaScript Object Notation (JSON). Tingkat atas objek JSON berisi string `notificationType`, objek `mail`, dan objek `bounce`, objek `complaint`, atau objek `delivery`.

Lihat bagian berikut untuk deskripsi berbagai tipe objek:

- Objek JSON tingkat atas
- objek `mail`
- Objek `bounce`
- Objek `complaint`
- Objek `delivery`

Berikut ini adalah beberapa catatan penting tentang isi notifikasi Amazon SNS untuk Amazon SES:

- Untuk tipe notifikasi tertentu, Anda mungkin menerima satu notifikasi Amazon SNS untuk beberapa penerima, atau Anda mungkin menerima satu notifikasi Amazon SNS per penerima. Kode Anda harus dapat mengurai notifikasi Amazon SNS dan menangani kedua kasus tersebut; Amazon SES tidak membuat jaminan pengurutan atau batching untuk notifikasi yang dikirim melalui Amazon SNS. Namun, tipe notifikasi Amazon SNS yang berbeda (misalnya, pentalan dan aduan) tidak digabungkan menjadi satu notifikasi.
- Anda mungkin menerima beberapa tipe notifikasi Amazon SNS untuk satu penerima. Misalnya, server email penerima mungkin menerima email tersebut (memicu notifikasi pengiriman), tapi setelah memproses email, server email penerima mungkin menentukan bahwa email tersebut benar-benar menghasilkan pentalan (memicu notifikasi pentalan). Namun, notifikasi ini selalu terpisah karena notifikasi tersebut merupakan tipe notifikasi yang berbeda.
- Amazon SES berhak untuk menambahkan bidang tambahan ke notifikasi. Dengan demikian, aplikasi yang mengurai notifikasi ini harus cukup fleksibel untuk menangani bidang yang tidak diketahui.
- Amazon SES menimpa header pesan ketika mengirimkan email. Anda dapat mengambil header pesan asli dari bidang `headers` dan `commonHeaders` dari objek `mail`.

Objek JSON Tingkat Atas

Objek JSON tingkat atas dalam notifikasi Amazon SES berisi bidang berikut.

Nama bidang	Deskripsi
`notificationType`	String yang memiliki tipe notifikasi yang diwakili oleh objek JSON. Kemungkinan nilai adalah: `Bounce`, `Complaint` , atau `Delivery`. Jika Anda menyiapkan penerbitan acara, bidang ini diberi nama`eventType` .
`mail`	Objek JSON yang berisi informasi tentang email asli yang dikaitkan dengan notifikasi. Untuk informasi lebih lanjut, lihat Objek surat.
`bounce`	Bidang ini muncul hanya jika `notificationType` adalah `Bounce` dan berisi objek JSON yang menyimpan informasi tentang pentalan. Untuk informasi lebih lanjut, lihat Objek pentalan.
`complaint`	Bidang ini muncul hanya jika `notificationType` adalah `Complaint` dan berisi objek JSON yang menyimpan informasi tentang aduan. Untuk informasi lebih lanjut, lihat Objek aduan.
`delivery`	Bidang ini muncul hanya jika `notificationType` adalah `Delivery` dan berisi objek JSON yang menyimpan informasi tentang pengiriman. Untuk informasi lebih lanjut, lihat Objek pengiriman.

Objek surat

Setiap notifikasi pentalan, aduan, atau pengiriman berisi informasi tentang email asli di objek `mail`. Objek JSON yang berisi informasi tentang objek `mail` memiliki bidang berikut.

Nama bidang	Deskripsi
`timestamp`	Waktu saat pesan asli dikirim (dalam format ISO8601).
`messageId`	ID unik yang ditetapkan Amazon SES ke pesan. Amazon SES mengembalikan nilai ini kepada Anda saat Anda mengirim pesan. **Note** ID pesan ini ditetapkan oleh Amazon SES. Anda dapat menemukan ID pesan dari email asli di `headers` bidang `mail` objek.
`source`	Alamat email tempat pesan asli dikirim (alamat envelope MAIL FROM).
`sourceArn`	Amazon Resource Name (ARN) dari identitas yang digunakan untuk mengirim email. Dalam hal otorisasi pengiriman, `sourceArn` adalah ARN identitas yang pemilik identitas nya mengotorisasi penggunaan pengirim delegasi untuk mengirim email. Untuk informasi selengkapnya tentang otorisasi pengiriman, lihat Metode autentikasi email.
`sourceIp`	Alamat IP publik asal dari klien yang melakukan permintaan pengiriman email ke Amazon SES.
`sendingAccountId`	ID Akun AWS dari akun yang digunakan untuk mengirim email. Dalam hal otorisasi pengirima

Nama bidang	Deskripsi
	n, `sendingAccountId` adalah ID akun pengirim delegasi.
`callerIdentity`	Identitas IAM pengguna Amazon SES yang mengirimkan email.
`destination`	Daftar alamat email yang merupakan penerima email asli.
`headersTruncated`	Objek ini ada hanya jika Anda mengonfig urasi pengaturan notifikasi untuk menyertakan header dari email asli. Menunjukkan jika header dipotong dalam notifikasi. Amazon SES memotong header dalam notifikasi ketika header dari pesan asli berukuran 10 KB atau lebih. Kemungkinan nilai adalah `true` dan `false`.
`headers`	Objek ini ada hanya jika Anda mengonfig urasi pengaturan notifikasi untuk menyertakan header dari email asli. Daftar header asli email. Setiap header dalam daftar memiliki bidang `name` dan bidang `value`. **Note** Setiap ID pesan dalam objek `headers` berasal dari pesan asli yang Anda kirimkan ke Amazon SES. ID pesan yang kemudian ditentukan Amazon SES untuk pesan berada di bidang `messageId` dari objek `mail`.

Nama bidang	Deskripsi
`commonHeaders`	Objek ini ada hanya jika Anda mengonfig urasi pengaturan notifikasi untuk menyertakan header dari email asli. Mencakup informasi tentang header email umum dari email asli, termasuk bidang Dari, Kepada, dan Subjek. Dalam objek ini, setiap header adalah kunci. Bidang Dari dan Kepada diwakili oleh array yang dapat berisi beberapa nilai. **Note** Untuk peristiwa, ID pesan apa pun di `commonHeaders` bidang adalah ID pesan yang kemudian ditentuka n Amazon SES untuk pesan di `messageId` bidang objek mail. Notifikasi akan berisi ID pesan dari email asli.

Berikut ini adalah contoh dari objek `mail` yang mencakup header email asli. Ketika tipe notifikasi ini tidak dikonfigurasi untuk menyertakan header email asli, objek `mail` tidak menyertakan bidang `headersTruncated`, `headers`, dan `commonHeaders`.

```
{
   "timestamp":"2018-10-08T14:05:45 +0000",
   "messageId":"000001378603177f-7a5433e7-8edb-42ae-af10-f0181f34d6ee-000000",
   "source":"sender@example.com",
   "sourceArn": "arn:aws:ses:us-east-1:888888888888: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"
   }
}
```

Objek pentalan

Objek JSON yang berisi informasi tentang pentalan berisi bidang berikut.

Nama bidang	Deskripsi
`bounceType`	Tipe pentalan, seperti yang ditentukan oleh Amazon SES. Untuk informasi lebih lanjut, lihat Tipe pentalan.
`bounceSubType`	Subtipe pentalan, seperti yang ditentukan oleh Amazon SES. Untuk informasi lebih lanjut, lihat Tipe pentalan.
`bouncedRecipients`	Daftar yang berisi informasi tentang penerima email asli yang terpental. Untuk informasi lebih lanjut, lihat Penerima yang terpental.
`timestamp`	Tanggal dan waktu pentalan dikirim (dalam format ISO8601). Perhatikan bahwa ini adalah waktu saat notifikasi dikirim oleh ISP, dan bukan waktu ketika notifikasi tersebut diterima oleh Amazon SES.
`feedbackId`	ID unik untuk pentalan.

Jika Amazon SES dapat menghubungi Message Transfer Authority (MTA) jarak jauh, bidang berikut ini juga ada.

Nama kolom	Deskripsi
`remoteMtaIp`	Alamat IP dari MTA tempat Amazon SES berusaha untuk mengirimkan email.

Jika notifikasi status pengiriman (DSN) terlampir pada pentalan, bidang berikut ini juga ada.

Nama kolom	Deskripsi
`reportingMTA`	Nilai bidang `Reporting-MTA` dari DSN. Ini adalah nilai MTA yang berusaha untuk melakukan pengiriman, relai, atau operasi gateway yang dijelaskan di DSN.

Berikut ini adalah contoh dari objek `bounce`.

```
{
   "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"
}
```

Penerima yang terpental

Notifikasi pentalan mungkin berkaitan dengan satu penerima atau beberapa penerima. Bidang `bouncedRecipients` memiliki daftar objek—satu per penerima yang berkaitan dengan notifikasi pentalan—dan selalu berisi bidang berikut.

Nama kolom	Deskripsi
`emailAddress`	Alamat email penerima. Jika DSN tersedia, ini adalah nilai bidang `Final-Recipient` dari DSN.

Secara opsional, jika DSN dilampirkan ke pentalan, bidang berikut mungkin juga ada.

Nama kolom	Deskripsi
`action`	Nilai bidang `Action` dari DSN. Hal ini menunjukkan tindakan yang dilakukan oleh Pelaporan-MTA sebagai hasil dari usahanya untuk mengirimkan pesan kepada penerima ini.
`status`	Nilai bidang `Status` dari DSN. Ini adalah kode status bebas-transportasi per penerima yang menunjukkan status penyampaian pesan.
`diagnosticCode`	Kode status yang dikeluarkan oleh MTA pelaporan. Ini adalah nilai bidang `Diagnostic-Code` dari DSN. Bidang ini mungkin tidak ada di DSN (sehingga juga tidak ada di JSON).

Berikut ini adalah contoh objek yang mungkin ada di daftar `bouncedRecipients`.

```
{
    "emailAddress": "recipient@example.com",
    "action": "failed",
    "status": "5.0.0",
    "diagnosticCode": "X-Postfix; unknown user"
}
```

Tipe pentalan

Objek pentalan berisi tipe pentalan `Undetermined`, `Permanent`, atau `Transient`. Tipe pentalan `Permanent` dan `Transient` juga dapat berisi salah satu dari beberapa subtipe pentalan.

Ketika Anda menerima notifikasi pentalan dengan tipe pentalan `Transient`, Anda mungkin dapat mengirim email ke penerima tersebut di masa mendatang jika masalah yang menyebabkan pesan terpental teratasi.

Ketika Anda menerima notifikasi pentalan dengan tipe pentalan `Permanent`, Anda tidak dapat mengirim email ke penerima tersebut di masa mendatang. Untuk alasan ini, Anda harus segera menghapus penerima alamat yang menghasilkan pentalan tersebut dari milis Anda.

> **Note**
>
> Ketika pentalan lunak (pentalan yang terkait dengan masalah sementara, seperti kotak masuk penerima yang penuh) terjadi, Amazon SES mencoba untuk mengirim ulang email untuk jangka waktu tertentu. Pada akhir jangka waktu tersebut, jika Amazon SES masih tidak dapat mengirimkan email, Amazon SES akan berhenti mencobanya.
> Amazon SES memberikan notifikasi untuk pentalan keras, dan untuk pentalan lunak yang Amazon SES berhenti mencoba untuk mengirimkan. Jika Anda ingin menerima notifikasi setiap kali terjadi pentalan lunak, aktifkan publikasi peristiwa dan konfigurasikan untuk mengirim notifikasi ketika peristiwa penundaan pengiriman terjadi.

bounceType	bounceSubType	Deskripsi
`Undetermined`	`Undetermined`	Penyedia email penerima mengirim pesan pentalan. Pesan pentalan tidak berisi informasi yang cukup bagi Amazon SES untuk menentukan alasan pentalan. Email pentalan, yang dikirim ke alamat di header Jalur Kembali email yang mengakibatkan pentalan, mungkin berisi informasi tambahan tentang masalah yang menyebabkan email terpental.
`Permanent`	`General`	Penyedia email penerima mengirim pesan pentalan. **Important** Ketika Anda menerima tipe notifikasi pentalan ini, Anda harus segera

bounceType	bounceSubType	Deskripsi
		menghapus alamat email penerima tersebut dari milis Anda. Mengirim pesan ke alamat yang menghasil kan pentalan keras dapat berdampak negatif pada reputasi Anda sebagai pengirim. Jika Anda terus mengirim email ke alamat yang menghasilkan pentalan keras, kami mungkin menjeda kemampuan Anda untuk mengirim email tambahan. Lihat the section called “Menggunakan daftar penekanan tingkat akun”.
`Permanent`	`NoEmail`	Itu tidak mungkin untuk mengambil alamat email penerima dari pesan bouncing.
`Permanent`	`Suppressed`	Alamat email penerima berada di dalam daftar penekanan Amazon SES karena memiliki riwayat terbaru memproduksi pentalan keras. Untuk mengganti daftar penindasan global, lihat. Menggunakan daftar SES penindasan tingkat akun Amazon
`Permanent`	`OnAccountSuppressionList`	Amazon SES telah menahan pengiriman ke alamat ini karena berada di daftar penahanan tingkat akun. Ini tidak dihitung terhadap metrik rasio pentalan Anda.

bounceType	bounceSubType	Deskripsi
`Transient`	`General`	Penyedia email penerima mengirim pesan pentalan umum. Anda mungkin dapat mengirim pesan ke penerima yang sama di masa mendatang jika masalah yang menyebabkan pesan terpental teratasi. **Note** Jika Anda mengirim email ke penerima yang memiliki aturan respons otomatis yang aktif (seperti pesan "di luar kantor"), Anda mungkin menerima tipe notifikasi ini. Meskipun respons tersebut memiliki tipe notifikasi `Bounce`, Amazon SES tidak menghitung respons otomatis saat menghitung tingkat pentalan untuk akun Anda.
`Transient`	`MailboxFull`	Penyedia email penerima mengirim pesan pentalan karena kotak masuk penerima penuh. Anda mungkin dapat mengirim ke penerima yang sama di masa mendatang ketika kotak pesan tidak lagi penuh.
`Transient`	`MessageTooLarge`	Penyedia email penerima mengirim pesan pentalan karena pesan yang Anda kirim terlalu besar. Anda mungkin dapat mengirim pesan ke penerima yang sama jika Anda mengurangi ukuran pesan.

bounceType	bounceSubType	Deskripsi
`Transient`	`ContentRejected`	Penyedia email penerima mengirim pesan pentalan karena pesan yang Anda kirim berisi konten yang tidak diizinkan oleh penyedia. Anda mungkin dapat mengirim pesan ke penerima yang sama jika mengubah konten pesan.
`Transient`	`AttachmentRejected`	Penyedia email penerima mengirim pesan pentalan karena pesan berisi lampiran yang tidak dapat diterima. Misalnya, beberapa penyedia email mungkin menolak pesan dengan lampiran tipe file tertentu, atau pesan dengan lampiran yang sangat besar. Anda mungkin dapat mengirim pesan ke penerima yang sama jika Anda menghapus atau mengubah konten lampiran.

Objek aduan

Objek JSON yang berisi informasi tentang aduan memiliki bidang berikut.

Nama kolom	Deskripsi
`complainedRecipients`	Daftar yang berisi informasi tentang penerima yang mungkin bertanggung jawab atas aduan tersebut. Untuk informasi lebih lanjut, lihat Penerima yang diadukan.
`timestamp`	Tanggal dan waktu saat ISP mengirimkan notifikasi aduan, dalam format ISO 8601. Tanggal dan waktu di bidang ini mungkin tidak sama dengan tanggal dan waktu ketika Amazon SES menerima notifikasi.
`feedbackId`	ID unik yang terkait dengan aduan tersebut.

Nama kolom	Deskripsi
`complaintSubType`	Nilai bidang `complaintSubType` dapat null atau `OnAccountSuppressionList`. Jika nilainya adalah `OnAccountSuppressionList`, Amazon SES menerima pesan tersebut, namun tidak mencoba mengirimkannya karena pesan tersebut berada dalam daftar penekanan tingkat akun.

Selanjutnya, jika laporan umpan balik dilampirkan ke aduan, bidang berikut mungkin ada.

Nama kolom	Deskripsi
`userAgent`	Nilai bidang `User-Agent` dari laporan umpan balik. Nilai ini menunjukkan nama dan versi sistem yang menghasilkan laporan.
`complaintFeedbackType`	Nilai bidang `Feedback-Type` dari laporan umpan balik yang diterima dari ISP. Ini berisi tipe umpan balik.
`arrivalDate`	Nilai bidang `Arrival-Date` atau `Received-Date` dari laporan umpan balik (dalam format ISO8601). Bidang ini mungkin tidak ada dalam laporan (sehingga juga tidak ada dalam JSON).

Berikut ini adalah contoh dari objek `complaint`.

```
{
   "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"
}
```

Penerima yang diadukan

Bidang `complainedRecipients` berisi daftar penerima yang mungkin telah mengajukan aduan. Anda harus menggunakan informasi ini untuk menentukan penerima yang mengajukan aduan, dan kemudian segera menghapus penerima tersebut dari milis Anda.

> **Important**
>
> Sebagian besar ISP menghapus alamat email penerima yang mengajukan aduan dari notifikasi aduan mereka. Untuk alasan ini, daftar ini berisi informasi tentang penerima yang mungkin telah mengirim aduan, berdasarkan penerima pesan asli dan ISP tempat kami menerima aduan tersebut. Amazon SES melakukan pencarian terhadap pesan asli untuk menentukan daftar penerima ini.

Objek JSON dalam daftar ini berisi bidang berikut.

Nama kolom	Deskripsi
`emailAddress`	Alamat email penerima.

Berikut ini adalah contoh objek penerima yang diadukan.

```
{ "emailAddress": "recipient1@example.com" }
```

> **Note**
>
> Karena perilaku ini, Anda dapat lebih yakin bahwa Anda mengetahui alamat email yang mengadu tentang pesan Anda jika Anda membatasi pengiriman ke satu pesan per penerima (daripada mengirim satu pesan dengan 30 alamat email yang berbeda di baris bcc).

Tipe aduan

Anda mungkin melihat tipe aduan berikut di bidang `complaintFeedbackType` seperti yang ditetapkan oleh ISP pelaporan, menurut situs web Internet Assigned Numbers Authority:

- `abuse`—Menunjukkan email yang tidak diminta atau jenis penyalahgunaan email lainnya.
- `auth-failure`—Laporan kegagalan autentikasi email.
- `fraud`—Menunjukkan beberapa jenis penipuan atau aktivitas pengelabuan.
- `not-spam`—Menunjukkan bahwa entitas yang menyediakan laporan tidak menganggap pesan tersebut sebagai spam. Tindakan ini dapat digunakan untuk memperbaiki pesan yang salah ditandai atau dikategorikan sebagai spam.
- `other`—Menunjukkan umpan balik lain yang tidak sesuai dengan tipe terdaftar lainnya.
- `virus`—Laporan bahwa virus ditemukan dalam pesan asal.

Objek pengiriman

Objek JSON yang berisi informasi tentang pengiriman selalu memiliki bidang berikut.

Nama kolom	Deskripsi
`timestamp`	Waktu Amazon SES mengirimkan email ke server email penerima (dalam format ISO8601).
`processingTimeMillis`	Waktu dalam milidetik antara ketika Amazon SES menerima permintaan dari pengirim hingga mengirimkan pesan ke server email penerima.
`recipients`	Daftar penerima yang dimaksudkan dari email yang diterapkan notifikasi pengiriman.
`smtpResponse`	Pesan respons SMTP dari ISP jarak jauh yang menerima email dari Amazon SES. Pesan ini bervariasi menurut email, menurut server email penerima, dan menurut ISP penerima.

Nama kolom	Deskripsi
`reportingMTA`	Nama host server email Amazon SES yang mengirim email.
`remoteMtaIp`	Alamat IP dari MTA tempat Amazon SES mengirimkan email.

Berikut ini adalah contoh dari objek `delivery`.

```
{
   "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"
}
```

Contoh notifikasi Amazon SNS untuk Amazon SES

Bagian berikut memberikan contoh tiga tipe notifikasi:

- Untuk contoh notifikasi pentalan, lihat Contoh notifikasi pentalan Amazon SNS.
- Untuk contoh notifikasi aduan, lihat Contoh notifikasi aduan Amazon SNS.
- Untuk contoh notifikasi pengiriman, lihat Contoh notifikasi pengiriman Amazon SNS.

Contoh notifikasi pentalan Amazon SNS

Bagian ini berisi contoh notifikasi pentalan dengan dan tanpa Delivery Status Notification (DSN) yang disediakan oleh penerima email yang mengirimkan umpan balik.

Notifikasi pentalan dengan DSN

Berikut ini adalah contoh dari notifikasi pentalan yang berisi DSN dan header email asli. Ketika notifikasi pentalan tidak dikonfigurasi untuk menyertakan header email asli, objek `mail` dalam notifikasi tidak menyertakan bidang `headersTruncated`, `headers`, dan `commonHeaders`.

```
   {
```

```
      "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-east-1:888888888888:identity/example.com",
         "sourceIp": "127.0.3.0",
         "sendingAccountId":"123456789012",
         "callerIdentity": "IAM_user_or_role_name",
         "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"
        }
      }
   }
```

Notifikasi pentalan tanpa DSN

Berikut ini adalah contoh dari notifikasi pentalan yang menyertakan header email asli tetapi tidak menyertakan DSN. Ketika notifikasi pentalan tidak dikonfigurasi untuk menyertakan header email asli, objek `mail` dalam notifikasi tidak menyertakan bidang `headersTruncated`, `headers`, dan `commonHeaders`.

```
  {
     "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-east-1:888888888888:identity/example.com",
         "sourceIp": "127.0.3.0",
         "sendingAccountId":"123456789012",
         "callerIdentity": "IAM_user_or_role_name",
         "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"
        }
     }
 }
```

Contoh notifikasi aduan Amazon SNS

Bagian ini berisi contoh notifikasi aduan dengan dan tanpa laporan umpan balik yang diberikan oleh penerima email yang mengirimkan umpan balik.

Notifikasi aduan dengan laporan umpan balik

Berikut ini adalah notifikasi aduan yang berisi laporan umpan balik dan header email asli. Ketika notifikasi aduan tidak dikonfigurasi untuk menyertakan header email asli, objek `mail` dalam notifikasi tidak menyertakan bidang `headersTruncated`, `headers`, dan `commonHeaders`.

```
  {
     "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-east-1:888888888888:identity/example.com",
        "sourceIp": "127.0.3.0",
        "sendingAccountId":"123456789012",
        "callerIdentity": "IAM_user_or_role_name",
        "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"
        }
     }
  }
```

Notifikasi aduan tanpa laporan umpan balik

Berikut ini adalah contoh notifikasi aduan yang menyertakan header email asli namun tidak menyertakan laporan umpan balik. Ketika notifikasi aduan tidak dikonfigurasi untuk menyertakan header email asli, objek `mail` dalam notifikasi tidak menyertakan bidang `headersTruncated`, `headers`, dan `commonHeaders`.

```
  {
     "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-east-1:888888888888:identity/example.com",
         "sourceIp": "127.0.3.0",
         "sendingAccountId":"123456789012",
         "callerIdentity": "IAM_user_or_role_name",
         "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"
         }
      }
  }
```

Contoh notifikasi pengiriman Amazon SNS

Berikut ini adalah contoh notifikasi pengiriman yang menyertakan header email asli. Ketika notifikasi pengiriman tidak dikonfigurasi untuk menyertakan header email asli, objek `mail` dalam notifikasi tidak menyertakan bidang `headersTruncated`, `headers`, dan `commonHeaders`.

```
  {
     "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-east-1:888888888888:identity/example.com",
        "sourceIp": "127.0.3.0",
        "sendingAccountId":"123456789012",
        "callerIdentity": "IAM_user_or_role_name",
        "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"
```

```
    }
  }
```

Menggunakan otorisasi identitas di Amazon SES

Kebijakan otorisasi identitas menentukan bagaimana identitas terverifikasi individu dapat menggunakan Amazon SES dengan menentukan tindakan SES API mana yang diizinkan atau ditolak untuk identitas dan dalam kondisi apa.

Anda dapat mempertahankan kendali atas identitas Anda melalui kebijakan otorisasi kebijakan Anda bahkan dapat otorisasi pengguna lain untuk menggunakan identitas yang Anda miliki (domain atau alamat email) dengan akun SES mereka sendiri.

Topik

- Anatomi kebijakan Amazon SES
- Membuat kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan di Amazon SES
- Contoh kebijakan identitas di Amazon SES
- Mengelola kebijakan otorisasi identitas Anda di Amazon SES

Anatomi kebijakan Amazon SES

Kebijakan mematuhi struktur tertentu, mengandung elemen, dan harus memenuhi persyaratan tertentu.

Struktur kebijakan

Setiap kebijakan otorisasi adalah dokumen JSON yang dilampirkan pada identitas. Setiap kebijakan mencakup bagian berikut:

- Informasi untuk seluruh kebijakan di bagian atas dokumen.
- Satu atau beberapa pernyataan individu, masing-masing menjelaskan satu set izin.

Contoh hibah kebijakan berikutAWSID akun123456789012izin yang ditentukan dalamAksibagian untuk domain yang diverifikasiexample.com.

```
{
  "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:GetEmailIdentity",
        "ses:UpdateEmailIdentityPolicy",
        "ses:ListRecommendations",
        "ses:CreateEmailIdentityPolicy",
        "ses:DeleteEmailIdentity"
      ]
    }
  ]
}
```

Anda dapat menemukan lebih banyak contoh kebijakan otorisasi diContoh kebijakan identitas.

Elemen kebijakan

Bagian ini menjelaskan elemen-elemen yang terkandung dalam kebijakan otorisasi identitas. Pertama kami menjelaskan elemen kebijakan secara keseluruhan, lalu kami menjelaskan elemen yang hanya berlaku untuk pernyataan tempat elemen tersebut disertakan. Kami selanjutnya mengadakan diskusi tentang cara menambahkan syarat untuk pernyataan Anda.

Untuk informasi spesifik tentang sintaksis elemen, lihat Tata Bahasa Kebijakan IAM di Panduan Pengguna IAM.

Informasi kebijakan keseluruhan

Ada dua elemen kebijakan secara keseluruhan: `Id` dan `Version`. Tabel berikut memberikan informasi tentang elemen-elemen ini.

Nama	Deskripsi	Wajib	Nilai valid
`Id`	Secara unik mengidentifikasi kebijakan.	Tidak	String apa pun
`Version`	Menentukan versi bahasa akses kebijakan.	Tidak	String apa pun. Sebagai praktik terbaik, sebaiknya sertakan bidang ini dengan nilai "2012-10-17".

Pernyataan khusus untuk kebijakan

Kebijakan otorisasi identitas memerlukan setidaknya satu pernyataan. Setiap pernyataan dapat mencakup elemen yang dijelaskan di tabel berikut.

Nama	Deskripsi	Wajib	Nilai valid
`Sid`	Secara unik mengidentifikasi pernyataan.	Tidak	String apa pun.
`Effect`	Menentukan hasil yang Anda inginkan dikembalikan oleh pernyataan kebijakan pada waktu evaluasi.	Ya	"Izinkan" atau "Tolak".
`Resource`	Menentukan identitas dengan kebijakan yang berlaku. (Untukmengirim otorisasi, ini adalah alamat email atau domain yang pemilik identitas memberi wewenang kepada	Ya	Nama Sumber Daya Amazon (ARN) dari identitas.

Nama	Deskripsi	Wajib	Nilai valid
	pengirim delegasi untuk digunakan.)		
`Principal`	MenentukanAkun AWS, pengguna, atauAWSlayanan yang menerima izin dalam pernyataan.	Ya	ValidAkun AWSID, pengguna ARN, atauAWSlayanan.Akun AWS ID dan ARN pengguna ditentukan menggunakan`"AWS"`(Sebagai contoh,`"AWS": ["123456789012"]` atau`"AWS": ["arn:aws:iam::123456789012:root"]`).AWSnama layanan ditentukan menggunakan`"Service"` (Sebagai contoh,`"Service": ["cognito-idp.amazonaws.com"]`). Untuk contoh format ARN pengguna, lihatReferensi Umum AWS.

Nama	Deskripsi	Wajib	Nilai valid
`Action`	Menentukan tindakan yang berlaku untuk pernyataan tersebut.	Ya	“ses:BatchGetMetricData“, “Ses:CancelExportJob“, “Ses:CreateDeliverabilityTestReport“, “Ses:CreateEmailIdentityPolicy“, “Ses:CreateExportJob“, “Ses:DeleteEmailIdentity“, “Ses:DeleteEmailIdentityPolicy“, “Ses:GetDomainStatisticsReport“, “Ses:GetEmailIdentity“, "Ses:GetEmailIdentityPolicies“, “Ses:GetExportJob“, “Ses:ListExportJobs“, “Ses:ListRecommendations“, “Ses:PutEmailIdentityConfigurationSetAttributes“, “Ses:PutEmailIdentityDkimAttributes“, “Ses:PutEmailIdentityDkimSigningAttributes“, “Ses:PutEmailIdentityFeedbackAttributes“, “Ses:PutEmailIdentityMailFromAttributes“, “Ses:TagResource“,

Nama	Deskripsi	Wajib	Nilai valid
			“Ses:UntagResource“, “Ses:UpdateEmailIdentityPolicy“ (Mengirim otorisasi tindakan: “ses:SendEmail“, “Ses:SendRawEmail“, “Ses:SendTemplatedEmail“, “Ses:SendBulkTemplatedEmail“) Anda dapat menentukan satu atau beberapa operasi ini.
`Condition`	Menentukan pembatasan atau detail tentang izin.	Tidak	Lihat informasi tentang syarat pada tabel berikut ini.

Kondisi

Syarat adalah pembatasan tentang izin di pernyataan. Bagian dari pernyataan yang menentukan syarat dapat menjadi yang paling detail dari semua bagian. Kunci adalah karakteristik spesifik yang menjadi dasar pembatasan akses, seperti tanggal dan waktu permintaan.

Anda menggunakan syarat maupun kunci secara bersama-sama untuk mengekspresikan pembatasan. Misalnya, jika Anda ingin membatasi pengirim delegasi membuat permintaan ke Amazon SES atas nama Anda setelah 30 Juli 2019, Anda menggunakan syarat yang disebut `DateLessThan`. Anda menggunakan kunci yang disebut `aws:CurrentTime` dan mengaturnya ke nilai `2019-07-30T00:00:00Z`.

SES hanya mengimplementasikan yang berikutAWSkunci kebijakan -wide:

- `aws:CurrentTime`
- `aws:EpochTime`

- `aws:SecureTransport`
- `aws:SourceIp`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:UserAgent`
- `aws:VpcSourceIp`

Untuk informasi selengkapnya tentang kunci ini, lihat Panduan Pengguna IAM.

Persyaratan kebijakan

Kebijakan harus memenuhi semua persyaratan berikut:

- Setiap kebijakan harus menyertakan setidaknya satu pernyataan.
- Setiap kebijakan harus menyertakan setidaknya satu prinsipiel yang valid.
- Setiap kebijakan harus menentukan satu sumber daya, dan sumber daya tersebut harus ARN dari identitas yang dilampirkan pada kebijakan.
- Pemilik identitas dapat mengaitkan hingga 20 kebijakan dengan setiap identitas unik.
- Kebijakan tidak boleh melebihi 4 kilobyte (KB).
- Nama kebijakan tidak boleh melebihi 64 karakter. Selain itu, kebijakan hanya dapat menyertakan karakter alfanumerik, tanda hubung, dan garis bawah.

Membuat kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan di Amazon SES

Kebijakan otorisasi identitas terdiri dari pernyataan yang menentukan tindakan API apa yang diizinkan atau ditolak untuk identitas dan dalam kondisi apa.

Untuk mengotorisasi identitas alamat email atau identitas alamat email yang Anda miliki, Anda membuat kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan atau identitas yang Anda miliki, lalu melampirkan kebijakan ke identitas. Identitas dapat memiliki nol, satu, atau banyak kebijakan. Namun, satu kebijakan hanya dapat dikaitkan dengan satu identitas.

Untuk daftar tindakan API yang dapat digunakan dalam kebijakan otorisasi identitas, lihat baris Tindakan dalamthe section called “Pernyataan khusus untuk kebijakan” tabel.

Anda dapat membuat kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan dengan

- Dengan menggunakan generator kebijakan — Anda dapat membuat kebijakan sederhana dengan menggunakan generator kebijakan — Anda dapat membuat kebijakan sederhana dengan menggunakan generator kebijakan — Anda dapat membuat kebijakan sederhana dengan menggunakan generator kebijakan — Anda dapat membuat kebijakan sederhana dengan menggunakan generator kebijakan — Anda dapat Selain mengizinkan atau menolak izin pada tindakan SES API, Anda dapat membatasi tindakan dengan kondisi. Anda juga dapat menggunakan generator kebijakan untuk membuat struktur dasar struktur dasar kebijakan kebijakan kebijakan kebijakan sederhana kebijakan kebijakan kebijakan untuk membuat struktur dasar kebijakan kebijakan kebijakan kebijakan kebijakan kebijakan kebijakan kebijakan kebijakan kebijakan untuk membuat struktur dasar kebijakan dasar kebijakan
- Dengan membuat kebijakan kustom — Jika Anda ingin menyertakan syarat lanjutan atau menggunakanAWS layanan sebagai prinsipiel, Anda dapat membuat kebijakan kustom dan melampirkannya ke identitas dengan menggunakan konsol SES atau API SES.

Topik

- Menggunakan generator kebijakan
- Membuat kebijakan kustom

Menggunakan generator kebijakan

Anda dapat menggunakan generator kebijakan untuk membuat kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi kebijakan untuk membuat kebijakan otorisasi kebijakan otorisasi kebijakan untuk membuat kebijakan otorisasi kebijakan otorisasi kebijakan otorisasi sederhana dengan

Untuk membuat kebijakan dengan menggunakan generator kebijakan

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Di kontainer Identitas di Identitas terverifikasi layar, pilih identitas terverifikasi yang ingin Anda buat kebijakan otorisasi.

4. Di layar detail identitas terverifikasi yang Anda pilih di langkah sebelumnya, pilih tab Otorisasi.
5. Di panel Kebijakan otorisasi, pilih Buat kebijakan, lalu pilih Gunakan pembuat kebijakan dari menu tarik-turun.
6. Di panel Create statement, pilih Allow in the Effect field. (Jika Anda ingin membuat kebijakan untuk membatasi identitas ini, pilih Deny sebagai gantinya.)
7. Di bidang Prinsipal, masukkan Akun AWSID, ARN pengguna IAM, atauAWS layanan untuk menerima izin yang ingin Anda otorisasi untuk identitas ini, lalu pilih Tambah. (Jika Anda ingin mengotorisasi lebih dari satu, ulangi langkah ini untuk masing-masing.)
8. Di bidang Tindakan, pilih kotak centang untuk setiap tindakan yang ingin Anda otorisasi untuk prinsipal Anda.
9. (Opsional) Perluas Tentukan kondisi jika Anda ingin menambahkan pernyataan kualifikasi ke izin.

    a. Pilih operator dari dropdown Operator.
    b. Pilih jenis dari dropdown Key.
    c. Masing-masing dengan jenis kunci yang Anda pilih, masukkan nilainya di bidang Nilai. (Jika Anda ingin menambahkan lebih banyak kondisi, pilih Tambahkan kondisi baru dan ulangi langkah ini untuk setiap kondisi tambahan.)
10. Pilih Simpan pernyataan.
11. (Opsional) Perluas Buat pernyataan lain jika Anda ingin menambahkan lebih banyak pernyataan ke kebijakan Anda dan ulangi langkah 6 - 10.
12. Pilih Berikutnya dan di layar Sesuaikan kebijakan, wadah Edit detail kebijakan memiliki bidang tempat Anda dapat mengubah atau menyesuaikan Nama kebijakan dan dokumen Kebijakan itu sendiri.
13. Pilih Berikutnya dan pada layar Tinjau dan terapkan, wadah Ikhtisar akan menampilkan identitas terverifikasi yang Anda otorisasi serta nama kebijakan ini. Di panel Dokumen kebijakan akan menjadi kebijakan aktual yang baru saja Anda tulis bersama dengan kondisi apa pun yang Anda tambahkan - tinjau kebijakan dan jika terlihat benar, pilih Terapkan kebijakan. (Jika Anda perlu mengubah atau memperbaiki sesuatu, pilih Sebelumnya dan bekerja di wadah Edit detail kebijakan.)

Membuat kebijakan kustom

Jika Anda ingin membuat kebijakan kustom dan melampirkannya ke identitas, Anda memiliki opsi berikut:

- Menggunakan API Amazon SES – Membuat kebijakan di editor teks lalu melampirkan kebijakan ke identitas dengan menggunakan API `PutIdentityPolicy` yang dijelaskan di Referensi API Amazon Simple Email Service.
- Menggunakan konsol Amazon SES – Membuat kebijakan di editor teks dan melampirkannya ke identitas dengan menempelkannya ke editor kebijakan kustom di konsol Amazon SES. Prosedur berikut menjelaskan metode ini.

Untuk membuat kebijakan kustom menggunakan editor kebijakan

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Di kontainer Identitas di Identitas terverifikasi layar, pilih identitas terverifikasi yang ingin Anda buat kebijakan otorisasi.
4. Di layar detail identitas terverifikasi yang Anda pilih di langkah sebelumnya, pilih tab Otorisasi.
5. Di panel Kebijakan otorisasi, pilih Buat kebijakan, lalu pilih Buat kebijakan kustom dari menu tarik-turun.
6. Di panel dokumen kebijakan, ketik atau tempel teks kebijakan kebijakan kebijakan kebijakan kebijakan Anda dalam format JSON. Anda juga dapat menggunakan generator kebijakan untuk membuat struktur dasar kebijakan kebijakan kebijakan kebijakan kebijakan kebijakan sederhana kebijakan kebijakan untuk membuat struktur dasar kebijakan kebijakan kebijakan untuk membuat struktur dasar kebijakan kebijakan kebijakan kebijakan kebijakan kebijakan sederhana kebijakan dasar kebijakan kebijakan sederhana kebijakan kebijakan kebijakan sederhana dan
7. Pilih Terapkan Kebijakan. (Jika Anda perlu mengubah kebijakan kustom Anda, cukup pilih kotak centang di bawah tab Otorisasi, pilih Edit, dan buat perubahan Anda di panel dokumen Kebijakan diikuti oleh Simpan perubahan).

Contoh kebijakan identitas di Amazon SES

Otorisasi identitas memungkinkan Anda untuk menentukan detail saat Anda mengizinkan atau menolak Tindakan API untuk identitas.

Contoh berikut menunjukkan cara menulis kebijakan untuk mengendalikan aspek Tindakan API yang berbeda:

- Menentukan Prpiel
- Membatasi Tindakan
- Menggunakan beberapa pernyataan

Menentukan Prpiel

Utama, yang merupakan entitas yang Anda berikan izin, bisa berupa akunAkun AWS, penggunaAWS Identity and Access Management (IAM), atauAWS layanan yang dimiliki akun yang sama.

Contoh berikut menunjukkan kebijakan sederhana yang memungkinkanAWS ID 1234589012 untuk mengendalikan identitas terverifikasi example.com yang juga dimiliki oleh 1234512 identitas terverifikasi example.com yang juga dimiliki olehAkun AWS 1234512.

```
{
  "Id":"SampleAuthorizationPolicy",
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"AuthorizeMarketer",
      "Effect":"Allow",
      "Resource":"arn:aws:ses:us-east-1:123456789012:identity/example.com",
      "Principal":{
        "AWS":[
          "123456789012"
        ]
      },
      "Action":[
        "ses:DeleteEmailIdentity",
        "ses:PutEmailIdentityDkimSigningAttributes"
      ]
    }
  ]
}
```

Contoh kebijakan berikut memberikan izin ke dua pengguna untuk mengendalikan identitas terverifikasi example.com. Pengguna ditentukan oleh Amazon Resource Name (ARN) mereka.

```
{
  "Id":"ExampleAuthorizationPolicy",
  "Version":"2012-10-17",
  "Statement":[
```

```
    {
      "Sid":"AuthorizeIAMUser",
      "Effect":"Allow",
      "Resource":"arn:aws:ses:us-east-1:123456789012:identity/example.com",
      "Principal":{
        "AWS":[
          "arn:aws:iam::123456789012:user/John",
          "arn:aws:iam::123456789012:user/Jane"
        ]
      },
      "Action":[
        "ses:DeleteEmailIdentity",
        "ses:PutEmailIdentityDkimSigningAttributes"
      ]
    }
  ]
}
```

Membatasi Tindakan

Ada beberapa tindakan yang dapat ditentukan dalam kebijakan otorisasi identitas tergantung pada tingkat kontrol yang ingin Anda otorisasi:

```
"BatchGetMetricData",
"ListRecommendations",
"CreateDeliverabilityTestReport",
"CreateEmailIdentityPolicy",
"DeleteEmailIdentity",
"DeleteEmailIdentityPolicy",
"GetDomainStatisticsReport",
"GetEmailIdentity",
"GetEmailIdentityPolicies",
"PutEmailIdentityConfigurationSetAttributes",
"PutEmailIdentityDkimAttributes",
"PutEmailIdentityDkimSigningAttributes",
"PutEmailIdentityFeedbackAttributes",
"PutEmailIdentityMailFromAttributes",
"TagResource",
"UntagResource",
"UpdateEmailIdentityPolicy"
```

Kebijakan otorisasi identitas otorisasi identitas juga memungkinkan Anda untuk membatasi prinsipal ke salah satu tindakan tersebut.

```
{
  "Id":"ExamplePolicy",
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"ControlAction",
      "Effect":"Allow",
      "Resource":"arn:aws:ses:us-east-1:123456789012:identity/example.com",
      "Principal":{
        "AWS":[
          "123456789012"
        ]
      },
      "Action":[
        "ses:PutEmailIdentityMailFromAttributes
      ]
    }
  ]
}
```

Menggunakan beberapa pernyataan

Kebijakan otorisasi identitas Anda dapat mencakup beberapa pernyataan. Contoh kebijakan berikut ini memiliki dua pernyataan. Pernyataan pertama menolak dua pengguna untuk mengaksesgetemailidentity dari sender@example.com dalam akun yang sama123456789012. Pernyataan kedua menyangkalUpdateEmailIdentityPolicy kepala sekolah, Jack, dalam akun yang sama123456789012.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"DenyGet",
      "Effect":"Deny",
      "Resource":"arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
      "Principal":{
        "AWS":[
          "arn:aws:iam::123456789012:user/John",
          "arn:aws:iam::123456789012:user/Jane"
        ]
      },
      "Action":[
```

```
        "ses:GetEmailIdentity"
      ]
    },
    {
      "Sid":"DenyUpdate",
      "Effect":"Deny",
      "Resource":"arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
      "Principal":{
        "AWS":"arn:aws:iam::123456789012:user/Jack"
      },
      "Action":[
        "ses:UpdateEmailIdentityPolicy"
      ]
    }
  ]
}
```

Mengelola kebijakan otorisasi identitas Anda di Amazon SES

Selain membuat dan melampirkan kebijakan ke identitas, Anda dapat mengedit, menghapus, mencantumkan, dan mengambil kebijakan identitas seperti yang dijelaskan di bagian berikut.

Mengelola kebijakan menggunakan konsol

Mengelola kebijakan Amazon SES memerlukan melihat, mengedit, atau menghapus kebijakan yang terlampir pada identitas dengan menggunakan konsol

Untuk mengelola kebijakan Amazon SES konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Identitas Terverifikasi.
3. Di daftar identitas, pilih identitas yang ingin Anda kelola.
4. Pada halaman detail identitas, navigasikan ke tab Otorisasi. Di sini Anda akan menemukan daftar semua kebijakan yang dilampirkan pada identitas ini.
5. Pilih kebijakan yang ingin Anda kelola dengan memilih kotak centang.
6. Tergantung pada tugas manajemen yang diinginkan, pilih tombol masing-masing sebagai berikut:

a. Untuk melihat kebijakan, pilih Lihat kebijakan. Jika Anda memerlukan salinannya, pilih tombol Salin dan itu akan disalin ke clipboard Anda.

b. Untuk mengedit kebijakan, pilih Edit. Di panel Dokumen kebijakan, edit kebijakan, lalu pilih Simpan perubahan.

> **Note**
>
> Untuk mencabut izin, Anda dapat mengedit kebijakan atau menghapusnya.

c. Untuk menghapus kebijakan, pilih Hapus.

> **Important**
>
> Menghapus kebijakan bersifat permanen. Sebaiknya buat cadangan kebijakan dengan menyalin dan menempelkannya ke file teks sebelum Anda menghapusnya.

Mengelola kebijakan menggunakan API

Mengelola kebijakan Amazon SES memerlukan melihat, mengedit, atau menghapus kebijakan yang terlampir pada identitas dengan menggunakan API

Untuk mencantumkan dan melihat kebijakan menggunakan API Amazon SES

- Anda dapat mencantumkan kebijakan yang dilampirkan ke identitas dengan menggunakan operasiListIdentityPolicies API. Anda juga dapat mengambil kebijakan sendiri dengan menggunakan operasiGetIdentityPolicies API.

Untuk mengedit kebijakan Amazon SES API

- Anda dapat mengedit kebijakan yang dilampirkan ke identitas dengan menggunakan operasiPutIdentityPolicy API.

Untuk menghapus kebijakan Amazon SES API

- Anda dapat menghapus kebijakan yang dilampirkan ke identitas dengan menggunakan operasiDeleteIdentityPolicy API.

Menggunakan otorisasi pengiriman dengan Amazon SES

Anda dapat mengonfigurasi Amazon SES untuk mengotorisasi pengguna lain untuk mengirim email dari identitas yang Anda miliki (domain atau alamat email) menggunakan akun Amazon SES mereka sendiri. Anda dapat mempertahankan kendali atas identitas Anda dengan fitur otorisasi pengiriman sehingga Anda dapat mengubah atau mencabut izin kapan saja. Misalnya, jika Anda adalah pemilik bisnis, Anda dapat menggunakan otorisasi pengiriman untuk mengaktifkan pihak ke tiga (seperti perusahaan pemasaran email) untuk mengirim email dari domain yang Anda miliki.

Bab ini mencakup spesifikasi otorisasi pengiriman yang menggantikan fitur notifikasi lintas akun lama. Anda harus terlebih dahulu memahami dasar-dasar otorisasi berbasis identitas menggunakan kebijakan otorisasi seperti yang dijelaskan di Menggunakan otorisasi identitas di Amazon SES mana mencakup topik-topik penting seperti anatomi kebijakan otorisasi dan bagaimana mengelola kebijakan Anda.

Dukungan lama pemberitahuan lintas-akun

Pemberitahuan umpan balik untuk pantulan, keluhan, dan pengiriman yang terkait dengan email yang dikirim dari pengirim delegasi yang telah diberi wewenang oleh pemilik identitas untuk dikirim dari salah satu identitas terverifikasi, secara tradisional telah dikonfigurasi menggunakan pemberitahuan lintas akun di mana pengirim delegasi akan mengaitkan topik dengan identitas yang tidak mereka miliki (itulah akun silang). Namun, notifikasi lintas akun telah diganti dengan menggunakan set konfigurasi dan identitas terverifikasi sehubungan dengan pengiriman delegasi di mana pengirim delegasi telah diberi wewenang oleh pemilik identitas untuk menggunakan salah satu identitas terverifikasi mereka untuk mengirim email. Metode baru ini memungkinkan fleksibilitas untuk mengonfigurasi bouncing, keluhan, pengiriman, dan pemberitahuan peristiwa lainnya dengan dua konstruksi berikut tergantung apakah Anda pengirim delegasi atau pemilik identitas terverifikasi:

- Set konfigurasi - Pengirim delegasi dapat mengatur penerbitan acara dalam set konfigurasinya sendiri yang dapat dia tentukan saat mengirim email dari identitas terverifikasi yang tidak dimilikinya, tetapi telah diberi wewenang untuk dikirim oleh pemilik identitas melalui kebijakan otorisasi. Penerbitan acara memungkinkan bouncing, keluhan, pengiriman, dan pemberitahuan acara lainnya dipublikasikan ke Amazon CloudWatch, Amazon Data Firehose, Amazon Pinpoint, dan Amazon SNS. Lihat Buat tujuan acara.
- Identitas terverifikasi - Selain memiliki pemilik identitas yang memberi wewenang kepada pengirim delegasi untuk menggunakan salah satu identitas terverifikasi untuk mengirim email, ia juga dapat, atas permintaan pengirim delegasi, mengonfigurasi pemberitahuan umpan balik pada identitas bersama untuk menggunakan topik SNS yang dimiliki oleh pengirim delegasi. Hanya pengirim

delegasi yang akan mendapatkan notifikasi ini karena mereka memiliki topik SNS. Lihat Langkah 14 untuk cara mengonfigurasi “topik SNS yang tidak Anda miliki” dalam prosedur kebijakan otorisasi.

> **Note**
>
> Untuk kompatibilitas, notifikasi lintas akun didukung untuk pemberitahuan lintas akun lama yang saat ini digunakan di akun Anda. Dukungan ini terbatas untuk dapat memodifikasi dan menggunakan akun lintas apa pun saat ini yang Anda buat di konsol klasik Amazon SES; namun, Anda tidak dapat lagi membuat notifikasi lintas akun baru. Untuk membuat yang baru di konsol baru Amazon SES, gunakan metode pengiriman delegasi baru baik dengan set konfigurasi menggunakan penerbitan acara, atau dengan identitas terverifikasi yang dikonfigurasi dengan topik SNS Anda sendiri.

Topik

- Gambaran umum otorisasi pengiriman Amazon SES
- Tugas pemilik identitas untuk otorisasi pengiriman Amazon SES
- Tugas pengirim delegasi untuk otorisasi pengiriman Amazon SES

Gambaran umum otorisasi pengiriman Amazon SES

Topik ini memberikan gambaran umum tentang proses otorisasi pengiriman dan kemudian menjelaskan cara fitur pengiriman email Amazon SES, seperti mengirim kuota dan notifikasi, bekerja dengan otorisasi pengiriman.

Bagian ini menggunakan syarat berikut:

- Identitas – Alamat email atau domain yang digunakan pengguna Amazon SES untuk mengirim email.
- Pemilik identitas – Pengguna Amazon SES yang telah memverifikasi kepemilikan alamat email atau domain dengan menggunakan prosedur yang dijelaskan di Identitas terverifikasi.
- Pengirim delegasi — AWS Akun, pengguna AWS Identity and Access Management (IAM), atau AWS layanan yang telah diotorisasi melalui kebijakan otorisasi untuk mengirim email atas nama pemilik identitas.

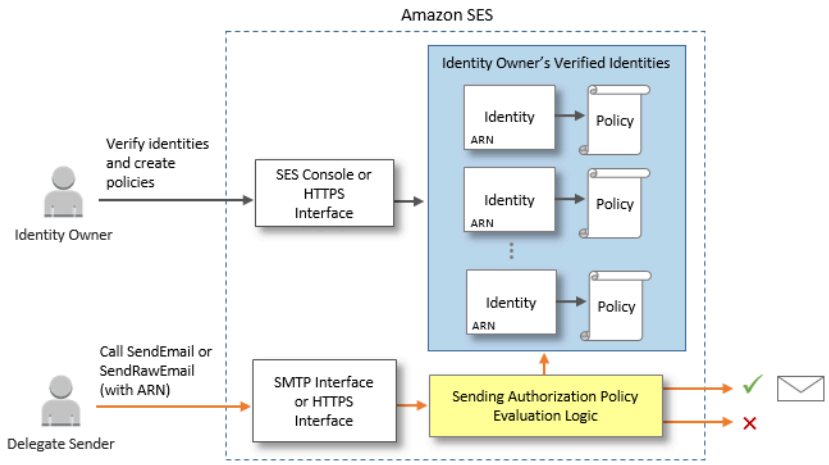
- Kebijakan otorisasi pengiriman – Dokumen yang Anda lampirkan pada identitas untuk menentukan siapa yang dapat mengirimkan identitas tersebut beserta syaratnya.
- Amazon Resource Name (ARN) – Cara terstandardisasi untuk secara unik mengidentifikasi sumber daya AWS di semua layanan AWS. Untuk mengirim otorisasi, sumber daya adalah identitas yang pemilik identitas telah mengizinkan pengirim delegasi untuk digunakan. Contoh ARN adalah arn:aws:ses:us-east - 1:123456789012: identity/example.com.

Proses otorisasi pengiriman

Otorisasi pengiriman didasarkan pada kebijakan otorisasi pengiriman. Jika Anda ingin mengaktifkan pengirim delegasi untuk mengirim atas nama Anda, Anda membuat kebijakan otorisasi pengiriman dan mengaitkan kebijakan ke identitas Anda dengan menggunakan konsol Amazon SES atau API Amazon SES. Ketika pengirim delegasi mencoba untuk mengirim email melalui Amazon SES atas nama Anda, pengirim delegasi meneruskan ARN identitas Anda di permintaan atau di header email.

Ketika Amazon SES menerima permintaan untuk mengirim email, Amazon SES memeriksa kebijakan identitas Anda (jika ada) untuk menentukan jika Anda telah mengotorisasi pengirim delegasi untuk mengirim atas nama identitas. Jika pengirim delegasi diotorisasi, Amazon SES menerima email; jika tidak, Amazon SES mengembalikan pesan kesalahan.

Diagram berikut menunjukkan hubungan tingkat tinggi antar konsep otorisasi pengiriman:



Proses otorisasi pengiriman terdiri dari langkah-langkah berikut:

1. Pemilik identitas memilih identitas terverifikasi untuk pengirim delegasi untuk digunakan. (Jika Anda belum memverifikasi identitas, lihatIdentitas terverifikasi.)

> **Note**
>
> Identitas terverifikasi yang Anda pilih untuk pengirim delegasi Anda tidak dapat memiliki konfigurasi default yang ditetapkan untuk itu.

2. Pengirim delegasi memungkinkan pemilik identitas mengetahui ID AWS akun atau ARN pengguna IAM mana yang ingin mereka gunakan untuk mengirim.
3. Jika pemilik identitas setuju untuk mengizinkan pengirim delegasi mengirim dari salah satu akun pemilik, pemilik membuat kebijakan otorisasi pengiriman dan melampirkan kebijakan ke identitas yang dipilih dengan menggunakan konsol Amazon SES atau Amazon SES API.
4. Pemilik identitas memberi pengirim delegasi ARN dari identitas resmi sehingga pengirim delegasi dapat memberikan ARN ke Amazon SES pada saat pengiriman email.
5. Pengirim delegasi dapat mengatur pemberitahuan pentalan dan keluhan melalui penerbitan acara yang diaktifkan dalam set konfigurasi yang ditentukan selama pengiriman delegasi. Pemilik identitas juga dapat mengatur pemberitahuan umpan balik email untuk peristiwa pentalan dan keluhan yang akan dikirim ke topik Amazon SNS pengirim delegasi.

> **Note**
>
> Jika pemilik identitas menonaktifkan pengiriman pemberitahuan peristiwa, pengirim delegasi harus menyiapkan penerbitan acara untuk mempublikasikan peristiwa pentalan dan keluhan ke topik Amazon SNS atau aliran Firehose. Pengirim juga harus menerapkan set konfigurasi yang berisi aturan publikasi peristiwa ke setiap email yang dikirim. Apabila baik pemilik identitas maupun pengirim delegasi tidak menyiapkan metode pengiriman notifikasi untuk peristiwa pentalan dan aduan, maka Amazon SES secara otomatis mengirimkan notifikasi peristiwa melalui email ke alamat di bidang Jalur Kembali email (atau alamat di bidang Sumber, jika Anda tidak menentukan alamat Jalur Kembali), bahkan jika pemilik identitas menonaktifkan penerusan umpan balik email.

6. Pengirim delegasi mencoba untuk mengirim email melalui Amazon SES atas nama pemilik identitas dengan meneruskan ARN identitas pemilik identitas di permintaan atau di header email. Pengirim delegasi dapat mengirim email dengan menggunakan antarmuka SMTP Amazon SES atau API Amazon SES. Setelah menerima permintaan, Amazon SES memeriksa setiap kebijakan yang terlampir pada identitas, dan menerima email jika pengirim delegasi diotorisasi untuk menggunakan alamat "Dari" dan alamat "Jalur Kembali" yang ditentukan; jika tidak, Amazon SES mengembalikan kesalahan dan tidak menerima pesan.

> **Important**
>
> AWSAkun pengirim delegasi harus dihapus dari kotak pasir sebelum dapat digunakan untuk mengirim email ke alamat yang tidak diverifikasi.

7. Jika pemilik identitas perlu membatalkan otorisasi pengirim delegasi, pemilik identitas mengedit kebijakan otorisasi pengiriman atau menghapus kebijakan sepenuhnya. Pemilik identitas dapat melakukan tindakan baik dengan menggunakan konsol Amazon SES atau API Amazon SES.

Untuk informasi selengkapnya tentang cara pemilik identitas atau pengirim delegasi melakukan tugas-tugas tersebut, lihat Tugas pemilik identitas atau Tugas pengirim delegasi, masing-masing.

Atribusi fitur pengiriman email

Penting untuk memahami peran pengirim delegasi dan pemilik identitas sehubungan dengan fitur pengiriman email Amazon SES seperti kuota pengiriman harian, pentalan dan aduan, penandatanganan DKIM, penerusan umpan balik, dan sebagainya. Atribusi adalah sebagai berikut:

- Kuota pengiriman – Email yang dikirim dari identitas pemilik identitas dihitung terhadap kuota pengirim delegasi.
- Pentalan dan aduan – Peristiwa pentalan dan aduan dicatat terhadap akun Amazon SES pengirim delegasi, sehingga dapat mempengaruhi reputasi pengirim delegasi.
- Penandatanganan DKIM – Jika pemilik identitas telah mengaktifkan penandatanganan Easy DKIM untuk identitas, semua email yang dikirim dari identitas tersebut akan ditandatangani oleh DKIM, termasuk email yang dikirim oleh pengirim delegasi. Hanya pemilik identitas yang dapat mengendalikan jika email ditandatangani oleh DKIM.
- Notifikasi – Baik pemilik identitas dan pengirim delegasi dapat menyiapkan notifikasi untuk pentalan dan aduan. Pemilik identitas email juga dapat mengaktifkan penerusan umpan balik email. Untuk informasi tentang penyiapan notifikasi, lihat Memantau aktivitas pengiriman Amazon SES.
- Verifikasi – Pemilik identitas bertanggung jawab untuk mengikuti prosedur di Identitas terverifikasi untuk memverifikasi bahwa mereka memiliki alamat email dan domain yang diotorisasi agar digunakan oleh pengirim delegasi. Pengirim delegasi tidak perlu memverifikasi alamat email atau domain khusus untuk otorisasi pengiriman.

> **Important**
>
> AWSAkun pengirim delegasi harus dihapus dari kotak pasir sebelum dapat digunakan untuk mengirim email ke alamat yang tidak diverifikasi.

- Wilayah AWS – Pengirim delegasi harus mengirim email dari Wilayah AWS tempat identitas dari pemilik identitas diverifikasi. Kebijakan otorisasi pengiriman yang memberikan izin kepada pengirim delegasi harus dilampirkan ke identitas di Wilayah tersebut.
- Penagihan – Semua pesan yang dikirim dari akun pengirim delegasi, termasuk email yang dikirim pengirim delegasi menggunakan alamat pemilik identitas, dibebankan ke pengirim delegasi.

Tugas pemilik identitas untuk otorisasi pengiriman Amazon SES

Bagian ini menjelaskan langkah-langkah yang harus diambil pemilik identitas ketika mengonfigurasi otorisasi pengiriman.

Topik

- Memverifikasi identitas untuk otorisasi pengiriman Amazon SES
- Menyiapkan notifikasi pemilik identitas untuk otorisasi pengiriman Amazon SES
- Mendapatkan informasi dari pengirim delegasi dengan otorisasi pengiriman Amazon SES
- Membuat kebijakan otorisasi pengiriman di Amazon SES
- Misalnya kebijakan mengirim kebijakan
- Menyediakan informasi identitas kepada pengirim delegasi untuk otorisasi pengiriman Amazon SES

Memverifikasi identitas untuk otorisasi pengiriman Amazon SES

Langkah pertama dalam mengonfigurasi otorisasi pengiriman adalah membuktikan bahwa Anda memiliki alamat email atau domain yang akan digunakan pengirim delegasi untuk mengirim email. Prosedur verifikasi dijelaskan di Identitas terverifikasi.

Anda dapat mengonfirmasi bahwa alamat email atau domain diverifikasi dengan memeriksa statusnya di bagian Identitas Terifikasihttps://console.aws.amazon.com/ses/atau dengan menggunakan`GetIdentityVerificationAttributes`Operasi API.

Sebelum Anda atau pengirim delegasi dapat mengirim email ke alamat email yang tidak diverifikasi, Anda harus mengirimkan permintaan agar akun Anda dihapus dari sandbox Amazon SES. Untuk informasi selengkapnya, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).

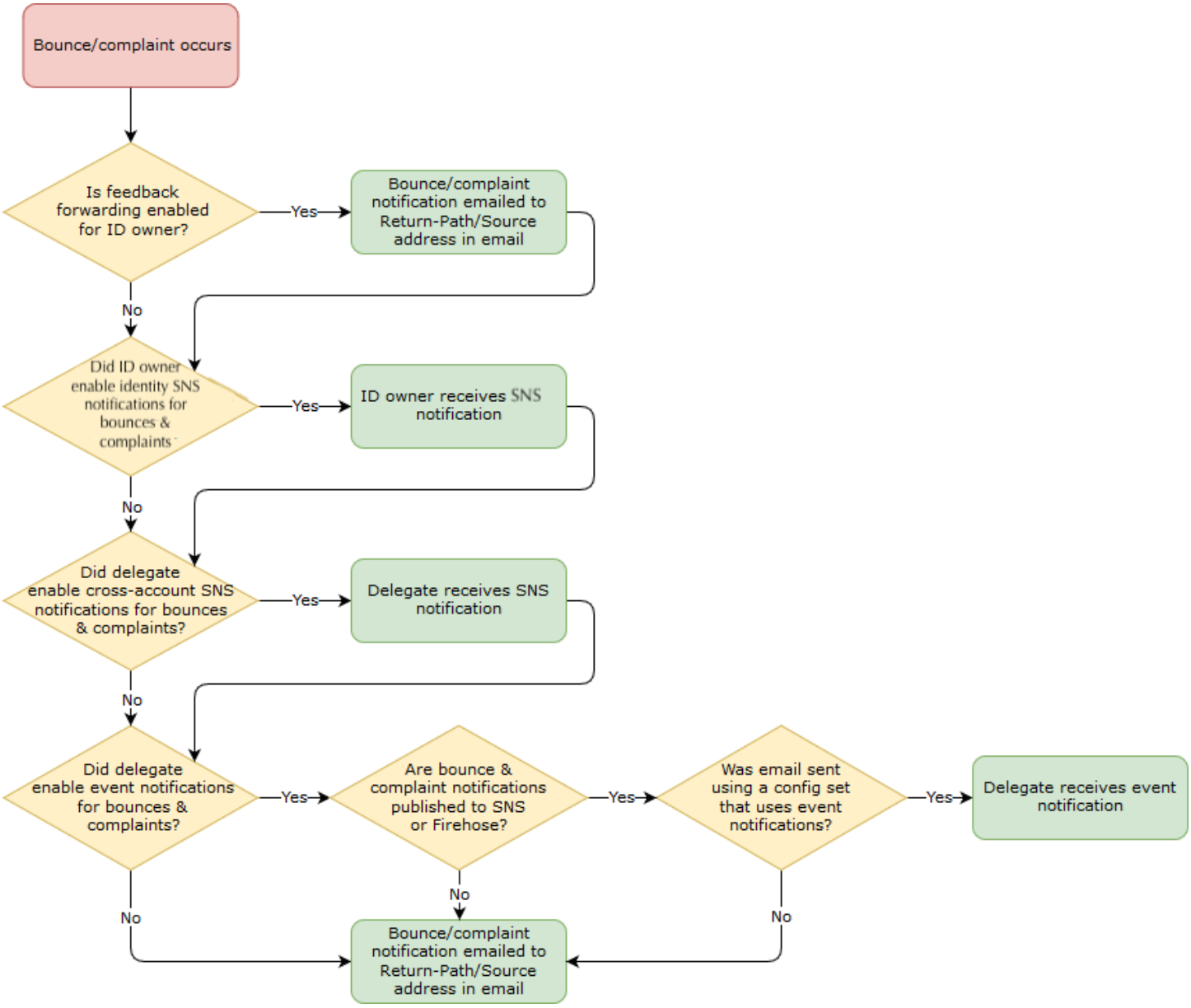
> **Important**
>
> ParameterAkun AWSdari pengirim delegasi harus dihapus dari sandbox sebelum dapat digunakan untuk mengirim email ke alamat yang tidak diverifikasi.

Menyiapkan notifikasi pemilik identitas untuk otorisasi pengiriman Amazon SES

Jika Anda mengotorisasi pengirim delegasi untuk mengirim email atas nama Anda, Amazon SES menghitung semua pentalan atau aduan yang dihasilkan email tersebut terhadap batas pentalan dan aduan pengirim delegasi, bukan milik Anda sendiri. Namun, jika alamat IP Anda muncul pada anti-spam pihak ketiga, Daftar Blackhole berbasis DNS (DNSBL) sebagai akibat dari pesan yang dikirim oleh pengirim delegasi, reputasi identitas Anda mungkin rusak. Untuk alasan ini, jika Anda adalah pemilik identitas, Anda harus mengatur penerusan umpan balik email untuk semua identitas Anda, termasuk yang telah Anda otorisasi untuk pengiriman delegasi. Untuk informasi selengkapnya, lihat Menerima notifikasi Amazon SES melalui email.

Pengirim delegasi dapat dan harus mengatur pemberitahuan bouncing dan keluhan mereka sendiri untuk identitas yang telah Anda izinkan untuk mereka gunakan. Mereka dapat mengatur penerbitan acara untuk mempublikasikan peristiwa bouncing dan keluhan ke topik Amazon SNS atau aliran Firehose.

Jika baik pemilik identitas maupun pengirim delegasi menyiapkan metode notifikasi pengiriman untuk peristiwa pentalan dan aduan, atau jika pengirim tidak menerapkan set konfigurasi yang menggunakan aturan publikasi peristiwa, maka Amazon SES secara otomatis mengirimkan notifikasi peristiwa melalui email ke alamat di bidang Jalur Kembali email (atau alamat di bidang Sumber, jika Anda tidak menentukan alamat Jalur Kembali), bahkan jika Anda menonaktifkan penerusan umpan balik email. Hal ini digambarkan dalam citra berikut.



Mendapatkan informasi dari pengirim delegasi dengan otorisasi pengiriman Amazon SES

Kebijakan otorisasi pengiriman Anda harus menentukan setidaknya satu pokok, yaitu entitas pengirim delegasi yang Anda berikan akses sehingga mereka dapat mengirim atas nama salah satu identitas terverifikasi Anda. Untuk kebijakan otorisasi pengiriman Amazon SES, prinsipiel dapat berupaAWS akun pengirim delegasi atau penggunaAWS Identity and Access Management (IAM),AWS layanan.

Cara mudah untuk berpikir tentang hal ini adalah bahwa kepala sekolah (delegasi pengirim) adalah penerima, dan Anda (pemilik identitas) adalah pemberi dalam kebijakan otorisasi di mana Anda memberi mereka Izinkan izin untuk mengirim kombinasi email, email mentah, email template, atau email template massal dari sumber daya (identitas terverifikasi) yang Anda miliki.

Jika Anda ingin kendali paling detail, minta pengirim delegasi untuk menyiapkan pengguna IAM sehingga hanya satu pengirim delegasi yang dapat mengirim untuk Anda daripada setiap pengguna diAWS akun pengirim delegasi. Pengirim delegasi dapat menemukan informasi tentang penyiapan pengguna IAM diAWS Akun di Panduan Pengguna IAM.

Minta IDAWS akun atau Amazon Resource Name (ARN) pengguna IAM dari pengirim delegasi sehingga Anda dapat menyertakannya ke dalam kebijakan otorisasi pengiriman Anda. Anda dapat merujuk pengirim delegasi Anda ke petunjuk untuk menemukan informasi ini di Memberikan informasi kepada pemilik identitas. Jika pengirim delegasi adalah layanan AWS, lihat dokumentasi untuk layanan tersebut untuk menentukan nama layanan.

Kebijakan contoh berikut menggambarkan elemen dasar dari apa yang diperlukan dalam kebijakan yang dibuat oleh pemilik identitas untuk mengotorisasi pengirim delegasi untuk mengirim dari sumber daya pemilik identitas. Pemilik identitas akan masuk ke alur kerja Identitas terverifikasi, dan di bawah Otorisasi, gunakan generator Kebijakan untuk membuat, dalam bentuknya yang paling sederhana, kebijakan dasar berikut yang memungkinkan pengirim delegasi untuk mengirim atas nama sumber daya yang dimiliki oleh pemilik identitas:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "stmt1632010098378",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "ses:SendEmail",
                "ses:SendRawEmail"
            ],
            "Resource": "arn:aws:ses:us-east-1:444455556666:identity/bob@example.com",
            "Condition": {}
        }
    ]
}
```

Untuk kebijakan di atas, legenda berikut menjelaskan elemen-elemen kunci dan siapa yang memilikinya:

- Principal - bidang ini diisi dengan ARN pengguna IAM pengirim delegasi.

- Tindakan - bidang ini diisi dengan dua tindakan SES (`SendEmail&SendRawEmail`) bahwa pemilik identitas mengizinkan pengirim delegasi untuk melakukan dari sumber daya pemilik identitas.
- Sumber daya — bidang ini diisi dengan sumber daya terverifikasi pemilik identitas yang mereka otorisasi untuk dikirim oleh pengirim delegasi.

Membuat kebijakan otorisasi pengiriman di Amazon SES

Mirip dengan membuat kebijakan untuk mengirim email menggunakan alamat email atau domain (identitas) yang Anda Amazon SES (dengan menggunakan alamat email atau domain (identitas) yang Anda miliki (identitas) yang Anda miliki, Anda membuat kebijakan ke identitas, dan kemudian melampirkan kebijakan ke identitas. Membuat kebijakan otorisasi identitas

Untuk daftar tindakan API yang dapat ditentukan dalam kebijakan otorisasi pengiriman, lihat baris Tindakan di the section called “Pernyataan khusus untuk kebijakan” tabel.

Anda dapat membuat kebijakan otorisasi pengiriman dengan menggunakan pembuat kebijakan atau dengan membuat kebijakan khusus. Prosedur khusus untuk membuat kebijakan otorisasi pengiriman disediakan untuk kedua metode.

> **Note**
>
> - Mengirim kebijakan otorisasi yang dilampirkan ke identitas alamat email lebih diutamakan daripada kebijakan yang dilampirkan ke identitas domain terkait. Misalnya, jika Anda membuat kebijakan untuk example.com yang melarang pengirim delegasi, dan Anda membuat kebijakan untuk sender@example.com yang mengizinkan pengirim delegasi, maka pengirim delegasi dapat mengirim email dari sender@example.com, tetapi tidak dari alamat lain di domain example.com.
> - Jika Anda membuat kebijakan untuk example.com yang mengizinkan pengirim delegasi, dan Anda membuat kebijakan untuk sender@example.com yang melarang pengirim delegasi, maka pengirim delegasi dapat mengirim email dari alamat mana pun di domain example.com, kecuali untuk sender@example.com.
> - Jika Anda tidak terbiasa dengan struktur kebijakan otorisasi SES, lihat. Anatomi kebijakan

Membuat kebijakan

Anda dapat menggunakan generator kebijakan untuk membuat kebijakan untuk membuat kebijakan.

Untuk membuat kebijakan

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Di kontainer Identitas di layar Identitas terverifikasi, pilih identitas terverifikasi yang ingin Anda otorisasi agar pengirim delegasi dikirim atas nama Anda.
4. Pilih tab Otorisasi identitas terverifikasi.
5. Di panel Kebijakan otorisasi, pilih Buat kebijakan, lalu pilih Gunakan pembuat kebijakan dari menu tarik-turun.
6. Di panel Create statement, pilih Allow in the Effect field. (Jika Anda ingin membuat kebijakan untuk membatasi pengirim delegasi Anda, pilih Tolak sebagai gantinya.)
7. Di bidang Prinsipal, masukkan Akun AWSID atau ARN pengguna IAM yang dibagikan pengirim delegasi Anda kepada Anda untuk mengizinkan mereka mengirim email atas nama akun Anda untuk identitas ini, lalu pilih Tambah. (Jika Anda ingin mengotorisasi lebih dari satu pengirim delegasi, ulangi langkah ini untuk masing-masing pengirim.)
8. Di bidang Tindakan, centang kotak untuk setiap jenis pengiriman yang ingin Anda otorisasi untuk pengirim delegasi Anda.
9. (Opsional) Perluas Tentukan kondisi jika Anda ingin menambahkan pernyataan kualifikasi ke izin pengirim delegasi.
    a. Pilih operator dari dropdown Operator.
    b. Pilih jenis dari dropdown Key.
    c. Masing-masing dengan jenis kunci yang Anda pilih, masukkan nilainya di bidang Nilai. (Jika Anda ingin menambahkan lebih banyak kondisi, pilih Tambahkan kondisi baru dan ulangi langkah ini untuk setiap kondisi tambahan.)
10. Pilih Simpan pernyataan.
11. (Opsional) Perluas Buat pernyataan lain jika Anda ingin menambahkan lebih banyak pernyataan ke kebijakan Anda dan ulangi langkah 6 - 10.
12. Pilih Berikutnya dan di layar Sesuaikan kebijakan, wadah Edit detail kebijakan memiliki bidang tempat Anda dapat mengubah atau menyesuaikan Nama kebijakan dan dokumen Kebijakan itu sendiri.
13. Pilih Berikutnya dan pada layar Tinjau dan terapkan, wadah Ikhtisar akan menampilkan identitas terverifikasi yang Anda otorisasi untuk pengirim delegasi serta nama kebijakan ini. Di panel

Dokumen kebijakan akan menjadi kebijakan aktual yang baru saja Anda tulis bersama dengan kondisi apa pun yang Anda tambahkan - tinjau kebijakan dan jika terlihat benar, pilih Terapkan kebijakan. (Jika Anda perlu mengubah atau memperbaiki sesuatu, pilih Sebelumnya dan bekerja di wadah Edit detail kebijakan.) Kebijakan yang baru saja Anda buat akan memungkinkan pengirim delegasi Anda untuk mengirim atas nama Anda.

14. (Opsional) Jika pengirim delegasi Anda juga ingin menggunakan topik SNS yang mereka miliki, untuk menerima pemberitahuan umpan balik saat mereka menerima pantulan atau keluhan, atau saat email dikirimkan, Anda harus mengonfigurasi topik SNS mereka dalam identitas terverifikasi ini. (Pengirim delegasi Anda perlu berbagi dengan Anda topik SNS ARN mereka.) Pilih tab Pemberitahuan dan pilih Edit di wadah pemberitahuan umpan balik:

    a. Pada panel Konfigurasi topik SNS, di salah satu bidang umpan balik, (Bounce, Complaint, atau Delivery), pilih topik SNS yang tidak Anda miliki dan masukkan topik SNS ARN yang dimiliki dan dibagikan dengan Anda oleh pengirim delegasi Anda. (Hanya pengirim delegasi Anda yang akan mendapatkan pemberitahuan ini karena mereka memiliki topik SNS - Anda, sebagai pemilik identitas, tidak akan melakukannya.)

    b. (Opsional) Jika Anda ingin pemberitahuan topik Anda menyertakan header dari email asli, centang kotak Sertakan header email asli langsung di bawah nama topik SNS dari setiap jenis umpan balik. Opsi ini hanya tersedia jika Anda telah menetapkan topik Amazon SNS untuk tipe notifikasi terkait. Untuk informasi tentang isi header email asli, lihat objek `mail` di Isi notifikasi.

    c. Pilih Save changes (Simpan perubahan). Perubahan yang Anda buat pada pengaturan notifikasi Anda mungkin memerlukan beberapa menit untuk diterapkan.

    d. (Opsional) Karena pengirim delegasi Anda akan mendapatkan pemberitahuan topik Amazon SNS untuk bouncing dan keluhan, Anda dapat menonaktifkan pemberitahuan email sepenuhnya jika Anda tidak ingin menerima umpan balik untuk pengiriman identitas ini. Untuk menonaktifkan umpan balik email untuk pantulan dan keluhan, di bawah tab Pemberitahuan, dalam wadah Penerusan Umpan Balik Email, pilih Edit, hapus centang pada kotak Diaktifkan, dan pilih Simpan perubahan. Pemberitahuan status pengiriman sekarang hanya akan dikirim ke topik SNS yang dimiliki oleh pengirim delegasi Anda.

Membuat kebijakan

Jika Anda ingin membuat kebijakan otorisasi pengiriman, dan melampirkannya ke identitas, Anda memiliki opsi berikut:

- Menggunakan API Amazon SES – Membuat kebijakan di editor teks lalu melampirkan kebijakan ke identitas dengan menggunakan API `PutIdentityPolicy` yang dijelaskan di Referensi API Amazon Simple Email Service.
- Menggunakan konsol Amazon SES – Membuat kebijakan di editor teks dan melampirkannya ke identitas dengan menempelkannya ke editor kebijakan kustom di konsol Amazon SES. Prosedur berikut menjelaskan metode ini.

Untuk membuat kebijakan

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Di kontainer Identitas di layar Identitas terverifikasi, pilih identitas terverifikasi yang ingin Anda otorisasi agar pengirim delegasi dikirim atas nama Anda.
4. Di layar detail identitas terverifikasi yang Anda pilih di langkah sebelumnya, pilih tab Otorisasi.
5. Di panel Kebijakan otorisasi, pilih Buat kebijakan, lalu pilih Buat kebijakan kustom dari menu tarik-turun.
6. Di panel panel dokumen, ketik atau tempel teks ke dalam format JSON. Anda juga dapat menggunakan generator kebijakan untuk membuat struktur dasar untuk membuat dengan cepat di sini.
7. Pilih Terapkan Kebijakan. (Jika Anda perlu mengubah kebijakan kustom Anda, cukup pilih kotak centang di bawah tab Otorisasi, pilih Edit, dan buat perubahan Anda di panel dokumen Kebijakan diikuti oleh Simpan perubahan).
8. (Opsional) Jika pengirim delegasi Anda juga ingin menggunakan topik SNS yang mereka miliki, untuk menerima pemberitahuan umpan balik saat mereka menerima pantulan atau keluhan, atau saat email dikirimkan, Anda harus mengonfigurasi topik SNS mereka dalam identitas terverifikasi ini. (Pengirim delegasi Anda perlu berbagi dengan Anda topik SNS ARN mereka.) Pilih tab Pemberitahuan dan pilih Edit di wadah pemberitahuan umpan balik:

    a. Pada panel Konfigurasi topik SNS, di salah satu bidang umpan balik, (Bounce, Complaint, atau Delivery), pilih topik SNS yang tidak Anda miliki dan masukkan topik SNS ARN yang dimiliki dan dibagikan dengan Anda oleh pengirim delegasi Anda. (Hanya pengirim delegasi Anda yang akan mendapatkan pemberitahuan ini karena mereka memiliki topik SNS - Anda, sebagai pemilik identitas, tidak akan melakukannya.)

b. (Opsional) Jika Anda ingin pemberitahuan topik Anda menyertakan header dari email asli, centang kotak Sertakan header email asli langsung di bawah nama topik SNS dari setiap jenis umpan balik. Opsi ini hanya tersedia jika Anda telah menetapkan topik Amazon SNS untuk tipe notifikasi terkait. Untuk informasi tentang isi header email asli, lihat objek `mail` di Isi notifikasi.

c. Pilih Save changes (Simpan perubahan). Perubahan yang Anda buat pada pengaturan notifikasi Anda mungkin memerlukan beberapa menit untuk diterapkan.

d. (Opsional) Karena pengirim delegasi Anda akan mendapatkan pemberitahuan topik Amazon SNS untuk bouncing dan keluhan, Anda dapat menonaktifkan pemberitahuan email sepenuhnya jika Anda tidak ingin menerima umpan balik untuk pengiriman identitas ini. Untuk menonaktifkan umpan balik email untuk pantulan dan keluhan, di bawah tab Pemberitahuan, dalam wadah Penerusan Umpan Balik Email, pilih Edit, hapus centang pada kotak Diaktifkan, dan pilih Simpan perubahan. Pemberitahuan status pengiriman sekarang hanya akan dikirim ke topik SNS yang dimiliki oleh pengirim delegasi Anda.

Misalnya kebijakan mengirim kebijakan

Mengirim otorisasi memungkinkan Anda untuk menentukan syarat detail saat Anda mengizinkan pengirim delegasi untuk mengirim atas nama Anda.

Misalnya, kondisi dan contoh berikut menunjukkan cara menulis kebijakan untuk mengontrol berbagai aspek pengiriman:

- Ketentuan khusus untuk mengirim otorisasi
- Menentukan pengirim delegasi
- Membatasi alamat "Dari"
- Membatasi waktu delegasi dapat mengirim email
- Membatasi tindakan pengiriman email
- Membatasi nama tampilan pengirim email
- Menggunakan beberapa pernyataan

Ketentuan khusus untuk mengirim otorisasi

Syarat adalah pembatasan tentang izin di pernyataan. Bagian dari pernyataan yang menentukan syarat dapat menjadi yang paling detail dari semua bagian. Kunci adalah karakteristik spesifik yang menjadi dasar pembatasan akses, seperti tanggal dan waktu permintaan.

Anda menggunakan syarat maupun kunci secara bersama-sama untuk mengekspresikan pembatasan. Misalnya, jika Anda ingin membatasi pengirim delegasi membuat permintaan ke Amazon SES atas nama Anda setelah 30 Juli 2019, Anda menggunakan syarat yang disebut `DateLessThan`. Anda menggunakan kunci yang disebut `aws:CurrentTime` dan mengaturnya ke nilai `2019-07-30T00:00:00Z`.

Anda dapat menggunakan salah satuAWS-kunci lebar terdaftar diKunci yang TersediadiPanduan Pengguna IAM, atau Anda dapat menggunakan salah satu kunci berikut khusus untuk SES yang berguna dalam mengirim kebijakan otorisasi:

Kunci syarat	Deskripsi
`ses:Recipients`	Membatasi alamat penerima, yang menyertakan alamat Kepada:, "CC", dan "BCC".
`ses:FromAddress`	Membatasi alamat "Dari".
`ses:FromDisplayName`	Membatasi isi string yang digunakan sebagai nama tampilan "Dari" (terkadang disebut "akrab dari"). Misalnya, nama tampilan "John Doe <johndoe@ example.com>" adalah John Doe.
`ses:FeedbackAddress`	Batasi alamat "Jalur Kembali", yang merupakan alamat tempat pentalan dan aduan dapat dikirim kepada Anda melalui penerusan umpan balik email. Untuk informasi tentang penerusan umpan balik email, lihat Menerima notifikasi Amazon SES melalui email.

Anda dapat menggunakan syarat `StringEquals` dan `StringLike` dengan kunci Amazon SES. Syarat ini untuk pencocokan string peka huruf besar kecil. Untuk `StringLike`, nilai dapat mencakup wildcard yang cocok dengan beberapa karakter (*) atau wildcard yang cocok dengan karakter tunggal (?) di mana pun di string. Misalnya, syarat berikut menentukan bahwa pengirim delegasi hanya dapat mengirim dari alamat "Dari" yang dimulai dengan faktur dan diakhiri dengan @example.com:

```
"Condition": {
    "StringLike": {
      "ses:FromAddress": "invoicing*@example.com"
    }
```

```
}
```

Anda juga dapat menggunakan syarat `StringNotLike` untuk mencegah pengirim delegasi mengirim email dari alamat email tertentu. Misalnya, Anda dapat melarang pengiriman dari admin@example.com, dan alamat serupa seperti "admin"@example.com, admin+1@example.com, atau sender@admin.example.com, dengan menyertakan syarat berikut di pernyataan kebijakan Anda:

```
"Condition": {
    "StringNotLike": {
      "ses:FromAddress": "*admin*example.com"
    }
 }
```

Untuk informasi selengkapnya tentang cara menentukan syarat, lihat Elemen Kebijakan JSON IAM: Syarat di Panduan Pengguna IAM.

Menentukan pengirim delegasi

Prinsipiel, yang merupakan entitas yang Anda berikan izin, bisa berupa akun Akun AWS, pengguna (IAM)AWS Identity and Access Management, atau layanan AWS.

Contoh berikut menunjukkan kebijakan sederhana yang memungkinkan ID AWS 123456789012 untuk mengirim email dari identitas terverifikasi example.com (yang dimiliki oleh Akun AWS 888888888888). The`Condition`pernyataan dalam kebijakan ini hanya mengizinkan delegasi (yaitu,AWSID123456789012) untuk mengirim email dari alamatpemasaran+. * @example .com, dimana*adalah string apa pun yang ingin ditambahkan pengirim setelahnyapemasaran+..

```
{
  "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"
        }
      }
    }
  ]
}
```

Contoh kebijakan berikut memberikan izin ke dua pengguna IAM untuk mengirim dari identitas example.com. Pengguna IAM ditentukan oleh Amazon Resource Name (ARN) mereka.

```
{
  "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"
      ]
    }
  ]
}
```

Contoh kebijakan berikut memberikan izin ke Amazon Cognito untuk mengirim dari identitas example.com.

```
{
  "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"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "888888888888",
          "aws:SourceArn": "arn:aws:cognito-idp:us-east-1:888888888888:userpool/your-
user-pool-id-goes-here"
        }
      }
    }
  ]
}
```

Contoh kebijakan berikut memberikan izin ke semua akun di Organisasi AWS untuk mengirim dari identitas example.com. TheAWSOrganisasi ditentukan menggunakanPrincipalOrgIDkunci kondisi global.

```
{
  "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-xxxxxxxxxxx"
        }
      }
    }
  ]
}
```

Membatasi alamat "Dari"

Jika Anda menggunakan domain terverifikasi, Anda mungkin ingin membuat kebijakan yang hanya mengizinkan pengirim delegasi untuk mengirim dari alamat email yang ditentukan. Untuk membatasi alamat “Dari”, Anda mengatur kondisi pada kunci yang disebutses:FromAddress. Kebijakan berikut memungkinkan ID Akun AWS 123456789012 untuk mengirim dari identitas example.com, tetapi hanya dari alamat email sender@example.com.

```
{
  "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"
        }
      }
    }
  ]
}
```

Membatasi waktu delegasi dapat mengirim email

Anda juga dapat mengonfigurasi kebijakan otorisasi pengirim sehingga pengirim delegasi hanya dapat mengirim email pada waktu tertentu dalam sehari, atau dalam rentang tanggal tertentu. Misalnya, jika Anda berencana mengirim email kampanye selama bulan September 2021, Anda dapat menggunakan kebijakan berikut untuk membatasi kemampuan delegasi untuk mengirim email ke bulan tersebut saja.

```
{
  "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":"2021-08-31T12:00Z"
        },
        "DateLessThan":{
          "aws:CurrentTime":"2021-10-01T12:00Z"
        }
      }
    }
  ]
}
```

Membatasi tindakan pengiriman email

Ada dua tindakan yang dapat digunakan pengirim untuk mengirim email dengan Amazon SES: `SendEmail` dan `SendRawEmail`, tergantung pada banyaknya kendali yang diinginkan pengirim atas format email. Kebijakan otorisasi pengiriman memungkinkan Anda untuk membatasi pengirim

delegasi ke salah satu dari dua tindakan tersebut. Namun, banyak pemilik identitas meninggalkan detail email yang mengirim panggilan ke pengirim delegasi dengan mengaktifkan kedua tindakan di kebijakan mereka.

> **Note**
>
> Jika Anda ingin memungkinkan pengirim delegasi untuk mengakses Amazon SES melalui antarmuka SMTP, Anda minimal harus memilih `SendRawEmail`.

Jika kasus penggunaan Anda sedemikian rupa sehingga Anda ingin membatasi tindakan, Anda dapat melakukannya dengan memasukkan hanya salah satu tindakan di kebijakan otorisasi pengiriman Anda. Contoh berikut menunjukkan cara membatasi tindakan untuk `SendRawEmail`.

```
{
  "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"
      ]
    }
  ]
}
```

Membatasi nama tampilan pengirim email

Beberapa klien email menampilkan nama "akrab" pengirim email (jika header email menyediakannya), bukan alamat "Dari" yang sebenarnya. Misalnya, nama tampilan "John Doe <johndoe@example.com>" adalah John Doe. Untuk instans, Anda dapat mengirim email dari user@example.com, namun Anda lebih suka penerima melihat bahwa email berasal dari Pemasaran daripada dari user@example.com. Kebijakan berikut memungkinkan ID Akun AWS 123456789012

untuk mengirim dari identitas example.com, tetapi hanya jika nama tampilan alamat "Dari" menyertakan Pemasaran.

```
{
  "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"
        }
      }
    }
  ]
}
```

Menggunakan beberapa pernyataan

Kebijakan otorisasi pengiriman Anda dapat mencakup beberapa pernyataan. Contoh kebijakan berikut ini memiliki dua pernyataan. Pernyataan pertama mengautorisasi dua Akun AWS untuk mengirim dari sender@example.com selama alamat "Dari" dan alamat umpan balik keduanya menggunakan domain example.com. Pernyataan kedua mengotorisasi pengguna IAM untuk mengirim email dari sender@example.com selama alamat email penerima berada di bawah domain example.com.

```
{
  "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"
        }
      }
    }
  ]
}
```

Menyediakan informasi identitas kepada pengirim delegasi untuk otorisasi pengiriman Amazon SES

Setelah membuat kebijakan otorisasi pengiriman dan melampirkannya ke identitas Anda, Anda dapat menyediakan pengirim delegasi dengan Amazon Resource Name (ARN) identitas. Pengirim delegasi akan meneruskan ARN ke Amazon SES di operasi pengiriman email atau di header email. Untuk menemukan ARN identitas Anda, ikuti langkah-langkah berikut.

Untuk menemukan ARN identitas

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Identitas yang terverifikasi.
3. Di daftar identitas, pilih identitas yang Anda lampirkan kebijakan otorisasi pengirimannya.
4. DiRingkasanpanel, kolom kedua,Amazon Resource Name (ARN) Amazon, akan berisi ARN identitas. Ini akan terlihat serupa dengan arn:aws:ses:us-east-1:123456789012:identity/user@example.com. Salin seluruh ARN dan berikan kepada pengirim delegasi Anda.

Tugas pengirim delegasi untuk otorisasi pengiriman Amazon SES

Sebagai pengirim delegasi, Anda mengirim email atas nama identitas yang tidak Anda miliki, tetapi berotorisasi untuk menggunakannya. Meskipun Anda mengirim atas nama pemilik identitas, pentalan dan aduan dihitung terhadap metrik pentalan dan aduan untuk akun AWS Anda, dan jumlah pesan yang Anda kirim dihitung terhadap kuota pengiriman Anda. Anda juga bertanggung jawab untuk meminta peningkatan kuota pengiriman yang mungkin Anda butuhkan untuk mengirim email pemilik identitas.

Sebagai pengirim delegasi, Anda harus menyelesaikan tugas berikut:

- Memberikan informasi kepada pemilik identitas
- Menggunakan notifikasi pengirim delegasi
- Mengirim email untuk pemilik identitas

Memberikan informasi kepada pemilik identitas untuk otorisasi pengiriman Amazon SES

Sebagai pengirim delegasi, Anda harus memberikan ID AWS akun kepada pemilik identitas atau nama sumber daya Amazon (ARN) pengguna IAM Anda karena Anda akan mengirim email atas nama pemilik identitas. Pemilik identitas memerlukan informasi akun Anda sehingga mereka dapat

membuat kebijakan yang memberi Anda izin untuk mengirim dari salah satu identitas terverifikasi mereka.

Jika Anda ingin menggunakan topik SNS Anda sendiri, Anda dapat meminta pemilik identitas Anda mengonfigurasi pemberitahuan umpan balik untuk pantulan, keluhan, atau pengiriman untuk dikirim ke satu atau beberapa topik SNS Anda. Untuk melakukan ini, Anda harus membagikan ARN topik SNS Anda dengan pemilik identitas Anda sehingga mereka dapat mengonfigurasi topik SNS Anda dalam identitas terverifikasi yang mereka otorisasi untuk Anda kirim.

Prosedur berikut menjelaskan cara menemukan informasi akun Anda dan ARN topik SNS untuk dibagikan dengan pemilik identitas Anda.

Untuk menemukan ID AWS akun Anda

1. Masuk ke AWS Management Console https://console.aws.amazon.com.
2. Di sudut kanan atas konsol, perluas nama/nomor akun Anda, dan pilih Akun Saya di dropdown.
3. Halaman Pengaturan akun akan membuka dan menampilkan semua informasi akun Anda termasuk ID AWS akun Anda.

Untuk menemukan ARN pengguna IAM Anda

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Pengguna.
3. Di daftar pengguna, pilih nama pengguna. Bagian Ringkasan menampilkan ARN pengguna IAM. ARN menyerupai contoh berikut: arn:aws:iam::123456789012:user/John.

Untuk menemukan topik SNS ARN Anda

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Di panel navigasi, pilih Pengguna.
3. Dalam daftar topik, ARN topik SNS ditampilkan di kolom ARN. ARN menyerupai contoh berikut: arn:aws:sns:us-east- 1:444455556666:. my-sns-topic

Menggunakan notifikasi pengirim delegasi untuk otorisasi pengiriman Amazon SES

Sebagai pengirim delegasi, Anda mengirim email atas nama identitas yang tidak Anda miliki, tetapi diizinkan untuk digunakan; namun, pantulan dan keluhan masih diperhitungkan terhadap metrik bouncing dan keluhan Anda, bukan metrik pemilik identitas.

Jika tingkat pentalan atau keluhan untuk akun Anda terlalu tinggi, akun Anda berisiko ditempatkan di bawah peninjauan atau memiliki kemampuan untuk mengirim email dijeda. Untuk alasan ini, penting bagi Anda untuk menyiapkan notifikasi dan memiliki proses untuk memantaunya. Anda juga perlu memiliki proses untuk menghapus alamat yang terpental atau diadukan dari milis Anda.

Oleh karena itu, sebagai pengirim delegasi, Anda dapat mengonfigurasi Amazon SES untuk mengirim pemberitahuan ketika peristiwa pentalan dan keluhan terjadi untuk email yang Anda kirim atas nama identitas apa pun yang tidak Anda miliki, tetapi telah diizinkan untuk digunakan oleh pemilik identitas. Anda juga dapat mengatur penerbitan acara untuk mempublikasikan pemberitahuan pentalan dan keluhan ke Amazon SNS atau Firehose.

> **Note**
>
> Jika Anda menyiapkan Amazon SES untuk mengirim notifikasi dengan menggunakan Amazon SNS, Anda akan dikenakan tarif Amazon SNS standar untuk notifikasi yang diterima. Untuk informasi lebih lanjut, lihat halaman harga Amazon SNS.

Membuat pemberitahuan pengirim delegasi baru

Anda dapat mengatur pemberitahuan pengiriman delegasi dengan set konfigurasi menggunakan penerbitan acara, atau dengan identitas terverifikasi yang dikonfigurasi dengan topik SNS Anda sendiri.

Prosedur diberikan di bawah ini untuk menyiapkan pemberitahuan pengiriman delegasi baru menggunakan salah satu metode:

- Penerbitan acara melalui set konfigurasi
- Pemberitahuan umpan balik ke topik SNS yang Anda miliki

Untuk mengatur penerbitan acara melalui konfigurasi yang ditetapkan untuk pengiriman delegasi Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Ikuti prosedur di Buat tujuan acara.
3. Setelah menyiapkan penerbitan acara dalam set konfigurasi, tentukan nama set konfigurasi saat Anda mengirim email sebagai pengirim delegasi menggunakan identitas terverifikasi yang diberi wewenang oleh pemilik identitas untuk Anda kirim. Lihat Mengirim email untuk pemilik identitas.

Untuk mengatur pemberitahuan umpan balik ke topik SNS yang Anda miliki untuk pengiriman delegasi Anda

1. Setelah Anda memutuskan topik SNS mana yang ingin Anda gunakan untuk pemberitahuan umpan balik, ikuti prosedur untuk menemukan ARN topik SNS Anda dan salin ARN lengkap dan bagikan dengan pemilik identitas Anda.
2. Minta pemilik identitas Anda untuk mengonfigurasi topik SNS Anda untuk pemberitahuan umpan balik tentang identitas bersama yang dia izinkan untuk Anda kirim. (Pemilik identitas Anda harus mengikuti prosedur yang diberikan untuk mengonfigurasi topik SNS dalam prosedur kebijakan otorisasi.)

Mengirim email untuk pemilik identitas untuk otorisasi pengiriman Amazon SES

Sebagai pengirim delegasi, Anda mengirim email dengan cara yang sama seperti yang dilakukan pengirim Amazon SES lainnya, kecuali bahwa Anda memberikan Amazon Resource Name (ARN) identitas yang telah diotorisasi pemilik identitas untuk Anda gunakan. Ketika Anda memanggil Amazon SES untuk mengirim email, Amazon SES memeriksa untuk melihat jika identitas yang Anda tentukan memiliki kebijakan yang mengotorisasi Anda untuk mengirimkannya.

Ada berbagai cara yang dapat Anda tentukan untuk ARN identitas ketika Anda mengirim email. Metode yang Anda gunakan tergantung jika Anda mengirim email dengan menggunakan operasi API Amazon SES atau antarmuka SMTP Amazon SES.

> **Important**
>
> Agar berhasil mengirim email, Anda harus terhubung ke titik akhir Amazon SES di Wilayah AWS tempat pemilik identitas memverifikasi identitas tersebut.

Selain itu, kedua akun AWS dari pemilik identitas dan pengirim delegasi harus dihapus dari sandbox sebelum salah satu akun dapat mengirim email ke alamat yang tidak diverifikasi. Untuk informasi lebih lanjut, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).

Menggunakan API Amazon SES

Seperti halnya pengirim email Amazon SES, jika Anda mengakses Amazon SES melalui Amazon SES API (baik secara langsung melalui HTTPS atau tidak langsung melaluiAWSSDK), Anda dapat memilih antara salah satu dari tiga tindakan pengiriman email:`SendEmail`,`SendTemplatedEmail`, dan`SendRawEmail`. Referensi API Amazon Simple Email Service menjelaskan detail API ini, tetapi kami memberikan gambaran umum parameter otorisasi pengiriman di sini.

SendRawEmail

Jika Anda ingin menggunakan`SendRawEmail`sehingga Anda dapat mengendalikan format email Anda, Anda dapat menentukan identitas resmi yang didelegasi dengan salah satu dari dua cara:

- Teruskan parameter opsional ke API **`SendRawEmail`**. Parameter yang diperlukan dijelaskan di tabel berikut:

Parameter	Deskripsi
`SourceArn`	ARN identitas yang terkait dengan kebijakan otorisasi pengiriman yang mengizinkan Anda untuk mengirim alamat email yang ditentukan di parameter `Source` dari `SendRawEmail` . **Note** Jika Anda hanya menentukan `SourceArn` , Amazon SES menetapkan alamat "Dari" dan alamat "Jalur Kembali" ke identitas yang Anda tentukan di `SourceArn` .

Parameter	Deskripsi
`FromArn`	ARN identitas yang terkait dengan kebijakan otorisasi pengiriman yang mengizinkan Anda untuk menentuka n alamat "Dari" tertentu di header email mentah.
`ReturnPathArn`	ARN identitas yang terkait dengan kebijakan otorisasi pengiriman yang mengizinkan Anda untuk menggunakan alamat email yang ditentukan di parameter `ReturnPath` dari `SendRawEmail` .

- Sertakan X-header di email. X-header adalah header kustom yang dapat Anda gunakan selain header email standar (seperti header Dari, Balas Ke, atau Subjek). Amazon SES mengenali tiga X-header yang dapat Anda gunakan untuk menentukan parameter otorisasi pengiriman:

> **Important**
>
> Jangan sertakan X-header ini di tanda tangan DKIM, karena header tersebut dihapus oleh Amazon SES sebelum mengirim email.

X-Header	Deskripsi
`X-SES-SOURCE-ARN`	Sesuai dengan `SourceArn` .
`X-SES-FROM-ARN`	Sesuai dengan `FromArn`.
`X-SES-RETURN-PATH-ARN`	Sesuai dengan `ReturnPathArn` .

Amazon SES menghapus semua X-header dari email sebelum mengirimnya. Jika Anda menyertakan beberapa instans dari X-header, Amazon SES hanya menggunakan instans pertama.

Contoh berikut menunjukkan email yang menyertakan X-header otorisasi pengiriman:

```
X-SES-SOURCE-ARN: arn:aws:ses:us-east-1:123456789012:identity/example.com
X-SES-FROM-ARN: arn:aws:ses:us-east-1:123456789012:identity/example.com
X-SES-RETURN-PATH-ARN: arn:aws:ses:us-east-1:123456789012: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 dan SendTemplatedEmail

Jika Anda menggunakan`SendEmail`atau`SendTemplatedEmail`operasi, Anda dapat menentukan identitas resmi yang didelegasi dengan meneruskan parameter opsional di bawah ini. Anda tidak dapat menggunakan metode X-header ketika Anda menggunakan`SendEmail`atau`SendTemplatedEmail`operasi.

Parameter	Deskripsi
`SourceArn`	ARN identitas yang terkait dengan kebijakan otorisasi pengiriman yang mengizinkan Anda untuk mengirim alamat email yang ditentukan di`Source`parameter dari salah satu`SendEmail` atau`SendTemplatedEmail` .
`ReturnPathArn`	ARN identitas yang terkait dengan kebijakan otorisasi pengiriman yang mengizinkan Anda untuk menggunakan alamat email yang ditentukan di`ReturnPath` parameter dari salah satu`SendEmail` atau`SendTemplatedEmail` .

Contoh berikut menunjukkan cara mengirim email yang mencakup`SourceArn`dan`ReturnPathArn`atribut menggunakan salah satu`SendEmail`atau`SendTemplatedEmail`operasi dan[SDK for Python](#).

```
import boto3
from botocore.exceptions import ClientError

# Create a new SES resource and specify a region.
client = boto3.client('ses',region_name="us-east-1")

# 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-east-1:123456789012:identity/example.com',
        ReturnPathArn='arn:aws:ses:us-east-1:123456789012: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'])
```

Menggunakan antarmuka SMTP Amazon SES

Ketika Anda menggunakan antarmuka Amazon SES SMTP untuk pengiriman delegasi, Anda harus menyertakan`X-SES-SOURCE-ARN`,`X-SES-FROM-ARN`, dan`X-SES-RETURN-PATH-ARN`header dalam pesan Anda. Teruskan header ini setelah Anda mengeluarkan perintah `DATA` di percakapan SMTP.

Mengirim email di Amazon SES di Amazon SES di Amazon SES di Amazon SES di Amazon SES

Kami merekomendasikan untuk menggunakan konsol Amazon SES untuk mengirim email dengan Amazon SES. Karena konsol tersebut mengharuskan Anda untuk secara manual memasukkan informasi, Anda biasanya hanya menggunakannya untuk mengirim email percobaan. Setelah Anda memulai dengan Amazon SES, Anda kemungkinan besar akan mengirim email dengan menggunakan salah satu dari antarmuka Amazon SES SMTP atau API. Namun, konsol tersebut berguna untuk memantau aktivitas pengiriman Anda.

Topik berikut menjelaskan cara menggunakan simulator kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol

- Menggunakan simulator kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari
- Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual

Menggunakan simulator kotak surat dari konsol kotak surat dari konsol kotak surat dari konsol kotak surat dari

> **Important**
>
> - Dalam tutorial ini, Anda mengirim email ke diri sendiri dari konsol sehingga Anda dapat memeriksa apakah Anda menerimanya. Untuk eksperimen atau pengujian beban lebih lanjut atau percobaan beban lebih lanjut, lihatMenggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual.
> - Email yang Anda kirim ke simulator kotak surat tidak diperhitungkan terhadap kuota pengiriman atau kecepatan pentalan dan aduan Anda, dan aduan Anda, atau apakah email

ke simulator kotak surat tidak diperhitungkan terhadap kuota pengiriman atau kecepatan pentalan dan aduan Anda.

Sebelum Anda mengikuti langkah-langkah ini, selesaikan tugas di Menyiapkan Amazon Simple Email Service.

Untuk mengirim pesan dari pesan dari konsol Amazon SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi di bawah Konfigurasi pilih Identitas Terverifikasi.
3. Dari tabel Identitas, pilih identitas email terverifikasi (dengan mengklik langsung nama identitas sebagai lawan memilih kotak centang). Jika Anda tidak memiliki identitas email terverifikasi, lihatMembuat identitas alamat email.
4. Pada halaman detail identitas email yang dipilih, pilih Kirim email uji.
5. Untuk Detail pesan, pilih Format Email. Dua pilihan tersebut sebagai berikut:

   - Diformat—Ini adalah opsi yang paling sederhana. Pilih opsi ini jika Anda hanya ingin mengetikkan teks pesan Anda ke kotak teks Badan. Ketika Anda mengirim email, Amazon SES menempatkan teks ke dalam format email untuk Anda.
   - Mentah— Pilih opsi ini jika Anda ingin mengirim pesan yang lebih rumit, seperti pesan yang menyertakan HTML atau lampiran. Karena fleksibilitas ini, Anda perlu memformat pesan, seperti yang dijelaskan di Mengirim email mentah menggunakan Amazon SES API v2, diri sendiri, dan kemudian tempel seluruh pesan yang diformat, termasuk header, ke kotak teks badan. Anda dapat menggunakan contoh berikut, yang berisi HTML, untuk mengirim email percobaan menggunakan format email Mentah. Salin dan tempel pesan ini secara keseluruhan ke dalam kotak teks Badan. Pastikan bahwa tidak ada garis kosong antara header `MIME-Version` dan header `Content-Type`; baris kosong antara dua baris ini menyebabkan email akan diformat sebagai teks biasa bukan HTML.

     ```
     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>
```

6. Pilih jenis skenario email simulasi yang ingin Anda uji dengan memperluas kotak daftar Skenario.
   - Jika Anda memilih Kustom dan Anda masih berada di sandbox Amazon SES, pastikan bahwa alamat di kolom Custom adalah alamat email terverifikasi. Untuk informasi selengkapnya, lihat Membuat identitas alamat email.
7. Isi bidang yang tersisa sesuai keinginan.
8. Pilih Kirim email uji.
9. Masuk ke alamat klien email yang Anda kirimi email. Anda akan menemukan pesan yang Anda kirim.

Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual

Amazon SES mencakup simulator kotak surat yang dapat Anda gunakan untuk menguji bagaimana aplikasi Anda menangani skenario pengiriman email yang berbeda. Simulator kotak surat berguna ketika, misalnya, Anda ingin menguji email mengirim aplikasi tanpa membuat alamat email fiktif, atau ketika Anda ingin menemukan throughput maksimum sistem Anda tanpa mempengaruhi kuota pengiriman harian Anda.

Pertimbangan penting

Pertimbangkan fitur dan batasan berikut ketika Anda menggunakan simulator kotak surat Amazon SES:

- Anda dapat menggunakan simulator kotak surat bahkan jika akun Anda berada di sandbox Amazon SES.
- Email yang Anda kirim ke simulator kotak surat dibatasi oleh laju pengiriman maksimum akun Anda, namun tidak memengaruhi kuota pengiriman harian Anda. Misalnya, jika akun Anda diotorisasi untuk mengirim 10.000 pesan per periode 24 jam, dan Anda mengirim 100 pesan ke

simulator kotak surat, Anda tetap dapat mengirim hingga 10.000 pesan ke penerima reguler tanpa mencapai kuota pengiriman.

- Email yang Anda kirim ke simulator kotak surat tidak memengaruhi metrik kemampuan pengiriman email atau reputasi Anda. Misalnya, jika Anda mengirim sejumlah besar pesan ke alamat pentalan simulator email, pesan tersebut tidak menampilkan peringatan bahwa rasio pentalan terlalu tinggi pada halaman konsol metrik reputasi.
- Untuk penagihan, email yang Anda kirim ke simulator kotak surat Amazon SES sama dengan email lain yang Anda kirim menggunakan Amazon SES. Dengan kata lain, kami menagih jumlah yang sama untuk pesan yang Anda kirim ke simulator kotak surat seperti yang Anda kirim ke penerima biasa.
- Simulator kotak surat mendukung pelabelan, yang mengizinkan Anda untuk mengirim email ke alamat simulator kotak surat yang sama dengan beberapa cara, atau untuk melihat cara aplikasi Anda menangani Variable Envelope Return Path (VERP). Misalnya, Anda dapat mengirim email ke bounce+label1@simulator.amazonses.com dan bounce+label2@simulator.amazonses.com untuk melihat jika aplikasi Anda dapat mencocokkan pesan pentalan dengan alamat email yang menyebabkan pentalan.
- Jika Anda menggunakan simulator kotak surat untuk mensimulasikan beberapa pentalan dari permintaan pengiriman yang sama, Amazon SES menggabungkan respons pentalan menjadi respons tunggal.

Menggunakan simulator kotak surat

Untuk menggunakan simulator email, temukan skenario di tabel berikut, dan kemudian kirim email ke alamat email yang sesuai.

> **Note**
>
> Ketika Anda mengirim email ke alamat simulator kotak surat, Anda harus mengirimkannya melalui Amazon SES, dengan menggunakan AWS CLI, sebuah SDK AWS, konsol Amazon SES, antarmuka Amazon SES SMTP, atau Amazon SES API. Simulator kotak surat tidak merespons email yang diterima dari sumber eksternal.

Skenario yang disimulasikan	Alamat Email
Pengiriman berhasil—Penerima penyedia email menerima email Anda. Jika Anda menyiapkan notifikasi pengiriman seperti yang dijelaskan di Menyiapkan pemberitahuan acara untuk Amazon SES, Amazon SES mengirimkan notifikasi pengiriman melalui Amazon Simple Notification Service (Amazon SNS).	success@simulator.amazonses.com
Pentalan—Penyedia email penerima menolak email Anda dengan kode respons SMTP 550 5.1.1 ("Pengguna Tidak Dikenali"). Amazon SES menghasilkan notifikasi pentalan dan, tergantung pada cara Anda menyiapkan akun Anda, mengirimkannya kepada Anda dalam email atau mengirimkan notifikasi ke topik Amazon SNS. Alamat email simulator kotak surat tidak ditempatkan di daftar penekanan Amazon SES, yang biasanya akan terjadi ketika pentalan keras terjadi. Respons pentalan yang Anda terima dari simulator kotak surat patuh dengan RFC 3464. Untuk informasi tentang cara menerima umpan balik pentalan, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.	bounce@simulator.amazonses.com
Respons otomatis—Penyedia email penerima menerima email Anda dan mengirimkannya ke kotak masuk penerima. Penyedia email mengirimkan respons otomatis, seperti pesan "out of the office" (OOTO), ke alamat di header Jalur Kembali email, atau alamat pengirim envelope ("MAIL FROM") jika header Jalur Kembali tidak ada. Respons otomatis yang	ooto@simulator.amazonses.com

Skenario yang disimulasikan	Alamat Email
Anda terima dari simulator kotak surat patuh dengan RFC 3834.	
Aduan—Penyedia email penerima menerima email Anda dan mengirimkannya ke kotak masuk penerima. Penerima memutuskan bahwa pesan Anda tidak diminta dan mengeklik "Tandai sebagai Spam" di klien emailnya. Amazon SES kemudian meneruskan notifikasi aduan kepada Anda melalui email atau dengan memberi tahu topik Amazon SNS, tergantung pada cara Anda mengatur akun Anda. Respons aduan yang Anda terima dari simulator kotak surat patuh dengan RFC 5965. Untuk informasi tentang cara menerima umpan balik aduan, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.	complaint@simulator.amazonses.com
Alamat penerima di daftar penekanan—Amazon SES menghasilkan pentalan keras seakan-akan alamat penerima ada di daftar penekanan global.	suppressionlist@simulator.amazonses.com

Pengujian peristiwa Tolak

Setiap pesan yang Anda kirim melalui Amazon SES dipindai virus. Jika Anda mengirim pesan yang berisi virus, Amazon SES menerima pesan, mendeteksi virus, dan menolak seluruh pesan. Ketika Amazon SES menolak pesan, pemrosesan pesan dihentikan, dan tidak berusaha untuk mengirimkannya ke server e-mail penerima. Amazon SES kemudian menghasilkan peristiwa Tolak.

Simulator kotak surat Amazon SES tidak menyertakan alamat untuk pengujian peristiwa Tolak. Namun, Anda dapat menguji peristiwa Tolak dengan menggunakan file uji European Institute for Computer Antivirus Research (EICAR). File ini adalah metode standar industri pengujian perangkat lunak antivirus dengan cara yang aman. Untuk membuat file uji EICAR, tempel teks berikut ke dalam file:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

Simpan file sebagai `sample.txt`, lampirkan ke email, dan kemudian kirim email ke alamat terverifikasi. Jika tidak ada masalah lain dengan email, Amazon SES menerima pesan, tetapi kemudian menolaknya seperti yang akan terjadi jika berisi virus yang sebenarnya.

> **Note**
>
> Email yang ditolak—termasuk email yang Anda kirim dengan menggunakan prosedur di atas—dihitung terhadap kuota pengiriman harian Anda. Kami menagih Anda untuk setiap pesan yang Anda kirim, termasuk pesan yang ditolak.

Pelajari selengkapnya tentang file uji EICAR, lihat Halaman file uji EICAR di Wikipedia.

Menggunakan set konfigurasi di Amazon SES

Set konfigurasi adalah grup aturan yang dapat Anda terapkan ke identitas terverifikasi Anda. Identitas terverifikasi adalah domain, subdomain, atau alamat email yang Anda gunakan untuk mengirim email melalui AmazonSES. Ketika Anda menerapkan set konfigurasi ke email, semua aturan dalam set konfigurasi tersebut akan diterapkan ke email.

Anda dapat menggunakan set konfigurasi untuk menerapkan jenis aturan berikut pada pengiriman email Anda dan dapat berisi salah satu, keduanya, atau tidak satu pun dari jenis ini:

- Tujuan acara - Memungkinkan Anda mempublikasikan metrik pengiriman email, termasuk jumlah pengiriman, pengiriman, pembukaan, klik, pantulan, dan keluhan ke AWS produk lain untuk setiap email yang Anda kirim. Misalnya, Anda dapat mengirim metrik email ke tujuan Amazon Data Firehose, lalu menganalisisnya menggunakan Amazon Managed Service for Apache Flink. Atau, Anda dapat mengirim informasi bouncing dan keluhan ke Amazon SNS dan segera menerima pemberitahuan saat peristiwa tersebut terjadi.
- Manajemen kumpulan IP - Jika Anda menyewakan alamat IP khusus untuk digunakan dengan AmazonSES, Anda dapat membuat grup alamat ini yang disebut kumpulan IP khusus untuk digunakan untuk mengirim jenis email tertentu. Misalnya, Anda dapat mengaitkan kumpulan IP khusus ini dengan set konfigurasi dan menggunakannya untuk mengirim komunikasi pemasaran, dan satu lagi untuk mengirim email transaksional. Reputasi pengirim Anda untuk email transaksional kemudian diisolasi dari email pemasaran Anda.

Untuk mengaitkan set konfigurasi dengan identitas terverifikasi dapat dilakukan dengan cara berikut:

- Sertakan referensi ke konfigurasi yang ditetapkan di header email. Untuk informasi selengkapnya tentang menentukan set konfigurasi di email Anda, lihat Menentukan set konfigurasi ketika Anda mengirim email.
- Tentukan set konfigurasi yang ada untuk digunakan sebagai set konfigurasi default identitas, baik pada saat pembuatan identitas, atau nanti saat mengedit identitas terverifikasi. Lihat Memahami set konfigurasi default.

Daftar Isi

- Membuat set konfigurasi di SES
- Mengelola set konfigurasi di Amazon SES

- Menentukan set konfigurasi ketika Anda mengirim email
- Melihat dan mengekspor metrik reputasi

Membuat set konfigurasi di SES

Anda dapat menggunakan SES konsol, `CreateConfigurationSet` tindakan di Amazon SES API v2, atau `aws sesv2 create-configuration-set` perintah di Amazon SES CLI v2 untuk membuat set konfigurasi baru. Bagian ini menunjukkan cara membuat set konfigurasi menggunakan SES konsol dan Amazon SES CLI v2.

Buat set konfigurasi (konsol)

Untuk membuat set konfigurasi menggunakan SES konsol, ikuti langkah-langkah ini:

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Set konfigurasi.
3. Pilih Buat set.
4. Masukkan detail berikut di bagian Detail umum:

   - Nama set konfigurasi - Nama untuk set konfigurasi Anda. Nama dapat berisi hingga 64 karakter alfanumerik, termasuk huruf, angka, tanda hubung (-) dan garis bawah (_) saja.
   - Mengirim kumpulan IP — Saat Anda mengirim email menggunakan set konfigurasi ini, pesan dikirim dari alamat IP khusus di kumpulan yang ditetapkan. Pilih kolam IP dari daftar.

     > **Note**
     >
     > Default (ses-default-dedicated-pool) berisi alamat IP khusus yang belum ditetapkan ke kumpulan lain. Untuk mempelajari selengkapnya tentang cara mengelola kolam IP, lihat Tetapkan kolam IP.

   - Opsi pelacakan
     - Gunakan domain pengalihan kustom — Pilih kotak centang untuk menggunakan domain pengalihan kustom untuk menangani open dan klik tracking untuk email yang dikirim dengan set konfigurasi ini.

- Domain pengalihan khusus - Pilih domain terverifikasi dari daftar Pilih domain terverifikasi untuk menjadi domain pengalihan kustom Anda. Anda juga dapat memasukkan subdomain di bidang Enter a subdomain.

  > **Note**
  >
  > Domain pengalihan kustom dapat ditentukan sebagai berikut:
  >
  > - Anda harus terlebih dahulu membuat dan memverifikasi domain pengalihan kustom di email yang ingin Wilayah AWS Anda kirim dan lacak, serta menyiapkan Jaringan Pengiriman Konten (CDN). Hal ini dijelaskan dalamMengonfigurasi domain kustom untuk menangani pelacakan buka dan klik.
  > - Kemudian, untuk menggunakan domain pengalihan khusus Anda untuk membuka dan mengklik pelacakan, Anda harus menunjukkannya saat membuat atau mengedit konfigurasi yang ditetapkan di sini pada langkah ini.
  > - Terakhir, setelah menentukan domain pengalihan kustom Anda, Lihat DNS catatan akan muncul di wadah Detail umum set konfigurasi. Jika Anda memperluasnya, Anda akan melihat CNAME catatan yang berisi domain pelacakan yang digunakan dalam domain Anda Wilayah AWS. Misalnya, jika subdomain kustom Anda disebut marketing.example.com dan dibuat di, memperluas DNScatatan Tampilan akan menampilkan CNAME catatan dengan nilai berikut: Name = marketing.example.com dan Value = r.us-east-1.awstrack.me. Wilayah AWS ***`us-east-1`***
  >
  >   Anda dapat menggunakan informasi ini hanya sebagai konfirmasi bahwa Anda memilih domain pelacakan yang benar dari tabel saat Anda mengatur CDN seperti yang dijelaskan diMengonfigurasi domain kustom untuk menangani pelacakan buka dan klik, atau Anda dapat melakukannya terlebih dahulu, dan menggunakan nilai CNAME catatan dari sini untuk digunakan dalam CDN pengaturan Anda.

- HTTPSkebijakan — Pilih opsi HTTPS kebijakan untuk protokol tautan pelacakan terbuka dan klik untuk domain pengalihan khusus Anda:
  - Opsional - (Perilaku default) Buka tautan pelacakan akan dibungkus menggunakanHTTP. Klik tautan pelacakan akan dibungkus menggunakan protokol asli tautan.
  - Diperlukan - Buka dan Klik tautan pelacakan keduanya akan dibungkus menggunakanHTTPS.

- Diperlukan untuk membuka - Tautan pelacakan terbuka akan dibungkus menggunakanHTTPS. Klik tautan pelacakan akan dibungkus menggunakan protokol asli tautan.

- Opsi pengiriman lanjutan - Pilih panah di sebelah kiri untuk memperluas bagian opsi pengiriman lanjutan.
  - Transport Layer Security (TLS) — SES Untuk membuat koneksi aman dengan server email penerima, dan mengirim email menggunakan TLS protokol, pilih kotak centang Diperlukan.

> **Note**
>
> SESmendukung TLS 1.2 dan merekomendasikan TLS 1.3. Untuk mempelajari selengkapnya, lihat Keamanan infrastruktur dalam SES.

5. Masukkan detail berikut di bagian Opsi reputasi:

   - Metrik reputasi — Digunakan untuk melacak metrik bouncing dan keluhan CloudWatch untuk email yang dikirim menggunakan set konfigurasi ini. (Biaya tambahan berlaku, lihat Harga per metrik untuk CloudWatch.)
     - Diaktifkan - Pilih kotak centang ini untuk mengaktifkan metrik reputasi untuk set konfigurasi.

6. Bagian Opsi daftar Supresi menyediakan set keputusan untuk menentukan penekanan khusus yang dimulai dengan opsi untuk menggunakan set konfigurasi ini untuk mengganti penekanan tingkat akun Anda. Peta logika penekanan set-level konfigurasi akan membantu Anda memahami efek dari kombinasi override. Pilihan penggantian multitier ini dapat digabungkan untuk menerapkan tiga tingkat penekanan yang berbeda:

   a. Gunakan penekanan tingkat akun: Jangan mengesampingkan penekanan tingkat akun Anda dan jangan menerapkan penekanan set-level konfigurasi apa pun - pada dasarnya, email apa pun yang dikirim menggunakan set konfigurasi ini hanya akan menggunakan penekanan tingkat akun Anda. Untuk melakukannya:

      - Di Pengaturan daftar Supresi, hapus centang pada kotak Ganti pengaturan tingkat akun.

   b. Jangan gunakan penekanan apa pun: Ganti penekanan tingkat akun Anda tanpa mengaktifkan penekanan set-level konfigurasi apa pun - ini berarti email apa pun yang dikirim menggunakan set konfigurasi ini tidak akan menggunakan penekanan tingkat akun Anda; dengan kata lain, semua penekanan dibatalkan. Untuk melakukannya:

i. Di Pengaturan daftar Supresi, centang kotak Ganti pengaturan tingkat akun.

ii. Dalam daftar Suppression, hapus centang pada kotak Diaktifkan.

c. Gunakan penekanan set-level konfigurasi: Ganti penekanan tingkat akun Anda dengan pengaturan daftar penekanan khusus yang ditentukan dalam set konfigurasi ini - ini berarti email apa pun yang dikirim menggunakan set konfigurasi ini hanya akan menggunakan pengaturan penindasan sendiri dan mengabaikan pengaturan penekanan tingkat akun apa pun. Untuk melakukannya:

i. Di Pengaturan daftar Supresi, centang kotak Ganti pengaturan tingkat akun.

ii. Dalam daftar Suppression, centang Diaktifkan.

iii. Di Tentukan alasannya... , pilih salah satu alasan penindasan untuk konfigurasi ini yang akan digunakan.

7.

Bagian opsi Virtual Deliverability Manager menyediakan cara bagi Anda untuk menentukan pengaturan khusus tentang bagaimana set konfigurasi ini akan menggunakan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan dengan mengesampingkan bagaimana pengaturan tersebut telah ditentukan dalam pengaturan Virtual Deliverability Manager di tingkat akun:

a. Untuk menonaktifkan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan untuk set konfigurasi ini:

i. Centang kotak Ganti pengaturan level akun.

ii. Pastikan Diaktifkan tidak dicentang untuk pelacakan Keterlibatan dan pengiriman bersama yang dioptimalkan, lalu pilih Simpan perubahan.

b. Untuk mengaktifkan atau menonaktifkan salah satu, atau keduanya, pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan untuk set konfigurasi ini:

i. Centang kotak Ganti pengaturan level akun.

ii. Centang atau hapus centang Diaktifkan untuk salah satu atau keduanya Pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan, lalu pilih Simpan perubahan.

c. Untuk kembali ke pengaturan tingkat akun Virtual Deliverability Manager untuk pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan untuk set konfigurasi ini:

- Hapus centang pada kotak Ganti pengaturan level akun, lalu pilih Simpan perubahan.

8. Anda dapat secara opsional menambahkan satu tanda atau lebih di bagian Tanda. Ulangi langkah-langkah berikut untuk setiap tanda yang ingin Anda tambahkan ke set konfigurasi Anda.

   a. Pilih Tambahkan tanda baru.

   b. Masukkan tanda Kunci.

   c. Masukkan tanda Nilai (opsional).

   Untuk menghapus tanda yang Anda masukkan, pilih Hapus untuk tanda tersebut. Anda bisa memasukkan maksimum 50 tanda.

9. Pilih Buat set untuk membuat set konfigurasi Anda.

Sekarang setelah Anda membuat set konfigurasi, Anda memiliki opsi untuk menentukan tujuan acara untuk set konfigurasi Anda yang memungkinkan penerbitan acara yang dipicu pada jenis acara yang Anda tentukan untuk tujuan acara. Set konfigurasi dapat memiliki beberapa tujuan acara dengan beberapa jenis acara yang ditentukan. Lihat Membuat tujuan SES acara Amazon.

Buat set konfigurasi. (AWS CLI)

Anda dapat membuat set konfigurasi menggunakan JSON file sebagai input ke `aws sesv2 create-configuration-set` perintah di file AWS CLI.

1. Buat JSON file CLI masukan

   Gunakan alat pengeditan file favorit Anda untuk membuat JSON file dengan kunci berikut, ditambah nilai yang valid untuk lingkungan Anda, atau gunakan `aws sesv2 create-configuration-set` perintah SES API v2 dengan `--generate-cli-skeleton` opsi tanpa nilai yang ditentukan untuk mencetak JSON struktur sampel ke output standar.

   Contoh ini menggunakan file bernama `create-configuration-set.json`:

```
{
    "ConfigurationSetName": "sample-configuration-set",
    "TrackingOptions": {
        "CustomRedirectDomain": "some.domain.com"
    },
    "DeliveryOptions": {
        "TlsPolicy": "REQUIRE",
        "SendingPoolName": "sending pool"
    },
```

```
    "ReputationOptions": {
        "ReputationMetricsEnabled": true,
        "LastFreshStart": timestamp
    },
    "SendingOptions": {
        "SendingEnabled": true
    },
    "Tags": [
        {
            "Key": "tag key",
            "Value": "tag value"
        }
    ],
    "SuppressionOptions": {
        "SuppressedReasons": ["BOUNCE","COMPLAINT"]
    }
}
```

> **Note**
>
> - Anda harus menyertakan `file://` notasi di awal jalur JSON file.
> - Jalur untuk JSON file harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws sesv2 create-configuration-set --cli-input-json file://create-configuration-set.json
```

> **Note**
>
> Untuk meninjau AWS CLI referensi untuk perintah ini, lihat create-configuration-set.

Mengelola set konfigurasi di Amazon SES

Setelah membuat set konfigurasi, Anda dapat mengelolanya dengan opsi tampilan, edit, dan hapus menggunakan SES konsol, Amazon SES API v2, dan Amazon SES CLI v2. Set konfigurasi juga dapat ditetapkan ke identitas terverifikasi sebagai set konfigurasi default yang diterapkan setiap kali email dikirim dari identitas.

Topik di bagian ini:

- Lihat, edit, & hapus set konfigurasi (konsol)
- Daftarkan set konfigurasi (AWS CLI)
- Dapatkan detail set konfigurasi (AWS CLI)
- Hapus set konfigurasi (AWS CLI)
- Hentikan pengiriman email dari set konfigurasi (AWS CLI)
- Memahami set konfigurasi default
- Membuat tujuan SES acara Amazon
- Menetapkan kolam IP di Amazon SES
- Mengonfigurasi domain kustom untuk menangani pelacakan buka dan klik

Lihat, edit, & hapus set konfigurasi (konsol)

Akses halaman detail set konfigurasi yang ada

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Set konfigurasi.
3. Untuk melihat detail set konfigurasi, pilih Nama dari daftar set konfigurasi. Langkah ini akan membawa Anda ke halaman detail.

Halaman detail Set konfigurasi memiliki dua tab untuk detail set konfigurasi dengan panel di setiap tab tempat Anda dapat melihat, mengedit, atau menghapus berikut ini:

- Tab Ikhtisar
  - Detail umum - panel ini menunjukkan detail umum untuk set konfigurasi:
    - Mengirim status (apakah saat ini diaktifkan)

  - Nama set konfigurasi
  - Mengirim kolam IP
  - Keamanan Lapisan Transportasi (TLS)
  - Domain pengalihan kustom
- Opsi reputasi— panel ini menampilkan detail yang terkait dengan reputasi pengiriman Anda:
  - Metrik reputasi (menunjukkan apakah Anda sedang melacak metrik)
  - Awal baru terakhir (tanggal dan waktu di mana metrik reputasi untuk set konfigurasi terakhir diatur ulang)
- Opsi daftar penindasan - panel ini menunjukkan jika Anda mengganti daftar penekanan tingkat akun Anda dengan set konfigurasi, dan jika demikian, detail penimpaannya:
  - Pengaturan daftar penindasan (menunjukkan pengaturan tingkat akun yang mengesampingkan—jika tidak, ini adalah satu-satunya item yang ditampilkan di panel)
  - Daftar penindasan (menunjukkan bagaimana Anda mengganti pengaturan tingkat akun Anda —baik dengan daftar penekanan diaktifkan atau dinonaktifkan)
  - Alasan penindasan (menunjukkan jika pantulan dan/atau keluhan adalah alasan untuk menambahkan alamat email penerima ke daftar penindasan Anda)
- Opsi Virtual Deliverability Manager — panel ini menunjukkan jika Anda mengganti pengaturan akun Virtual Deliverability Manager untuk pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan dengan set konfigurasi, dan jika demikian, detail penimpaannya:
  - Pelacakan keterlibatan (menunjukkan apakah pelacakan keterlibatan diaktifkan atau dinonaktifkan)
  - Pengiriman bersama yang dioptimalkan (menunjukkan apakah pengiriman bersama yang dioptimalkan diaktifkan atau dinonaktifkan)
- Tanda — panel ini menampilkan semua tanda yang telah Anda lampirkan pada set konfigurasi.
  - Kunci
  - Nilai

Anda dapat melakukan tindakan berikut dari panel ini:

- Pilih tombol Edit, atau dalam kasus panel Tanda, tombol Kelola tanda untuk mengedit detail masing-masing panel.
- Untuk informasi selengkapnya tentang bidang, lihat bagian terkait di langkah Buat set konfigurasi (konsol).

> **Tip**
>
> Ingatlah untuk Simpan perubahan setelah selesai mengedit. Pilih Batalkan untuk kembali ke halaman detail set konfigurasi tanpa menyimpan.

- Tab Tujuan peristiwa
  - Semua destinasi (***count of event destinations***) - panel ini mencantumkan semua tujuan acara yang telah Anda masukkan untuk set konfigurasi Anda. Untuk setiap tujuan, Anda dapat melihat:
    - Nama
    - Tujuan
    - Jenis acara
    - Penerbitan acara

  Anda dapat melakukan tindakan berikut dari panel ini:
  - Tambahkan tujuan acara baru dengan memilih tombol Tambah tujuan. Untuk informasi selengkapnya tentang menambahkan tujuan peristiwa, lihat Membuat tujuan acara.
  - Ubah tujuan acara yang ada dengan memilih namanya yang akan membuka layar edit.
  - Hapus tujuan acara yang ada dengan memilih kotak centang di samping namanya lalu pilih tombol Hapus.

Di bagian atas setiap halaman detail set konfigurasi, dan yang terlihat dari tab Gambaran umum atau Tujuan peristiwa, adalah opsi berikut:

- Hapus — tombol ini akan menghapus set konfigurasi Anda.
- Menonaktifkan pengiriman — tombol ini akan berhenti mengirim email dari set konfigurasi Anda.

Daftarkan set konfigurasi (AWS CLI)

Anda dapat menggunakan list-configuration-sets perintah di AWS CLI untuk menghasilkan daftar semua set konfigurasi yang terkait dengan akun Anda di Wilayah saat ini, sebagai berikut:

```
aws sesv2 list-configuration-sets
```

Dapatkan detail set konfigurasi (AWS CLI)

Anda dapat menggunakan get-configuration-set perintah di AWS CLI untuk mendapatkan rincian untuk set konfigurasi tertentu, sebagai berikut:

```
aws sesv2 get-configuration-set --configuration-set-name name
```

Hapus set konfigurasi (AWS CLI)

Anda dapat menggunakan delete-configuration-set perintah di AWS CLI untuk menghapus set konfigurasi tertentu, sebagai berikut:

```
aws sesv2 delete-configuration-set --configuration-set-name name
```

Hentikan pengiriman email dari set konfigurasi (AWS CLI)

Anda dapat menggunakan put-configuration-set-sending-options perintah di AWS CLI untuk berhenti mengirim email dari set konfigurasi tertentu, sebagai berikut:

```
aws sesv2 put-configuration-set-sending-options --configuration-set-name name --no-
sending-enabled
```

Untuk mulai mengirim lagi, jalankan perintah yang sama dengan opsi `--sending-enabled` sebagai gantinya, seperti berikut:

```
aws sesv2 put-configuration-set-sending-options --configuration-set-name name --
sending-enabled
```

Memahami set konfigurasi default

Konsep menetapkan set konfigurasi sebagai default yang akan digunakan oleh identitas terverifikasi dijelaskan di bagian ini untuk membantu memahami manfaat dan kasus penggunaan.

Set konfigurasi default secara otomatis menerapkan aturannya ke semua pesan yang Anda kirim dari identitas email yang terkait dengan set konfigurasi tersebut. Anda dapat menerapkan set konfigurasi default ke alamat email dan identitas domain selama pembuatan identitas atau setelah fakta sebagai fungsi edit dari identitas yang ada.

Konfigurasi default mengatur pertimbangan

- Set konfigurasi harus dibuat terlebih dahulu sebelum mengaitkannya dengan identitas.
- Set konfigurasi default hanya akan diterapkan jika identitas diverifikasi.
- Identitas email dapat dikaitkan dengan hanya satu set konfigurasi dalam satu waktu. Namun, Anda dapat menerapkan set konfigurasi yang sama ke beberapa identitas.
- Set konfigurasi default pada tingkat alamat email akan menimpa set konfigurasi default pada tingkat domain. Sebagai contoh, set konfigurasi default yang terkait dengan joe@example.com menimpa set konfigurasi untuk domain dari example.com.
- Set konfigurasi default pada tingkat domain berlaku untuk semua alamat email untuk domain tersebut (kecuali jika Anda memverifikasi alamat tertentu untuk domain tersebut).
- Jika Anda menghapus set konfigurasi yang ditetapkan sebagai konfigurasi default yang ditetapkan untuk identitas, dan kemudian mencoba mengirim email melalui identitas itu, panggilan Anda ke Amazon SES gagal dengan kesalahan “permintaan buruk”.
- Set konfigurasi default tidak dapat ditetapkan ke identitas terverifikasi yang digunakan oleh pengirim delegasi.
- Cara menentukan set konfigurasi yang ada untuk digunakan sebagai set konfigurasi default identitas sebenarnya adalah fungsi dari identitas yang diverifikasi, sehingga instruksi diberikan dalam alur kerja identitas yang sesuai:
  - Tentukan set konfigurasi default selama pembuatan identitas - ikuti petunjuk yang diberikan dalam Langkah 6 opsional untuk set konfigurasi default identitas Domain atau set konfigurasi default identitas Email yang terletak di bagian Membuat dan memverifikasi identitas di Amazon SES ini.
  - Tentukan konfigurasi default yang ditetapkan untuk identitas yang ada — ikuti langkah-langkah Mengedit identitas menggunakan konsol bersama dengan rincian berikut untuk Langkah 5:
    a. Pilih tab Set Konfigurasi.
    b. Pilih Edit dalam kontainer set konfigurasi Default.
    c. Pilih kotak daftar dan pilih set konfigurasi yang ada untuk digunakan sebagai default.
    d. Lanjutkan dengan langkah-langkah yang tersisaMengedit identitas menggunakan konsol.

> **Note**
>
> Jika konfigurasi yang ditetapkan sebagai default mengaktifkan metrik reputasi, biaya tambahan akan dikenakan untuk setiap email yang dikirim menggunakan set konfigurasi default, lihat Harga per metrik untuk. CloudWatch

Membuat tujuan SES acara Amazon

Tujuan acara memungkinkan Anda mempublikasikan tindakan pelacakan email keluar berikut ke AWS layanan lain untuk pemantauan:

- Mengirim
- Kegagalan rendering
- Menolak
- Pengiriman
- pentalan keras
- Aduan
- Penundaan penyampaian
- Langganan
- Membuka
- Klik

Untuk mempelajari selengkapnya tentang meyiapkan publikasi peristiwa, lihat the section called “Pantau pengiriman email menggunakan penerbitan acara”.

Membuat tujuan acara

Setelah membuat set konfigurasi, Anda memiliki opsi untuk membuat tujuan acara untuk set konfigurasi Anda yang memungkinkan penerbitan acara yang dipicu pada jenis acara yang Anda tentukan untuk tujuan acara. Set konfigurasi dapat memiliki beberapa tujuan acara dengan beberapa jenis acara yang ditentukan.

Jika Anda belum membuat set konfigurasi, lihatthe section called “Buat set konfigurasi”.

Langkah-langkah berikut menunjukkan cara membuat atau menambahkan tujuan acara ke set konfigurasi.

Untuk membuat atau menambah dan tujuan acara menggunakan SES konsol:

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Set konfigurasi.

3. Pilih nama set konfigurasi dari kolom Nama untuk mengakses detailnya.
4. Pilih tab Tujuan acara.
5. Pilih Tambahkan tujuan.
6. Pilih tipe peristiwa

   Peristiwa pengiriman email adalah metrik yang berkaitan dengan aktivitas pengiriman yang dapat Anda ukur menggunakan AmazonSES. Pada langkah ini, Anda memilih jenis acara pengiriman email yang Anda ingin Amazon publikasikan SES ke tujuan acara Anda.

   Untuk mempelajari selengkapnya tentang tipe peristiwa, lihat Memantau aktivitas pengiriman Amazon SES.

   a. Pilih Tipe peristiwa untuk dipublikasikan

      - Pengiriman dan penyampaian – untuk memilih tipe peristiwa untuk dipublikasikan, pilih kotak centang masing-masing, atau pilih Centang semua untuk mempublikasikan semua tipe peristiwa.

        Jenis peristiwa

        - Mengirim — permintaan kirim berhasil dan Amazon SES akan mencoba mengirimkan pesan ke server email penerima.
        - Kegagalan Rendering – Email tidak terkirim dikarenakan masalah rendering templat. Tipe peristiwa ini dapat terjadi saat data templat tidak ada, atau jika ada ketidakcocokan antara parameter templat dan data. (Jenis peristiwa ini hanya terjadi ketika Anda mengirim email menggunakan `SendTemplatedEmail`atau `SendBulkTemplatedEmail`APIoperasi.)
        - Menolak — Amazon SES menerima email tersebut, tetapi memutuskan bahwa email tersebut berisi virus dan tidak berusaha mengirimkannya ke server email penerima.
        - Pengiriman - Amazon SES berhasil mengirimkan email ke server email penerima.
        - Pentalan keras – server surat penerima menolak email secara permanen. (Pantulan lunak hanya disertakan ketika Amazon SES gagal mengirimkan email setelah mencoba lagi untuk jangka waktu tertentu.)
        - Aduan – email berhasil disampaikan ke server email penerima, namun penerima menandainya sebagai spam.
        - Penundaan penyampaian – Email tidak dapat dikirim ke server surat penerima karena terjadi masalah sementara. Penundaan penyamapaian dapat terjadi, misalnya, saat

kotak masuk penerima penuh, atau saat server email penerima mengalami masalah sementara. (Jenis acara ini tidak didukung oleh Amazon Pinpoint.)

- Langganan – email berhasil diantarkan, namun penerima memperbarui preferensi langganan dengan mengeklik `List-Unsubscribe` di header email atau tautan `Unsubscribe` di footer. (Jenis acara ini tidak didukung oleh Amazon Pinpoint.)

- Pelacakan buka dan klik — untuk mengukur keterlibatan pelanggan, pilih salah satu atau kedua kotak centang untuk melacak Membuka dan Mengeklik.
  - Pembukaan – Penerima telah menerima pesan dan membukanya di klien email mereka.
  - Klik – penerima mengeklik satu atau beberapa tautan dalam email.

> **Note**
>
> Buka dan klik penerbitan acara yang ditentukan di sini, atau dalam set konfigurasi lainnya, tidak memengaruhi opsi pelacakan keterlibatan untuk dasbor Virtual Deliverability Manager; ini ditentukan melalui pengaturan akun Virtual Deliverability Manager atau penggantian set konfigurasi. Misalnya, jika pelacakan keterlibatan Anda dinonaktifkan melalui Virtual Deliverability Manager, itu tidak akan menonaktifkan publikasi acara buka dan klik yang telah Anda siapkan di sini di tujuan SES acara.

  - Konfigurasi mengatur domain pengalihan — bidang ini akan muncul dan diisi sebelumnya dengan nama domain pengalihan kustom jika Anda menetapkan satu saat membuat set konfigurasi.

> **Note**
>
> Anda dapat memperbarui domain pengalihan kustom dalam konfigurasi yang ditetapkan untuk membuka dan mengklik pelacakan di bawah domain tersebut —lihat Opsi pelacakan di Langkah 4 dari. Buat set konfigurasi Untuk informasi selengkapnya tentang mengonfigurasi domain buka dan klik kustom, lihat Mengonfigurasi domain kustom untuk menangani pelacakan buka dan klik.

b. Pilih Next untuk melanjutkan.

7. Tentukan tujuan

   Tujuan acara adalah AWS layanan tempat acara pengiriman email dapat dipublikasikan. Memilih tujuan yang sesuai bergantung pada tingkat detail yang ingin Anda dapatkan dan bagaimana Anda ingin menerima data tersebut.

   a. Opsi tujuan

      - Jenis tujuan - ketika Anda memilih tombol radio di sebelah AWS layanan untuk mempublikasikan acara Anda, panel detail akan muncul dengan bidang yang terkait dengan layanan. Dengan memilih tautan di bawah ini akan memberikan petunjuk tentang panel detail layanan:
        - Amazon CloudWatch (Biaya tambahan berlaku, lihat Harga per metrik untuk CloudWatch.)
        - Amazon Data Firehose
        - Amazon EventBridge
        - Amazon Pinpoint (Tidak mendukung penundaan Pengiriman atau jenis acara Langganan.)
        - Amazon SNS

        Untuk mempelajari selengkapnya tentang menggunakan model publikasi peristiwa untuk memantau operasi email, lihat Pantau pengiriman email menggunakan penerbitan SES acara Amazon.
      - Nama – masukkan nama tujuan untuk set konfigurasi ini. Nama tersebut dapat menggunakan huruf, angka, tanda pisah, dan tanda hubung.
      - Publikasi peristiwa— untuk mengaktifkan publikasi peristiwa untuk tujuan ini, beri tanda centang pada kotak centang Diaktifkan.

   b. Pilih Next untuk melanjutkan.

8. Ulasan

   Ketika Anda puas bahwa entri Anda benar, pilih Tambahkan tujuan untuk menambahkan tujuan peristiwa Anda.

Anda juga dapat membuat tujuan acara menggunakan SES konsol Amazon, Amazon SES API v2, atau Amazon SES CLI v2.

Untuk membuat tujuan acara menggunakan SESAPI:

- Untuk membuat tujuan acara menggunakan SESAPI, lihat `CreateConfigurationSetEventDestination`.

Mengedit, menonaktifkan/mengaktifkan, atau menghapus tujuan acara

Ikuti langkah-langkah berikut untuk mengedit, menonaktifkan/mengaktifkan, atau menghapus tujuan acara menggunakan konsol: SES

Untuk mengedit, menonaktifkan/mengaktifkan, atau menghapus tujuan acara menggunakan konsol: SES

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Set konfigurasi.
3. Pilih nama set konfigurasi dari kolom Nama untuk mengakses detailnya.
4. Pilih tab Tujuan acara set konfigurasi.
5. Pilih nama tujuan acara di bawah kolom Nama.
6. - Untuk mengedit - Pilih tombol Edit pada panel masing-masing untuk kumpulan bidang yang ingin Anda edit dan buat perubahan diikuti oleh Simpan perubahan.
   - Untuk menonaktifkan atau mengaktifkan — Pilih tombol yang berlabel Nonaktifkan atau Aktifkan di sudut kanan atas.
   - Untuk menghapus - Pilih tombol Hapus di sudut kanan atas.

Anda juga dapat mengedit, menonaktifkan/mengaktifkan, atau menghapus tujuan acara menggunakan konsol AmazonSES, Amazon v2, atau Amazon SES API v2. SES CLI

Untuk mengedit, menonaktifkan/mengaktifkan, atau menghapus tujuan acara menggunakan: SES API

1. Untuk menonaktifkan/mengaktifkan tujuan acara menggunakan, lihat. SES API `UpdateConfigurationSetEventDestination`
2. Untuk menghapus tujuan acara menggunakan SESAPI, lihat `DeleteConfigurationSetEventDestination`.

Menetapkan kolam IP di Amazon SES

Anda dapat menggunakan kolam IP untuk membuat grup alamat IP khusus untuk mengirim tipe email tertentu. Anda juga dapat menggunakan kumpulan alamat IP yang dibagikan oleh semua SES pelanggan Amazon.

Saat menetapkan kumpulan IP ke set konfigurasi, Anda dapat memilih dari opsi berikut:

- Kolam IP khusus tertentu – Ketika Anda memilih kolam IP khusus yang telah ada, email yang menggunakan set konfigurasi dikirim hanya dengan menggunakan alamat IP khusus yang termasuk dalam kolam tersebut. Untuk prosedur tentang cara membuat:
  - kolam IP standar baru, lihatMembuat kolam IP khusus standar untuk IP khusus (standar).
  - kumpulan IP terkelola baru, lihatMembuat kumpulan IP terkelola untuk mengaktifkan IP khusus (dikelola).
- ses-default-dedicated-pool— Pool ini berisi semua alamat IP khusus untuk akun Anda yang belum termasuk dalam kumpulan IP. Jika Anda mengirim email menggunakan set konfigurasi yang tidak terkait dengan kumpulan, atau jika Anda mengirim email tanpa menentukan set konfigurasi sama sekali, email akan dikirim dari salah satu alamat di kumpulan default ini. Pool ini dikelola secara otomatis oleh SES dan tidak dapat diedit.
- ses-shared-pool— Kumpulan ini berisi sekumpulan besar alamat IP yang dibagikan di antara semua SES pelanggan Amazon. Opsi ini mungkin berguna ketika Anda perlu mengirim email yang tidak sesuai dengan perilaku pengiriman biasa Anda.

Menetapkan kolam IP untuk set konfigurasi

Bagian ini mereferensikan prosedur untuk menetapkan dan memodifikasi kumpulan IP dalam set konfigurasi menggunakan konsol AmazonSES.

- Untuk menetapkan kolam IP untuk set konfigurasi...
  - saat membuat set konfigurasi baru – lihat Kolam IP pengiriman di Langkah 4 dari Buat set konfigurasi
  - saat memodifikasi set konfigurasi yang ada - pilih tombol Edit di panel Detail umum dari set konfigurasi yang dipilih, dan ikuti petunjuk untuk Mengirim kumpulan IP di Langkah 4 Buat set konfigurasi

Mengonfigurasi domain kustom untuk menangani pelacakan buka dan klik

Saat Anda menggunakan penerbitan acara untuk menangkap peristiwa terbuka dan klik, Amazon SES membuat perubahan kecil pada email yang Anda kirim. Untuk menangkap peristiwa terbuka, SES tambahkan GIF gambar transparan 1 piksel kali 1 piksel di setiap email yang dikirim SES yang menyertakan nama file unik untuk setiap email, dan di-host di server yang dioperasikan olehSES; ketika gambar diunduh, SES dapat mengetahui dengan tepat pesan mana yang dibuka dan oleh siapa.

Secara default, piksel ini disisipkan di bagian bawah email; namun, beberapa aplikasi penyedia email memotong pratinjau email ketika melebihi ukuran tertentu dan dapat memberikan tautan untuk melihat sisa pesan. Dalam skenario ini, gambar pelacakan SES piksel tidak dimuat dan akan membuang tarif terbuka yang Anda coba lacak. Untuk menyiasatinya, Anda dapat secara opsional menempatkan piksel di awal email, atau di mana pun, dengan memasukkan `{{ses:openTracker}}` placeholder ke dalam badan email. Setelah SES menerima pesan dengan placeholder, itu akan diganti dengan gambar piksel pelacakan terbuka.

> **Important**
>
> Cukup tambahkan satu `{{ses:openTracker}}` placeholder, karena lebih dari satu akan menghasilkan kode `400 BadRequestException` kesalahan dikembalikan.

Untuk menangkap peristiwa klik tautan, SES ganti tautan di email Anda dengan tautan ke server yang dioperasikan olehSES. Tindakn ini segera mengalihkan penerima ke tujuan yang dimaksudkan.

Anda juga memiliki opsi untuk menggunakan domain Anda sendiri, bukan domain yang dimiliki dan dioperasikan olehSES, untuk menciptakan pengalaman yang lebih kohesif bagi penerima Anda, yang berarti semua indikator dihapus. SES Anda dapat mengonfigurasi beberapa domain kustom untuk menangani peristiwa pelacakan buka dan klik. Domain kustom ini terkait dengan set konfigurasi. Ketika Anda mengirim email menggunakan set konfigurasi, jika set konfigurasi tersebut dikonfigurasi untuk menggunakan domain kustom, maka tautan buka dan klik di email tersebut secara otomatis menggunakan domain kustom yang ditentukan di set konfigurasi.

Bagian ini berisi prosedur untuk menyiapkan subdomain di server yang Anda miliki untuk secara otomatis mengarahkan pengguna ke server pelacakan terbuka dan klik yang dioperasikan oleh. SES Ada tiga langkah yang terlibat dalam pengaturan domain ini. Pertama, Anda mengonfigurasi subdomain itu sendiri, lalu mengatur konfigurasi yang ditetapkan untuk menggunakan domain

kustom, dan kemudian mengatur tujuan acara untuk mempublikasikan peristiwa terbuka dan klik. Topik ini berisi prosedur untuk menyelesaikan semua langkah ini.

Namun, jika Anda hanya ingin mengaktifkan pelacakan terbuka atau klik tanpa menyiapkan domain khusus, Anda dapat melanjutkan langsung ke menentukan tujuan acara untuk set konfigurasi Anda yang memungkinkan penerbitan acara yang dipicu pada jenis acara yang Anda tentukan, termasuk acara buka dan klik. Set konfigurasi dapat memiliki beberapa tujuan acara dengan beberapa jenis acara yang ditentukan. Lihat Membuat tujuan SES acara Amazon.

Bagian 1: Menyiapkan domain untuk menangani pengalihan tautan buka dan klik

Prosedur khusus untuk menyiapkan domain pengalihan bervariasi tergantung pada penyedia hosting web Anda (dan Jaringan Pengiriman Konten Anda, jika Anda menggunakan HTTPS server). Prosedur di bagian berikut menyediakan panduan umum bukannya langkah-langkah tertentu.

Opsi 1: Mengkonfigurasi domain HTTP

Jika Anda berencana untuk menggunakan HTTP domain untuk menangani tautan terbuka dan klik (sebagai lawan dari HTTPS domain), proses untuk mengonfigurasi subdomain hanya melibatkan beberapa langkah.

> **Note**
>
> Jika Anda menyiapkan domain kustom yang menggunakan HTTP protokol, dan Anda mengirim email yang berisi tautan yang menggunakan HTTPS protokol, pelanggan Anda mungkin akan melihat pesan peringatan ketika mereka mengklik tautan di email Anda. Jika Anda berencana untuk mengirim email yang berisi tautan yang menggunakan HTTPS protokol, Anda harus menggunakan HTTPS domain untuk menangani peristiwa pelacakan klik.

Untuk menyiapkan HTTP subdomain untuk menangani tautan terbuka dan klik

1. Buat subdomain yang akan digunakan untuk membuka dan mengklik tautan pelacakan. SESmerekomendasikan bahwa subdomain ini secara khusus didedikasikan untuk menangani tautan ini, dan subdomain dibuat untuk setiap Wilayah AWS Anda mengirim email yang ingin Anda lacak.
2. Verifikasi subdomain untuk digunakan denganSES. Untuk informasi selengkapnya, lihat Membuat identitas domain.

3. Tambahkan CNAME catatan baru ke DNS pengaturan subdomain Anda yang mengalihkan permintaan ke domain pelacakan. SES Alamat yang Anda alihkan harus sama Wilayah AWS dengan subdomain kustom Anda.

   Tabel berikut berisi daftar domain pelacakan Wilayah AWS tempat SES tersedia—pilih domain yang berada di wilayah yang sama dengan domain kustom Anda:

AWS Wilayah	AWS domain pelacakan
AS Timur (Ohio)	`r.us-east-2.awstrack.me`
AS Timur (Virginia Utara)	`r.us-east-1.awstrack.me`
AS Barat (California Utara)	`r.us-west-1.awstrack.me`
AS Barat (Oregon)	`r.us-west-2.awstrack.me`
Afrika (Cape Town)	`r.af-south-1.awstrack.me`
Asia Pasifik (Jakarta)	`r.ap-southeast-3.awstrack.me`
Asia Pasifik (Mumbai)	`r.ap-south-1.awstrack.me`
Asia Pasifik (Osaka)	`r.ap-northeast-3.awstrack.me`
Asia Pasifik (Seoul)	`r.ap-northeast-2.awstrack.me`
Asia Pasifik (Singapura)	`r.ap-southeast-1.awstrack.me`
Asia Pasifik (Sydney)	`r.ap-southeast-2.awstrack.me`
Asia Pasifik (Jakarta)	r.ap-tenggara 3.awstrack.me
Asia Pasifik (Jakarta)	r.ap-tenggara 3.awstrack.me
Asia Pasifik (Tokyo)	`r.ap-northeast-1.awstrack.me`
Kanada (Pusat)	`r.ca-central-1.awstrack.me`
Eropa (Frankfurt)	`r.eu-central-1.awstrack.me`

AWS Wilayah	AWS domain pelacakan
Eropa (Irlandia)	`r.eu-west-1.awstrack.me`
Eropa (London)	`r.eu-west-2.awstrack.me`
Eropa (Milan)	`r.eu-south-1.awstrack.me`
Eropa (Stockholm)	`r.eu-north-1.awstrack.me`
Israel (Tel Aviv)	`r.il-central-1.awstrack.me`
Timur Tengah (Bahrain)	`r.me-south-1.awstrack.me`
Amerika Selatan (Sao Paulo)	`r.sa-east-1.awstrack.me`
AWS GovCloud (AS-Barat)	`r.us-gov-west-1.awstrack.me`
AWS GovCloud (AS-Timur)	`r.us-gov-east-1.awstrack.me`

> **Note**
>
> Bergantung pada penyedia hosting web Anda, mungkin diperlukan beberapa menit agar perubahan yang Anda buat pada DNS catatan subdomain berlaku. Penyedia hosting web atau organisasi IT Anda dapat memberikan informasi tambahan tentang penundaan ini.

Opsi 2: Mengkonfigurasi domain HTTPS

Anda juga dapat menggunakan HTTPS domain untuk melacak klik terbuka dan tautan. Untuk menyiapkan HTTPS domain untuk melacak klik terbuka dan tautan, Anda harus melakukan beberapa langkah tambahan, di luar yang diperlukan untuk menyiapkan HTTP domain.

Untuk menyiapkan HTTPS subdomain untuk menangani tautan terbuka dan klik

1. Buat subdomain yang akan digunakan untuk membuka dan mengklik tautan pelacakan. SESmerekomendasikan bahwa subdomain ini secara khusus didedikasikan untuk menangani

tautan ini, dan subdomain dibuat untuk setiap Wilayah AWS Anda mengirim email yang ingin Anda lacak.

2. Verifikasi subdomain untuk digunakan denganSES. Untuk informasi selengkapnya, lihat Membuat identitas domain.
3. Membuat akun baru dengan Jaringan Pengiriman Konten (CDN), seperti Amazon CloudFront, lihat Memulai CloudFront distribusi dasar.
4. Konfigurasikan CDN ke asal yang merupakan domain SES pelacakan, seperti `r.us-east-1.awstrack.me` misalnya. CDNHarus menunjuk ke domain AWS pelacakan yang berada di wilayah yang sama dengan domain kustom Anda. CDNHarus meneruskan `Host` header yang disediakan oleh pemohon ke asal, lihat artikel AWS re:Post ini untuk informasi lebih lanjut.

   Tabel berikut berisi daftar domain pelacakan Wilayah AWS tempat SES tersedia—pilih domain yang berada di wilayah yang sama dengan domain kustom Anda:

AWS Wilayah	AWS domain pelacakan
AS Timur (Ohio)	`r.us-east-2.awstrack.me`
AS Timur (Virginia Utara)	`r.us-east-1.awstrack.me`
AS Barat (California Utara)	`r.us-west-1.awstrack.me`
AS Barat (Oregon)	`r.us-west-2.awstrack.me`
Afrika (Cape Town)	`r.af-south-1.awstrack.me`
Asia Pasifik (Jakarta)	`r.ap-southeast-3.awstrack.me`
Asia Pasifik (Mumbai)	`r.ap-south-1.awstrack.me`
Asia Pasifik (Osaka)	`r.ap-northeast-3.awstrack.me`
Asia Pasifik (Seoul)	`r.ap-northeast-2.awstrack.me`
Asia Pasifik (Singapura)	`r.ap-southeast-1.awstrack.me`
Asia Pasifik (Sydney)	`r.ap-southeast-2.awstrack.me`

AWS Wilayah	AWS domain pelacakan
Asia Pasifik (Tokyo)	`r.ap-northeast-1.awstrack.me`
Kanada (Pusat)	`r.ca-central-1.awstrack.me`
Eropa (Frankfurt)	`r.eu-central-1.awstrack.me`
Eropa (Irlandia)	`r.eu-west-1.awstrack.me`
Eropa (London)	`r.eu-west-2.awstrack.me`
Eropa (Milan)	`r.eu-south-1.awstrack.me`
Eropa (Stockholm)	`r.eu-north-1.awstrack.me`
Israel (Tel Aviv)	`r.il-central-1.awstrack.me`
Timur Tengah (Bahrain)	`r.me-south-1.awstrack.me`
Amerika Selatan (Sao Paulo)	`r.sa-east-1.awstrack.me`
AWS GovCloud (AS-Barat)	`r.us-gov-west-1.awstrack.me`
AWS GovCloud (AS-Timur)	`r.us-gov-east-1.awstrack.me`

5. Jika Anda menggunakan Route 53 untuk mengelola DNS konfigurasi domain Anda dan CloudFront sebagai milik AndaCDN, buat catatan Alias di Route 53 yang merujuk ke CloudFront distribusi Anda (seperti d111111abcdef8.cloudfront.net). Untuk informasi lebih lanjut, lihat Membuat Catatan Menggunakan Konsol Amazon Route 53 di Panduan Developer Amazon Route 53.

   Jika tidak, dalam DNS konfigurasi untuk subdomain Anda, tambahkan CNAME catatan yang merujuk ke alamat AndaCDN.

6. Dapatkan SSL sertifikat dari Otoritas Sertifikat tepercaya. Sertifikat harus mencakup subdomain yang Anda buat di langkah 1 serta yang CDN Anda konfigurasikan dalam langkah 3—5. Unggah sertifikat keCDN.

7. Anda dapat menggunakan perintah curl berikut untuk memvalidasi bahwa domain kustom Anda yang baru dibuat menggunakan wilayah dan HTTPS protokol yang benar. Dalam contoh berikut, semuanya adalah literal kecuali untuk nama domain Anda:

```
curl --head https://custom.domain.com/favicon.ico
```

Respons dikembalikan seperti pada contoh berikut:

```
(python-sdk-test) jdoe@12a34567b89c BaconRedirectService % curl --head https://
custom.domain.com/favicon.ico
HTTPS/1.1 200 OK
x-amz-ses-region: us-east-1
x-amz-ses-request-protocol: https
Content-Type: image/x-icon
Transfer-Encoding: chunked
Date: Fri, 30 Aug 2024 13:50:14 GMT
```

Tanggapan ini berisi properti berikut:

- Nilai `x-amz-request-region` header adalah SES wilayah yang menerima permintaan.
- Nilai `x-amz-request-protocol` header adalah protokol yang digunakan untuk permintaan antara CDN dan SES di header.

Jika pengaturan Anda benar, wilayah harus mencerminkan wilayah tempat domain Anda dibuat dan protokolnya seharusnyaHTTPS.

Bagian 2: Menentukan domain dan HTTPS kebijakan pengalihan kustom Anda melalui set konfigurasi

Setelah mengonfigurasi domain untuk menangani pengalihan pelacakan terbuka dan klik, Anda harus menentukan domain dan HTTPS kebijakan kustom Anda dalam set konfigurasi.

Saat Anda mengirim email menggunakan set konfigurasi, jika set konfigurasi tersebut dikonfigurasi untuk menggunakan domain pengalihan kustom, tautan buka dan klik di email tersebut secara otomatis menggunakan opsi domain dan HTTPS kebijakan khusus yang ditentukan dalam set konfigurasi.

Anda dapat menyelesaikan ini menggunakan SES konsol atau API operasi `CreateConfigurationSet`v2.

Untuk menentukan domain dan HTTPS kebijakan pengalihan kustom menggunakan konsol

- Saat membuat atau mengedit set konfigurasi, gunakan opsi Pelacakan di Langkah 4 Buat set konfigurasi untuk menentukan opsi domain dan HTTPS kebijakan pengalihan kustom Anda.

Untuk menentukan domain dan HTTPS kebijakan pengalihan kustom menggunakan AWS CLI

Anda dapat menggunakan `CreateConfigurationSet`operasi di SES API v2 dan menggunakan `TrackingOptions` properti untuk menentukan domain pengalihan kustom dan HTTPS kebijakan. Anda dapat memanggil operasi ini dari AWS CLI seperti yang ditunjukkan pada contoh berikut.

- Buat konfigurasi yang diatur di Wilayah AWS tempat Anda ingin mengirim dan melacak email:

```
aws sesv2 create-configuration-set --cli-input-json file://create.json
```

- Dalam contoh ini, file input menggunakan parameter `TrackingOptions`properti— `CustomRedirectDomain` menentukan domain kustom yang akan digunakan untuk melacak tautan terbuka dan klik, dan `HttpsPolicy` menentukan opsi HTTPS kebijakan:

```
{
    "ConfigurationSetName": "my-config-set",
    "TrackingOptions": {
        "CustomRedirectDomain": "marketing.example.com",
        "HttpsPolicy": "REQUIRE"
    },
    "SendingOptions": {
        "SendingEnabled": true
    }
}
```

  Untuk `HttpsPolicy` parameter, nilai berikut dapat ditentukan untuk mengatur protokol tautan pelacakan terbuka dan klik untuk domain pengalihan khusus Anda:

  - `OPTIONAL`— (Perilaku default) Buka tautan pelacakan akan dibungkus menggunakanHTTP. Klik tautan pelacakan akan dibungkus menggunakan protokol asli tautan.
  - `REQUIRE`— Buka dan Klik tautan pelacakan keduanya akan dibungkus menggunakanHTTPS.
  - `REQUIRE_OPEN_ONLY`— Tautan pelacakan terbuka akan dibungkus menggunakanHTTPS. Klik tautan pelacakan akan dibungkus menggunakan protokol asli tautan.

Bagian 3: Menentukan jenis acara terbuka dan klik melalui set konfigurasi

Setelah menentukan domain dan HTTPS kebijakan kustom Anda dalam konfigurasi yang ditetapkan pada langkah sebelumnya, Anda harus menentukan jenis peristiwa terbuka dan/atau klik untuk melacak tujuan peristiwa melalui set konfigurasi.

Anda dapat menyelesaikan ini menggunakan SES konsol atau API operasi `CreateConfigurationSetEventDestination`v2.

Untuk memilih jenis acara terbuka dan/atau klik menggunakan konsol

- Saat membuat atau memodifikasi tujuan acara, gunakan Buka dan klik pelacakan di Langkah 6 the section called “Membuat tujuan acara” untuk menentukan jenis acara.

Menentukan set konfigurasi ketika Anda mengirim email

Untuk menggunakan set konfigurasi saat mengirim email, Anda harus melewati nama set konfigurasi di header email. Semua metode pengiriman SES email Amazon—termasuk, SESSMTPantarmuka AWS CLI, dan Amazon—memungkinkan Anda meneruskan konfigurasi yang ditetapkan di header email yang Anda kirim. AWS SDKs

Jika Anda menggunakan SMTPantarmuka atau `SendRawEmail`APIoperasi, Anda dapat menentukan set konfigurasi dengan memasukkan header berikut di email Anda (mengganti *`ConfigSet`* dengan nama set konfigurasi yang ingin Anda gunakan):

```
X-SES-CONFIGURATION-SET: ConfigSet
```

Panduan ini mencakup contoh kode untuk mengirim email menggunakan AWS SDKs dan SES SMTP antarmuka Amazon. Masing-masing contoh ini mencakup metode penentuan set konfigurasi. Untuk melihat step-by-step prosedur pengiriman email yang menyertakan referensi ke set konfigurasi, lihat berikut ini:

- Mengirim email melalui Amazon SES menggunakan AWS SDK
- Menggunakan SES SMTP antarmuka Amazon untuk mengirim email

Melihat dan mengekspor metrik reputasi

Amazon SES secara otomatis mengekspor informasi tentang tingkat pentalan dan keluhan keseluruhan untuk seluruh akun Anda ke Amazon. CloudWatch Anda dapat menggunakan metrik ini untuk membuat alarm CloudWatch, atau menjeda pengiriman email secara otomatis menggunakan fungsi Lambda.

Anda juga dapat mengekspor metrik reputasi untuk CloudWatch set konfigurasi individual. Mengekspor data reputasi pada tingkat set konfigurasi memberi Anda kendali lebih besar atas reputasi pengirim Anda.

Bagian ini mencakup prosedur untuk mengekspor data reputasi untuk set konfigurasi individual CloudWatch dengan menggunakan Amazon SESAPI.

Mengaktifkan ekspor metrik reputasi

Untuk mulai mengekspor metrik reputasi untuk set konfigurasi, gunakan operasi. `UpdateConfigurationSetReputationMetricsEnabled` API Untuk mengakses Amazon SESAPI, kami sarankan menggunakan AWS CLI atau salah satu dari AWS SDKs.

Prosedur ini mengasumsikan bahwa diinstal pada komputer Anda dan dikonfigurasi dengan benar. AWS CLI Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Untuk mengaktifkan ekspor metrik reputasi untuk set konfigurasi

- Di baris perintah, ketik perintah berikut:

  ```
  aws ses update-configuration-set-reputation-metrics-enabled --configuration-set-
  name ConfigSet --enabled
  ```

  Ganti *`ConfigSet`* di perintah sebelumnya dengan nama set konfigurasi yang ingin Anda mulai mengekspor metrik reputasi.

Menonaktifkan ekspor metrik reputasi

Anda juga dapat menggunakan `UpdateConfigurationSetReputationMetricsEnabled` API operasi untuk menonaktifkan ekspor metrik reputasi untuk set konfigurasi.

Untuk menonaktifkan ekspor metrik reputasi pada set konfigurasi

- Di baris perintah, ketik perintah berikut:

```
aws ses update-configuration-set-reputation-metrics-enabled --configuration-set-
name ConfigSet --no-enabled
```

  Ganti *ConfigSet* dalam perintah sebelumnya dengan nama set konfigurasi yang ingin Anda nonaktifkan ekspor metrik reputasi.

Alamat IP khusus untuk Amazon SES

Saat Anda membuat akun Amazon SES baru, secara default email Anda dikirim dari alamat IP yang dibagikan dengan pengguna SES lainnya. Anda juga dapat menggunakan alamat IP khusus yang dicadangkan untuk penggunaan eksklusif Anda dengan menyewanya dengan biaya tambahan. Ini memberi Anda kendali penuh atas reputasi pengirim Anda dan memungkinkan Anda untuk mengisolasi reputasi Anda untuk segmen yang berbeda dalam program email. Amazon SES menawarkan dua cara untuk menyediakan dan mengelola alamat IP khusus:

- Standar — mengacu pada alamat IP khusus yang Anda atur dan kelola secara manual, termasuk opsi untuk menghangatkannya secara manual dan menskalakannya, dan memindahkannya secara manual masuk dan keluar dari kumpulan IP. (Ini sebelumnya disebut sebagai alamat IP khusus di SES.)
- Dikelola — mengacu pada alamat IP khusus yang secara otomatis diatur atas nama Anda oleh SES untuk menyediakan cara cepat dan mudah untuk mulai menggunakan alamat IP khusus yang dikelola oleh SES; mereka secara otomatis melakukan pemanasan untuk setiap ISP secara individual dan skala otomatis berdasarkan volume pengiriman Anda untuk membantu memastikan bahwa alamat IP khusus Anda digunakan secara optimal berdasarkan cara Anda mengirim email.

Saat memutuskan antara alamat IP bersama atau dua jenis alamat IP khusus yang ditentukan di atas, pilih salah satu yang memberikan manfaat paling besar untuk jenis, volume, dan pola email yang Anda kirim. Untuk membantu Anda membuat keputusan, manfaat ini dirangkum dalam tabel berikut. Pilih item di kolom Manfaat untuk informasi tambahan.

Manfaat	Alamat IP bersama	Alamat IP khusus (standar)	Alamat IP khusus (dikelola)
Siap digunakan segera	Ya	Tidak	Tidak
Diperlukan pengaturan tambahan	Tidak	Ya	Ya
Alamat IP & reputasi terisolasi dari	Tidak	Ya	Ya

Manfaat	Alamat IP bersama	Alamat IP khusus (standar)	Alamat IP khusus (dikelola)
pelanggan SES lainnya			
Kapasitas meningkat secara otomatis saat lalu lintas meningkat	Tidak	Tidak	Ya
Baik untuk pelanggan dengan pola pengiriman yang berkelanjutan dan dapat diprediksi	Ya	Ya	Ya
Baik untuk pelanggan dengan pola pengiriman yang kurang dapat diprediksi	Ya	Tidak	Ya
Baik untuk pengirim volume tinggi	Ya	Ya	Ya
Baik untuk pengirim volume rendah	Ya	Tidak	Tidak
Biaya bulanan tambahan	Tidak	Ya	Ya
Kontrol penuh atas reputasi pengirim	Tidak	Ya	Ya
Mengisolasi reputasi berdasarkan jenis email, penerima, atau faktor lainnya	Tidak	Ya	Ya

Manfaat	Alamat IP bersama	Alamat IP khusus (standar)	Alamat IP khusus (dikelola)
Menyediakan alamat IP yang dikenal yang tidak pernah berubah	Tidak	Ya	Tidak

> **Important**
>
> Jika Anda tidak berencana mengirim email dalam jumlah besar secara teratur dan dapat diprediksi, sebaiknya gunakan alamat IP bersama. Jika Anda ingin menggunakan alamat IP khusus dalam situasi di mana pola pengiriman Anda sangat tidak teratur, menggunakan IP Khusus (dikelola) adalah pilihan yang lebih baik.

Kemudahan penyiapan

Alamat IP bersama — Anda tidak perlu melakukan konfigurasi tambahan apa pun. Akun SES Anda siap mengirim email segera setelah Anda memverifikasi alamat email dan keluar dari kotak pasir.

Alamat IP khusus (standar) —Anda harus mengirimkan permintaan melalui AWS Support Center dan secara opsional mengonfigurasi kumpulan IP khusus.

Alamat IP khusus (terkelola) —Anda tidak perlu mengirimkan permintaan untuk alamat IP khusus. Mereka akan secara otomatis dialokasikan saat Anda ikut serta dan melakukan penelusuran satu kali untuk membuat kumpulan khusus terkelola Anda.

Manajemen reputasi

reputasi alamat IP sebagian besar didasarkan pada pola dan volume pengiriman historis. Alamat IP yang mengirimkan volume email yang konsisten selama jangka waktu yang lama biasanya memiliki reputasi yang bagus.

Alamat IP bersama — dibagi antara beberapa pelanggan SES, alamat ini secara kolektif mengirim volume email yang besar dan AWS dengan hati-hati mengelola lalu lintas keluar untuk memaksimalkan reputasi alamat IP bersama.

Alamat IP khusus (standar) —setelah pemanasan, alamat IP Anda diisolasi dari kumpulan bersama SES dan Anda mempertahankan reputasi pengirim Anda sendiri dengan mengirimkan volume email yang konsisten dan dapat diprediksi.

Alamat IP khusus (dikelola) —setelah pemanasan IP baru Anda, IP tersebut diisolasi dari kumpulan bersama SES dan Anda mempertahankan reputasi pengirim Anda sendiri. Ada manfaat tambahan dari melacak reputasi untuk setiap ISP dan secara optimal menjadwalkan pengiriman keluar yang sesuai. Jadi, sementara Anda masih mempertahankan reputasi pengirim Anda, otomatisasi ini membantu meningkatkan kemampuan pengiriman secara keseluruhan dan mengurangi rasio pentalan jika dibandingkan dengan beban kerja yang setara pada alamat IP khusus yang dikonfigurasi secara manual.

> **Note**
>
> Untuk informasi tentang data Smart Network Data Services (SNDS) untuk IP khusus Anda, lihat. Metrik SNDS untuk IP khusus

Prediktabilitas pola pengiriman

Alamat IP dengan riwayat pengiriman email yang konsisten memiliki reputasi yang lebih baik daripada alamat email yang tiba-tiba mulai mengirimkan email dalam volume besar tanpa riwayat pengiriman sebelumnya.

Alamat IP bersama - baik untuk pola pengiriman email yang tidak mengikuti pola yang dapat diprediksi. Dengan alamat IP bersama, Anda dapat menambah atau mengurangi pola pengiriman email Anda sesuai tuntutan situasi.

Alamat IP khusus (standar) — Anda harus menghangatkan alamat dengan mengirimkan sejumlah email yang secara bertahap meningkat setiap hari. Proses penyiapan alamat IP baru dijelaskan dalam Pemanasan alamat IP khusus (standar). Setelah alamat IP khusus Anda disiapkan, Anda harus mempertahankan pola pengiriman yang konsisten.

Alamat IP khusus (dikelola) —alamat IP khusus Anda dihangatkan secara otomatis untuk setiap IP di kumpulan terkelola dengan menggunakan strategi pemanasan adaptif (bersama dengan kumpulan bersama SES) yang memperhitungkan pola pengiriman aktual untuk mengoptimalkan pemanasan untuk setiap ISP secara individual. Kumpulan IP terkelola secara otomatis menskalakan per ISP berdasarkan penggunaan dan pertimbangan kebijakan khusus ISP.

Volume email keluar

Alamat IP bersama — terbaik untuk pelanggan yang mengirim email dengan volume rendah.

Alamat IP khusus (standar) | Alamat IP khusus (dikelola) —keduanya cocok untuk pelanggan yang mengirim email dalam jumlah besar. Sebagian besar ISP hanya melacak reputasi alamat IP tertentu jika mereka menerima volume email yang signifikan dari alamat itu. Untuk setiap ISP yang ingin Anda tumbuhkan reputasinya, Anda harus mengirim beberapa ratus email dalam jangka waktu 24 jam setidaknya satu kali per bulan. Dalam beberapa kasus, kedua jenis alamat IP khusus juga dapat berfungsi untuk volume email yang lebih kecil. Misalnya, mereka dapat bekerja dengan baik jika Anda mengirim ke sekelompok kecil penerima yang terdefinisi dengan baik yang server emailnya menerima atau menolak email menggunakan daftar alamat IP tertentu, bukan reputasi alamat IP.

Biaya tambahan

Alamat IP bersama —termasuk dalam harga SES standar.

Alamat IP khusus (standar) —tersedia dengan biaya bulanan tambahan per alamat IP yang Anda sewa. Untuk informasi harga, lihat halaman harga SES.

Alamat IP khusus (dikelola) —tersedia dengan biaya bulanan standar (terlepas dari jumlah IP yang dibutuhkan) dan biaya penggunaan per pesan. Untuk informasi harga, lihat halaman harga SES.

Kontrol atas reputasi pengirim

Alamat IP bersama —reputasi pengirim Anda dikendalikan oleh SES.

Alamat IP khusus (standar) | Alamat IP khusus (dikelola) —reputasi pengirim Anda sepenuhnya berada di bawah kendali Anda. Akun SES Anda adalah satu-satunya yang dapat mengirim email dari alamat tersebut. Untuk alasan ini, reputasi pengirim ditentukan oleh praktik pengiriman email Anda. Selain itu, IP khusus (dikelola) secara aktif memantau alamat IP keluar yang digunakan untuk pengiriman email dengan menggunakan alamat IP berkinerja tertinggi untuk meningkatkan pengiriman email ke penerima Anda. Data pemanfaatan dapat muncul dengan menggunakan layanan tambahan seperti CloudWatch metrik Amazon dan dasbor bawaan yang ada di Amazon SES.

Kemampuan untuk mengisolasi reputasi pengirim

Alamat IP bersama — reputasi pengirim Anda ditetapkan pada tingkat akun dan tidak dapat diisolasi.

Alamat IP khusus (standar) | Alamat IP khusus (dikelola) —Anda dapat mengisolasi reputasi pengirim untuk berbagai komponen dalam program email Anda dengan membuat kumpulan IP khusus — grup alamat IP khusus yang dapat digunakan untuk mengirim jenis email tertentu. Misalnya, Anda dapat membuat satu kolam alamat IP khusus untuk mengirim email pemasaran, dan satu lagi untuk mengirim email transaksional.

Diketahui, alamat IP yang tidak berubah

Alamat IP bersama — Anda tidak tahu alamat IP yang digunakan SES untuk mengirim email Anda, dan mereka dapat berubah kapan saja.

Alamat IP khusus (standar) —Anda dapat menemukan nilai alamat yang mengirim email Anda di halaman IP Khusus konsol SES. Ini karena alamat IP khusus bersifat statis.

Alamat IP khusus (dikelola) —SES akan secara otomatis mengonfigurasi jumlah optimal alamat IP khusus berdasarkan pola pengiriman Anda. Ini berarti bahwa alamat IP khusus di kumpulan Anda tidak terlihat dan secara dinamis akan meningkat atau menurun berdasarkan permintaan.

Alamat IP khusus (standar) di Amazon SES

Alamat IP khusus (standar) adalah alamat IP khusus yang Anda atur dan kelola secara manual di SES. Mereka berbeda dari yang diatur dan dikelola secara otomatis menggunakan fitur SESthe section called “Dikelola”. Selain memungkinkan Anda kontrol penuh atas reputasi pengiriman Anda menggunakan alamat IP khusus, IP khusus (standar) memungkinkan Anda untuk sepenuhnya mengelola IP khusus Anda, termasuk pemanasan mereka, skala mereka keluar, dan manajemen kolam IP.

IP khusus (standar) dan IP Khusus (dikelola) keduanya merujuk ke alamat IP khusus yang Anda sewa di SES untuk harga tambahan, tetapi berbeda dalam cara mereka diimplementasikan dan dikelola. Meskipun ada manfaat bersama yang umum untuk keduanya, mereka masing-masing memiliki keuntungan unik untuk ditawarkan tergantung pada jenis pengiriman email Anda, seperti yang dibahas dalamAlamat IP khusus.

Topiknya adalah bagian ini menjelaskan cara mengatur dan mengelola IP khusus (standar) secara manual di SES.

Topik

- Meminta dan melepaskan alamat IP khusus (standar)

- Pemanasan alamat IP khusus (standar)
- Membuat kolam IP khusus standar untuk IP khusus (standar)

Meminta dan melepaskan alamat IP khusus (standar)

Untuk menggunakan alamat IP khusus (standar), Anda harus terlebih dahulu memintanya. Jika Anda tidak lagi membutuhkannya, Anda harus melepaskan mereka. Meminta dan melepaskan IP khusus (standar) melalui AWS SupportPusat. Akun Anda dikenai biaya bulanan tambahan untuk setiap alamat IP khusus yang Anda sewa untuk digunakan dengan Amazon SES. Tidak ada komitmen minimum saat menggunakan IP khusus (standar).

Untuk informasi selengkapnya tentang biaya yang terkait dengan IP khusus (standar), lihat Harga Amazon SES.

Untuk daftar semua Wilayah tempat Amazon SES saat ini tersedia, lihat Wilayah AWSdan Endpoint di Referensi Umum Amazon Web. Untuk mempelajari selengkapnya tentang jumlah Availability Zone yang tersedia di setiapWilayah AWS, lihat InfrastrukturAWS Global.

Minta IP khusus (standar)

Anda dapat meminta IP khusus (standar) yang Anda butuhkan dengan membuat kasus peningkatan kuota layanan di PusatAWS Support.

Untuk meminta IP khusus (standar)

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi di sebelah kiri, pilih IP khusus.
3. Lakukan salah satu dari berikut:

    a. Jika Anda tidak memiliki IP khusus yang ada di akun Anda:

        - Halaman orientasi IP khusus ditampilkan. Di panel ikhtisar IP Khusus (standar), pilih Minta IP khusus.

          Halaman Create case terbuka diAWS Support Console.

    b. Jika Anda memiliki IP khusus yang ada di akun Anda:

        i. Pilih tab Standard IP pool pada halaman IP Khusus.

ii. Di panel Ringkasan standar, pilih Minta atau lepaskan IP khusus Standar.

Halaman Create case terbuka diAWS Support Console.

4. Di bawah Buat kasus, pilih kartu peningkatan batas layanan di bagian atas halaman.
5. Di bawah Detail kasus, lengkapi bagian berikut:
   - Untuk jenis Limit, simpan Batas Layanan SES.
   - Untuk Tipe Surat, pilih tipe email yang akan dikirim menggunakan alamat IP khusus Anda. Jika beberapa nilai berlaku, pilih opsi yang berlaku untuk sebagian besar email yang akan dikirim.
   - Untuk URL Situs Web, masukkan URL situs web Anda. Memberikan informasi ini membantu kami memahami tipe konten yang ingin Anda kirim dengan lebih baik.
   - Untuk Penjelasan, secara detail, tentang cara Anda mengirim ke penerima yang secara khusus meminta alamat email Anda, berikan respons yang konsisten dengan kasus penggunaan Anda.
   - Untuk Penjelasan, secara detail, tentang proses yang akan Anda ikuti saat menerima notifikasi pentalan dan aduan, berikan respons yang konsisten dengan kasus penggunaan Anda.
   - Untuk Akankah Anda mematuhi Syarat Layanan dan AUP AWS, pilih opsi yang berlaku untuk kasus penggunaan Anda.
6. Di bawah Permintaan, lengkapi bagian berikut:
   - Untuk Wilayah, pilih Wilayah AWS yang berlaku untuk permintaan Anda.
   - Untuk Limit, simpan IP Khusus yang Diinginkan.
   - Untuk Nilai batas baru, masukkan jumlah alamat IP khusus yang Anda butuhkan untuk menerapkan kasus penggunaan Anda.

> **Note**
>
> Jika Anda ingin meminta alamat IP khusus untuk digunakan di lainnyaWilayah AWS, pilih Tambahkan permintaan lain, dan kemudian selesaikan bidang Wilayah, Batasan, dan Nilai batas baru untuk mendapatkan tambahanWilayah AWS. Ulangi proses ini untuk setiapWilayah AWS tempat Anda ingin gunakan alamat IP khusus.

7. Di bawah Deskripsi kasus, untuk Deskripsi kasus penggunaan, nyatakan bahwa Anda ingin meminta alamat IP khusus. Sebutkan juga, Jika Anda ingin meminta sejumlah alamat IP khusus. Jika Anda tidak menentukan jumlah alamat IP khusus, kami akan memberikan jumlah alamat IP

khusus yang diperlukan untuk memenuhi persyaratan tingkat pengiriman yang Anda tentukan di langkah sebelumnya.

Selanjutnya, jelaskan rencana Anda menggunakan alamat IP khusus untuk mengirim email menggunakan Amazon SES. Sertakan informasi tentang alasan Anda ingin menggunakan alamat IP khusus ketimbang alamat IP bersama. Informasi ini membantu kami memahami kasus penggunaan Anda dengan lebih baik.

8. Di bawah Opsi kontak, untuk Bahasa kontak pilihan, pilih jika Anda ingin menerima komunikasi untuk kasus ini dalam Bahasa Inggris atau Bahasa Jepang.
9. Setelah selesai, pilih Kirim.

Setelah Anda mengirimkan formulir, kami akan mengevaluasi permintaan Anda. Jika kami mengabulkan permintaan Anda, maka kami akan membalas kasus Anda di Pusat Support guna mengonfirmasi bahwa alamat IP khusus terkait dengan akun Anda.

Lepaskan alamat IP khusus

Jika Anda menggunakan alamat IP khusus dan tidak lagi menginginkannya terkait dengan akun Anda, prosedur berikut menunjukkan cara melepaskannya dengan membuat kasus di PusatAWS Support.

> **Important**
>
> Proses pelepasan alamat IP khusus tidak dapat dibalik. Jika Anda melepaskan alamat IP khusus di pertengahan bulan, kami memprorata biaya penggunaan IP khusus di bulan yang berjalan.

Untuk melepaskan IP khusus (standar)

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi di sebelah kiri, pilih IP khusus.
3. Pilih tab Standard IP pool pada halaman IP Khusus.
4. Di panel Ringkasan standar, pilih Minta atau lepaskan IP khusus Standar.
5. Di bawah Rincian kasus, untuk jenis Limit, simpan Batas Layanan SES

> **Note**
>
> Kotak yang tersisa di bagian ini tidak berlaku untuk melepaskan IP khusus. Biarkan kosong.

6. Di bawah Permintaan, lengkapi bagian berikut:

   - Untuk Wilayah, pilih permintaan relinquish Anda yang berlaku.Wilayah AWS

   > **Note**
   >
   > Alamat IP khusus yang unik untuk setiapWilayah AWS, penting untuk memilih yang terkait dengan alamat IP khusus yang terkait dengan alamat IP khususWilayah AWS yang unik untuk setiap, penting untuk memilih alamat IP khusus yang terkait dengan alamat IP khusus.

   - Untuk Limit, simpan IP Khusus yang Diinginkan.
   - Untuk Nilai batas baru, masukkan nomor apa pun. Nomor yang Anda masukkan di sini tidak penting, Anda menentukan jumlah IP khusus yang ingin Anda lepaskan di langkah berikutnya.

   > **Note**
   >
   > Alamat IP khusus tunggal hanya dapat digunakan dalam Wilayah AWS. Jika Anda ingin melepaskan alamat IP khusus yang Anda gunakan di lainnyaWilayah AWS, pilih Tambahkan permintaan lain. Kemudian lengkapi bidang nilai Region, Limit, dan New limit untuk tambahanWilayah AWS. Ulangi proses ini untuk setiap alamat IP khusus yang ingin Anda lepaskan.

7. Di bawah Deskripsi kasus, untuk Gunakan deskripsi kasus, sebutkan bahwa Anda ingin melepaskan alamat IP khusus yang ada. Jika Anda saat ini menyewa lebih dari satu alamat IP khusus, sertakan jumlah alamat IP khusus yang ingin Anda lepaskan.
8. Di bawah Opsi kontak, untuk Bahasa kontak pilihan, pilih jika Anda ingin menerima komunikasi untuk kasus ini dalam Bahasa Inggris atau Bahasa Jepang.
9. Setelah selesai, pilih Kirim.

Setelah kami menerima permintaan Anda, kami mengirimkan pesan yang meminta Anda untuk mengonfirmasi bahwa Anda ingin melepaskan alamat IP khusus Anda. Setelah mengonfirmasi bahwa Anda ingin melepaskan alamat IP, kami akan menghapusnya dari akun Anda.

Pemanasan alamat IP khusus (standar)

Ketika menentukan apakah menerima atau menolak sebuah pesan, penyedia layanan email mempertimbangkan reputasi alamat IP yang mengirimkan pesannya. Salah satu faktor yang berkontribusi terhadap reputasi alamat IP adalah apakah alamat IP tersebut memiliki riwayat pengiriman email berkualitas tinggi. Penyedia email cenderung tidak menerima email dari alamat IP baru yang memiliki sedikit atau sama sekali tidak memiliki riwayat. E-mail yang dikirim dari alamat IP yang memiliki sedikit atau sama sekali tidak memiliki riwayat mungkin diletakkan di folder email sampah penerima, atau mungkin diblokir.

Ketika Anda mulai mengirim email dari alamat IP khusus baru, Anda harus secara bertahap meningkatkan jumlah email yang Anda kirim dari alamat itu sebelum menggunakannya ke kapasitas penuh. Proses ini disebut penyiapan Alamat IP.

Jumlah waktu yang diperlukan untuk menghangatkan alamat IP bervariasi antara penyedia email. Untuk beberapa penyedia email, Anda dapat membangun reputasi positif dalam waktu sekitar dua minggu, sedangkan untuk yang lain mungkin memerlukan waktu hingga enam minggu. Saat memanaskan alamat IP khusus baru, Anda harus mengirim email ke pengguna Anda yang paling aktif untuk memastikan bahwa tingkat keluhan Anda tetap rendah. Anda juga harus berhati-hati memeriksa pesan pentalan dan mengirim lebih sedikit email jika Anda menerima sejumlah besar notifikasi pemblokiran atau throttling. Untuk informasi selengkapnya tentang cara memantau pentalan Anda, lihat Memantau aktivitas pengiriman Amazon SES.

Pemanasan otomatis untuk khusus IPs (standar)

Saat Anda meminta alamat IP khusus (standar), Amazon SES secara otomatis menghangatkannya untuk meningkatkan pengiriman email yang Anda kirim. Fitur pemanasan alamat IP otomatis diaktifkan secara default. SESsecara otomatis menghangatkan dedicated Anda IPs dengan secara bertahap meningkatkan jumlah email yang Anda kirim melalui dedicated Anda IPs berdasarkan rencana pemanasan yang telah ditentukan. Peningkatan bertahap ini membantu Anda IPs membangun reputasi positif dengan penyedia layanan internet (ISPs).

Langkah-langkah yang terjadi selama proses pemanasan otomatis tergantung pada apakah Anda sudah memiliki alamat IP khusus.

- Ketika Anda meminta khusus IPs (standar) untuk pertama kalinya, SES mendistribusikan pengiriman email Anda antara alamat IP khusus Anda dan serangkaian alamat yang dibagikan dengan SES pelanggan lain. SESsecara bertahap meningkatkan jumlah pesan yang dikirim dari alamat IP khusus Anda dari waktu ke waktu.
- Jika Anda sudah memiliki alamat IP khusus, SES mendistribusikan pengiriman email Anda antara dedicated yang ada IPs (yang sudah dihangatkan) dan dedicated baru Anda IPs (yang tidak dihangatkan). SESsecara bertahap meningkatkan jumlah pesan yang dikirim dari alamat IP khusus baru Anda dari waktu ke waktu.

> **Note**
>
> Pemanasan IP otomatis adalah proses berbasis waktu. Persentase pemanasan terus meningkat selama 45 hari, terlepas dari volume pengiriman Anda.

Setelah Anda menyiapkan alamat IP khusus, Anda harus mengirim sekitar 1.000 email setiap hari ke setiap penyedia email yang ingin Anda pertahankan untuk reputasi positif. Anda harus melakukan tugas ini pada setiap alamat IP khusus yang Anda gunakanSES.

Anda harus menghindari pengiriman email dalam jumlah besar segera setelah proses pemanasan selesai. Sebagai gantinya, tingkatkan perlahan jumlah email yang Anda kirim hingga mencapai target volume. Jika penyedia email melihat peningkatan besar dan tiba-tiba dalam jumlah email yang dikirim dari alamat IP, mereka dapat memblokir atau membatasi pengiriman pesan dari alamat itu.

Nonaktifkan proses pemanasan otomatis pada dedicated IPs (standar)

Saat Anda membeli alamat IP khusus standar baru, Amazon SES secara otomatis menghangatkannya untuk Anda karena fitur pemanasan alamat IP otomatis diaktifkan secara default untuk akun Anda. Jika Anda lebih suka memanaskan alamat IP khusus sendiri, Anda dapat menonaktifkan fitur pemanasan otomatis di tingkat akun untuk semua alamat IP Anda.

Jika Anda menonaktifkan fitur pemanasan otomatis, setiap dedikasi yang disewakan selanjutnya IPs akan ditambahkan ke akun Anda dengan status pemanasan Lengkap yang membuatnya tersedia untuk digunakan tanpa pemanasan — ini berarti Anda bertanggung jawab untuk memastikan ini IPs dihangatkan dengan benar sebelum menggunakannya untuk pengiriman reguler. Apa pun IPs yang saat ini berada di tengah pemanasan pada saat Anda menonaktifkan fitur pemanasan otomatis tidak akan terpengaruh.

> **Important**
>
> Jika Anda menonaktifkan fitur penyiapan otomatis, Anda bertanggung jawab untuk mempersiapkan alamat IP khusus Anda sendiri. Jika Anda mengirim email dari alamat yang belum disiapkan, Anda mungkin mengalami laju penyampaian yang buruk.

Untuk menonaktifkan (atau mengaktifkan kembali) fitur pemanasan otomatis untuk semua khusus IPs (standar) di akun Anda

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Khusus IPs.
3. Pilih tab Standard IP pool pada IPs halaman Dedicated.
4. Pilih Nonaktifkan pemanasan otomatis di panel ikhtisar Standar untuk menonaktifkan pemanasan otomatis, atau pilih Aktifkan pemanasan otomatis untuk mengaktifkan kembali pemanasan otomatis.

Pemanasan secara manual khusus IPs (standar)

Anda dapat secara manual menambah atau mengurangi volume pengiriman khusus IPs (standar) saat ini dengan mengedit persentase pemanasannya, mengakhiri proses pemanasan sebelum waktunya, dan mengatur volume pengiriman saat ini menjadi 0% dan memulai kembali proses pemanasan.

Untuk pemanasan secara manual khusus IPs (standar)

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Khusus IPs.
3. Pilih tab Standard IP pool pada IPs halaman Dedicated.
4. Di IPs panel khusus Semua Standar, pilih alamat IP dan pilih Edit pemanasan dan pilih salah satu opsi berikut:

    a. Edit persentase —masukkan nilai di bidang Persentase pemanasan untuk menambah atau mengurangi volume pengiriman IP Anda saat ini dengan mengedit persentase pemanasan diikuti dengan Simpan perubahan.

Kolom status Pemanasan akan mengatakan `In progress` dan kolom Persentase pemanasan akan menampilkan nilai yang Anda masukkan.

b. Tandai sebagai Selesai —baca pemanasan Mark sebagai Selesai? dialog untuk mengonfirmasi bahwa Anda memahami implikasi dari mengakhiri proses pemanasan otomatis sebelum waktunya, lalu pilih Tandai sebagai Lengkap.

   Kolom status Pemanasan akan mengatakan `Complete` dan kolom persentase Pemanasan akan mengatakan. `100%`

c. Setel ulang persentase —baca persentase Reset pemanasan? dialog untuk mengonfirmasi bahwa Anda menyetel volume pengiriman IP saat ini menjadi 1% dan harus memulai ulang proses pemanasan otomatis atau mengatur persentase pemanasan secara manual, lalu pilih Reset.

   Kolom status Pemanasan akan mengatakan `In progress` dan kolom persentase Pemanasan akan mengatakan. `1%`

Membuat kolam IP khusus standar untuk IP khusus (standar)

Jika Anda membeli beberapa alamat IP khusus untuk digunakan dengan Amazon SES, Anda dapat membuat grup alamat tersebut, yang disebut kumpulan IP khusus. Mengelompokkan IP khusus (standar) bersama-sama di kolam membuat mereka lebih mudah untuk mengelola. Skenario umum adalah membuat satu kolam untuk mengirim komunikasi pemasaran, dan satu lagi untuk mengirim email transaksional. Reputasi pengirim Anda untuk email transaksional kemudian diisolasi dari email pemasaran Anda. Dalam skenario ini, jika kampanye pemasaran menghasilkan sejumlah besar keluhan, pengiriman email transaksional Anda tidak terpengaruh.

Bagian ini berisi prosedur untuk membuat kolam IP khusus.

> **Note**
>
> Anda juga dapat membuat set konfigurasi yang menggunakan kolam alamat IP yang dibagi oleh semua pelanggan SES. kolam IP bersama berguna untuk situasi di mana Anda perlu mengirim email yang tidak sesuai dengan perilaku pengiriman biasa Anda. Untuk informasi tentang penggunaan kolam IP dengan satu set konfigurasi, lihat Menetapkan kolam IP di Amazon SES.

Untuk membuat kolam IP khusus untuk IP khusus (standar) menggunakan konsol SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Pada panel navigasi kiri, pilih IP khusus.

   > **Note**
   >
   > Jika saat ini Anda tidak memiliki IP khusus (standar) di akun Anda, halaman orientasi IP Khusus ditampilkan memberi Anda kesempatan untuk membeli IP khusus (standar). Untuk informasi selengkapnya, lihat the section called “Minta IP khusus (standar)”.

3. Pilih tab Standard IP pool pada halaman IP Khusus.
4. Di panel kolam All Dedicated IP (standar), pilih Create Standard IP pool.

   Halaman Create IP Pool terbuka.
5. Di panel rincian Pool,

   a. Pilih Standar (dikelola sendiri) di bidang mode Scaling.

   b. Masukkan nama untuk kolam IP Anda di kolom nama kolam IP.

      > **Note**
      >
      > Nama IP pool harus unik dan tidak dapat berupa duplikat nama kumpulan IP terkelola di akun Anda.

   c. (Opsional) Jika Anda memiliki alamat IP khusus standar yang ingin Anda tambahkan ke kumpulan IP ini, pilih dari daftar dropdown di bidang Alamat IP Khusus.

      > **Note**
      >
      > Jika Anda memilih alamat IP yang sudah terkait dengan kolam IP ini, sekarang hanya terkait dengan kolam IP ini.

6. (Opsional) Anda dapat mengasosiasikan kolam IP ini dengan set konfigurasi dengan memilih salah satu dari daftar dropdown di bidang Configuration sets.

> **Note**
>
> - Jika Anda memilih set konfigurasi yang sudah dikaitkan dengan kolam IP, sekarang hanya akan dikaitkan dengan kolam IP ini.
> - Untuk menambah atau menghapus set konfigurasi terkait setelah kumpulan IP ini dibuat, edit parameter kumpulan Mengirim IP kumpulan set konfigurasi.
> - Jika Anda belum membuat rangkaian konfigurasi apa pun, lihat Set konfigurasi.

7. (Opsional) Anda dapat menambahkan satu atau beberapa Tanda ke kolam IP ini dengan menyertakan kunci tanda dan nilai opsional untuk kunci.

   a. Pilih Tambahkan tanda baru dan masukkan Kunci. Anda juga dapat menambahkan opsi Nilai untuk tanda.

   b. Untuk menambahkan tanda, pilih Simpan perubahan.

      Anda dapat menambahkan hingga 50 tanda. Anda dapat menghapus tanda dengan memilih Hapus.

8. Pilih Create pool.

> **Note**
>
> Setelah Anda membuat kolam IP standar, Anda memiliki opsi untuk mengubahnya menjadi kumpulan IP terkelola. Lihat Membuat kumpulan IP terkelola.

Alamat IP khusus (dikelola) untuk Amazon SES

Alamat IP khusus (dikelola) adalah fitur Amazon SES yang secara otomatis mengatur dan mengelola alamat IP khusus atas nama Anda untuk memberikan cara cepat dan mudah untuk mulai menggunakan alamat IP khusus yang dikelola oleh SES. Ini membantu memastikan bahwa alamat IP khusus Anda digunakan secara efisien dan optimal untuk cara Anda mengirim email.

Untuk mengaktifkan IP khusus (dikelola) di akun Anda, Anda cukup membuat kumpulan IP terkelola dan SES melakukan sisanya. SES akan menentukan berapa banyak IP khusus yang Anda butuhkan berdasarkan pola pengiriman Anda, membuatnya untuk Anda, dan kemudian mengelola bagaimana mereka menskalakannya berdasarkan persyaratan pengiriman Anda.

Setelah diaktifkan, Anda dapat menggunakan IP khusus (terkelola) dalam pengiriman email Anda dengan mengaitkan kumpulan IP terkelola dengan set konfigurasi, dan kemudian menentukan konfigurasi yang ditetapkan saat mengirim email. Set konfigurasi juga dapat diterapkan ke identitas pengiriman dengan menggunakan set konfigurasi default.

Manfaat dan fitur IP khusus (dikelola)

Alamat IP khusus yang Anda buat dengan IP khusus (terkelola) mengotomatiskan tugas manajemen untuk membantu memastikan bahwa alamat IP khusus Anda digunakan dengan cara yang optimal untuk cara Anda mengirim email:

- Orientasi mudah — Untuk memulai dengan IP khusus (terkelola), Anda membuat kumpulan IP terkelola langsung dari konsol SES. Alamat IP khusus secara otomatis dialokasikan ke kolam renang. Anda dapat mulai mengirim dengan kumpulan IP terkelola tanpa harus membuka kasus permintaan melalui AWS Support Center.
- Penskalaan otomatis per ISP - Anda tidak perlu memantau atau menskalakan kumpulan IP khusus Anda secara manual karena kumpulan IP terkelola keluar secara otomatis berdasarkan penggunaan. Ini juga mempertimbangkan kebijakan khusus ISP. Misalnya, jika SES mendeteksi bahwa ISP mendukung kuota pengiriman harian yang rendah, kumpulan skala untuk mendistribusikan lalu lintas ke ISP tersebut dengan lebih baik di lebih banyak alamat IP.
- Intelligent warmup — IP khusus (terkelola) mulai mengirim email ke ISP berdasarkan kapasitasnya. Artinya, seberapa banyak mereka saat ini dihangatkan. Mereka secara otomatis melacak tingkat pemanasan untuk setiap ISP secara individual. Selain itu, fitur IP khusus (terkelola) memberikan informasi tentang reputasi Anda dengan tarif harian yang efektif dengan ISP teratas dalam bentuk CloudWatch metrik Amazon dan dasbor bawaan.
  - Warmup per ISP — SES melacak reputasi untuk setiap IP di kumpulan IP terkelola untuk setiap ISP secara individual. Misalnya, jika Anda telah mengirim semua lalu lintas Anda ke Gmail, alamat IP dianggap hangat hanya untuk Gmail dan dingin untuk ISP lainnya. Jika Anda mengubah pola lalu lintas Anda dengan meningkatkan email yang dikirim ke Hotmail, SES meningkatkan lalu lintas secara perlahan untuk Hotmail, karena alamat IP belum dihangatkan.
  - Adaptive warmup & Shared pool transiitioning — Penyesuaian pemanasan bersifat adaptif dan memperhitungkan pola pengiriman aktual. Saat mengirim volume ke ISP turun, persentase pemanasan juga turun untuk ISP itu. Pada fase awal pemanasan, setiap pengiriman yang berlebihan berdasarkan tingkat pemanasan saat ini dikirim melalui alamat IP yang dibagikan dengan pengguna Amazon SES lainnya — kumpulan bersama SES. Pada tahap pemanasan selanjutnya, setiap pengiriman yang berlebihan secara proaktif diperlambat dan dicoba lagi nanti.

> **Important**
>
> Sementara IP khusus (dikelola) secara otomatis menghangatkan alamat IP khusus Anda, bagian dari proses otomatis itu bekerja secara interaktif dengan kumpulan IP bersama SES.
>
> - Jika tingkat pengiriman Anda terlalu agresif untuk IP khusus baru Anda saat sedang dihangatkan, SES akan secara otomatis menumpahkan sebagian pengiriman Anda ke kumpulan IP bersama SES untuk melindungi reputasi IP khusus baru Anda.
> - Bahkan setelah IP khusus baru Anda sepenuhnya dipanaskan, tidak dijamin bahwa semua pengiriman Anda akan melalui mereka 100% dari waktu. Misalnya, jika tingkat pengiriman Anda tiba-tiba naik dan IP khusus (dikelola) menentukan itu harus mengalokasikan alamat IP khusus tambahan, itu akan memulai proses pemanasan yang mencakup penggunaan kolam bersama. Demikian juga, jika tingkat pengiriman Anda tiba-tiba turun sangat rendah, semua pengiriman Anda dapat beralih ke kumpulan IP bersama SES, lihatthe section called “Pentingnya pemanasan”.

- Permintaan otomatis & pelepasan alamat IP khusus - Anda tidak perlu meminta atau melepaskan alamat IP khusus yang dikelola melalui AWS Support Center, seperti yang diperlukan saat menggunakan IP khusus (standar). Saat melakukan onboarding dengan IP khusus (dikelola) langsung dari konsol SES, CLI, atau API, Anda secara otomatis dialokasikan alamat IP khusus dan dikenakan biaya berdasarkan volume pesan yang Anda kirim. Ketika Anda menghapus kumpulan IP yang dibuat oleh IP khusus (dikelola) atau memilih keluar dari IP khusus (dikelola), alamat IP yang dialokasikan Anda secara otomatis dilepaskan dan biaya segera dihentikan.
  - Mendapatkan alamat IP khusus pertama Anda - Fitur IP khusus (terkelola) akan secara otomatis mengalokasikan alamat IP khusus pertama Anda setelah volume pengiriman Anda mencapai ratusan email selama beberapa hari. Ini memastikan bahwa IP yang Anda kirim dapat membangun reputasi pengiriman dan meningkatkan kemampuan pengiriman. (Jika Anda tidak mengharapkan volume pengiriman Anda berada pada level ini, Anda harus menggunakan alamat IP bersama. Lihat tabel perbandingan Alamat IP khusus untuk meninjau jenis alamat IP yang terbaik untuk cara Anda mengirim email.)

Mengapa pemanasan IP yang tepat itu penting

Untuk memastikan bahwa email Anda akan dikirim melalui alamat IP khusus Anda, itu harus memiliki reputasi yang baik dengan ISP penerima. ISP hanya akan menerima sejumlah kecil email dari IP

yang tidak mereka kenali. Ketika Anda pertama kali mengalokasikan IP, itu baru dan tidak akan dikenali oleh ISP penerima karena tidak memiliki reputasi yang terkait dengannya. Agar reputasi IP dapat ditetapkan, secara bertahap harus membangun kepercayaan dengan menerima ISP — proses membangun kepercayaan bertahap ini disebut sebagai pemanasan. Segera setelah IP khusus (dikelola) mengalokasikan IP, ia memulai proses pemanasan Cerdas.

Dengan fitur Warmup per ISP dan Adaptive warmup dari IP khusus (dikelola), kelangsungan bisnis dipertahankan sepanjang siklus pemanasan dengan memastikan bahwa email Anda akan dikirimkan. Setelah fase pemanasan selesai, setiap kelebihan kapasitas antri dan dikirim hanya melalui kumpulan IP khusus. Namun, jika Anda memiliki satu alamat IP khusus dan pengiriman Anda berada di bawah volume minimum yang diperlukan untuk mempertahankan reputasi IP, IP khusus (dikelola) dapat menghapus IP khusus Anda dan pengiriman Anda akan diarahkan melalui kumpulan IP bersama SES.

> **Note**
>
> Jika Anda mengirim email dalam volume kecil (kurang dari beberapa ratus per hari selama beberapa hari), akan lebih bermanfaat untuk mengirim melalui kumpulan IP bersama SES. Lihat apakah IP khusus (dikelola) tepat untuk cara Anda mengirim email dengan meninjau tabel perbandingan di. Alamat IP khusus

Membuat kumpulan IP terkelola untuk mengaktifkan IP khusus (dikelola)

Untuk mengaktifkan IP khusus (dikelola), pertama-tama Anda membuat kumpulan IP terkelola. Setelah Anda membuat kumpulan terkelola, fitur menentukan berapa banyak IP khusus yang Anda butuhkan berdasarkan pola pengiriman Anda dan secara dinamis akan menskalakannya sesuai kebutuhan Anda.

Untuk menggunakan kumpulan terkelola untuk mengirim email, Anda harus mengaitkan kumpulan terkelola dengan set konfigurasi, lalu menentukan set konfigurasi tersebut saat mengirim email. Set konfigurasi juga dapat diterapkan ke identitas pengiriman dengan menggunakan set konfigurasi default.

Ada dua cara Anda dapat membuat kumpulan IP terkelola:

- Buat kolam baru.
- Konversikan kumpulan yang ada dari standar menjadi terkelola.

Dalam prosedur berikut, instruksi disediakan untuk kedua metode tersebut.

Untuk membuat atau mengonversi ke kumpulan IP terkelola menggunakan konsol SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih IP Khusus.
3. Bergantung pada apakah Anda ingin membuat kumpulan IP terkelola baru atau mengonversi kumpulan IP khusus standar ke kumpulan IP terkelola, ikuti instruksi masing-masing:

   Create new pool

   Untuk membuat kumpulan IP terkelola baru

   1. Lakukan salah satu hal berikut:

      a. Jika Anda tidak memiliki IP khusus di akun Anda:

         - Halaman orientasi IP Khusus ditampilkan. Di panel ikhtisar IP Khusus (terkelola), pilih Aktifkan IP khusus.

           Halaman Create IP Pool terbuka.

      b. Jika Anda memiliki IP khusus yang ada di akun Anda:

         i. Pilih tab Managed IP pools pada halaman IP Khusus.

         ii. Di panel kolam All Dedicated IP (managed), pilih Create Managed IP pool.

           Halaman Create IP Pool terbuka.

   2. Di panel detail Pool,

      a. Pilih Dikelola (dikelola secara otomatis) di bidang mode Penskalaan.

      b. Masukkan nama untuk kumpulan terkelola Anda di bidang nama kumpulan IP.

      > **Note**
      >
      > - Nama IP pool harus unik. Ini tidak bisa menjadi duplikat dari nama kolam IP khusus standar di akun Anda.

- Anda tidak dapat memiliki lebih dari 50 kumpulan IP khusus per Wilayah AWS akun Anda termasuk kolam IP terkelola dan standar.

3. (Opsional) Anda dapat mengaitkan kumpulan IP terkelola ini dengan set konfigurasi dengan memilih salah satu dari daftar tarik-turun di bidang Set konfigurasi.

> **Note**
>
> - Jika Anda memilih set konfigurasi yang sudah dikaitkan dengan kumpulan IP, itu akan dikaitkan dengan kumpulan terkelola ini, dan tidak lagi dikaitkan dengan kumpulan sebelumnya.
> - Untuk menambah atau menghapus set konfigurasi terkait setelah kumpulan terkelola ini dibuat, edit parameter kumpulan IP Mengirim set konfigurasi di panel Detail umum.
> - Jika Anda belum membuat rangkaian konfigurasi apa pun, lihat Set konfigurasi.

4. (Opsional) Anda dapat menambahkan satu atau beberapa Tanda ke kolam IP Anda dengan memasukkan kunci tanda dan nilai opsional untuk kunci.

   a. Pilih Tambahkan tanda baru dan masukkan Kunci. Anda juga dapat menambahkan opsi Nilai untuk tanda. Anda dapat menambahkan hingga 50 tag, jika Anda membuat kesalahan, pilih Hapus.

   b. Untuk menambahkan tag, pilih Simpan perubahan.

   Setelah membuat pool, Anda dapat menambahkan, menghapus, atau mengedit tag dengan memilih kumpulan terkelola dan memilih Edit.

5. Pilih Buat kolam.

> **Note**
>
> - Setelah Anda membuat kumpulan IP terkelola, itu tidak dapat dikonversi ke kolam IP standar.

- Saat menggunakan IP khusus (dikelola), Anda tidak dapat memiliki lebih dari 10.000 identitas pengiriman (domain dan alamat email, dalam kombinasi apa pun) per Wilayah AWS akun Anda.

Convert standard to managed

Untuk mengonversi kolam IP khusus standar menjadi terkelola

1. Pilih tab Standard IP pool pada halaman IP Khusus.
2. Di panel kolam Semua IP Khusus (standar), pilih kotak centang kumpulan IP khusus yang ingin Anda konversi dari standar ke terkelola.
3. Pilih Konversikan ke kumpulan terkelola —baca dialog Konversikan ke kumpulan IP terkelola untuk mengonfirmasi bahwa Anda memahami kondisi mengonversi kumpulan IP khusus standar Anda menjadi kumpulan terkelola.

> **Note**
>
> Sebelum mengonversi kumpulan IP khusus Anda dari standar ke terkelola, perhatikan hal berikut:
>
> 1. Semua IP khusus Anda saat ini (standar) akan dipindahkan ke kumpulan terkelola.
> 2. Jika saat ini Anda menyewakan terlalu banyak IP khusus (standar) untuk volume pengiriman Anda, maka IP khusus (dikelola) akan menghapus IP yang berlebihan.
> 3. Jika salah satu IP khusus Anda (standar) adalah bagian dari daftar izin untuk aplikasi lain, Anda tidak boleh mentransfernya ke kumpulan terkelola karena akan dihapus jika menjadi berlebihan — lihat poin 2.
> 4. Anda tidak akan lagi dikenakan biaya per IP, tetapi akan dikenakan biaya berdasarkan volume yang Anda kirim melalui kumpulan yang dikelola. Lihat harga Amazon SES.

4. Jika Anda menyetujui ketentuan seperti yang dinyatakan, pilih Konfirmasi —sebuah spanduk muncul, mengonfirmasi bahwa kumpulan IP khusus standar Anda telah diubah menjadi kumpulan terkelola.

> **Note**
>
> Set konfigurasi atau tag apa pun yang Anda kaitkan dengan kumpulan standar sebelum konversi sekarang akan dikaitkan dengan kumpulan terkelola yang menyediakan transisi tanpa batas untuk pengiriman email apa pun menggunakan set konfigurasi.

Penerbitan acara dapat digunakan untuk melacak kinerja pengiriman kumpulan terkelola. Untuk informasi selengkapnya, lihat the section called “Pantau pengiriman email menggunakan penerbitan acara”.

Melihat pengiriman dan kapasitas kumpulan IP terkelola di konsol Amazon SES

Untuk kumpulan IP terkelola yang telah Anda buat, konsol SES menyediakan cara mudah bagi Anda untuk mengamati bagaimana mereka digunakan untuk pengiriman email Anda melalui penggunaan kartu dan grafik deret waktu yang menunjukkan metrik pengiriman dan pemanfaatan dan kapasitas ISP.

Untuk melihat pengiriman dan kapasitas kumpulan IP terkelola menggunakan konsol SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih IP Khusus.
3. Pilih tab Managed IP pools pada halaman IP Khusus.
4. Bergantung pada apakah Anda ingin melihat metrik pengiriman dan kapasitas di konsol Amazon SES atau CloudWatch konsol Amazon, ikuti instruksi masing-masing:

   Amazon SES console

   Untuk melihat metrik pengiriman dan kapasitas di konsol Amazon SES

   1. Dalam tabel pool All Dedicated IP (managed), pilih nama kumpulan IP terkelola yang tercantum di kolom kolam IP untuk melihat detailnya.

Halaman detail kumpulan IP yang dipilih terbuka dengan kartu dan grafik deret waktu berikut:

a. Kartu:

- Status pengiriman - Menunjukkan apakah volume dan frekuensi pengiriman Anda cukup untuk memanfaatkan IP khusus dengan menampilkan salah satu dari dua status:
  - Volume tidak mencukupi - Volume pengiriman Anda terlalu rendah.
  - Mengirim melalui IP Khusus - Satu atau lebih IP khusus digunakan di kumpulan terkelola Anda.
- Volume pengiriman IP khusus yang dikelola — Volume email yang dikirim melalui IP khusus di kumpulan terkelola Anda dalam 7 hari terakhir.
- Persentase pengiriman IP khusus yang dikelola — Persentase email yang dikirim melalui IP khusus di kumpulan terkelola Anda dalam 7 hari terakhir.

b. Grafik:

- Volume terkirim — Volume email yang dikirim dalam 7 hari terakhir melalui IP khusus yang dikelola dibandingkan dengan IP bersama.
- Persentase volume terkirim — Persentase email yang dikirim dalam 7 hari terakhir melalui IP khusus yang dikelola dibandingkan dengan IP bersama.
- Kapasitas ISP - Menampilkan berapa banyak email yang dikirim melalui IP khusus di kumpulan terkelola Anda per 10 ISP teratas yang paling banyak digunakan dan kapasitas yang tersedia selama pengiriman Anda:
  - Mengirim untuk ISP (bar merah) - Volume email yang Anda kirim dalam 24 jam terakhir melalui ISP yang dipilih.
  - Kapasitas untuk ISP (garis biru) — Kapasitas ISP yang tersedia selama 24 jam terakhir.

2. Untuk memfilter ISP tertentu untuk grafik kapasitas ISP, pilih kotak daftar ISP dan pilih ISP—grafik akan diperbarui dengan metrik untuk ISP yang dipilih. (Jika Anda tidak memfilter pada ISP, Gmail ditampilkan secara default).

Amazon CloudWatch console

Untuk melihat metrik pengiriman dan kapasitas di konsol Amazon CloudWatch

- Dalam tabel kumpulan Semua IP Khusus (terkelola), pilih <pool_name>tautan Lihat CloudWatch metrik di kolom CloudWatchmetrik untuk melihat detailnya.

  Halaman kumpulan IP yang dipilih terbuka di CloudWatch konsol yang menampilkan metrik berikut:

  - Kirim — Volume email yang dikirim melalui IP khusus yang dikelola dan IP bersama.
  - ApproximateDedicatedSendingPercentage— Menunjukkan perkiraan persentase lalu lintas yang telah dikirimkan melalui IP khusus.
  - SentLast24 Jam — Volume email yang Anda kirim dalam 24 jam terakhir melalui ISP yang dipilih. (Berlabel Mengirim untuk ISP di konsol SES.)
  - Available24 HourSend — Kapasitas ISP yang dipilih tersedia selama 24 jam terakhir. (Kapasitas Berlabel untuk ISP di konsol SES.)

Menghapus kumpulan IP terkelola dan memilih keluar dari IP khusus (dikelola)

Saat Anda menghapus kumpulan IP terkelola, semua alamat IP yang dialokasikan secara otomatis dilepaskan. Jika Anda hanya memiliki satu kumpulan IP terkelola dan Anda menghapusnya, atau Anda menghapus kumpulan IP terkelola terakhir yang tersisa, Anda akan memilih keluar dari fitur IP khusus (terkelola) dan biaya akan segera dihentikan.

Untuk menghapus kumpulan IP terkelola menggunakan konsol SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih IP Khusus.
3. Pilih tab Managed IP pools pada halaman IP Khusus.
4. Di tabel kolam All Dedicated IP (managed), pilih tombol radio di sebelah nama kumpulan IP dari kumpulan terkelola yang ingin Anda hapus dan pilih Hapus.

5. Dalam modal pop-up, Anda akan memiliki kesempatan untuk mengonfirmasi pilihan Anda dengan memilih Hapus, atau Batal untuk menjaga kumpulan terkelola Anda.

> **Note**
>
> Jika Anda hanya memiliki satu kumpulan terkelola atau menghapus kumpulan terkelola terakhir Anda, modal pop-up akan mengingatkan Anda bahwa dengan menghapus kumpulan terkelola yang tersisa, Anda akan memilih keluar dari fitur IP (terkelola) khusus dan tidak akan lagi dikenakan biaya untuk itu. Anda akan diminta untuk masuk `Disable` di kolom konfirmasi sebelum Anda dapat memilih Hapus.

Menggunakan alamat IP Anda sendiri untuk mengirim email menggunakan Amazon SES

Amazon SES mencakup fitur yang disebut Bawa Alamat IP Anda Sendiri (BYOIP), yang memungkinkan untuk menggunakan alamat IP Anda sendiri untuk mengirim email melalui Amazon SES. Jika Anda sudah menggunakan rentang alamat IP untuk mengirim email, Anda dapat meminta agar kami membuat rentang IP Anda tersedia untuk mengirim email melalui Amazon SES.

> **Note**
>
> BYOIP hanya tersedia untuk alamat IP khusus yang Anda konfigurasikan secara manual—BYOIP tidak dapat digunakan dengan IP Khusus (dikelola).

BYOIP sangat membantu, misalnya, ketika Anda telah mengembangkan reputasi IP positif menggunakan sistem pengiriman email internal, tetapi Anda ingin bermigrasi ke Amazon SES. Dengan menggunakan BYOIP, Anda dapat mulai mengirim email melalui Amazon SES segera, tanpa harus membangun kembali reputasi alamat IP Anda.

Persyaratan

Untuk menggunakan BYOIP, rentang alamat IP Anda harus memenuhi persyaratan sebagai berikut:

- Rentang alamat harus didaftarkan dengan Regional internet registry (RIR), seperti American Registry for Internet Numbers (ARIN), Réseaux IP Européens Network Coordination Centre (RIPE

NCC), atau Asia-Pacific Network Information Centre (APNIC). Rentang alamat ini harus didaftarkan untuk entitas bisnis atau kelembagaan dan tidak dapat didaftarkan untuk perorangan.

- Anda harus dapat memberikan bukti bahwa Anda memiliki rentang alamat dengan mengirimkan pesan otorisasi yang ditandatangani.
- Alamat dalam rentang alamat IP harus memiliki riwayat yang bersih. Kami dapat menginvestigasi reputasi rentang alamat IP, dan berhak menolak rentang alamat IP jika berisi alamat IP yang memiliki reputasi buruk atau terkait dengan perilaku jahat.
- Rentang alamat IP tidak dapat menyertakan rentang alamat IP yang dibawa ke yang lainLayanan AWS untuk BYOIP, seperti Amazon EC2.

Pertimbangan-pertimbangan

Ada beberapa faktor yang harus Anda pertimbangkan sebelum meminta transfer rentang IP Anda ke Amazon SES:

- Rentang alamat paling spesifik yang dapat Anda tentukan adalah /24. Dengan kata lain, jika Anda mentransfer rentang IP 203.0.113.0/24 ke akun Amazon SES Anda, maka Anda dapat mengirim dari total alamat 256, mulai dari 203.0.113.0 hingga 203.0.113.255. Anda harus mentransfer seluruh rentang—Amazon SES yang saat ini tidak memungkinkan Anda untuk mentransfer alamat IP individu.
- Jika Anda menggunakan BYOIP untuk rentang tertentu alamat IP, Anda hanya dapat mengakses rentang dari satu Wilayah AWS.
- Anda dapat membawa lima rentang alamat per Wilayah ke Akun AWS Anda.
- Jika Anda menggunakan alamat IP Anda sendiri, Anda tidak dapat menggunakan alamat di kolam alamat IP Amazon SES bersama. Jika Anda perlu menggunakan alamat IP bersama ini, Anda dapat menggunakan Amazon SES di Wilayah AWS berbeda, atau buat Akun AWS baru.
- Ada biaya bulanan untuk setiap alamat IP yang Anda gunakan dengan BYOIP. Untuk informasi lebih lanjut, lihat Harga Amazon SES.

Menggunakan alamat IP Anda sendiri dengan Amazon SES

Untuk mencegah sistem kami digunakan untuk mengirim konten yang tidak diminta atau berbahaya, kami harus mempertimbangkan setiap permintaan BYOIP dengan hati-hati.

Jika Anda ingin menggunakan rentang IP Anda sendiri dengan Amazon SES, kirim informasi berikut ke ses-byoip-request@amazon .com:

- ID akun AWS Anda.
- Wilayah AWS yang ingin Anda gunakan rentang IP-nya, seperti ap-south-1.
- Deskripsi kasus penggunaan Anda.
- Rentang IP yang ingin Anda gunakan dengan Amazon SES.
- Nama registri internet yang rentangnya didaftarkan.

Kami akan merespons permintaan Anda dalam waktu 48 jam kerja. Dalam komunikasi kami dengan Anda, kami dapat meminta informasi tambahan, termasuk dokumen yang membuktikan kepemilikan Anda atas rentang IP.

Manajer Pengiriman Virtual untuk Amazon SES

Deliverability, atau memastikan email Anda mencapai kotak masuk penerima alih-alih folder spam atau sampah, adalah elemen inti dari strategi email yang sukses.

Virtual Deliverability Manager adalah fitur Amazon SES yang membantu Anda meningkatkan kemampuan pengiriman email, seperti meningkatkan pengiriman kotak masuk dan konversi email, dengan memberikan wawasan tentang data pengiriman dan pengiriman Anda, dan memberikan saran tentang cara memperbaiki masalah yang berdampak negatif pada tingkat keberhasilan pengiriman dan reputasi Anda.

Mengapa pengiriman kotak masuk dan reputasi pengirim Anda penting

Pengiriman kotak masuk adalah faktor kunci ketika datang ke konversi email (ketika penerima mengambil tindakan setelah membuka email) —pelanggan yang tidak menerima pesan Anda tidak akan dapat melihatnya, apalagi dapat terlibat dengan mereka.

Reputasi pengiriman memiliki pengaruh terbesar pada pengiriman kotak masuk di tingkat pengalaman pelanggan—hal ini menentukan apakah pesan yang tidak diinginkan menjangkau penerima atau pesan yang diperlukan diarahkan ke folder spam atau diblokir sebelum mendapatkan kesempatan untuk menjangkau kotak pesan penerima.

Bagaimana Virtual Deliverability Manager dapat membantu meningkatkan kemampuan pengiriman dan reputasi

Virtual Deliverability Manager membantu Anda meningkatkan kemampuan pengiriman dan reputasi Anda dengan dasbor yang menawarkan tampilan tingkat tinggi dan terperinci dari program email akun Anda untuk membantu fokus pada area bermasalah dan penasihat yang memberikan solusi untuk memulihkan masalah infrastruktur yang berdampak buruk pada pengiriman dan reputasi email Anda.

- Dasbor - Memberikan wawasan tentang data pengiriman Anda yang berfokus pada akun, ISP, identitas pengiriman, dan tingkat set konfigurasi. Ini membantu Anda untuk dengan cepat melihat area dan tren yang bermasalah, dan untuk menangkap kemungkinan masalah sebelum mereka berubah menjadi masalah pengiriman yang lebih besar seperti penolakan sementara (penangguhan) atau blok. Wawasan ini juga akan membantu Anda meningkatkan reputasi pengirim Anda dengan menghitung waktu dan tanggal yang ideal untuk keterlibatan pelanggan dan konversi yang lebih baik untuk kampanye email Anda.

- Advisor - Memberikan rekomendasi untuk meningkatkan pengiriman email Anda dengan menandai masalah konfigurasi yang berdampak negatif terhadap pengiriman dan reputasi email Anda. Ini akan merekomendasikan solusi untuk menyelesaikan masalah spesifik dalam infrastruktur domain pengiriman Anda, ruang IP, dan catatan otentikasi seperti ketika catatan SPF, DMARC, atau DKIM tidak ada, atau jika panjang kunci DKIM terlalu pendek.

Memulai dengan Virtual Deliverability Manager

Untuk mulai menggunakan Virtual Deliverability Manager, panduan orientasi di konsol Amazon SES akan memandu Anda melalui langkah-langkah mengaktifkan Virtual Deliverability Manager untuk akun Anda. Lihat the section called “Memulai”.

Topik

- Memulai dengan Virtual Deliverability Manager
- Dasbor Manajer Pengiriman Virtual
- Penasihat Manajer Pengiriman Virtual
- Pengaturan Virtual Deliverability Manager

Memulai dengan Virtual Deliverability Manager

Untuk mulai menggunakan Virtual Deliverability Manager dengan akun Anda, Anda harus mengaktifkannya menggunakan panduan orientasi di konsol Amazon SES, tempat Anda akan menyiapkan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan. Virtual Deliverability Manager menggunakan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan untuk memantau pengiriman Anda dan untuk membantu Anda meningkatkan kemampuan pengiriman dan reputasi Anda.

- Pelacakan keterlibatan - Kemampuan untuk memantau perilaku keterlibatan penerima melalui peristiwa terbuka dan klik dengan menggunakan piksel pelacakan dalam tautan yang dibungkus. Saat dipicu, piksel pelacakan memberikan stempel waktu kapan pesan dibuka, dan menunjukkan tautan mana yang diklik oleh penerima. Mengaktifkan ini mengubah URL dan tautan Anda untuk menyertakan pembungkus pelacakan keterlibatan Amazon SES.
- Pengiriman bersama yang dioptimalkan - Secara otomatis memilih IP optimal untuk digunakan saat mengirim email, meningkatkan pengiriman pesan titik akhir ke penerima email target. Ini tidak berlaku untuk alamat IP khusus.

Meskipun pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan diaktifkan secara default di panduan orientasi, Anda memiliki opsi untuk mematikannya. Kami sangat menyarankan agar Anda mengaktifkan kedua fitur tersebut untuk mendapatkan hasil maksimal dari Virtual Deliverability Manager.

Memulai Virtual Deliverability Manager menggunakan konsol Amazon SES

Prosedur berikut menunjukkan cara memulai Virtual Deliverability Manager menggunakan konsol Amazon SES.

Untuk memulai dengan Virtual Deliverability Manager menggunakan konsol Amazon SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Virtual Deliverability Manager.
3. Pilih salah satu tombol Mulai dengan Virtual Deliverability Manager di halaman ringkasan Virtual Deliverability Manager.
4. Pada halaman Select Engagement tracking, terima default atau pilih Nonaktifkan pelacakan keterlibatan, lalu pilih Berikutnya.

> **Note**
>
> Mengaktifkan pelacakan keterlibatan mengubah URL dan tautan Anda untuk menyertakan pembungkus pelacakan keterlibatan Amazon SES.

5. Pada halaman Pilih pengiriman bersama yang dioptimalkan, terima default atau pilih Matikan pengiriman bersama yang dioptimalkan, lalu pilih Berikutnya.

> **Important**
>
> Pengiriman bersama yang dioptimalkan dapat mengakibatkan penundaan preemptive untuk email Anda dikirim dalam upaya untuk melindungi reputasi pengiriman Anda. Jika Anda memiliki beban kerja penting yang harus dikirim tanpa penundaan, sebaiknya Anda tidak mengaktifkan pengaturan ini. Sebagai gantinya, gunakan set konfigurasi untuk mengirim, dan hanya aktifkan pengiriman bersama yang dioptimalkan untuk set konfigurasi di mana Anda dapat mengalami penundaan.

6. Tinjau pilihan Anda untuk melacak keterlibatan dan pengiriman bersama yang dioptimalkan di halaman Tinjauan dan aktifkan. Pilih Sebelumnya jika Anda ingin kembali dan membuat perubahan; jika tidak, pilih Aktifkan Virtual Deliverability Manager.

   Halaman pengaturan Virtual Deliverability Manager terbuka. Panel ikhtisar Langganan menunjukkan status Virtual Deliverability Manager dan panel Setelan tambahan menunjukkan status Pelacakan Keterlibatan dan pengiriman bersama yang dioptimalkan.

Setelah mengaktifkan Virtual Deliverability Manager untuk akun Anda, Anda dapat menentukan setelan kustom tentang bagaimana set konfigurasi akan menggunakan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan dengan mengesampingkan cara mereka didefinisikan di Virtual Deliverability Manager. Ini memberi Anda fleksibilitas untuk menyesuaikan pengiriman email Anda untuk kampanye email tertentu. Misalnya, Anda dapat mengaktifkan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan untuk email pemasaran Anda dan menonaktifkannya untuk email transaksional Anda. Lihat opsi Virtual Deliverability Manager saat membuat atau mengedit set konfigurasi.

Memulai dengan Virtual Deliverability Manager menggunakan AWS CLI

Contoh berikut menunjukkan kepada Anda cara memulai dengan Virtual Deliverability Manager menggunakan file. AWS CLI

Untuk memulai dengan Virtual Deliverability Manager menggunakan AWS CLI

Anda dapat menggunakan `PutAccountVdmAttributes`operasi di Amazon SES API v2 untuk memulai dengan Virtual Deliverability Manager. Anda dapat memanggil operasi ini dari AWS CLI, seperti yang ditunjukkan pada contoh berikut.

- Aktifkan Virtual Deliverability Manager di akun Anda:

```
aws --region us-east-1 sesv2 put-account-vdm-attributes --vdm-attributes
 VdmEnabled=ENABLED
```

- Aktifkan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan menggunakan file input:

```
aws --region us-east-1 sesv2 put-account-vdm-attributes --cli-input-json file://
attributes.json
```

File input terlihat mirip dengan ini:

```
{
    "VdmAttributes": {
        "VdmEnabled": "ENABLED",
        "DashboardAttributes": {
            "EngagementMetrics": "ENABLED"
        },
        "GuardianAttributes": {
            "OptimizedSharedDelivery": "ENABLED"
        }
    }
}
```

Nilai parameter dan tipe data terkait dapat ditemukan dengan menautkan dari tipe `VdmAttributes`data di referensi Amazon SES API v2.

> **Note**
>
> Mengaktifkan pelacakan keterlibatan mengubah URL dan tautan Anda untuk menyertakan pembungkus pelacakan keterlibatan Amazon SES.

> **Important**
>
> Pengiriman bersama yang dioptimalkan dapat mengakibatkan penundaan preemptive untuk email Anda dikirim dalam upaya untuk melindungi reputasi pengiriman Anda. Jika Anda memiliki beban kerja penting yang harus dikirim tanpa penundaan, sebaiknya Anda tidak mengaktifkan pengaturan ini. Sebagai gantinya, gunakan set konfigurasi untuk mengirim, dan hanya aktifkan pengiriman bersama yang dioptimalkan untuk set konfigurasi di mana Anda dapat mengalami penundaan.

- Untuk memverifikasi hasilnya:

```
aws --region us-east-1 sesv2 get-account
```

- Untuk menentukan setelan kustom tentang bagaimana set konfigurasi akan menggunakan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan dengan mengesampingkan

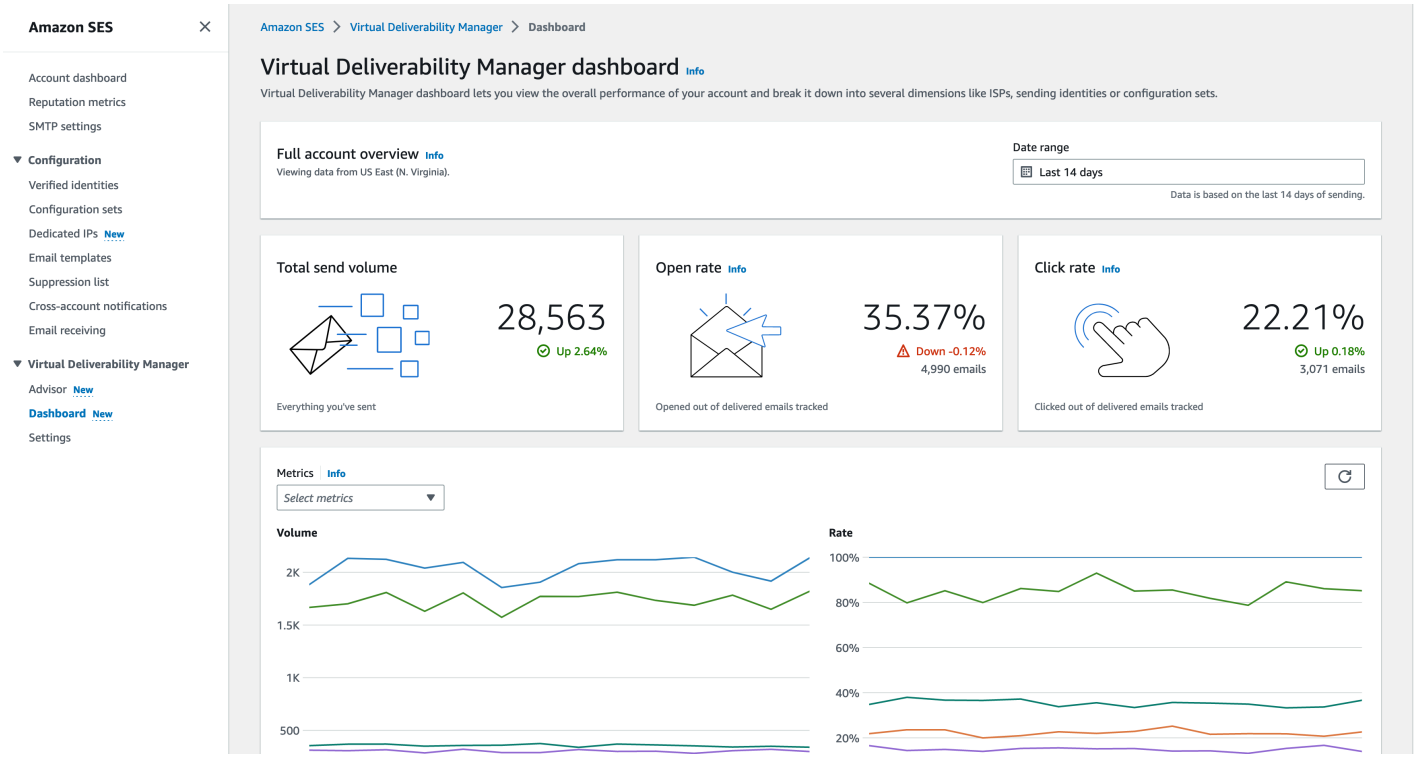
cara mereka didefinisikan di Virtual Deliverability Manager, lihat AWS CLI contoh di. the section called “Pengaturan”

Dasbor Manajer Pengiriman Virtual

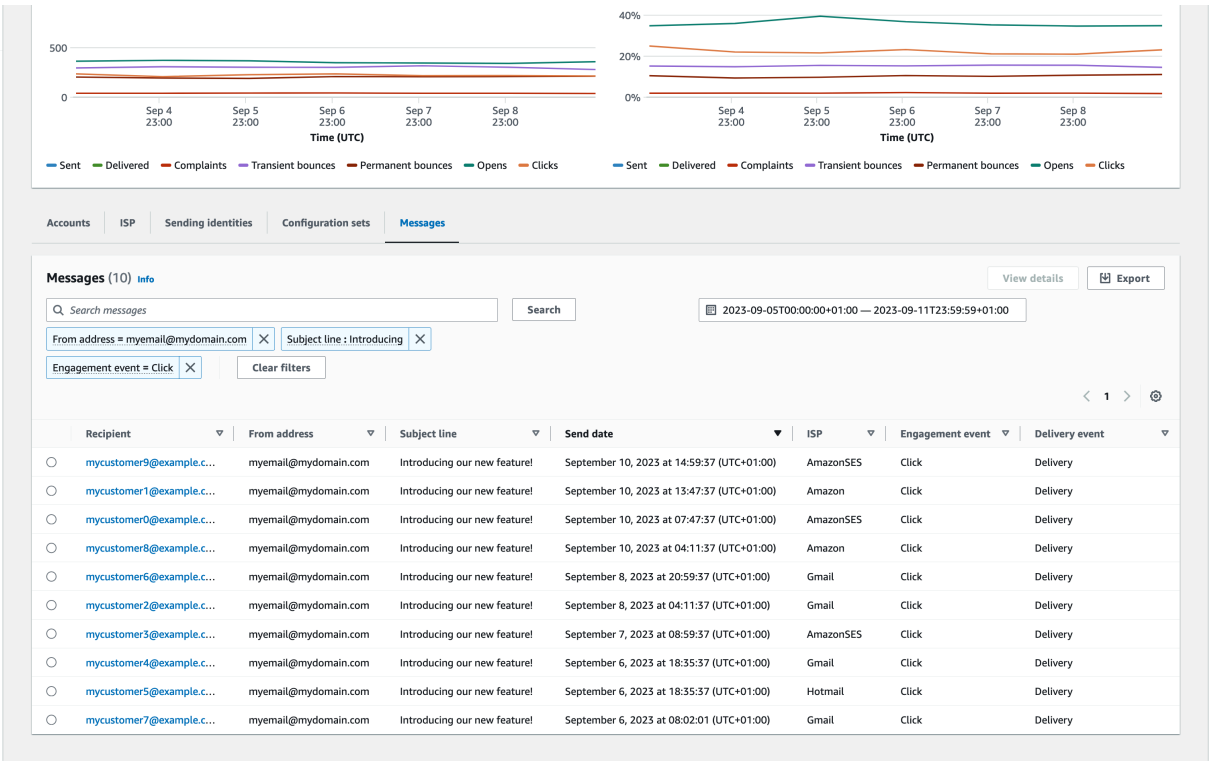
Dasbor menawarkan tampilan tingkat tinggi dari program pengiriman akun Anda, seperti kartu yang mudah dibaca dan grafik deret waktu yang menunjukkan kemampuan pengiriman dan reputasi melalui tingkat keterbukaan/klik dan pengiriman serta statistik pantulan/keluhan. Dasbor juga menawarkan tampilan yang lebih rinci, memungkinkan Anda menelusuri data tabel spesifik yang lebih rinci ketika ada masalah yang terkait dengan ISP tertentu, identitas pengiriman, atau set konfigurasi yang terkait dengan kampanye email.

Mampu melihat hal-hal dari tingkat keseluruhan yang tinggi dengan kemampuan untuk juga melihat detail spesifik memungkinkan Anda untuk fokus pada area bermasalah pengiriman Anda daripada perlu meninjau program email Anda secara keseluruhan. Tingkat wawasan ini juga memberi Anda kemampuan untuk menangkap tren dan kemungkinan masalah sebelum berubah menjadi masalah pengiriman yang lebih besar, seperti penangguhan atau blok.

Ikhtisar akun di dasbor Virtual Deliverability Manager yang menunjukkan grafik kartu dan deret waktu.



Tabel Pesan yang dipilih di dasbor Virtual Deliverability Manager yang menampilkan pesan terkirim yang cocok dengan rentang tanggal dan kriteria filter.

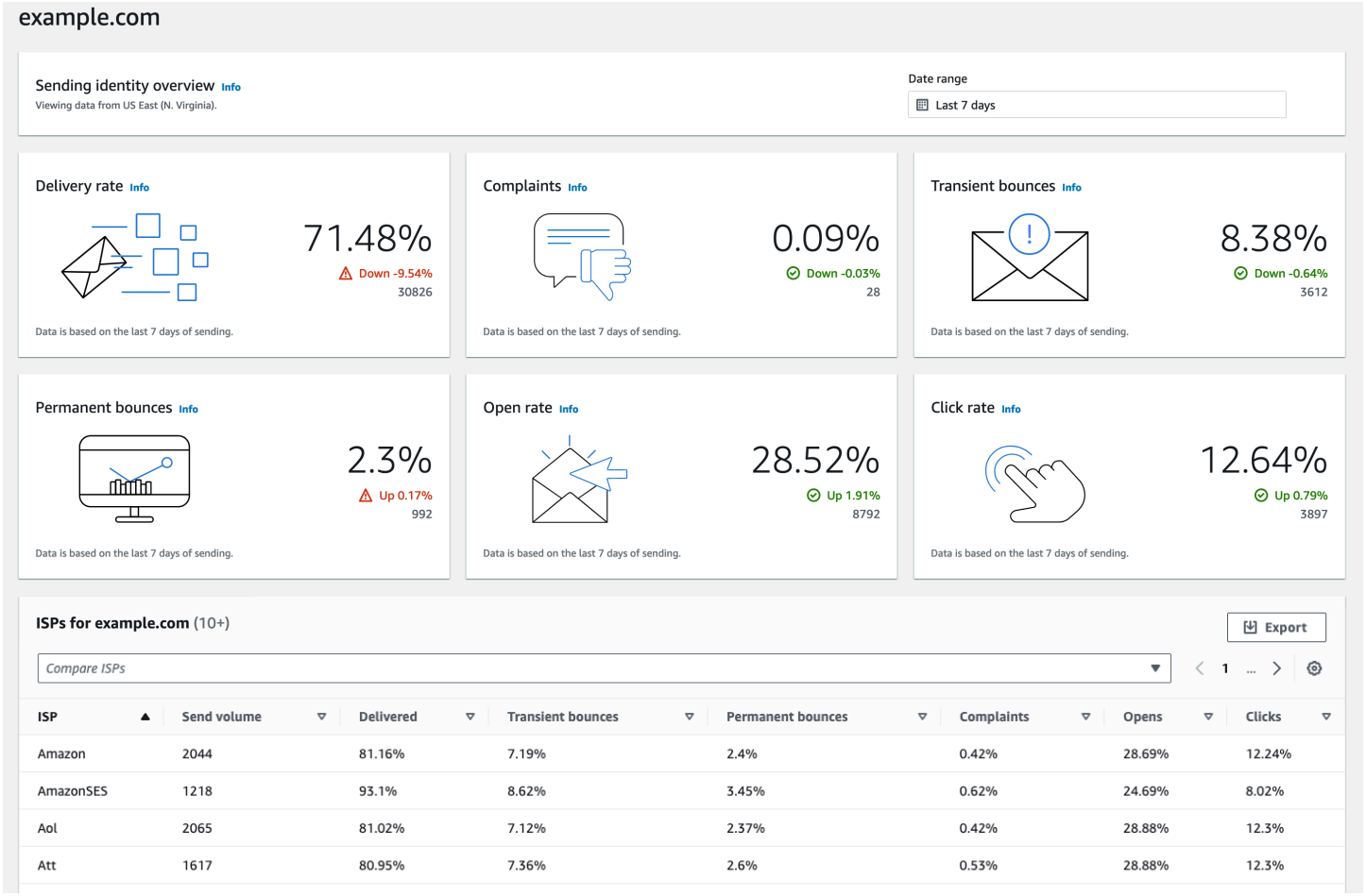


Data terperinci yang disediakan oleh dasbor dapat membantu Anda meningkatkan reputasi pengirim dan menghitung waktu dan tanggal ideal untuk keterlibatan dan konversi yang lebih baik untuk program email Anda dengan kemampuan untuk menelusuri kumpulan data tertentu:

- Data ISP — Berharga jika Anda memiliki masalah pengiriman ke ISP atau penyedia kotak pesan tertentu—alih-alih mencoba menyesuaikan seluruh akun Anda, yang mungkin berjalan dengan baik, Anda dapat fokus pada titik akhir yang bermasalah dan menyelaraskan dengan praktik terbaiknya untuk meningkatkan reputasi pengirim ke ISP tersebut dan mengembalikan kemampuan pengiriman kotak masuk yang baik untuk menjangkau penerima Anda. Penting juga untuk memahami distribusi ISP Anda—karena Anda dapat mengirim lebih banyak ke satu ISP atau penyedia kotak surat daripada yang lain. Anda perlu memastikan bahwa lalu lintas selalu dikirim dan dilibatkan oleh penerima akhir untuk memiliki dampak positif konversi email Anda.
- Mengirim identitas & data set konfigurasi — Berguna dalam membantu Anda mengidentifikasi identitas pengiriman dan set konfigurasi yang berkontribusi terhadap masalah pengiriman akun Anda secara keseluruhan. Anda dapat fokus pada mereka secara khusus, menyesuaikan konfigurasi Anda, dan mungkin mengurangi pengiriman dengan identitas tertentu hingga masalah teratasi. Misalnya, identitas pengirim secara tidak sengaja dikirim ke daftar penindasan, mengakibatkan semua lalu lintas melalui identitas itu. Identitas itu dikaitkan dengan set konfigurasi,

menyebabkan masalah pengiriman. Sangat berharga dalam kasus seperti itu untuk dapat mengidentifikasi identitas pengiriman atau set konfigurasi sehingga Anda dapat fokus pada perbaikan masalah itu secara khusus, daripada menyisir seluruh akun Anda untuk mencoba mengidentifikasi akar penyebab masalah pengiriman.

Telusuri data yang ditampilkan di dasbor Virtual Deliverability Manager untuk identitas pengiriman yang dipilih, example.com —kartu menampilkan metrik pengiriman dan reputasi. Tabel menampilkan semua ISP tempat identitas pengirim mengirim email dengan tarif metrik untuk setiap ISP dalam rentang tanggal yang dimasukkan.



Menggunakan dasbor Virtual Deliverability Manager di konsol Amazon SES

Prosedur berikut menunjukkan kepada Anda cara menggunakan dasbor Virtual Deliverability Manager di konsol Amazon SES untuk melihat statistik pengiriman dan reputasi Anda secara keseluruhan serta menelusuri area bermasalah.

Untuk menggunakan dasbor Virtual Deliverability Manager untuk melihat data tingkat tinggi dan lebih rinci dari metrik pengiriman akun Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Dasbor di bawah Virtual Deliverability Manager.

> **Note**
>
> Dasbor tidak akan terlihat jika Anda belum mengaktifkan Virtual Deliverability Manager untuk akun Anda. Untuk informasi selengkapnya, lihat the section called “Memulai”.

3. Di panel ikhtisar akun lengkap, pilih rentang tanggal yang akan digunakan untuk semua metrik dalam kartu, grafik deret waktu, dan tabel penelusuran.

   - Di bidang Rentang tanggal, pilih Rentang relatif (default) atau Rentang absolut.
     - Rentang relatif — Pilih tombol radio yang sesuai dengan jumlah hari yang diinginkan.
       - Rentang kustom - Masukkan rentang dalam beberapa hari (hingga 60), minggu (hingga 8), atau bulan (hingga 2).
     - Rentang absolut - Tanggal pertama yang Anda pilih adalah tanggal Mulai, tanggal kedua adalah tanggal Akhir, tidak melebihi total 60 hari. Untuk menentukan satu hari, pilih untuk tanggal Mulai dan Akhir.

> **Note**
>
> Berikut ini berlaku untuk semua rentang tanggal di dasbor:
>
> - Semua tanggal & waktu adalah UTC.
> - Untuk tanggal rentang Relatif, hari terakhir berakhir pada stempel waktu tengah malam UTC. Misalnya, jika Anda memilih 7 hari terakhir, hari ketujuh adalah kemarin, berakhir pada tengah malam.
> - Jika rentang tanggal lebih besar dari 30 hari, kolom% Perbedaan dalam tabel statistik Akun dan persentase perubahan dalam kartu tidak akan memiliki nilai (ditunjukkan dengan tanda hubung-).

4. Kartu, grafik deret waktu, dan semua tabel penelusuran, statistik Akun, ISP, identitas pengiriman, dan set Konfigurasi, menampilkan total metrik yang dihitung dari rentang tanggal yang dimasukkan, dan menggunakan matematika metrik yang dijelaskan dalam. Bagaimana metrik dasbor dihitung

   - Untuk membuat `.csv` file lokal dari data yang sedang Anda lihat di ISP, Mengirim identitas, atau tabel set Konfigurasi, pilih tombol Ekspor.

5. Grafik deret waktu yang memetakan Volume dan Perkembangan tingkat untuk rentang tanggal yang Anda masukkan ditampilkan di panel Metrik. Melayang di atas interval tanggal dalam grafik akan menunjukkan jumlah volume atau persentase tingkat yang tepat berdasarkan agregasi harian. Anda dapat memfilter metrik yang ingin Anda lihat menggunakan menu tarik-turun Pilih metrik.

6. Pilih tab Accounts untuk menampilkan tabel statistik Accounts.

   - Tabel ini memberikan ikhtisar metrik deliverability dan reputasi Anda, yang menunjukkan total Volume,% Rate, dan% Difference untuk Terkirim, Terkirim, Keluhan, Transient & Permanen, Opens & Clicks yang dihitung dari rentang tanggal yang dimasukkan.

   > **Note**
   >
   > Jika rentang tanggal lebih besar dari 30 hari, kolom% Perbedaan tidak akan memiliki nilai (ditunjukkan oleh tanda hubung`-`).

7. Pilih tab ISP untuk menampilkan tabel ISP.

   - Tabel ini menampilkan metrik untuk volume Kirim, Terkirim, Pantulan Transien & Permanen, Keluhan, Buka & Klik untuk setiap ISP yang Anda kirim sesuai perhitungan dari rentang tanggal yang dimasukkan.
   - Untuk memfilter ISP tertentu, di dalam kotak pencarian Bandingkan ISP, pilih kotak centang yang sesuai untuk setiap ISP yang akan disertakan.
   - Untuk membuat `.csv` file lokal dari data yang sedang Anda lihat di tabel ini, pilih tombol Ekspor.

8. Pilih tab Mengirim identitas untuk menampilkan tabel Mengirim identitas.

   - Tabel ini menampilkan metrik untuk volume Kirim, Terkirim, Pantulan Transien & Permanen, Keluhan, Buka & Klik untuk setiap identitas pengiriman yang Anda gunakan yang dihitung dari rentang tanggal yang dimasukkan.

- Untuk memfilter identitas pengiriman tertentu, di dalam kotak pencarian Bandingkan identitas, pilih kotak centang yang sesuai untuk setiap identitas yang akan disertakan.
- Untuk menelusuri identitas pengiriman tertentu, pilih namanya di kolom Mengirim identitas.
  - Kartu akan muncul menampilkan Delivery rate, Complaints, Transient & Permanent bounce, Open & Click rates untuk identitas pengiriman yang dipilih sebagaimana dihitung dari rentang tanggal yang dimasukkan.
  - Grafik deret waktu akan disegarkan menampilkan semua metrik untuk identitas pengiriman yang dipilih sebagaimana dihitung dari rentang tanggal yang dimasukkan.
  - Tabel ISP akan ditampilkan daftar semua ISP identitas pengirim email yang dikirim dengan metrik yang diberikan untuk setiap ISP yang dihitung dari rentang tanggal yang dimasukkan.
- Untuk membuat `.csv` file lokal dari data yang sedang Anda lihat di tabel ini, pilih tombol Ekspor.

9. Pilih tab Set konfigurasi untuk menampilkan tabel set Konfigurasi.

   - Tabel ini menampilkan metrik untuk volume Kirim, Terkirim, Pantulan Transien & Permanen, Keluhan, Buka & Klik untuk setiap set konfigurasi yang telah digunakan untuk mengirim email sebagaimana dihitung dari rentang tanggal yang dimasukkan.
   - Untuk memfilter set konfigurasi tertentu, di dalam kotak pencarian Bandingkan set konfigurasi, pilih kotak centang yang sesuai untuk setiap set konfigurasi yang akan disertakan.
   - Untuk menelusuri set konfigurasi tertentu, pilih namanya di kolom set Konfigurasi.
     - Kartu akan muncul menampilkan Delivery rate, Complaints, Transient & Permanent bounce, Open & Click rates untuk konfigurasi yang dipilih yang ditetapkan sebagaimana dihitung dari rentang tanggal yang dimasukkan.
     - Grafik deret waktu akan disegarkan menampilkan semua metrik untuk set konfigurasi yang dipilih sebagaimana dihitung dari rentang tanggal yang dimasukkan.
     - Tabel ISP akan ditampilkan mencantumkan semua ISP yang digunakan set konfigurasi untuk mengirim email dengan metrik yang diberikan untuk setiap ISP yang dihitung dari rentang tanggal yang dimasukkan.
   - Untuk membuat `.csv` file lokal dari data yang sedang Anda lihat di tabel ini, pilih tombol Ekspor.

10. Pilih tab Pesan untuk menampilkan tabel Pesan.

    Ini adalah tabel interaktif yang menyediakan cara bagi Anda untuk mencari dan menemukan pesan yang Anda kirim. Untuk setiap pesan, Anda dapat melacak status pengiriman dan

keterlibatan saat ini, riwayat peristiwa, dan melihat respons yang ditampilkan oleh penyedia kotak pesan. Poin-poin berikut mencakup cara Anda dapat mencari pesan tertentu:

- Memilih di dalam pemilih rentang tanggal, Anda dapat memfilter pesan yang telah Anda kirim dalam 30 hari terakhir. Jika Anda tidak memilih rentang tanggal, pencarian Anda akan default ke 7 hari terakhir termasuk hari ini dalam zona waktu Anda.
- Di bidang Pesan pencarian, Anda dapat memfilter pada Penerima, Dari alamat, Baris subjek, ISP, Acara Keterlibatan, Acara pengiriman, dan ID Pesan - properti berikut berlaku:
  - Bergantung pada jenis filter, Anda memasukkan string teks peka huruf besar/kecil, atau memilih nilai dari daftar.
  - Acara keterlibatan terbatas pada satu nilai, baris Subjek dapat memiliki hingga dua nilai, dan semua filter lainnya dapat memiliki hingga lima nilai per pencarian. Pemfilteran berdasarkan ID Pesan akan mengecualikan filter lain yang mungkin telah Anda pilih termasuk rentang tanggal.
  - Kolom ID Pesan disembunyikan secara default, tetapi dapat ditampilkan dengan memilih ikon roda gigi untuk menyesuaikan cara Anda melihat tabel Pesan.
- Setelah Anda memilih filter dan rentang tanggal, pilih Cari dan tabel akan diisi dengan pesan yang cocok dengan kriteria pencarian Anda. Tabel dapat memuat hingga 100 pesan. Jika pencarian Anda mengembalikan lebih dari 100 pesan, 100 pesan dalam tabel adalah sampel acak dari total yang dikembalikan.
- Memilih tombol radio pesan yang diikuti dengan memilih Lihat detail akan menghasilkan bilah sisi Info pesan yang berisi detail riwayat peristiwa lengkap pesan, yang terbaru di bagian atas, dan tanggapan atau kode diagnostik apa pun yang dikembalikan oleh penyedia kotak pesan.
- Untuk membuat `.csv` file lokal dari data yang sedang Anda lihat di tabel ini, pilih tombol Ekspor.

Mengakses data metrik Virtual Deliverability Manager Anda menggunakan AWS CLI

Contoh berikut menunjukkan cara mengakses data metrik Virtual Deliverability Manager menggunakan. AWS CLI Ini adalah data yang sama yang digunakan di dasbor Virtual Deliverability Manager di konsol.

Untuk mengakses data metrik kiriman Anda menggunakan AWS CLI

Anda dapat menggunakan BatchGetMetricDataoperasi di Amazon SES API v2 untuk mengakses data metrik pengiriman Anda. Anda dapat memanggil operasi ini dari AWS CLI seperti yang ditunjukkan pada contoh berikut.

- Akses data metrik pengiriman Anda:

```
aws --region us-east-1 sesv2 batch-get-metric-data --cli-input-json file://sends.json
```

- File input terlihat mirip dengan ini:

```
{
 "Queries": [
   {
     "Id": "Retrieve-Account-Sends",
     "Namespace": "VDM",
     "Metric": "SEND",
     "StartDate": "2022-11-04T00:00:00",
     "EndDate": "2022-11-05T00:00:00"
    }
 ]
}
```

Informasi selengkapnya tentang nilai parameter dan tipe data terkait dapat ditemukan dengan menautkan dari tipe BatchGetMetricDataQuerydata di referensi Amazon SES API v2.

Memfilter dan mengekspor data metrik pengiriman Anda menggunakan AWS CLI

Contoh ini menunjukkan cara menggunakan CreateExportJoboperasi untuk memfilter dan mengekspor data metrik deliverability Anda ke file.csv atau .json menggunakan file. AWS CLI Ini adalah data yang sama yang digunakan di ISP dasbor Virtual Deliverability Manager, Mengirim identitas, dan tabel set Konfigurasi.

Untuk memfilter dan mengekspor data metrik deliverability Anda ke file.csv atau .json menggunakan AWS CLI

Anda dapat menggunakan CreateExportJoboperasi bersama dengan tipe MetricsDataSourcedata di Amazon SES API v2 untuk memfilter dan mengekspor data metrik

Anda ke file.csv atau .json. Anda memanggil operasi ini dari AWS CLI seperti yang ditunjukkan pada contoh berikut.

- Filter dan ekspor data metrik pengiriman Anda menggunakan file input:

```
aws --region us-east-1 sesv2 create-export-job --cli-input-json file://metric-export-input.json
```

- Dalam contoh ini, file input menggunakan `MetricsDataSource`parameter untuk memfilter semua ISP yang Anda kirimi email, menunjukkan tingkat keberhasilan pengiriman dalam rentang tanggal tertentu, dan format.csv yang ditentukan untuk file keluaran:

```
{
    "ExportDataSource": {
        "MetricsDataSource": {
            "Dimensions": {
                "ISP": ["*"]
            },
            "Namespace": "VDM",
            "Metrics": [
                {
                    "Name": "DELIVERY",
                    "Aggregation": "RATE"
                }
            ],
            "StartDate": "2023-06-13T00:00:00",
            "EndDate": "2023-06-20T00:00:00"
        }
    },
    "ExportDestination": {
        "DataFormat": "CSV"
    }
}
```

Informasi selengkapnya tentang nilai parameter dan tipe data terkait dapat ditemukan `MetricsDataSource`sebagai objek tipe `ExportDataSource`di referensi Amazon SES API v2.

Menemukan pesan terkirim, status pengiriman & keterlibatannya, dan mengekspor hasilnya menggunakan AWS CLI

Contoh ini menunjukkan cara menggunakan `CreateExportJob`operasi untuk mencari dan menemukan pesan tertentu yang telah Anda kirim, melihat status pengiriman dan keterlibatan mereka saat ini, dan mengekspor hasil pencarian Anda ke file.csv atau .json menggunakan file. AWS CLI Ini adalah data yang sama yang digunakan dalam tabel Pesan dasbor Virtual Deliverability Manager.

Untuk menemukan pesan terkirim, status pengiriman dan keterlibatannya, dan mengekspor hasilnya ke file.csv atau .json menggunakan AWS CLI

Anda dapat menggunakan `CreateExportJob`operasi bersama dengan tipe `MessageInsightsDataSource`data di Amazon SES API v2 untuk menerapkan filter guna menemukan pesan tertentu yang telah Anda kirim, melihat status pengiriman dan keterlibatannya, dan mengekspor hasilnya ke file.csv atau .json. Anda memanggil operasi ini dari AWS CLI seperti yang ditunjukkan dalam contoh berikut.

> **Note**
>
> Jika penelusuran Anda yang difilter menampilkan lebih dari 10.000 pesan, 10.000 pesan dalam kumpulan hasil API adalah contoh acak dari total yang dikembalikan.

- Temukan pesan terkirim, lihat statusnya saat ini, dan ekspor hasil menggunakan file input:

```
aws --region us-east-1 sesv2 create-export-job --cli-input-json file://message-insights-export-input.json
```

- Dalam contoh ini, file input menggunakan `MessageInsightsDataSource`parameter untuk memfilter pada subjek yang sama dengan “Penjualan Berakhir Malam Ini!” , dan format.csv yang ditentukan untuk file output:

```
{
    "ExportDataSource": {
        "MessageInsightsDataSource": {
            "StartDate": "2023-07-01T00:00:00",
            "EndDate": "2023-07-10T00:00:00",
            "Include": {
                "Subject": [
                    "Sale Ends Tonight!"
```

```
                ]
            }
        }
    },
    "ExportDestination": {
        "DataFormat": "CSV"
    }
}
```

- Dalam contoh ini, file input menggunakan `MessageInsightsDataSource`parameter untuk memfilter subjek yang dimulai dengan “Halo”, dikirim dengan FromEmailAddress berisi “informasi” ke tujuan yang diakhiri dengan “@example .com”, dan format.json yang ditentukan untuk file keluaran:

```
{
    "ExportDataSource": {
        "MessageInsightsDataSource": {
            "StartDate": "2023-07-01T00:00:00",
            "EndDate": "2023-07-10T00:00:00",
            "Include": {
                "Subject": [
                    "Hello*"
                ],
                "FromEmailAddress": [
                    "*information*"
                ],
                "Destination": [
                    "*@example.com"
                ]
            }
        }
    },
    "ExportDestination": {
        "DataFormat": "JSON"
    }
}
```

- Dalam contoh ini, file input menggunakan `MessageInsightsDataSource`parameter untuk memfilter subjek yang dimulai dengan “Halo”, kecualikan hasil yang memiliki

"noreply@example.com" sebagai FromEmailAddress, dan format.csv yang ditentukan untuk file keluaran:

```
{
    "ExportDataSource": {
        "MessageInsightsDataSource": {
            "StartDate": "2023-07-01T00:00:00",
            "EndDate": "2023-07-10T00:00:00",
            "Include": {
                "Subject": [
                    "Hello*"
                ]
            },
            "Exclude": {
                "FromEmailAddress": [
                    "noreply@example.com"
                ]
            }
        }
    },
    "ExportDestination": {
        "DataFormat": "CSV"
    }
}
```

- Dalam contoh ini, file input menggunakan `MessageInsightsDataSource`parameter untuk memfilter subjek yang dimulai dengan “Halo”, dikirim dengan FromEmailAddress berisi “informasi” ke tujuan yang diakhiri dengan “@example .com”, menggunakan Gmail sebagai ISP, acara pengiriman terakhir “PENGIRIMAN”, acara keterlibatan terakhir yang “BUKA” atau “KLIK”, dan format.json yang ditentukan untuk file keluaran:

```
{
    "ExportDataSource": {
        "MessageInsightsDataSource": {
            "StartDate": "2023-07-01T00:00:00",
            "EndDate": "2023-07-10T00:00:00",
            "Include": {
                "Subject": [
                    "Hello*"
                ],
                "FromEmailAddress": [
```

```
                    "*information*"
                ],
                "Destination": [
                    "*@example.com"
                ],
                "Isp": [
                    "Gmail"
                ],
                "LastDeliveryEvent": [
                    "DELIVERY"
                ],
                "LastEngagementEvent": [
                    "OPEN", "CLICK"
                ]
            }
        }
    },
    "ExportDestination": {
        "DataFormat": "JSON"
    }
}
```

- Dalam contoh ini, file input menggunakan `MessageInsightsDataSource`parameter untuk memfilter tujuan yang diakhiri dengan “@example1 .com”, atau “@example2 .com”, atau “@example3 .com”, mengecualikan pesan yang LastDeliveryEvent sama dengan “KIRIM” atau “PENGIRIMAN”, dan format.csv yang ditentukan untuk file keluaran:

```
{
    "ExportDataSource": {
        "MessageInsightsDataSource": {
            "StartDate": "2023-07-01T00:00:00",
            "EndDate": "2023-07-10T00:00:00",
            "Include": {
                "Destination": [
                    "*@example1.com",
                    "*@example2.com",
                    "*@example3.com"
                ]
            },
            "Exclude": {
                "LastDeliveryEvent": [
                    "SEND",
```

```
                    "DELIVERY"
                ]
            }
        }
    },
    "ExportDestination": {
        "DataFormat": "CSV"
    }
}
```

Informasi selengkapnya tentang nilai parameter dan tipe data terkait dapat ditemukan `MessageInsightsDataSource`sebagai objek tipe `ExportDataSource`di referensi Amazon SES API v2.

Mengelola pekerjaan ekspor Anda menggunakan AWS CLI

Contoh-contoh ini menunjukkan kepada Anda cara mengelola pekerjaan ekspor Anda dengan mencantumkannya, mendapatkan informasi tentang mereka, dan membatalkannya menggunakan. AWS CLI

Untuk membuat daftar pekerjaan ekspor Anda menggunakan AWS CLI

Anda dapat menggunakan `ListExportJobs`operasi di Amazon SES API v2 untuk daftar pekerjaan ekspor Anda. Anda dapat memanggil operasi ini dari AWS CLI seperti yang ditunjukkan pada contoh berikut.

- Daftar pekerjaan ekspor Anda:

```
aws --region us-east-1 sesv2 list-export-jobs --export-source-type=METRICS_DATA
```

```
aws --region us-east-1 sesv2 list-export-jobs --job-status=CREATED
```

```
aws --region us-east-1 sesv2 list-export-jobs --cli-input-json file://list-export-jobs-input.json
```

- File input terlihat mirip dengan ini:

```
{
  "NextToken": "",
```

```
  "PageSize": 0,
  "ExportSourceType": "METRICS_DATA",
  "JobStatus": "CREATED"
}
```

Informasi selengkapnya tentang nilai parameter untuk `ListExportJobs`operasi dapat ditemukan di referensi Amazon SES API v2.

Untuk mendapatkan informasi tentang pekerjaan ekspor Anda menggunakan AWS CLI

Anda dapat menggunakan `GetExportJob`operasi di Amazon SES API v2 untuk mendapatkan informasi tentang pekerjaan ekspor Anda. Anda dapat memanggil operasi ini dari AWS CLI seperti yang ditunjukkan pada contoh berikut.

- Dapatkan informasi tentang pekerjaan ekspor Anda:

```
aws --region us-east-1 sesv2 get-export-job --job-id=<JobId>
```

```
aws --region us-east-1 sesv2 get-export-job --cli-input-json file://get-export-job-input.json
```

- File input terlihat mirip dengan ini:

```
{
    "JobId": "e2220d6b-dce5-45f2-bf60-3287a465b732"
}
```

Informasi selengkapnya tentang nilai parameter untuk `GetExportJob`operasi dapat ditemukan di referensi Amazon SES API v2.

Untuk membatalkan pekerjaan ekspor Anda menggunakan AWS CLI

Anda dapat menggunakan `CancelExportJob`operasi di Amazon SES API v2 untuk membatalkan pekerjaan ekspor Anda. Anda dapat memanggil operasi ini dari AWS CLI seperti yang ditunjukkan pada contoh berikut.

- Batalkan pekerjaan ekspor Anda:

```
aws --region us-east-1 sesv2 cancel-export-job --job-id=<JobId>
```

```
aws --region us-east-1 sesv2 cancel-export-job --cli-input-json file://cancel-export-job-input.json
```

- File input terlihat mirip dengan ini:

```
{
    "JobId": "e2220d6b-dce5-45f2-bf60-3287a465b732"
}
```

Informasi selengkapnya tentang nilai parameter untuk `CancelExportJob`operasi dapat ditemukan di referensi Amazon SES API v2.

Melihat riwayat acara lengkap pesan dan tanggapan ISP menggunakan AWS CLI

Contoh berikut menunjukkan cara melihat detail riwayat peristiwa lengkap pesan dan respons atau kode diagnostik apa pun yang ditampilkan oleh penyedia kotak pesan menggunakan. AWS CLI Ini adalah data yang sama yang digunakan di sidebar Info pesan setelah memilih tombol radio pesan di tabel Pesan dasbor Virtual Deliverability Manager.

Untuk melihat riwayat acara pesan dan respons ISP menggunakan AWS CLI

Anda dapat menggunakan `GetMessageInsights`operasi di Amazon SES API v2 untuk melihat detail pesan terkirim. Anda dapat memanggil operasi ini dari AWS CLI seperti yang ditunjukkan pada contoh berikut.

- Lihat detail pesan tentang email terkirim yang diidentifikasi oleh id pesannya:

```
aws --region us-east-1 sesv2 get-message-insights --message-id
 01000100001000dd-2a19190d-99d4-0000-9f00-deb5bbf2bfbe-000001
```

Informasi selengkapnya tentang nilai parameter untuk `GetMessageInsights`operasi dapat ditemukan di referensi Amazon SES API v2.

Bagaimana metrik dasbor Virtual Deliverability Manager dihitung

Semua kartu tarif dan tabel penelusuran yang ditampilkan di dasbor Virtual Deliverability Manager menghitung metrik untuk rentang tanggal yang dimasukkan dalam panel ikhtisar akun lengkap.

Persentase tingkat metrik yang ditampilkan di dasbor dihitung seperti yang dijelaskan dalam tabel. Empat kolom terakhir mewakili kualifikasi untuk matematika dasar yang digunakan untuk menurunkan metrik yang ditampilkan. Misalnya, Rasio terbuka Anda dihitung sebagai total terbuka dibagi dengan total pengiriman untuk pesan HTML yang dikirimkan dengan pelacakan keterlibatan diaktifkan. Mereka tidak mencerminkan pesan apa pun yang Anda kirim tanpa pelacakan keterlibatan dan tidak dikodekan HTML.

Tingkat%	Bagaimana itu dihitung	Dengan pelacakan keterlibatan diaktifkan & HTML	Dan dengan setidaknya 1 tautan yang dilacak	Dikirim ke ISP dengan SES FBL	Dikecualikan jika ada di daftar penekanan tingkat akun Anda
Tarif terbuka	total terbuka/total terkirim	✗			
Tingkat klik	klik total/total terkirim	✗	✗		
Tingkat keluhan	total keluhan/total terkirim			✗	✗
Tingkat pengiriman	total dikirim/total terkirim				
Tingkat pentalan sementara	total pantulan transien/total terkirim				✗
Tingkat pentalan permanen	total bouncing permanen/total terkirim				✗
Total volume kirim	Nilai% tidak ditampilkan (semua yang Anda kirim; selalu 100%)				

Bagaimana tingkat perbedaan dan total volume dihitung untuk semua metrik:

- Selisih% — Perbedaan total metrik dibandingkan dengan total metrik sebelumnya untuk rentang tanggal yang diberikan. Misalnya, jika 7 hari terakhir adalah rentang tanggal yang ditentukan, Tingkat metrik 7 hari terakhir - Tingkat metrik 7 hari sebelumnya.
  - Perbedaan% untuk Total volume kirim dihitung secara berbeda. Misalnya, (Kirim volume 7 hari terakhir - Kirim volume 7 hari sebelumnya)/Kirim volume 7 hari sebelumnya.
- Volume — Jumlah total setiap metrik.

Note

- Kolom Terkirim dalam tabel penelusuran menampilkan volume pengiriman langsung tanpa kualifikasi yang dikirim yang digunakan untuk menghitung tingkat buka, klik, dan keluhan.
- Virtual Deliverability Manager hanya melacak metrik dari email yang memiliki satu penerima—email dengan beberapa penerima tidak dihitung dalam metrik dasbor Virtual Deliverability Manager mana pun.
  - Dalam kasus ini, jumlah metrik Virtual Deliverability Manager Anda akan lebih rendah daripada jumlah CloudWatch metrik Amazon Anda karena CloudWatch metrik menyertakan email dengan beberapa penerima.
- Email yang dikirim ke simulator kotak surat SES tidak dihitung dalam metrik dasbor Virtual Deliverability Manager mana pun.
- Email yang dikirim melalui akun pengirim delegasi (sebelumnya pengiriman lintas akun) tidak dihitung dalam metrik dasbor Virtual Deliverability Manager mana pun.

Important

Perlindungan Privasi Apple Mail dan dampaknya terhadap tingkat keterlibatan: Sebagai hasil dari Apple menerapkan fitur Perlindungan Privasi Mail (MPP) mereka untuk perangkat Apple pada iOS15, nomor keterlibatan telah meningkat saat pemicu MPP terbuka saat aplikasi Apple Mail dimulai, tidak harus ketika penerima membuka dan/atau mengklik pesan. Hal ini menyebabkan data keterlibatan terlihat jauh lebih tinggi daripada biasanya dan ini adalah sesuatu yang pemasar email harus mempertimbangkan ketika meninjau keterlibatan. Ada beberapa cara lain untuk mengidentifikasi keterlibatan, seperti aktivitas web, penggunaan aplikasi/portal dan juga menggunakan data proxy dari perangkat non-Apple

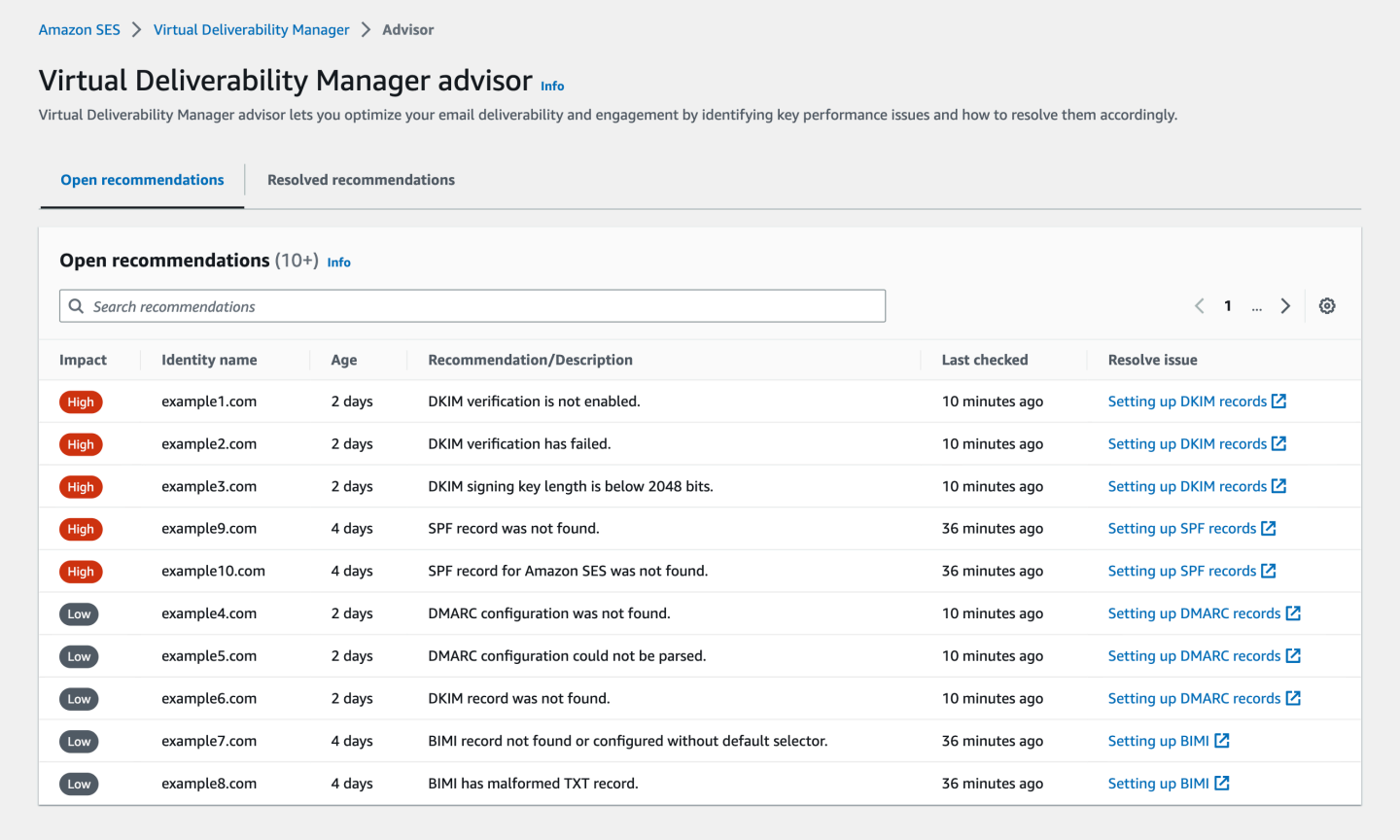
untuk membangun metrik agregat. Yang penting untuk difokuskan adalah tren keterlibatan karena itu dapat menunjukkan jika ada masalah dengan pengiriman email Anda. Untuk informasi selengkapnya, lihat Perlindungan Privasi Apple Mail.

Penasihat Manajer Pengiriman Virtual

Penasihat Virtual Deliverability Manager membantu mengoptimalkan pengiriman dan keterlibatan email Anda dengan mengidentifikasi masalah kinerja dan infrastruktur utama di akun dan mengirimkan tingkat identitas yang berdampak buruk pada pengiriman dan reputasi email Anda. Ini memberikan solusi dengan memberikan panduan khusus tentang cara menyelesaikan masalah yang diidentifikasi.

Rekomendasi infrastruktur penasihat tercantum dalam tabel Rekomendasi terbuka. Rekomendasi mengidentifikasi masalah otentikasi email standar, seperti ketika data SPF, DKIM, DMARC, atau BIMI tidak ada atau memiliki masalah dengan konfigurasi mereka seperti cacat atau memiliki panjang kunci yang terlalu pendek. Mereka dikategorikan berdasarkan tingkat keparahan Dampak, Nama identitas domain pengirim, dan Usia peringatan. Di bilah pencarian, kotak daftar menyediakan opsi untuk memfilter pada tingkat dampak, kategori infrastruktur, atau mengirim nama identitas. Kolom terakhir yang dicentang menunjukkan waktu relatif kapan rekomendasi terakhir diperbarui, seperti “Baru saja” atau “15 menit yang lalu”. Kolom terakhir, Selesaikan masalah, menyediakan tautan ke bagian yang relevan di Panduan Pengembang Amazon SES dengan panduan tentang cara mengatasi masalah yang diidentifikasi.

Rekomendasi terbuka ditampilkan di penasihat Virtual Deliverability Manager yang diurutkan berdasarkan tingkat dampak.



Jika Anda tidak memiliki pemberitahuan penasihat yang sedang berlangsung, sebuah pesan akan menunjukkan bahwa Anda tidak memiliki rekomendasi terbuka. Kami menyarankan Anda memeriksa penasihat secara teratur. Secara opsional, Anda dapat mengintegrasikan peristiwa notifikasi penasihat ini dengan Amazon EventBridge untuk membuat aplikasi berbasis peristiwa yang dapat diskalakan seperti yang dijelaskan dalam. Pemantauan menggunakan EventBridge

Anda juga dapat mengakses tabel Rekomendasi Terselesaikan dari halaman penasihat Virtual Deliverability Manager, yang mencantumkan masalah infrastruktur yang telah Anda selesaikan dengan menerapkan panduan penasihat. Rekomendasi yang diselesaikan dicantumkan dengan status awal yang menjelaskan masalah sebelum diselesaikan. Rekomendasi yang diselesaikan kedaluwarsa setelah 30 hari.

Apa yang dicari oleh penasihat Virtual Deliverability Manager

Pada bagian sebelumnya, kami membahas bahwa penasihat Virtual Deliverability Manager melakukan pemeriksaan terhadap domain pengirim Anda untuk menentukan apakah Anda telah mengonfigurasi infrastruktur yang diautentikasi dengan aman untuk memastikan Anda mempertahankan tingkat pengiriman email yang tinggi dan mempertahankan reputasi pengirim yang

baik. Sebelum Anda mengaktifkan penasihat Virtual Deliverability Manager, kami pikir akan sangat membantu bagi Anda untuk mengetahui dengan tepat apa yang diperiksa penasihat dan apa yang dicari dalam pemeriksaan tersebut.

Anda dapat menggunakan tabel ini sebagai referensi untuk menelusuri konfigurasi domain pengiriman Anda dan memperbaiki elemen-elemen ini yang tidak selaras dengan standar yang tercantum dalam tabel ini sebelum menjadi masalah yang harus diperingatkan oleh penasihat.

Jenis cek	Pesan penasihat	Mengapa penasihat memperingatkan Anda	Pelajari selengkapnya
Pemeriksaan tingkat keluhan	*ISP_name ISP memiliki tingkat keluhan tinggi/med/rendah.*	Identitas telah melampaui ambang batas rekomendasi keluhan untuk ISP ini.	Pemantauan reputasi pengirim
Konfigurasi DKIM	Verifikasi DKIM tidak diaktifkan.	DKIM tidak diaktifkan per identitas.	DKIM Mudah di SES
Kekuatan kunci DKIM	Panjang kunci penandatanganan DKIM di bawah 2048 bit.	Panjang kunci penandatanganan DKIM tidak menggunakan setidaknya 2048 bit.	DKIM Mudah di SES
Validasi catatan DNS DKIM	Verifikasi DKIM gagal.	Catatan DKIM CNAME ditentukan tidak valid setelah mencari dan mencoba memvalidasi kunci.	Memverifikasi identitas domain DKIM dengan penyedia DNS Anda
Konfigurasi DMARC	Konfigurasi DMARC tidak ditemukan.	Catatan DMARC TXT hilang.	Menyiapkan kebijakan DMARC di domain Anda

Jenis cek	Pesan penasihat	Mengapa penasihat memperingatkan Anda	Pelajari selengkapnya
Pemeriksaan format catatan DNS DMARC	Konfigurasi DMARC tidak dapat diurai.	Format tidak valid ditemukan untuk catatan DMARC TXT.	Menyiapkan kebijakan DMARC di domain Anda
Konfigurasi DKIM DMARC	Catatan DKIM tidak ditemukan.	Tidak ada catatan DKIM yang ditemukan untuk mematuhi DMARC.	Mematuhi DMARC melalui DKIM
Konfigurasi DKIM DMARC	Catatan DKIM tidak selaras.	Domain yang ditentukan dalam tanda tangan DKIM tidak sejajar (cocok) dengan domain di alamat Dari.	Mematuhi DMARC melalui DKIM
Konfigurasi SPF	Catatan SPF tidak ditemukan.	Rekaman SPF TXT hilang untuk domain Custom MAIL FROM.	Mengonfigurasi domain MAIL FROM kustom
SPF “termasuk” dikonfigurasi	Catatan SPF untuk Amazon SES tidak ditemukan.	`include:amazonses.com` hilang dari catatan SPF TXT.	Mengonfigurasi domain MAIL FROM kustom
Penegakan SPF dikonfigurasi	SPF semua kualifikasi tidak ada.	`~all`hilang dari catatan SPF TXT.	Mengonfigurasi domain MAIL FROM kustom
Validasi penegakan SPF	Masalah konfigurasi SPF ditemukan.	Upaya untuk mendeteksi catatan SPF MX yang diperlukan dalam waktu 72 jam gagal.	KUSTOM MAIL DARI status pengaturan domain

Jenis cek	Pesan penasihat	Mengapa penasihat memperingatkan Anda	Pelajari selengkapnya
BIMI dikonfigurasi	Catatan BIMI tidak ditemukan atau dikonfigurasi tanpa pemilih default.	Catatan BIMI TXT hilang atau tidak memiliki atribut pemilih.	Menyiapkan BIMI
Validasi format BIMI	BIMI memiliki catatan TXT yang salah.	Catatan BIMI TXT ditentukan sebagai salah konfigurasi setelah memeriksa keberadaan dan format yang valid dari: versi, URL sertifikat, dan URL logo.	Menyiapkan BIMI

Menggunakan penasihat Virtual Deliverability Manager di konsol Amazon SES

Prosedur berikut menunjukkan kepada Anda cara menggunakan penasihat Virtual Deliverability Manager di konsol Amazon SES untuk mengatasi masalah pengiriman yang teridentifikasi menggunakan konsol Amazon SES.

Untuk menggunakan penasihat Virtual Deliverability Manager untuk menyelesaikan masalah deliverability dan reputasi

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Advisor di bawah Virtual Deliverability Manager.

> **Note**
>
> Advisor tidak akan terlihat jika Anda belum mengaktifkan Virtual Deliverability Manager untuk akun Anda. Untuk informasi selengkapnya, lihat the section called “Memulai”.

3. Tabel rekomendasi terbuka ditampilkan secara default. Rekomendasi dikategorikan berdasarkan Dampak (Tinggi/Rendah), Nama identitas (domain pengirim), Usia (peringatan), dan Rekomendasi/Deskripsi (masalah yang diidentifikasi). Di bilah pencarian, filter pada tingkat Dampak, Kategori masalah infrastruktur, atau nama Identitas domain pengirim.
4. Untuk memperbaiki masalah yang dijelaskan di kolom Rekomendasi/Deskripsi, pilih tautan di kolom Selesaikan masalah untuk baris tersebut, dan terapkan solusi yang disarankan.

> **Note**
>
> Setelah Anda menerapkan solusi, masalah yang diselesaikan dapat memakan waktu hingga enam jam untuk tercermin. Anda dapat melihat masalah yang diselesaikan di tab Rekomendasi Terselesaikan.

Mengakses rekomendasi Virtual Deliverability Manager Anda menggunakan AWS CLI

Contoh berikut menunjukkan kepada Anda cara mengakses rekomendasi Virtual Deliverability Manager Anda menggunakan. AWS CLI

Untuk mengakses rekomendasi Virtual Deliverability Manager Anda menggunakan AWS CLI

Anda dapat menggunakan `ListRecommendations`operasi di Amazon SES API v2 untuk mencantumkan rekomendasi pengiriman Anda. Anda dapat memanggil operasi ini dari AWS CLI, seperti yang ditunjukkan pada contoh berikut.

- Buat daftar rekomendasi untuk melihat masalah pengiriman:

```
aws --region us-east-1 sesv2 list-recommendations
```

- Terapkan filter untuk mengambil rekomendasi untuk domain tertentu yang Anda miliki:

```
aws --region us-east-1 sesv2 list-recommendations --cli-input-json file://list-
recommendations.json
```

- File input terlihat mirip dengan ini:

```
{
  "PageSize":100,
```

```
  "Filter":{
    "RESOURCE_ARN": "arn:aws:ses:us-east-1:123456789012:identity/example.com"
   }
}
```

Pengaturan Virtual Deliverability Manager

Anda dapat melihat atau mengubah pengaturan Virtual Deliverability Manager di akun Anda kapan saja. Anda dapat mengaktifkan atau menonaktifkan Virtual Deliverability Manager, dan dapat menentukan mode on atau off untuk pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan di tingkat akun Virtual Deliverability Manager melalui konsol Amazon SES atau AWS CLI

Opsi Virtual Deliverability Manager juga disediakan pada tingkat set konfigurasi sehingga Anda dapat menentukan pengaturan kustom untuk bagaimana set konfigurasi akan menggunakan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan dengan mengesampingkan bagaimana mereka telah didefinisikan di Virtual Deliverability Manager. Ini memberi Anda fleksibilitas untuk menyesuaikan pengiriman email Anda untuk kampanye email tertentu. Misalnya, Anda dapat mengaktifkan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan untuk email pemasaran Anda dan menonaktifkannya untuk email transaksional Anda.

Mengubah setelan akun Virtual Deliverability Manager menggunakan konsol Amazon SES

Prosedur berikut menunjukkan cara mengubah pengaturan akun Virtual Deliverability Manager menggunakan konsol Amazon SES.

Untuk mengubah setelan akun Virtual Deliverability Manager menggunakan konsol Amazon SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Pengaturan di bawah Virtual Deliverability Manager.

   Halaman pengaturan Virtual Deliverability Manager terbuka. Panel ikhtisar Langganan menunjukkan status Virtual Deliverability Manager dan panel Setelan tambahan menunjukkan status Pelacakan Keterlibatan dan pengiriman bersama yang dioptimalkan.
3. Untuk mengubah pelacakan Keterlibatan atau pengaturan pengiriman bersama yang dioptimalkan:

a. Di panel Pengaturan tambahan, pilih Edit.

b. Pilih tombol radio yang sesuai untuk mengaktifkan atau menonaktifkan fitur, lalu pilih Kirim pengaturan.

   Halaman pengaturan Virtual Deliverability Manager menampilkan ringkasan perubahan Anda di panel Pengaturan tambahan.

> **Note**
>
> Opsi pelacakan keterlibatan yang Anda tentukan di sini atau di pengaturan pengaturan Virtual Deliverability Manager, mengontrol apakah akan melaporkan pembukaan dan klik di dasbor Virtual Deliverability Manager atau tidak; opsi tersebut tidak memengaruhi konfigurasi tujuan acara yang mempublikasikan peristiwa terbuka dan klik. Misalnya, jika pelacakan keterlibatan Anda dinonaktifkan di sini, itu tidak akan menonaktifkan publikasi acara buka dan klik yang telah Anda siapkan di tujuan acara SES.

4. (Opsional) Untuk menentukan pengaturan kustom tentang cara set konfigurasi menggunakan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan dengan mengesampingkan bagaimana mereka didefinisikan dalam Virtual Deliverability Manager, referensi opsi Virtual Deliverability Manager saat membuat atau mengedit set konfigurasi.
5. Untuk menonaktifkan Virtual Deliverability Manager:

   a. Di panel Ikhtisar langganan, pilih Nonaktifkan Virtual Deliverability Manager.

   b. Di Disable Virtual Deliverability Manager? jendela pop-up, masukkan *`Disable`* di bidang konfirmasi, lalu pilih Nonaktifkan Virtual Deliverability Manager.

   c. Spanduk muncul, mengonfirmasi bahwa Anda telah menonaktifkan Virtual Deliverability Manager.

6. Untuk mengaktifkan kembali Virtual Deliverability Manager, lihat. the section called “Memulai”

Mengubah setelan akun Virtual Deliverability Manager menggunakan AWS CLI

Anda dapat mengubah pengaturan akun Virtual Deliverability Manager menggunakan file. AWS CLI

Untuk mengubah setelan akun Virtual Deliverability Manager menggunakan AWS CLI

Anda dapat menggunakan PutAccountVdmAttributesdan PutConfigurationSetVdmOptionsoperasi di Amazon SES API v2 untuk mengubah pengaturan Virtual Deliverability Manager. Anda dapat memanggil operasi ini dari AWS CLI, seperti yang ditunjukkan pada contoh berikut.

- Mengaktifkan atau menonaktifkan pelacakan keterlibatan, pengiriman bersama yang dioptimalkan, atau keduanya menggunakan file input:

```
aws --region us-east-1 sesv2 put-account-vdm-attributes --cli-input-json file://
attributes.json
```

Dalam contoh ini, di mana pelacakan keterlibatan ENABLED dan pengiriman bersama yang dioptimalkanDISABLED, file input terlihat mirip dengan ini:

```
{
    "VdmAttributes": {
        "VdmEnabled": "ENABLED",
        "DashboardAttributes": {
            "EngagementMetrics": "ENABLED"
        },
        "GuardianAttributes": {
            "OptimizedSharedDelivery": "DISABLED"
        }
    }
}
```

Anda dapat menemukan informasi selengkapnya tentang nilai parameter dan jenis data terkaitdengan menautkan dari tipe VdmAttributesdata di referensi Amazon SES API v2.

- Tentukan setelan kustom untuk bagaimana set konfigurasi akan menggunakan pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan dengan mengesampingkan bagaimana mereka telah didefinisikan di Virtual Deliverability Manager:

```
aws --region us-east-1 sesv2 put-configuration-set-vdm-options --cli-input-json
 file://config-set.json
```

Dalam contoh ini, di mana set konfigurasi bernama example memiliki pelacakan keterlibatan dan pengiriman bersama yang dioptimalkan diaktifkan, file input terlihat mirip dengan ini:

```
{
```

```
    "ConfigurationSetName": "example",
    "VdmOptions": {
        "DashboardOptions": {
            "EngagementMetrics": "ENABLED"
        },
        "GuardianOptions": {
            "OptimizedSharedDelivery": "ENABLED"
        }
    }
}
```

Untuk informasi selengkapnya tentang nilai parameter dan tipe data terkait, lihat tipe `VdmOptions`data dalam referensi Amazon SES API v2.

- Untuk memverifikasi hasilnya:

```
aws –region us-east-1 sesv2 get-configuration-set –configuration-set-name example
```

- Tidak menentukan `DashboardOptions`atau `GuardianOptions`opsi pada tingkat set konfigurasi menghasilkan pengaturan tingkat akun Virtual Deliverability Manager Anda yang berlaku untuk lalu lintas yang dikirim melalui set konfigurasi tersebut.

Manajer Email untuk Amazon SES

Mail Manager adalah serangkaian fitur gateway email Amazon SES yang dirancang untuk membantu Anda memperkuat infrastruktur email organisasi Anda, menyederhanakan manajemen alur kerja email, dan merampingkan kontrol kepatuhan email. Ini terintegrasi dengan infrastruktur Anda yang ada, dapat menghubungkan berbagai aplikasi bisnis, dan mengotomatiskan pemrosesan email masuk. Mail Manager juga bertindak sebagai garis pertahanan pertama dalam menjaga sistem email yang sehat dengan mengelola lalu lintas email Anda secara efisien dan meningkatkan kepatuhan dengan kemampuan arsip emailnya.

Seiring dengan kemampuan Amazon SES saat ini, Mail Manager terdiri dari fitur-fitur berikut yang mendukung lalu lintas masuk:

- Endpoint Ingress — Komponen infrastruktur utama yang menggunakan kebijakan dan aturan penyaringan yang dapat Anda konfigurasikan untuk menentukan email mana yang harus diizinkan masuk ke organisasi Anda dan mana yang harus ditolak.
- Kebijakan lalu lintas dan kumpulan aturan - Aktifkan administrator email untuk menentukan dan menegakkan aturan untuk mengelola lalu lintas email masuk dengan kebijakan dan aturan yang sangat dapat disesuaikan yang dapat mengurutkan, mengkategorikan, memprioritaskan, dan melakukan tindakan pada email berdasarkan serangkaian kondisi dan pengecualian yang Anda tentukan. Pemfilteran cerdas ini dikombinasikan dengan alur kerja otomatis membantu merampingkan manajemen email, meningkatkan efisiensi, dan memastikan kepatuhan terhadap kebijakan email organisasi Anda.
- Relai SMTP - Mengalihkan lalu lintas email ke server SMTP lain berdasarkan kriteria yang Anda tentukan dalam aturan dengan menghubungkan sistem email internal, dan merampingkan manajemen email dengan penerusan otomatis. Mampu mendistribusikan lalu lintas di beberapa server dan gateway memungkinkan organisasi Anda mengelola lalu lintas email volume tinggi secara efektif, bahkan di lingkungan hybrid.
- Pengarsipan email - Menyimpan dan melindungi email Anda dengan menyimpan data dalam penyimpanan jangka panjang yang persisten dan aman, dan memberi Anda cara untuk mencari dan mengarsipkan email dengan cepat. Ini menyediakan pengarsipan tingkat perusahaan penuh waktu tanpa meningkatkan persyaratan penyimpanan server kotak surat Anda.
- Email Add Ons - Kumpulan alat keamanan khusus dari penyedia yang disetujui SES yang dapat digunakan untuk mengelola email yang masuk ke titik akhir ingress Anda serta menyediakan opsi perutean berdasarkan hasil keamanan. Alat-alat ini adalah intelijen keamanan bersertifikat

dan solusi penegakan hukum yang siap diintegrasikan ke dalam alur kerja email Anda dan dapat diaktifkan langsung dari konsol Mail Manager.

Memulai dengan Mail Manager

Untuk mulai menggunakan Mail Manager, panduan orientasi di konsol Amazon SES akan memandu Anda melalui langkah-langkah mengaktifkan Mail Manager untuk akun Anda. Lihat the section called “Memulai”.

Topik

- Memulai dengan Mail Manager
- Titik akhir masuknya
- Kebijakan lalu lintas dan pernyataan kebijakan
- Set aturan dan aturan
- SMTPestafet
- Pengarsipan email
- Email Tambah Ons
- Kebijakan izin untuk Mail Manager

Memulai dengan Mail Manager

Untuk mulai menggunakan Amazon SES Mail Manager, Anda dapat menggunakan panduan Memulai dengan Mail Manager di konsol Amazon SES, tempat Anda akan membuat titik akhir ingress dan mengonfigurasinya dengan kebijakan lalu lintas dan aturan yang ditetapkan.

Endpoint ingress adalah blok bangunan pertama Anda dalam menyiapkan Mail Manager—ini adalah komponen infrastruktur utama yang menggunakan:

- Kebijakan lalu lintas — Kebijakan lalu lintas berisi pernyataan kebijakan yang Anda tentukan untuk mengurutkan email masuk dengan mengizinkan atau memblokir jenis email tertentu saat kondisi pernyataan kebijakan terpenuhi.
- Kumpulan aturan — Kumpulan aturan berisi aturan yang Anda tetapkan untuk melakukan tindakan pada email yang Anda izinkan saat kondisi aturan terpenuhi.

Namun, bagian dari pembuatan titik akhir ingress adalah memilih kebijakan lalu lintas dan kumpulan aturan yang telah dibuat dan kemudian menetapkannya ke titik akhir ingress. Langkah-langkah dalam prosedur berikut akan memandu Anda melalui urutan yang benar untuk mengonfigurasi titik akhir masuk pertama Anda.

Memulai Mail Manager menggunakan konsol SES

Prosedur berikut menunjukkan kepada Anda cara memulai dengan Mail Manager menggunakan konsol SES.

Untuk memulai dengan Mail Manager menggunakan konsol Amazon SES

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Mail Manager dan pilih salah satu tombol Memulai dengan Mail Manager pada halaman ikhtisar Mail Manager.
3. Pada halaman Menyiapkan, pilih Buat kebijakan lalu lintas di Buat kartu kebijakan lalu lintas.

    a. Lengkapi alur kerja di halaman Buat kebijakan lalu lintas. Jika Anda membutuhkan informasi tambahan, lihatthe section called “Membuat kebijakan lalu lintas & pernyataan kebijakan (konsol)”.

    b. Setelah membuat kebijakan lalu lintas dan pernyataan kebijakan pertama Anda, gunakan tombol kembali browser Anda untuk kembali ke halaman Dapatkan penyiapan atau pilih Siapkan di bawah Manajer Email di panel navigasi kiri.
4. Pada halaman Dapatkan set up, pilih Buat aturan set pada Create a rule set card.

    a. Selesaikan alur kerja pada halaman Buat set aturan. Jika Anda membutuhkan informasi tambahan, lihatthe section called “Membuat set aturan & aturan (konsol)”.

    b. Setelah membuat aturan dan aturan pertama Anda, gunakan tombol kembali browser Anda untuk kembali ke halaman Get set up atau pilih Get set up di bawah Mail Manager di panel navigasi kiri.
5. Sekarang setelah Anda membuat kebijakan lalu lintas dan aturan set pertama Anda, Anda akan dapat membuat titik akhir ingress pertama Anda. Pada halaman Get set up, pilih Create ingress endpoint pada Create an ingress endpoint card.

    - Bagian dari alur kerja pada halaman titik akhir masuknya Email adalah menetapkan kebijakan lalu lintas dan aturan yang baru saja Anda buat ke titik akhir ingress. Jika Anda

membutuhkan informasi tambahan, lihatthe section called “Membuat titik akhir ingress (konsol)”.

Dengan titik akhir ingress pertama Anda dibuat, Anda dapat mulai menggunakan Mail Manager dan memanfaatkan fitur lainnya seperti relay SMTP dan pengarsipan email. Anda juga dapat membuat titik akhir ingress tambahan dengan kebijakan lalu lintas unik dan set aturan untuk menyesuaikan lebih lanjut cara Anda mengelola semua email masuk Anda.

Titik akhir masuknya

Endpoint ingress adalah komponen infrastruktur utama di Mail Manager yang menerima, merutekan, dan mengelola email Anda dengan memanfaatkan kebijakan dan aturan yang Anda konfigurasikan untuk menentukan email mana yang harus ditolak, mana yang harus diizinkan, dan mana yang harus ditindaklanjuti.

Setiap titik akhir ingress memiliki kebijakan lalu lintas sendiri untuk menentukan email mana yang akan diblokir atau diizinkan, dan aturannya sendiri ditetapkan untuk melakukan tindakan pada email yang Anda izinkan; oleh karena itu, dengan membuat beberapa titik akhir ingress, Anda dapat mendelegasikan masing-masing untuk mengelola dan merutekan jenis email tertentu. Tingkat granularitas ini akan membantu Anda membangun sistem manajemen email yang disesuaikan dengan kebutuhan bisnis Anda.

Alur kerja prasyarat untuk membuat titik akhir ingress

Pada saat membuat titik akhir ingress Anda, Anda harus menetapkan kebijakan lalu lintas dan aturan yang telah dibuat. Oleh karena itu, alur kerja untuk membuat titik akhir ingress harus dalam urutan sebagai berikut:

1. Mulailah dengan membuat kebijakan lalu lintas untuk menentukan email yang ingin Anda blokir atau izinkan. Untuk detailnya, lihat the section called “Membuat kebijakan lalu lintas & pernyataan kebijakan (konsol)”.
2. Selanjutnya, buat aturan yang ditetapkan untuk melakukan tindakan pada email yang Anda izinkan. Untuk detailnya, lihat the section called “Membuat set aturan & aturan (konsol)”.
3. Terakhir, buat titik akhir ingress Anda dan tetapkan kebijakan lalu lintas dan aturan yang baru saja Anda buat atau yang lain yang sebelumnya Anda buat.

Setelah Anda membuat titik akhir ingress, Anda harus mengonfigurasinya dengan lingkungan yang Anda gunakan untuk menerima email, apakah itu konfigurasi SMTP klien on-premise atau host domain berbasis web. DNS Ini dibahas di bawah ini dithe section called “Mengkonfigurasi lingkungan Anda”.

Mengonfigurasi lingkungan Anda untuk menggunakan titik akhir ingress

Menggunakan catatan “A”

Pada saat Anda membuat titik akhir ingress, catatan “A” untuk titik akhir akan dihasilkan dan nilainya ditampilkan di layar ringkasan titik akhir ingress di konsol. SES Cara Anda menggunakan nilai catatan ini bergantung pada jenis titik akhir yang Anda buat dan kasus penggunaan Anda:

- Open endpoint — Mail yang dikirim ke domain Anda akan diselesaikan langsung ke titik akhir ingress Anda—tidak diperlukan autentikasi.
  - Salin dan tempel nilai catatan “A” baik langsung ke SMTP konfigurasi SMTP klien di lokasi atau ke catatan MX untuk domain Anda dalam konfigurasi Anda. DNS
- Titik akhir yang diautentikasi — Mail yang dikirim ke domain Anda harus berasal dari pengirim resmi yang telah Anda bagikan SMTP kredensialnya, seperti server email lokal Anda.
  - Salin dan tempel nilai catatan “A” langsung ke SMTP konfigurasi SMTP klien di lokasi serta nama pengguna dan kata sandi Anda.

Jika Anda menggunakan data MX dalam konfigurasi Anda, ingatlah bahwa meskipun setiap DNS penyedia memiliki prosedur dan antarmuka yang berbeda untuk mengonfigurasi catatan, bagian penting dari informasi yang perlu Anda masukkan ke dalam DNS pengaturan Anda tercantum dalam contoh berikut:

Semua email yang dikirim ke recipient@marketing.example.com akan masuk ke titik akhir ingress Anda karena Anda memasukkan catatan “A” titik akhir ingress sebagai nilai untuk catatan MX di pengaturan domain Anda: DNS

- Domain — `marketing.example.com`
- Nilai catatan MX — `890123abcdef.ghijk.mail-manager-smtp.amazonaws.com` (Ini adalah nilai catatan “A” yang disalin dari titik akhir ingress Anda.)
- Prioritas – `10`

Menghubungkan ke titik akhir yang diautentikasi

Untuk pengirim resmi yang telah Anda bagikan SMTP kredensialnya agar dapat terhubung ke titik akhir yang diautentikasi, protokol berikut harus diikuti untuk nama pengguna dan kata sandi untuk membuat koneksi yang berhasil ke server:

- Nama pengguna - Ini adalah ID titik akhir ingress dan harus dikodekan di Base64. (Lihat Langkah 10. dalam prosedur konsol untuk mempelajari cara menemukan ID titik akhir ingress.)
- Kata sandi — Ini adalah yang digunakan selama pembuatan titik akhir ingress dan harus dikodekan di Base64.

Contoh berikut menunjukkan SMTP AUTH server tipikal dan pertukaran klien membangun koneksi:

```
S: 250 AUTH LOGIN PLAIN
C: AUTH LOGIN
S: 334 VXNlcm5hbWU6
C: SW5ncmVzc1BvaW50
S: 334 UGFzc3dvcmQ6
C: SW5ncmVzc1Bhc3N3b3Jk
S: 235 Authentication successful
```

Contoh ini berisi properti berikut:

- Sberarti “server"—server yang menerima pesan. SMTP
- Cberarti “Klien” — SMTP klien membangun koneksi dengan server dan mengirim pesan ke server.
- `250 AUTH LOGIN PLAIN`adalah respons dari server dengan AUTH metode yang didukung, `AUTH LOGIN` atau`AUTH PLAIN`, pengirim dapat memilih salah satunya, dan mengirim SMTP perintah yang sesuai dengan spesifikasi Ekstensi SMTP Layanan untuk Otentikasi 2554. RFC `AUTH LOGIN`digunakan di sini.
- `334 VXNlcm5hbWU6`— Server meminta nama pengguna di Base64.
- `SW5ncmVzc1BvaW50`- Klien merespons dengan ID titik akhir ingress di Base64.
- `334 UGFzc3dvcmQ6`— Server meminta kata sandi di Base64.
- `SW5ncmVzc1Bhc3N3b3Jk`- Klien merespons dengan kata sandi titik akhir ingress di Base64.

Prosedur di bagian selanjutnya akan memandu Anda membuat titik akhir ingress di konsol. SES

Membuat titik akhir ingress di konsol SES

Prosedur berikut menunjukkan cara menggunakan halaman endpoint Ingress di SES konsol untuk membuat titik akhir ingress dan mengelola yang sudah Anda buat.

Untuk membuat endpoint kelola ingress menggunakan konsol

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Endpoint Ingress di bawah Mail Manager.
3. Pada halaman Endpoint Ingress, pilih Create ingress endpoint.
4. Pada halaman Create new ingress endpoint, masukkan nama unik untuk titik akhir ingress Anda.
5. Pilih apakah itu akan menjadi titik akhir Terbuka atau Terotentikasi.

   - Jika Anda memilih Autentikasi, pilih salah satu SMTPkata sandi dan masukkan kata sandi (untuk dibagikan dengan pengirim resmi), atau Rahasia dan pilih salah satu rahasia Anda dari Rahasia. ARN Jika Anda memilih rahasia yang dibuat sebelumnya, itu harus berisi kebijakan yang ditunjukkan dalam langkah-langkah berikut untuk membuat rahasia baru.
   - Anda memiliki opsi untuk membuat rahasia baru dengan memilih Create new —the AWS Secrets Manager konsol akan terbuka di mana Anda dapat terus membuat kunci baru:

   a. Pilih Jenis rahasia lainnya dalam tipe Rahasia.

   b. Dalam pasangan kunci/nilai, masukkan `password` kunci, dan kata sandi Anda yang sebenarnya untuk nilainya.

      > **Note**
      >
      > Untuk Key, Anda hanya harus memasukkan `password` (hal lain akan menyebabkan otentikasi gagal).

   c. Pilih Tambahkan kunci baru untuk membuat kunci terkelola KMS pelanggan (CMK) di kunci Enkripsi —the AWS KMS konsol akan terbuka.

   d. Pilih Create key pada halaman Customer manged keys.

   e. Simpan nilai default pada halaman tombol Configure dan pilih Next.

   f. Masukkan nama untuk kunci Anda di Alias (opsional, Anda dapat menambahkan deskripsi dan tag), diikuti oleh Berikutnya.

g. Pilih pengguna (selain Anda sendiri) atau peran yang ingin Anda izinkan untuk mengelola kunci di Administrator kunci diikuti oleh Berikutnya.

h. Pilih pengguna (selain Anda sendiri) atau peran yang ingin Anda izinkan untuk menggunakan kunci di Pengguna kunci diikuti oleh Berikutnya.

i. Salin dan tempel KMSCMKkebijakan ke editor JSON teks kebijakan Kunci di `"statement"` tingkat dengan menambahkannya sebagai pernyataan tambahan yang dipisahkan oleh koma. Ganti wilayah dan nomor akun dengan milik Anda.

j. Pilih Selesai.

k. Pilih tab browser Anda di mana Anda memiliki AWS Secrets Manager Simpan halaman rahasia baru terbuka dan pilih ikon penyegaran (panah melingkar) di sebelah bidang kunci Enkripsi, lalu klik di dalam bidang dan pilih kunci yang baru Anda buat.

l. Masukkan nama di bidang Nama rahasia di halaman Konfigurasi rahasia.

m. Pilih Edit izin di Izin sumber daya.

n. Salin dan tempel Rahasia kebijakan sumber daya ke editor JSON teks izin Sumber daya dan ganti wilayah dan nomor akun dengan milik Anda. (Pastikan untuk menghapus kode contoh apa pun di editor.)

o. Pilih Simpan diikuti oleh Berikutnya.

p. Opsional mengkonfigurasi rotasi diikuti oleh Berikutnya.

q. Tinjau dan simpan rahasia baru Anda dengan memilih Store.

r. Pilih tab browser Anda di mana Anda memiliki halaman SES Create new ingress endpoint terbuka dan pilih Refresh list, lalu pilih rahasia yang baru Anda buat di Secret. ARN

6. Pilih kebijakan lalu lintas untuk menentukan email yang ingin Anda blokir atau izinkan.
7. Pilih kumpulan aturan yang berisi tindakan aturan yang ingin Anda lakukan pada email yang Anda izinkan.
8. Pilih Buat titik akhir ingress.
9. Secara umum, “Penyediaan” akan ditampilkan saat titik akhir masuk Anda sedang dibuat—segarkan halaman hingga “Aktif” ditampilkan dan bidang berisi nilai. ARecord Salin nilai catatan “A” dan tempelkan ke DNS konfigurasi Anda atau SMTP klien Anda seperti yang dibahas diMengkonfigurasi lingkungan Anda.
10. Tepat di atas wadah Detail umum di konsol, ada nomor besar yang tidak berlabel yang diawali dengan “inp” (juga direplikasi di jejak remah roti di bagian atas halaman), misalnya, inp-1abc2de3fghi4jkl5mnop6qr. Ini disebut sebagai ID titik akhir ingress, nilainya digunakan

sebagai nama pengguna untuk masuk ke server ingress Anda. (Anda harus berbagi ini dengan pengirim resmi Anda untuk terhubung ke titik akhir Anda.)

11. Anda dapat melihat dan mengelola titik akhir ingress yang telah Anda buat dari halaman titik akhir Ingress. Jika ada titik akhir ingress yang ingin Anda hapus, pilih tombol radionya diikuti oleh Delete.
12. Untuk mengedit titik akhir ingress, pilih namanya untuk membuka halaman ringkasannya:

    - Anda dapat mengubah status aktif titik akhir dengan memilih Edit di Detail umum diikuti dengan Simpan perubahan.
    - Anda dapat memilih kumpulan aturan atau kebijakan lalu lintas yang berbeda dengan memilih Edit dalam kumpulan Aturan atau kebijakan Lalu lintas diikuti dengan Simpan perubahan.

Kebijakan lalu lintas dan pernyataan kebijakan

Kebijakan lalu lintas adalah wadah untuk pernyataan kebijakan yang Anda tetapkan ke titik akhir ingress sehingga dapat mengurutkan email masuk dengan mengizinkan atau memblokir jenis email tertentu ketika kondisi pernyataan kebijakan terpenuhi. Kebijakan lalu lintas dapat digunakan oleh beberapa titik akhir ingress.

> **Tip**
>
> Anda dapat menganggap kebijakan lalu lintas sebagai “set filter”, dan pernyataan kebijakan sebagai “filter”. Kebijakan lalu lintas (set filter) berisi kebijakan (filter) yang Anda gunakan untuk memfilter email masuk Anda.

Saat membuat kebijakan lalu lintas, Anda memiliki opsi untuk menetapkan ukuran pesan maksimum (dalam byte). Ketika sebuah pesan melebihi ukuran itu, pesan itu segera dibuang. Ini bertindak sebagai filter “pass pertama” saat disetel. Selanjutnya, Anda menetapkan tindakan default untuk mengizinkan atau memblokir email yang berada di luar kondisi pernyataan kebijakan Anda—anggap ini sebagai tindakan “catch all” untuk kebijakan lalu lintas.

Pernyataan kebijakan juga dibuat dengan tindakan izinkan atau blok yang diambil ketika kondisi pernyataan terpenuhi. Anda membuat kondisi dengan memilih protokol email dan operator bersyarat untuk nilai yang Anda masukkan yang harus dicocokkan dengan pesan masuk sebelum pernyataan kebijakan mengizinkan atau memblokirnya. Setiap pernyataan kebijakan dapat memiliki beberapa kondisi.

Kebijakan lalu lintas dapat berisi beberapa pernyataan kebijakan dan mengeksekusinya dalam urutan yang didasarkan pada hierarki implisit tentang cara mengevaluasi email:

- Ukuran pesan maksimum — Jika parameter opsional ini disetel, pesan apa pun yang lebih besar dari ukuran ini akan segera dibuang, melewati pernyataan kebijakan.
- Pernyataan kebijakan yang memblokir — Pernyataan ini dievaluasi terlebih dahulu dan memblokir pesan apa pun yang memenuhi persyaratan pernyataan.
- Pernyataan kebijakan yang memungkinkan — Pernyataan ini dievaluasi selanjutnya dan memungkinkan pesan apa pun yang memenuhi persyaratan pernyataan tersebut.
- Tindakan default kebijakan lalu lintas — Sisa pesan yang berada di luar pernyataan kebijakan diizinkan atau diblokir berdasarkan cara Anda mendefinisikan parameter ini.

Kebijakan lalu lintas adalah sumber daya independen yang dapat digunakan oleh lebih dari satu titik akhir ingress, tetapi pernyataan kebijakan hanya dimiliki oleh kebijakan lalu lintas di mana mereka dibuat. Dengan demikian, Anda harus terlebih dahulu membuat kebijakan lalu lintas, atau mengedit yang sudah ada, sebelum Anda dapat membuat pernyataan kebijakan untuk mengevaluasi email yang masuk ke titik akhir ingress Anda.

Prosedur di bagian selanjutnya menjelaskan cara membuat kebijakan lalu lintas dan pernyataan kebijakan mereka di konsol SES.

Membuat kebijakan lalu lintas dan pernyataan kebijakan di konsol SES

Prosedur berikut menunjukkan cara menggunakan halaman Kebijakan lalu lintas di konsol SES untuk membuat kebijakan lalu lintas dan pernyataan kebijakan mereka, dan mengelola yang telah Anda buat.

Untuk membuat dan mengelola kebijakan lalu lintas dan pernyataan kebijakan menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Kebijakan lalu lintas di bawah Manajer Email.
3. Pada halaman Kebijakan lalu lintas, pilih Buat kebijakan lalu lintas.
4. Pada halaman Buat kebijakan lalu lintas, masukkan nama unik untuk kebijakan lalu lintas Anda.
5. (Opsional) Jika Anda ingin membuang pesan apa pun di atas ukuran tertentu, masukkan nilai dalam byte di bidang Ukuran pesan maksimum.

6. Dalam tindakan Default, pilih apakah kebijakan lalu lintas adalah Izinkan atau Tolak (blokir) pesan yang berada di luar (tidak ditangani oleh) kondisi pernyataan kebijakan Anda.
7. Pilih Tambahkan pernyataan kebijakan baru untuk membuat pernyataan untuk kebijakan lalu lintas Anda.
8. Pilih Izinkan atau Tolak (blokir) untuk tindakan yang akan diambil ketika kondisi pernyataan terpenuhi.
9. Buat kondisi dengan memilih protokol email dan operator bersyarat untuk nilai yang Anda masukkan. Pilih Tambahkan kondisi baru jika Anda ingin menambahkan lebih banyak kondisi ke pernyataan kebijakan ini. Untuk mempelajari lebih lanjut tentang properti kondisi dan operatornya serta nilai yang valid, lihat referensi Kondisi pernyataan kebijakan.

    - Jika Anda berlangganan Email Add On, Anda akan dapat memilihnya di sini sebagai protokol email.
10. Jika Anda ingin menambahkan lebih banyak pernyataan dan ketentuan kebijakan, ulangi langkah 7 - 9 di atas.
11. Setelah selesai membuat pernyataan kebijakan dan kondisinya, pilih Buat kebijakan lalu lintas.
12. Anda dapat melihat dan mengelola kebijakan lalu lintas yang telah Anda buat dari halaman Kebijakan lalu lintas. Jika ada kebijakan lalu lintas yang ingin Anda hapus, pilih tombol radionya diikuti oleh Hapus.
13. Untuk mengedit properti kebijakan lalu lintas atau pernyataan kebijakannya, pilih namanya untuk membuka halaman ikhtisar, dari sini, pilih Edit.
14. Dalam detail kebijakan lalu lintas, Anda dapat mengubah ukuran pesan maksimum dan tindakan default.
15. Di salah satu wadah pernyataan Policy, Anda dapat mengubah properti allow/deny dan mengedit salah satu kondisi. Anda juga dapat menghapus pernyataan dan ketentuan kebijakan, serta menambahkan yang baru.
16. Setelah selesai dengan semua pengeditan, simpan perubahan Anda dengan memilih Simpan perubahan.

Referensi untuk kondisi pernyataan kebijakan

Kondisi pernyataan kebijakan

Tabel referensi berikut mencantumkan semua protokol pernyataan kebijakan yang tersedia untuk membangun kondisi pernyataan kebijakan. Memilih jenis ekspresi protokol akan membawa Anda ke

halaman referensi di SES Mail Manager API Referensi yang mencantumkan semua operator yang tersedia dan nilai yang valid untuk protokol tersebut.

Kondisi pernyataan kebijakan: Protokol, operator, dan nilai

Protokol	Jenis ekspresi
Alamat penerima	Operator dan nilai yang valid untuk ekspresi string
Rentang IP pengirim	Operator dan nilai yang valid untuk ekspresi IP
Versi protokol TLS	Operator dan nilai yang valid untuk ekspresi protokol TLS
Abusix Mail Intelligence (jika berlangganan) Daftar Blok Domain Spamhaus (jika berlangganan)	Operator dan nilai yang valid untuk ekspresi boolean

Set aturan dan aturan

Set aturan adalah wadah untuk aturan yang Anda tetapkan ke titik akhir ingress sehingga dapat melakukan tindakan pada email yang diizinkan masuk dari kebijakan lalu lintas titik akhir ingress. Sebuah set aturan dapat digunakan oleh beberapa titik akhir ingress.

Aturan memberi tahu titik akhir ingress Anda cara menangani email masuk dengan menjalankan tindakan yang ditentukan dalam aturan saat pesan memenuhi ketentuan aturan. Setiap aturan dapat memiliki beberapa kondisi dan tindakan. Aturan yang Anda buat dalam kumpulan aturan dieksekusi dalam urutan yang Anda tentukan dalam kumpulan aturan.

Anda membangun kondisi aturan dengan memilih properti email dan operator bersyarat untuk nilai yang Anda masukkan yang harus dicocokkan dengan pesan sebelum aturan akan menjalankan tindakannya—Anda menentukan tindakan yang akan diambil serta urutan pelaksanaannya.

Untuk perincian yang lebih besar, aturan Anda juga dapat berisi pengecualian yang didefinisikan mirip dengan kondisi, tetapi di sini, Anda menentukan kondisi bahwa pesan tidak boleh cocok. Kondisi dan pengecualian beroperasi secara independen—Anda dapat membuat aturan hanya dengan pengecualian jika Anda mau, serta kondisi dan pengecualian intermix.

Karena perincian halus tentang bagaimana aturan dapat didefinisikan dalam kumpulan aturan, daftar berikut disediakan untuk membantu menggambarkan hubungan komponen kumpulan aturan:

- Set aturan berisi:
  - Aturan — Anda dapat menentukan urutan di mana aturan dieksekusi dalam aturan yang ditetapkan.

    Aturan mengandung:

    - Ketentuan - Aturan berlaku jika pesan cocok dengan evaluasi kondisi; dan jika aturan memiliki pengecualian, lihat di bawah.
    - Pengecualian — Aturan berlaku jika pesan tidak cocok dengan evaluasi pengecualian; dan jika aturan memiliki kondisi, lihat di atas.
    - Tindakan — Tindakan dipicu saat aturan berlaku—semua kondisi cocok dan tidak ada pengecualian.

      Anda dapat menentukan urutan di mana tindakan dijalankan dalam aturan.

Karena setiap aturan dapat memiliki beberapa kondisi, pengecualian, dan tindakan, dan fakta bahwa Anda dapat menentukan urutan bagaimana aturan dan tindakan dijalankan, ini memungkinkan Anda untuk membangun solusi penanganan email yang sangat disesuaikan dan otomatis yang disesuaikan dengan kebutuhan bisnis spesifik Anda.

Kumpulan aturan adalah sumber daya independen yang dapat digunakan oleh lebih dari satu titik akhir ingress, tetapi aturan hanya dimiliki oleh kumpulan aturan di mana mereka dibuat. Dengan demikian, Anda harus terlebih dahulu membuat kumpulan aturan, atau mengedit yang sudah ada, sebelum Anda dapat membuat aturan untuk bertindak atas email yang masuk ke titik akhir ingress Anda.

Prosedur di bagian selanjutnya akan memandu Anda membuat set aturan dan aturannya di SES konsol.

Membuat set aturan dan aturan di SES konsol

Prosedur berikut menunjukkan cara menggunakan halaman Rule sets di SES konsol untuk membuat kumpulan aturan dan aturannya, dan mengelola yang sudah Anda buat.

Untuk membuat kumpulan aturan kelola dan aturan menggunakan konsol

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Set aturan di bawah Manajer Mail.
3. Pada halaman Rule sets, pilih Create rule set dan masukkan nama unik untuk set aturan Anda.
4. Pada halaman ikhtisar set aturan, pilih Edit, lalu pilih Buat aturan baru di halaman edit.
5. Di bilah sisi Rule details, masukkan nama unik untuk aturan Anda.
6. Pilih Tambahkan kondisi baru untuk membuat kondisi bahwa pesan harus cocok; atau centang kotak EXCEPTdalam kasus: diikuti oleh Tambahkan pengecualian baru untuk membuat kondisi bahwa pesan tidak boleh cocok.
7. Buat kondisi atau pengecualian dengan memilih properti email dan operator bersyarat untuk nilai yang Anda masukkan. Pilih Tambahkan kondisi baru atau Tambahkan pengecualian baru jika Anda ingin menambahkan lebih banyak kondisi atau pengecualian ke aturan ini. Untuk mempelajari lebih lanjut tentang properti kondisi dan operatornya serta nilai yang valid, lihat referensi Kondisi aturan.

   - Jika Anda berlangganan Email Add On, Anda akan dapat memilihnya di sini sebagai properti email.

8. Pilih Tambahkan tindakan baru untuk menentukan tindakan yang akan diambil ketika kondisi aturan cocok dan/atau pengecualian tidak cocok. Untuk menambahkan lebih banyak tindakan yang akan diambil, pilih Tambahkan tindakan baru. Untuk mempelajari lebih lanjut tentang tindakan dan parameternya, lihat referensi Tindakan aturan.

   - Saat Anda membuat dua tindakan atau lebih, panah atas/bawah ditampilkan sehingga Anda dapat mengatur urutan eksekusi.
   - Untuk menjalankan tindakan aturan Tulis ke S3, Kirim ke kotak pesan, atau Kirim ke internet, Anda harus mengaktifkan kebijakan izin masing-masing untuk akun Anda; jika tidak, tindakan aturan akan gagal.

     Anda dapat menerapkan kebijakan izin untuk salah satu tindakan ini langsung dari panel Rincian aturan setelah memilih tindakan:

     a. Pilih Buat peran baru di bidang IAMperan dan masukkan nama diikuti oleh Buat peran. (Kebijakan IAM kepercayaan untuk peran ini akan dibuat secara otomatis di latar belakang.)

b. Karena kebijakan IAM kepercayaan dibuat secara otomatis, Anda hanya perlu menambahkan kebijakan izin tindakan ke peran—pilih Lihat peran di bawah bidang IAMperan untuk membuka konsol. IAM

c. Di bawah tab Izin, pilih Tambahkan izin dan pilih Buat kebijakan sebaris.

d. Pada halaman Tentukan izin, pilih JSONdi Editor kebijakan.

e. Salin dan tempel kebijakan izin masing-masing dari Kebijakan tindakan aturan ke editor Kebijakan dan ganti data dalam teks merah dengan milik Anda. (Pastikan untuk menghapus kode contoh apa pun di editor.)

f. Pilih Berikutnya.

g. Tinjau dan buat kebijakan izin Anda untuk IAM peran tersebut dengan memilih Buat kebijakan.

h. Pilih tab browser Anda di mana Anda memiliki halaman pengaturan aturan Edit Manajer SES Email terbuka dan lanjutkan dengan langkah-langkah yang tersisa untuk membuat aturan.

9. Setelah selesai membuat kondisi, pengecualian, dan tindakan untuk aturan, Anda menyimpannya ke set aturannya dengan memilih Simpan set aturan yang terletak di panel Edit aturan set di sebelah kiri.

10. Jika Anda ingin menambahkan lebih banyak aturan ke set aturan, ulangi langkah 4 - 9 di atas.

    - Saat Anda membuat dua aturan atau lebih, panah atas/bawah ditampilkan di kolom Reorder set aturan sehingga Anda dapat mengatur urutan eksekusi.

11. Anda dapat melihat dan mengelola set aturan yang telah Anda buat dari halaman Rule sets. Jika ada aturan yang ingin Anda hapus, pilih tombol radionya diikuti oleh Hapus.

12. Untuk mengedit kumpulan aturan, pilih namanya untuk membuka halaman ikhtisar, dari sini, pilih Edit di mana Anda dapat menyusun ulang eksekusi aturannya, menambahkan lebih banyak aturan dengan memilih Buat aturan baru, atau hapus aturan dengan memilih tombol radio diikuti oleh Hapus.

13. Untuk mengedit aturan, pilih tombol radionya. Di salah satu wadah di bilah sisi detail Aturan, Anda dapat mengedit kondisi atau pengecualian apa pun dan mengubah atau menyusun ulang tindakan apa pun. Anda juga dapat menghapus kondisi, pengecualian, dan tindakan, serta menambahkan yang baru.

14. Setelah selesai dengan semua pengeditan, simpan perubahan Anda dengan memilih Simpan aturan set yang terletak di panel Edit aturan set di sebelah kiri.

Referensi untuk kondisi dan tindakan aturan

Kondisi aturan

Tabel referensi berikut mencantumkan semua properti aturan yang tersedia untuk membangun kondisi aturan (atau pengecualian) dan dikategorikan berdasarkan jenis ekspresinya. Properti aturan yang berbagi jenis ekspresi yang sama juga berbagi operator dan nilai yang sama. Memilih jenis ekspresi properti akan membawa Anda ke halaman referensi di Referensi Manajer SES API Email yang mencantumkan semua operator yang tersedia dan nilai yang valid untuk properti tersebut.

Kondisi aturan: Properti, operator, dan nilai

Properti	Jenis ekspresi
Dari alamat Untuk mengatasi Alamat CC Mail dari Alamat penerima Subjek Helo MIMEsundulan	Operator dan nilai yang valid untuk ekspresi string
Rentang IP	Operator dan nilai yang valid untuk ekspresi IP
Ukuran maks pesan	Operator dan nilai yang valid untuk ekspresi angka
DKIM SPF Pemindaian Virus Trend Micro (jika berlangganan)	Operator dan nilai yang valid untuk ekspresi putusan

Properti	Jenis ekspresi
TLS TLSdibungkus Baca tanda terima	Operator dan nilai yang valid untuk ekspresi boolean
DMARCkebijakan	Operator dan nilai yang valid untuk DMARC ekspresi

Tindakan aturan

Tabel referensi berikut mencantumkan semua tindakan aturan yang dapat diambil ketika kondisi aturan terpenuhi atau pengecualiannya tidak terpenuhi. Dengan memilih tindakan, Anda akan dibawa ke halaman referensi tindakan di Referensi Manajer SES API Email yang mencantumkan parameter dan formatnya untuk tindakan tersebut. Tabel menggunakan nama tindakan yang diadopsi di konsol Manajer API Mail—nama mungkin sedikit berbeda.

> **Note**
>
> Di beberapa API referensi, akan ada *`ActionFailurePolicy`* parameter yang dapat diatur ke Lanjutkan atau Jatuhkan jika tindakan gagal—ini hanya berlaku saat menggunakanAPI; saat menggunakan konsol, *`ActionFailurePolicy`* telah disetel ke nilai default Lanjutkan.

Tindakan aturan: Tindakan dan parameter

Tindakan & parameternya	Deskripsi
Menulis ke S3	Menulis MIME konten email ke ember S3.
SMTPaksi relai	Relay email melalui SMTP ke SMTP server tertentu lainnya.
Tindakan arsip	Mengarsipkan email dengan mengirimkannya ke SES arsip Amazon.

Tindakan & parameternya	Deskripsi
Tambahkan header	Menambahkan header kustom ke email yang diterima.
Penerima email menulis ulang	Mengganti penerima amplop email dengan daftar penerima yang diberikan. Jika kondisi tindakan ini hanya berlaku untuk subset penerima, hanya penerima yang diganti.
Kirim ke kotak surat	Mengirimkan email ke WorkMail kotak surat Amazon.
Kirim ke internet	Menggunakan SES untuk mengirim email ke penerima pada daftar penerima email.
Jatuhkan tindakan	Untuk email dengan beberapa penerima, jika tindakan ini berlaku untuk satu atau lebih (tetapi tidak semua) penerima tersebut, mereka akan dihapus dari daftar penerima email, dan pemrosesan aturan yang berkelanjutan akan berlaku untuk penerima yang tersisa. Jika tindakan ini berlaku untuk semua penerima, pemrosesan aturan berhenti karena semua penerima dikeluarkan dari daftar penerima dan tidak akan menerima email.

SMTPestafet

Karena Mail Manager digunakan di antara lingkungan email Anda (seperti Microsoft 365, Google Workspace, atau On-Premise Exchange) dan internet, Mail Manager menggunakan SMTP relay untuk merutekan email masuk yang diproses oleh Mail Manager ke lingkungan email Anda. Ini juga dapat merutekan email keluar ke infrastruktur email lain seperti server Exchange lain atau gateway email pihak ketiga sebelum mengirim ke penerima akhir.

SMTPRelai adalah komponen penting dari infrastruktur email Anda, yang bertanggung jawab untuk merutekan email antar server secara efisien ketika ditunjuk oleh tindakan aturan yang ditentukan dalam kumpulan aturan.

Secara khusus, SMTP relay dapat mengarahkan email masuk antara SES Mail Manager dan infrastruktur email eksternal seperti Exchange, gateway email on-premise atau pihak ketiga, dan lainnya. Email masuk ke titik akhir ingress akan diproses oleh aturan yang akan merutekan email tertentu ke SMTP relai yang ditunjuk, yang pada gilirannya, akan meneruskannya ke infrastruktur email eksternal yang ditentukan dalam relai. SMTP

Saat titik akhir ingress Anda menerima email, ia menggunakan kebijakan lalu lintas untuk menentukan email mana yang akan diblokir atau diizinkan. Email yang Anda izinkan diteruskan ke kumpulan aturan yang menerapkan aturan bersyarat untuk menjalankan tindakan yang telah Anda tetapkan untuk jenis email tertentu. Salah satu tindakan aturan yang dapat Anda tentukan adalah SMTPRelaytindakan —jika Anda memilih tindakan ini, email akan diteruskan ke SMTP server eksternal yang ditentukan dalam SMTP relay Anda.

Misalnya, Anda dapat menggunakan SMTPRelaytindakan untuk mengirim email dari titik akhir ingress ke Microsoft Exchange Server lokal Anda. Anda akan mengatur server Exchange Anda untuk memiliki SMTP titik akhir publik yang hanya dapat diakses menggunakan kredensi tertentu. Ketika Anda membuat SMTP relay, Anda memasukkan nama server, port, dan kredenal server Exchange Anda dan memberikan SMTP relay Anda nama unik, katakanlah, ""RelayToMyExchangeServer. Kemudian, Anda membuat aturan di set aturan titik akhir ingress Anda yang mengatakan, “Ketika Dari alamat berisi 'gmail.com', lalu lakukan SMTPRelaytindakan menggunakan relay yang dipanggil”. SMTP RelayToMyExchangeServer

Sekarang, ketika email dari gmail.com tiba ke titik akhir ingress Anda, aturan akan memicu SMTPRelaytindakan dan menghubungi server Exchange Anda menggunakan kredensi yang Anda berikan saat membuat SMTP relay dan mengirimkan email ke server Exchange Anda. Dengan demikian, email yang diterima dari gmail.com diteruskan ke server Exchange Anda.

Anda harus terlebih dahulu membuat SMTP relay sebelum dapat ditunjuk dalam tindakan aturan. Prosedur di bagian selanjutnya akan memandu Anda membuat SMTP relai di SES konsol.

Membuat SMTP relay di SES konsol

Prosedur berikut menunjukkan cara menggunakan halaman SMTPrelay di SES konsol untuk membuat SMTP relay dan mengelola yang sudah Anda buat.

Untuk membuat dan mengelola SMTP relay menggunakan konsol

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih SMTPrelay di bawah Mail Manager.
3. Pada halaman SMTPrelay, pilih Buat SMTP relai.
4. Pada halaman Buat SMTP relai, masukkan nama unik untuk SMTP relay Anda.
5. Bergantung pada apakah Anda ingin mengonfigurasi SMTP relai masuk (tidak diautentikasi) atau keluar (diautentikasi), ikuti instruksi masing-masing:

   Inbound

   Untuk mengkonfigurasi relai masuk SMTP

   1. Ketika SMTP relay digunakan sebagai gateway masuk untuk merutekan email masuk yang diproses oleh Mail Manager ke lingkungan email eksternal Anda, Anda harus terlebih dahulu mengonfigurasi lingkungan hosting email. Meskipun setiap penyedia hosting email memiliki alur kerja mereka sendiri GUI dan konfigurasi yang unik bagi mereka, prinsip mengonfigurasinya untuk bekerja dengan gateway masuk, seperti relai Manajer Mail Anda, akan serupa. SMTP

      Untuk membantu mengilustrasikan hal ini, kami telah memberikan contoh cara mengonfigurasi Google Workspaces dan Microsoft Office 365 agar berfungsi dengan SMTP relay Anda sebagai gateway masuk di bagian berikut:

      - Menyiapkan Google Workspaces
      - Menyiapkan Microsoft Office 365

      > **Note**
      >
      > Pastikan bahwa domain tujuan penerima yang Anda tuju adalah identitas domain yang SES diverifikasi. Misalnya, jika Anda ingin mengirimkan email ke penerima abc@example.com dan support@acme.com, domain example.com dan acme.com perlu diverifikasi. SES Jika domain penerima tidak diverifikasi, tidak SES akan mencoba untuk mengirimkan email ke SMTP server publik.

Untuk informasi selengkapnya, lihat the section called “Membuat & memverifikasi identitas”.

2. Setelah mengonfigurasi Google Workspaces atau Microsoft Office 365 agar berfungsi dengan gateway masuk, masukkan nama host SMTP server publik dengan nilai di bawah ini masing-masing ke penyedia Anda:

   - Ruang Kerja Google: `aspmx.l.google.com`
   - Microsoft Office 365: `<your_domain>.mail.protection.outlook.com`

     Ganti titik-titik dengan “-” di nama domain Anda. Misalnya, jika domain Anda adalah acme.com, Anda akan memasukkan `acme-com.mail.protection.outlook.com`

3. Masukkan nomor port 25 untuk SMTP server publik.
4. Biarkan bagian Otentikasi kosong (jangan pilih atau buat rahasiaARN).

Outbound

Untuk mengkonfigurasi relai keluar SMTP

1. Masukkan nama host dari SMTP server publik yang ingin Anda sambungkan.
2. Masukkan nomor port untuk SMTP server publik.
3. Siapkan otentikasi untuk SMTP server Anda dengan memilih salah satu rahasia Anda dari ARNRahasia. Jika Anda memilih rahasia yang dibuat sebelumnya, itu harus berisi kebijakan yang ditunjukkan dalam langkah-langkah berikut untuk membuat rahasia baru.

   - Anda memiliki opsi untuk membuat rahasia baru dengan memilih Create new —the AWS Secrets Manager konsol akan terbuka di mana Anda dapat terus membuat kunci baru:

     a. Pilih Jenis rahasia lainnya dalam tipe Rahasia.

     b. Masukkan kunci dan nilai berikut dalam pasangan kunci/Nilai:

Kunci	nilai
`username`	my_username

Kunci	nilai
`password`	my_kata sandi

> **Note**
>
> Untuk kedua kunci, Anda hanya harus memasukkan `username` dan `password` seperti yang ditunjukkan (hal lain akan menyebabkan otentikasi gagal). Untuk nilainya, masukkan nama pengguna dan kata sandi Anda sendiri.

c. Pilih Tambahkan kunci baru untuk membuat kunci terkelola KMS pelanggan (CMK) di kunci Enkripsi —the AWS KMS konsol akan terbuka.

d. Pilih Create key pada halaman Customer manged keys.

e. Simpan nilai default pada halaman tombol Configure dan pilih Next.

f. Masukkan nama untuk kunci Anda di Alias (opsional, Anda dapat menambahkan deskripsi dan tag), diikuti oleh Berikutnya.

g. Pilih pengguna (selain Anda sendiri) atau peran yang ingin Anda izinkan untuk mengelola kunci di Administrator kunci diikuti oleh Berikutnya.

h. Pilih pengguna (selain Anda sendiri) atau peran yang ingin Anda izinkan untuk menggunakan kunci di Pengguna kunci diikuti oleh Berikutnya.

i. Salin dan tempel KMSCMKkebijakan ke editor JSON teks kebijakan Kunci di `"statement"` tingkat dengan menambahkannya sebagai pernyataan tambahan yang dipisahkan oleh koma. Ganti wilayah dan nomor akun dengan milik Anda.

j. Pilih Selesai.

k. Pilih tab browser Anda di mana Anda memiliki AWS Secrets Manager Simpan halaman rahasia baru terbuka dan pilih ikon penyegaran (panah melingkar) di sebelah bidang kunci Enkripsi, lalu klik di dalam bidang dan pilih kunci yang baru Anda buat.

l. Masukkan nama di bidang Nama rahasia di halaman Konfigurasi rahasia.

m. Pilih Edit izin di Izin sumber daya.

n. Salin dan tempel Rahasia kebijakan sumber daya ke editor JSON teks izin Sumber daya dan ganti wilayah dan nomor akun dengan milik Anda. (Pastikan untuk menghapus kode contoh apa pun di editor.)

o. Pilih Simpan diikuti oleh Berikutnya.

p. Opsional mengkonfigurasi rotasi diikuti oleh Berikutnya.

q. Tinjau dan simpan rahasia baru Anda dengan memilih Store.

r. Pilih tab browser Anda di mana Anda memiliki halaman SES Create new ingress endpoint terbuka dan pilih Refresh list, lalu pilih rahasia yang baru Anda buat di Secret. ARN

6. Pilih Buat SMTP relai.
7. Anda dapat melihat dan mengelola SMTP relay yang telah Anda buat dari halaman SMTPrelay. Jika ada SMTP relay yang ingin Anda hapus, pilih tombol radionya diikuti oleh Hapus.
8. Untuk mengedit SMTP relay, pilih namanya. Pada halaman detail, Anda dapat mengubah nama relai, nama SMTP server eksternal, port, dan kredensi login dengan memilih tombol Edit atau Perbarui yang sesuai diikuti dengan Simpan perubahan.

Menyiapkan Google Workspaces untuk relai masuk (tidak diautentikasi) SMTP

Contoh panduan berikut menunjukkan cara mengatur Google Workspaces agar berfungsi dengan relay masuk (tidak diautentikasi) Mail Manager. SMTP

Prasyarat

- Akses ke konsol administrator Google (konsol administrator Google > Aplikasi> Google Workspace > Gmail).
- Akses ke server nama domain yang menghosting catatan MX untuk domain yang akan digunakan untuk penyiapan Mail Manager.

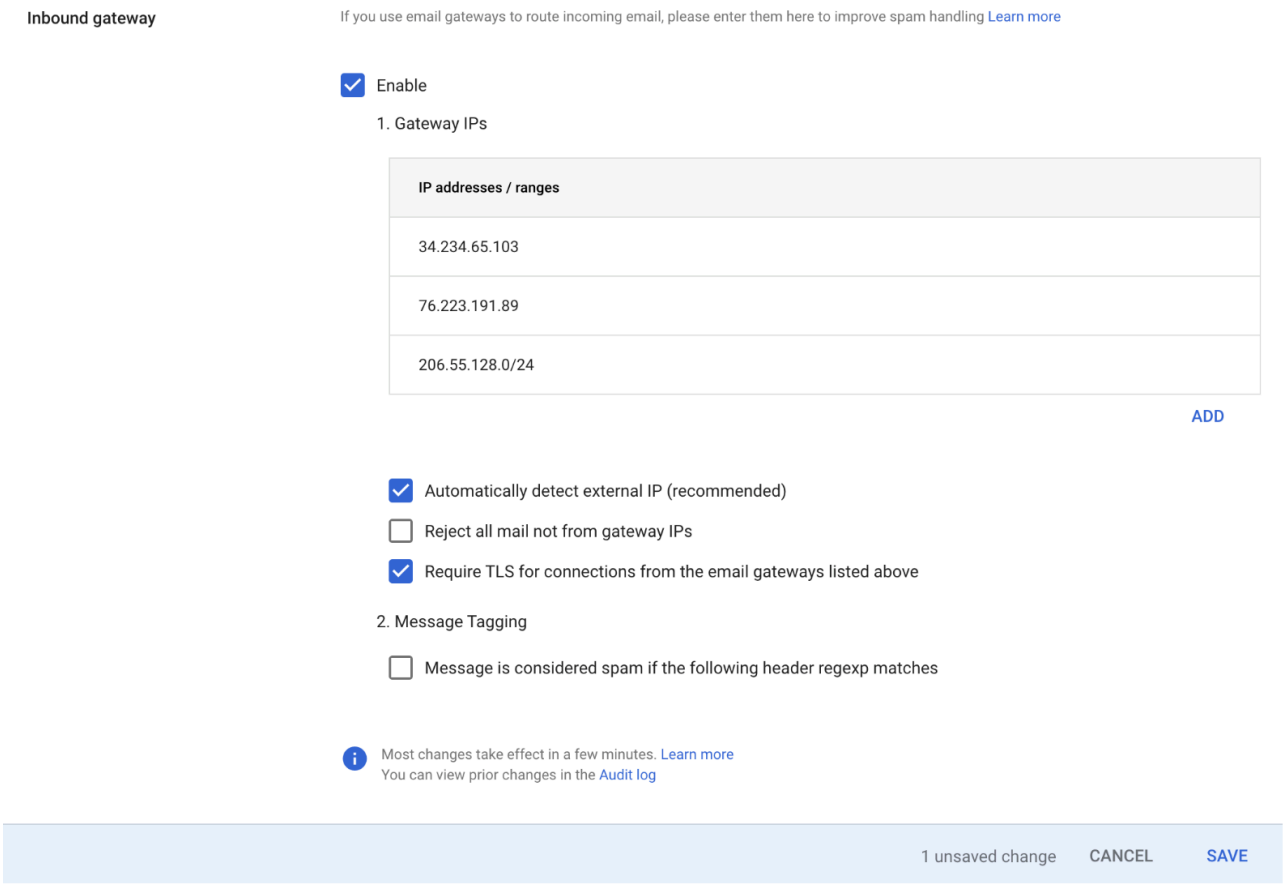
Untuk mengatur Google Workspaces agar berfungsi dengan relai masuk SMTP

- Tambahkan alamat IP Mail Manager ke konfigurasi gateway masuk

  a. Di konsol administrator Google, buka Apps > Google Workspace > Gmail.

  b. Pilih Spam, Phishing, dan Malware, lalu buka konfigurasi gateway masuk.

  c. Aktifkan Inbound gateway, dan konfigurasikan dengan detail berikut:



- Di Gateway IPs, pilih Tambah, dan tambahkan titik akhir ingress IPs khusus ke wilayah Anda dari tabel berikut:

Wilayah	Rentang IP
eu-barat-1/ DUB	206.55.133.0/24
eu-sentral-1/ FRA	206.55.132.0/24
kami-barat-2/ PDX	206.55.131.0/24
ap-timur laut-1/NRT	206.55.130.0/24
kami-timur-1/ IAD	206.55.129.0/24
ap-tenggara 2/SYD	206.55.128.0/24

- Pilih Secara otomatis mendeteksi IP eksternal.

- Pilih TLSMemerlukan koneksi dari gateway email yang tercantum di atas.
- Pilih Simpan di bagian bawah kotak dialog untuk menyimpan konfigurasi. Setelah disimpan, konsol administrator akan menampilkan gateway Inbound sebagai diaktifkan.

Menyiapkan Microsoft Office 365 untuk relay masuk (tidak diautentikasi) SMTP

Contoh panduan berikut menunjukkan cara mengatur Microsoft Office 365 agar bekerja dengan relay masuk (tidak diautentikasi) Manajer Mail. SMTP

Prasyarat

- Akses ke pusat admin Microsoft Security (Pusat admin Microsoft Security > Email & kolaborasi > Kebijakan & Aturan > Kebijakan ancaman).
- Akses ke server nama domain yang menghosting catatan MX untuk domain yang akan digunakan untuk penyiapan Mail Manager.

Untuk mengatur Microsoft Office 365 agar bekerja dengan relai masuk SMTP

1. Tambahkan alamat IP Mail Manager ke daftar Izinkan
   a. Di pusat admin Microsoft Security, buka Email & kolaborasi > Kebijakan & Aturan > Kebijakan Ancaman.
   b. Pilih Anti-spam di bawah Kebijakan.
   c. Pilih Kebijakan filter koneksi diikuti dengan Edit kebijakan filter koneksi.
      - Dalam dialog Selalu izinkan pesan dari alamat IP atau rentang alamat berikut, tambahkan titik akhir ingress IPs khusus ke wilayah Anda dari tabel berikut:

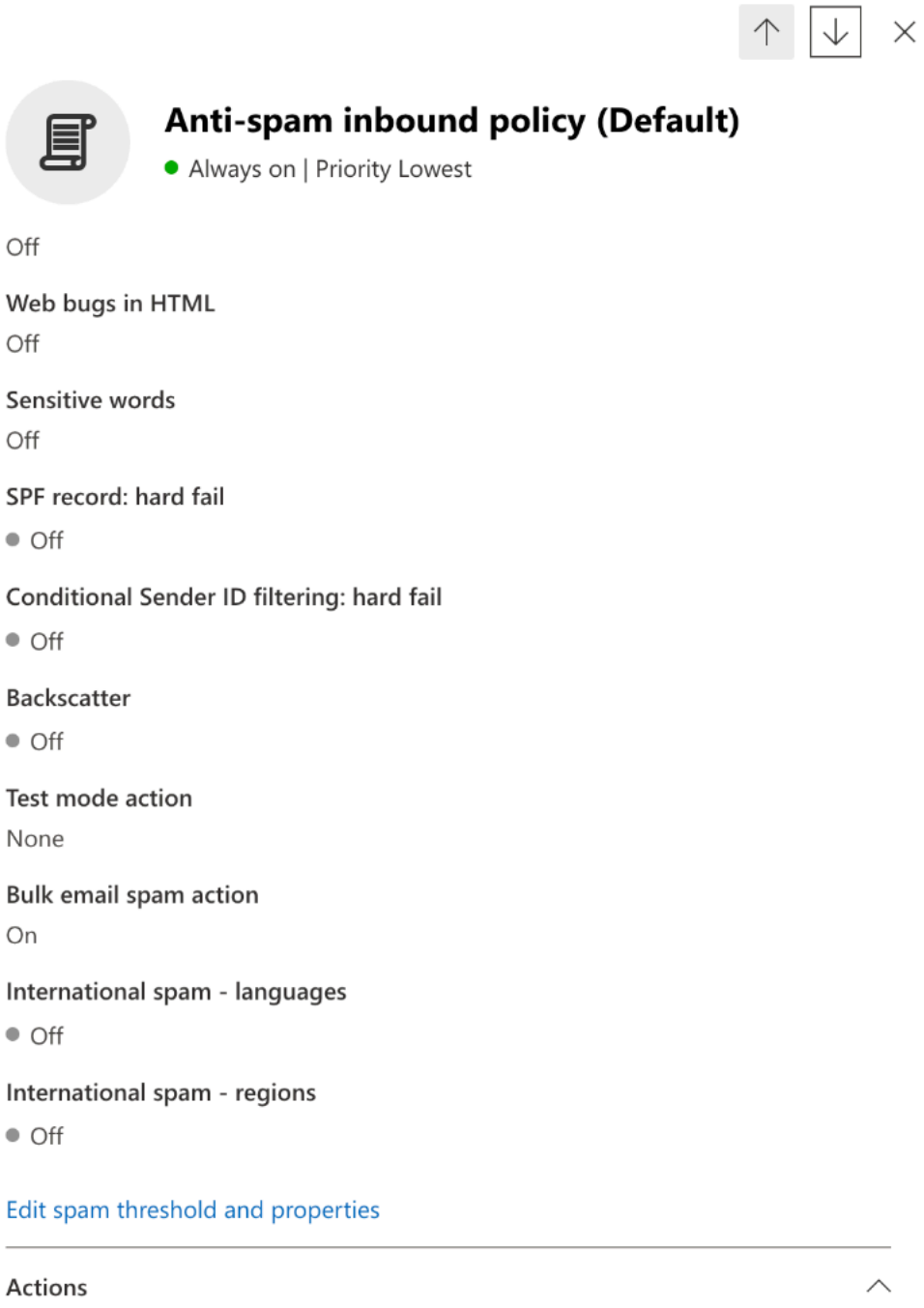
Wilayah	Rentang IP
eu-barat-1/ DUB	206.55.133.0/24
eu-sentral-1/ FRA	206.55.132.0/24
kami-barat-2/ PDX	206.55.131.0/24

Wilayah	Rentang IP
ap-timur laut-1/NRT	206.55.130.0/24
kami-timur-1/ IAD	206.55.129.0/24
ap-tenggara 2/SYD	206.55.128.0/24

- Pilih Simpan.

d. Kembali ke opsi Anti-spam dan pilih Kebijakan masuk Anti-spam.

- Di bagian bawah dialog, pilih Edit ambang batas dan properti spam:



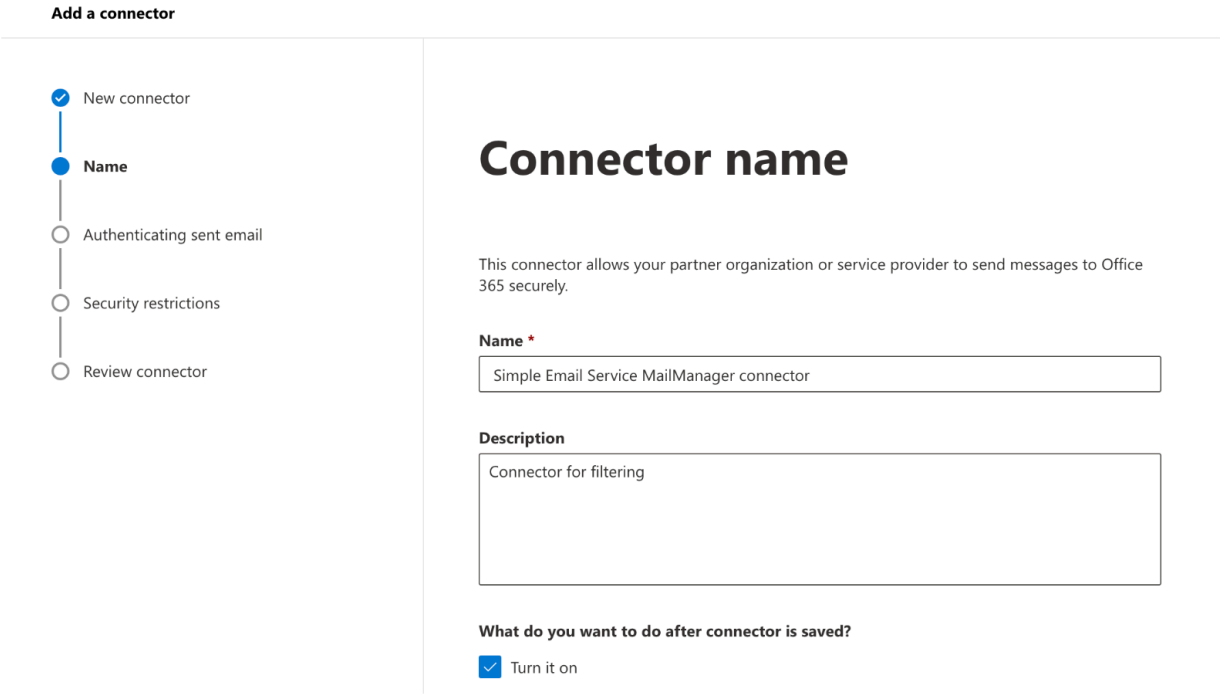
- Gulir ke Tandai sebagai spam dan pastikan bahwa SPFrecord: hard fail diatur ke Off.
- Pilih Simpan.

2. Konfigurasi penyaringan yang disempurnakan (disarankan)

   Opsi ini akan memungkinkan Microsoft Office 365 untuk mengidentifikasi IP penghubung asli dengan benar sebelum pesan diterima oleh SES Mail Manager.

   a. Buat konektor masuk

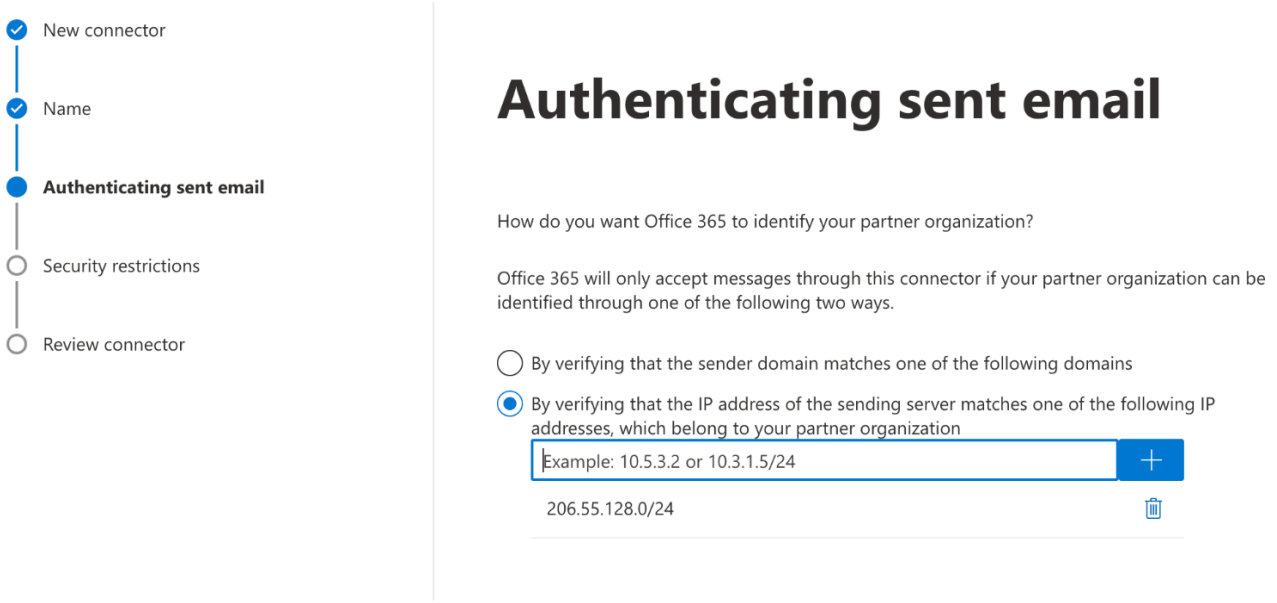
- Masuk ke pusat admin Exchange yang baru dan buka Alur surat > Konektor.
- Pilih Tambahkan konektor.
- Dalam Koneksi dari, pilih Organisasi mitra diikuti oleh Berikutnya.
- Isi kolom sebagai berikut:
  - Nama - Konektor Manajer Surat Layanan Email Sederhana
  - Deskripsi - Konektor untuk penyaringan



- Pilih Selanjutnya.
- Dalam Mengautentikasi email terkirim, pilih Dengan memverifikasi bahwa alamat IP server pengirim cocok dengan salah satu alamat IP berikut, yang merupakan milik organisasi mitra Anda dan tambahkan titik akhir ingress IPs khusus ke wilayah Anda dari tabel berikut:

Wilayah	Rentang IP
eu-barat-1/ DUB	206.55.133.0/24
eu-sentral-1/ FRA	206.55.132.0/24
kami-barat-2/ PDX	206.55.131.0/24

Wilayah	Rentang IP
ap-timur laut-1/NRT	206.55.130.0/24
kami-timur-1/ IAD	206.55.129.0/24
ap-tenggara 2/SYD	206.55.128.0/24

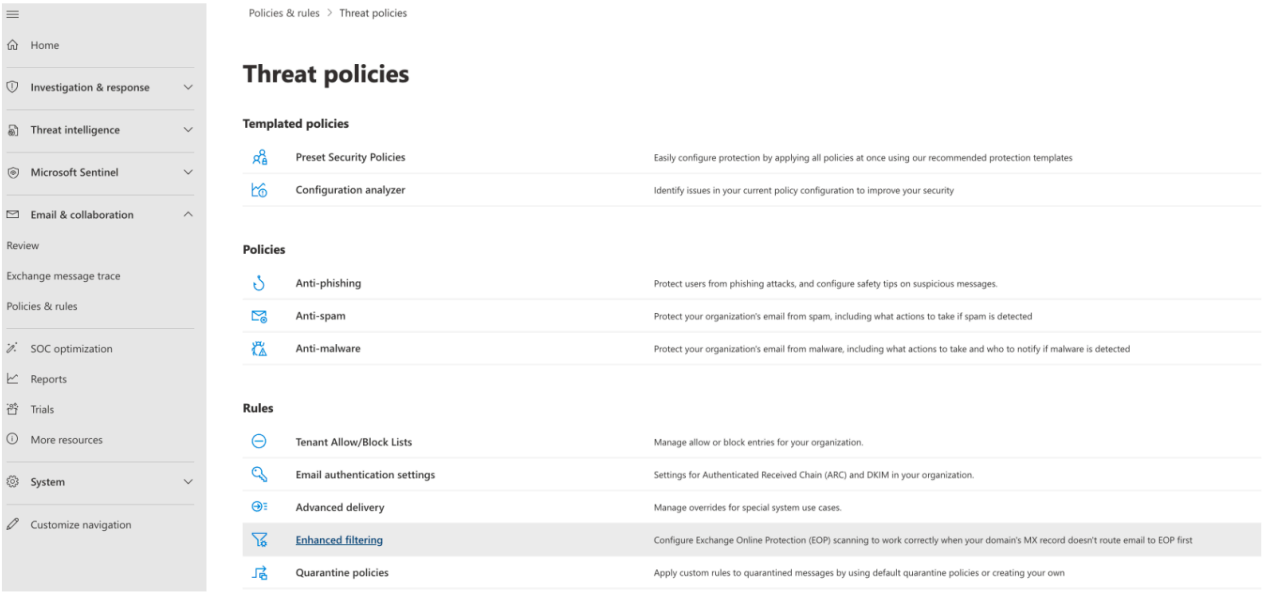


- Pilih Selanjutnya.
- Di Pembatasan keamanan, terima default Tolak pesan email jika tidak dikirim melalui TLS pengaturan, diikuti oleh Berikutnya.
- Tinjau pengaturan Anda dan pilih Buat konektor.

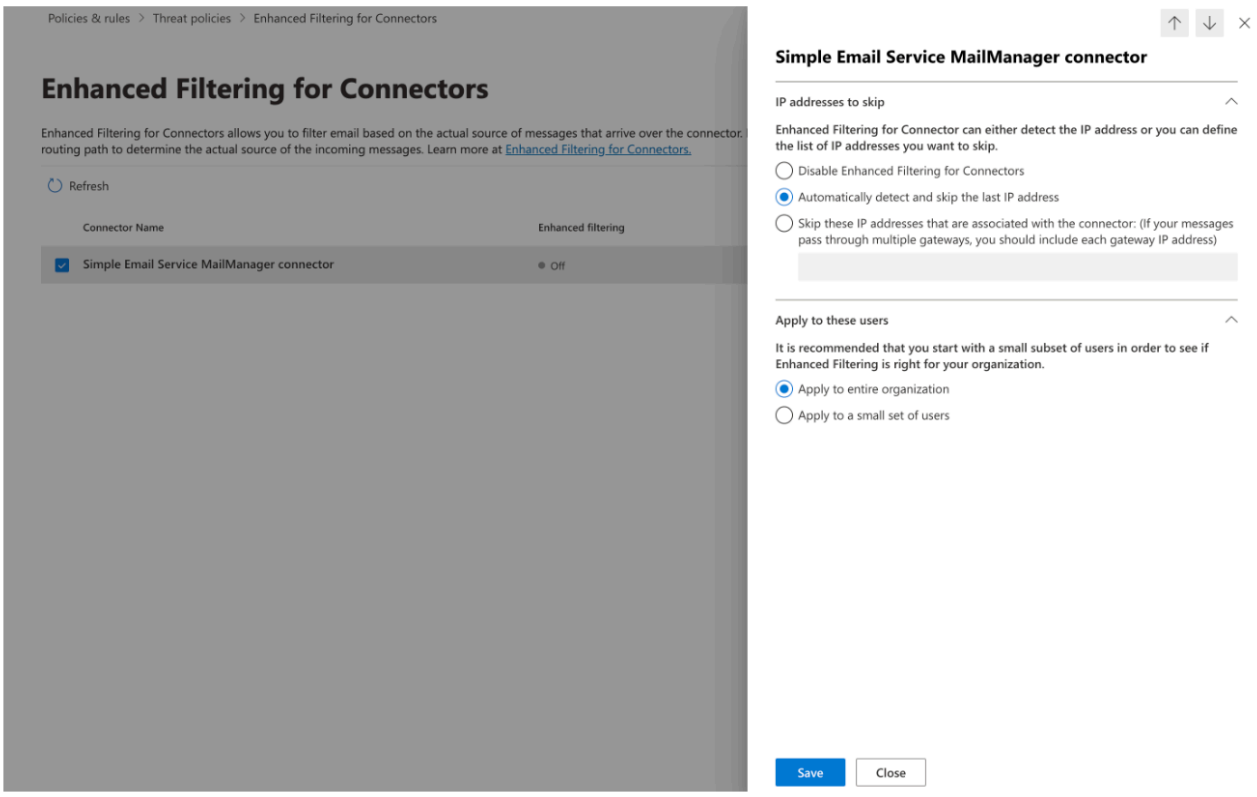
b. Aktifkan penyaringan yang disempurnakan

Sekarang konektor masuk telah dikonfigurasi, Anda harus mengaktifkan konfigurasi penyaringan konektor yang disempurnakan di pusat admin Microsoft Security.

- Di pusat admin Microsoft Security, buka Email & kolaborasi > Kebijakan & Aturan > Kebijakan Ancaman.
- Pilih Pemfilteran yang disempurnakan di bawah Aturan.



- Pilih konektor Pengelola Surat Layanan Email Sederhana yang Anda buat sebelumnya untuk mengedit parameter konfigurasinya.
- Pilih keduanya Deteksi secara otomatis dan lewati alamat IP terakhir dan Terapkan ke seluruh organisasi.



- Pilih Simpan.

Pengarsipan email

Pengarsipan email menyediakan cara bagi Anda untuk mengarsipkan jenis email yang Anda tentukan masuk ke titik akhir ingress Anda serta menyediakan cara untuk menemukan pesan yang diarsipkan melalui serangkaian filter pencarian lanjutan yang kaya dan kemampuan untuk mengekspor hasilnya.

Pengarsipan email menyimpan dan melindungi email Anda dengan menyimpan data dalam penyimpanan jangka panjang yang persisten dan aman, dan memberi Anda cara untuk mencari dan mengarsipkan email dengan cepat. Ini menyediakan pengarsipan tingkat perusahaan penuh waktu tanpa meningkatkan persyaratan penyimpanan server kotak surat Anda.

Saat titik akhir ingress Anda menerima email, ia menggunakan kebijakan lalu lintas untuk menentukan email mana yang akan diblokir atau diizinkan. Email yang Anda izinkan diteruskan ke kumpulan aturan yang menerapkan aturan bersyarat untuk menjalankan tindakan yang telah Anda tetapkan untuk jenis email tertentu. Salah satu tindakan aturan yang dapat Anda tentukan adalah tindakan Arsip —jika Anda memilih tindakan ini, email akan diarsipkan ke arsip email yang Anda tetapkan.

Anda harus terlebih dahulu membuat arsip sebelum dapat ditunjuk dalam tindakan aturan. Prosedur di bagian selanjutnya akan memandu Anda membuat arsip di konsol SES.

Menggunakan pengarsipan email di konsol Amazon SES

Halaman pengarsipan Email di konsol SES terdiri dari empat tabel interaktif, Arsip pencarian, Riwayat pencarian, riwayat Ekspor, dan Kelola arsip, yang dapat Anda gunakan untuk mencari email di arsip Anda, mengekspor hasil, dan mengelola arsip Anda. Dalam prosedur berikut, instruksi disediakan untuk setiap tabel.

Untuk menggunakan halaman pengarsipan Email untuk mencari, mengekspor, dan mengelola arsip Anda

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Pengarsipan email di bawah Manajer Email.
3. Halaman pengarsipan Email terdiri dari empat tabel Arsip pencarian, Riwayat pencarian, riwayat Ekspor, dan Kelola arsip. Untuk instruksi khusus untuk masing-masing tabel ini, pilih tab yang sesuai di bawah ini:

Search archive

Arsip pencarian adalah tabel interaktif yang menyediakan cara bagi Anda untuk mencari dan menemukan pesan yang diarsipkan dengan filter kaya dan set tanggal yang menawarkan kriteria pencarian terperinci untuk menemukan apa pun dari email tertentu hingga banyak email yang cocok dengan kategori yang lebih luas. Pesan yang sesuai dengan kriteria pencarian Anda dapat diunduh satu per satu atau dapat diekspor secara massal ke bucket S3.

Untuk mencari, mengunduh, atau mengekspor email yang diarsipkan

1. Pada halaman Pengarsipan email, pilih tab Arsip pencarian untuk menampilkan tabel arsip pencarian.
2. Klik di dalam bidang Arsip dan pilih arsip dari daftar yang diikuti oleh Pencarian, atau perbaiki pencarian Anda menggunakan langkah-langkah berikut.
3. Pilih bidang Rentang tanggal untuk memperluas opsi rentang tanggal untuk pencarian Anda:

   - Rentang relatif (default) - Pilih tombol radio yang sesuai dengan jumlah hari yang diinginkan, atau pilih rentang Kustom dengan memilih unit waktu dan rentang tanggal hingga 30 hari.
   - Rentang absolut - Masukkan tanggal Mulai dan tanggal Akhir (dan waktu jika diinginkan) hingga 30 hari.

   > **Note**
   >
   > - Pencarian dalam arsip dibatasi hingga 30 hari sekaligus. Misalnya, jika Anda ingin mencari pesan dari 1 Juni hingga 31 Juli, Anda harus memecahnya menjadi tiga pencarian sebagai berikut:
   >
   >   1. 30 hari di bulan Juni.
   >   2. 30 hari pertama di bulan Juli.
   >   3. Hari 31 Juli.
   >
   > - Untuk tanggal rentang Relatif, hari terakhir berakhir pada tengah malam. Misalnya, jika Anda memilih 7 hari terakhir, hari ketujuh adalah kemarin, berakhir pada tengah malam.

4. (Opsional) Pilih bidang Filter untuk dipilih di antara filter berikut: Dari, Ke, CC, Baris subjek, dan Memiliki lampiran —properti berikut berlaku:

- Anda dapat membuat hingga 10 filter.
- Filter dapat diedit dengan mengkliknya, atau dihapus dengan memilih X.

5. Pilih Cari dan email yang diarsipkan yang cocok dengan kriteria pencarian Anda akan diisi dalam tabel Hasil pencarian.

   - Kolom ID Pesan disembunyikan secara default, tetapi dapat ditampilkan dengan memilih ikon roda gigi untuk menyesuaikan cara Anda melihat tabel.
   - Setiap pencarian yang Anda jalankan secara otomatis disimpan dengan id pencarian unik dan akan tercantum dalam tabel riwayat Pencarian.

6. Untuk melihat teks pesan beserta informasi amplop dan headernya, pilih tombol radio pesan diikuti oleh Lihat detail untuk membuka bilah sisi Detail pesan.
7. Untuk membuat file lokal pesan, pilih tombol radio pesan diikuti oleh Unduh pesan.
8. Pencarian Anda yang difilter dapat disimpan ke bucket Amazon S3 dengan memilih Ekspor ke S3.

   a. Jika Anda mengetahui URI bucket S3 yang ingin Anda gunakan, masukkan di bidang URI S3; jika tidak, pilih Browse S3 dan pilih bucket dan folder S3 untuk digunakan di halaman S3.

   b. (Opsional) Anda dapat mengenkripsi pesan yang diekspor baik dengan memasukkan AWS KMS kunci Anda sendiri ke bidang ARN kunci KMS, atau dengan memilih Buat kunci baru. Jika tidak, enkripsi akan disetel ke metode apa pun yang digunakan pada bucket S3 tujuan (meskipun tidak ada).

   c. Pilih Ekspor dan semua pesan yang ditemukan dalam pencarian Anda yang difilter akan disimpan sebagai file individual di folder S3 yang Anda pilih.

> **Note**
>
> Meskipun tidak ada batasan berapa banyak pesan yang dapat berisi arsip Anda, hasil penelusuran dibatasi hingga 1000 baris dalam tabel hasil Penelusuran.

Search history

Riwayat pencarian Anda tercantum dalam tabel ini sehingga Anda dapat mengembalikan set hasil atau mengakses set filter kompleks yang dibuat sebelumnya. Anda juga dapat membuat

pencarian baru berdasarkan pencarian asli dengan mengedit filter dan tanggal. Setiap pencarian baru secara otomatis disimpan dengan ID pencarian unik dan akan tercantum dalam tabel ini.

Untuk melihat dan bekerja dengan penelusuran Anda sebelumnya

1. Pada halaman pengarsipan Email, pilih tab Riwayat pencarian untuk menampilkan tabel riwayat pencarian yang mencantumkan riwayat semua penelusuran email yang diarsipkan dengan yang terbaru di atas. Tabel ini memuat data saat pertama kali mengunjunginya — jika Anda beralih tab dan kembali, gunakan ikon penyegaran untuk mengambil data terbaru.
2. Klik di dalam bidang Arsip dan pilih arsip dari daftar—semua pencarian milik arsip itu akan diisi dalam tabel. Anda dapat melihat dan melakukan lebih banyak dengan pencarian individual dalam langkah-langkah di bawah ini.
3. Pilih tombol radio dari pencarian sebelumnya diikuti oleh Lihat hasil pencarian untuk mengembalikan hasil pencarian aslinya — halaman arsip Pencarian akan terbuka menampilkan set filter dan rentang tanggal yang digunakan untuk pencarian asli bersama dengan semua pesan yang ditemukan sebelumnya berdasarkan kriteria tersebut. Anda dapat memperluas pencarian asli dengan cara berikut:
   - Buat pencarian baru dengan memodifikasi rentang tanggal dan filter diikuti oleh Penelusuran.
   - Setiap pencarian baru yang Anda lakukan secara otomatis disimpan dengan ID pencarian unik dan akan tercantum dalam tabel riwayat Pencarian.

Export history

Riwayat ekspor Anda tercantum dalam tabel ini yang memungkinkan akses mudah ke konten folder ekspor di konsol S3.

Untuk melihat ekspor terbaru Anda

1. Pada halaman Pengarsipan email, pilih tab Riwayat ekspor untuk menampilkan tabel riwayat Ekspor yang mencantumkan semua penelusuran email yang diarsipkan yang Anda ekspor ke bucket S3 dalam 30 hari terakhir. Tabel ini memuat data saat pertama kali mengunjunginya — jika Anda beralih tab dan kembali, gunakan ikon penyegaran untuk mengambil data terbaru.
2. Jika status ekspor Antrean, Preprocessing atau Processing, Anda dapat membatalkannya dengan memilih Batal.

3. Pilih URI S3 untuk membuka folder bucket ekspor di konsol S3 tempat Anda dapat melihat file yang dikandungnya.

Manage archives

Tabel ini mencantumkan arsip Anda di mana Anda memiliki opsi untuk membuat arsip baru, mencari arsip tertentu dan melihat detailnya, mengedit arsip, atau menghapus arsip.

Untuk membuat dan mengelola arsip

1. Pada halaman pengarsipan Email, pilih tab Kelola arsip untuk menampilkan tabel Arsip yang mencantumkan semua arsip email Anda. Tabel ini memuat data saat pertama kali mengunjunginya — jika Anda beralih tab dan kembali, gunakan ikon penyegaran untuk mengambil data terbaru.
2. Untuk mencari arsip tertentu, mulailah mengetik di bidang Arsip.
3. Untuk melihat detail arsip, pilih namanya di kolom Nama arsip.
4. Untuk membuat arsip, pilih Buat arsip.

   a. Masukkan nama unik di bidang Nama arsip.

   b. (Opsional) Pilih periode retensi di bidang Periode retensi untuk mengganti periode retensi default 180 hari.

   c. (Opsional) Anda dapat mengenkripsi arsip Anda baik dengan memasukkan AWS KMS kunci Anda sendiri ke bidang ARN kunci KMS, atau dengan memilih Buat kunci baru.

   Pilih Buat arsip.
5. Untuk mengedit arsip, pilih tombol radionya diikuti oleh Edit.

   a. Edit atau ubah nama di bidang Nama arsip.

   b. Ubah periode retensi di bidang Periode retensi.

   Pilih Perbarui arsip.
6. Untuk menghapus arsip, pilih tombol radionya diikuti oleh Hapus.

   - Ketik `delete` kolom Konfirmasi diikuti dengan Hapus.

Status arsip akan beralih ke penghapusan Tertunda di tabel Arsip dan akan dihapus secara otomatis setelah 30 hari.

> **Note**
>
> Jika Anda ingin membatalkan penghapusan ini, buat tiket ke Amazon SES dalam waktu 30 hari.

Email Tambah Ons

Email Add Ons adalah kumpulan alat keamanan khusus dari penyedia yang disetujui SES yang dapat digunakan untuk mengelola jenis email yang Anda izinkan ke titik akhir masuk Anda dan untuk menentukan tindakan yang akan diambil pada jenis email tertentu. Alat-alat ini adalah intelijen keamanan bersertifikat dan solusi penegakan hukum yang siap diintegrasikan ke dalam alur kerja email Anda dan dapat diaktifkan langsung dari konsol Mail Manager.

Add Ons ini menawarkan fleksibilitas untuk memilih di antara solusi keamanan email yang diperiksa sesuai dengan kasus penggunaan pribadi Anda yang dapat digunakan berdasarkan harga terukur, sebagai lawan membeli solusi produk tunggal yang besar yang mungkin tidak dioptimalkan untuk kebutuhan Anda. Email Add Ons memperluas fitur intelijen ancaman inti dan penegakan keamanan berdasarkan per beban kerja, jadi tidak ada dugaan tentang kapasitas yang diperlukan. Manfaat ini memungkinkan Anda untuk fokus untuk tetap berada di depan masalah keamanan email dan mempertahankan standar layanan yang tinggi untuk organisasi Anda.

Anda dapat mempelajari lebih lanjut tentang setiap Add On langsung dari halaman Add On Email yang terletak di konsol Mail Manager tempat Anda akan memiliki akses ke deskripsi produk, manfaat utama, dan informasi harga. Setelah Anda memutuskan Add On yang ingin Anda gunakan, cukup berlangganan dari konsol Mail Manager. Setelah berlangganan, Anda akan dapat memilihnya sebagai kondisi kebijakan lalu lintas dalam menentukan email yang diizinkan masuk ke titik akhir masuk, atau sebagai kondisi yang ditetapkan aturan untuk menentukan tindakan yang akan diambil pada email tertentu. Dukungan utama untuk semua Add Ons disediakan oleh AWS dan juga dapat diakses dari konsol Mail Manager.

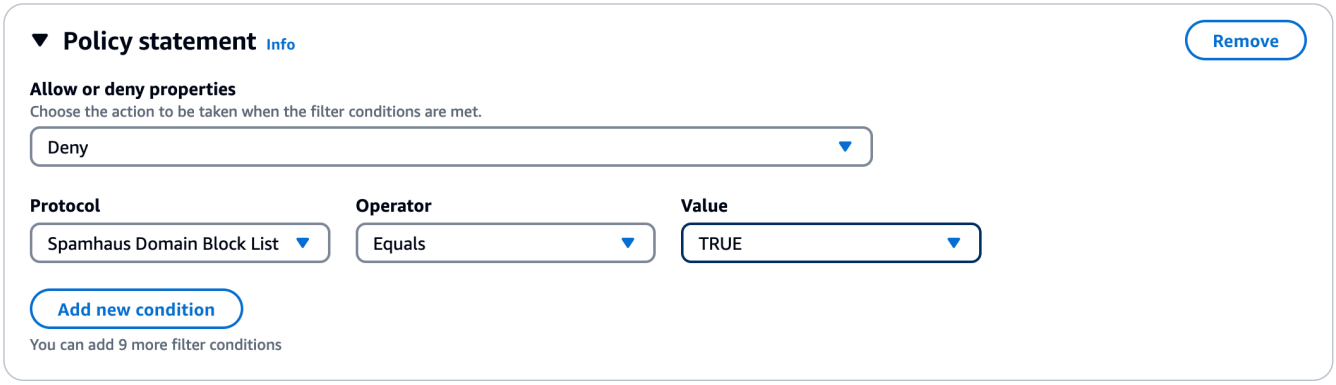
Prosedur di bagian selanjutnya akan memandu Anda berlangganan Email Add On di konsol Mail Manager.

Berlangganan Email Add Ons di konsol Mail Manager

Prosedur berikut menunjukkan cara menggunakan halaman Add On Email di konsol Mail Manager untuk berlangganan Add On sehingga dapat digunakan dalam kebijakan lalu lintas atau kumpulan aturan apa pun.

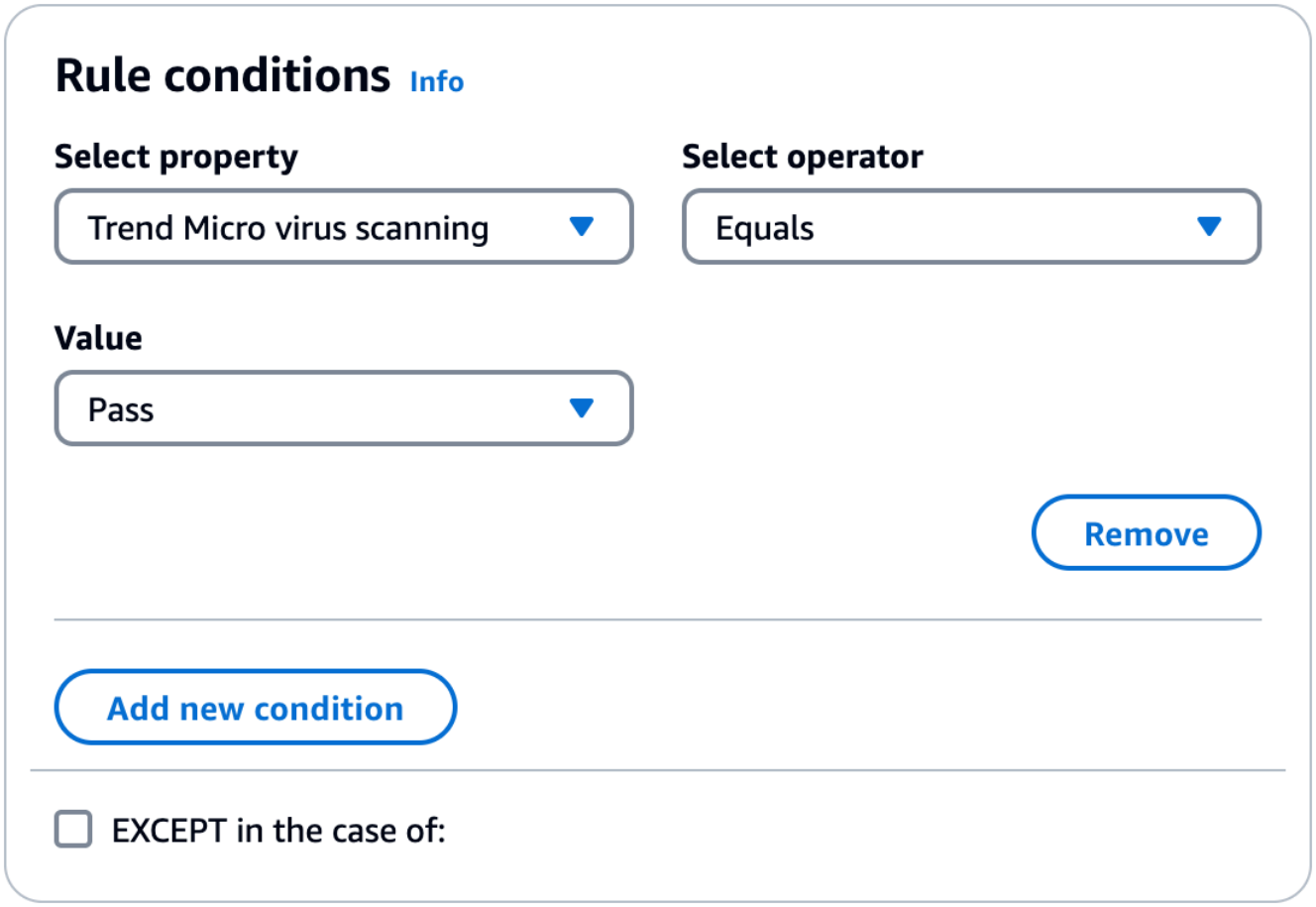
Untuk berlangganan Email Add On menggunakan konsol

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi kiri, pilih Email Add Ons di bawah Mail Manager.
3. Pada halaman Add On Email, pilih judul kartu Add On untuk membuka halaman ikhtisar di mana Anda dapat mempelajari lebih lanjut tentang apa yang dilakukannya, manfaat utamanya, dan informasi harga. Jika Anda ingin menggunakan Add On ini, pilih Berlangganan.
   - Baca Syarat dan Ketentuan yang disajikan dan centang kotak Saya terima diikuti oleh Berlangganan.
4. Setelah berlangganan Add On, Anda akan dapat mengintegrasikannya ke dalam alur kerja email Anda dengan memilihnya sebagai kondisi kebijakan lalu lintas untuk menolak atau mengizinkan email ke titik akhir ingress Anda, atau kondisi yang ditetapkan aturan untuk menentukan tindakan yang harus diambil pada pesan yang memenuhi syarat. Contoh berikut menggambarkan penggunaan Add On dalam kondisi pernyataan kebijakan dan dalam kondisi aturan:
   - Menggunakan Daftar Blok Domain Spamhaus Add On dalam kondisi pernyataan kebijakan untuk memblokir email yang masuk ke titik akhir ingress Anda yang berasal dari domain yang terdaftar di Spamhaus:



   - Untuk detail tentang cara membuat kebijakan lalu lintas dan membuat kondisi pernyataan kebijakan dengan Email Add Ons, lihatthe section called “Membuat kebijakan lalu lintas & pernyataan kebijakan (konsol)”.

- Menggunakan Trend Micro Virus Scanning Add On dalam kondisi aturan untuk menentukan tindakan aturan untuk email yang melewati pemindaian virus:



- Untuk detail tentang cara membuat kumpulan aturan dan membangun kondisi aturan dengan Email Add Ons, lihatthe section called “Membuat set aturan & aturan (konsol)”.

5. Untuk melihat detail umum atau mengakses dukungan untuk Add On yang Anda berlangganan, pilih namanya di halaman Email Add Ons untuk membuka halaman ikhtisar:

   - Secara umum, Anda dapat melihat tanggal kapan Anda berlangganan dan Amazon Resource Name (ARN) dari Add On Anda.
   - Pilih tab Support untuk mengakses link ke AWS Support.

6. Untuk berhenti berlangganan dari Add On:

   a. Anda harus terlebih dahulu menghapusnya dari salah satu kebijakan lalu lintas atau set aturan di mana Anda telah mendefinisikannya dalam suatu kondisi; jika tidak, langkah-langkah berhenti berlangganan berikut akan gagal.

b. Pilih namanya di halaman Email Add Ons untuk membuka halaman ikhtisar diikuti dengan Berhenti Berlangganan.

c. Ketik `confirm` kolom Konfirmasi diikuti dengan Berhenti Berlangganan.

Kebijakan izin untuk Mail Manager

Kebijakan dalam Bab ini disediakan sebagai titik acuan tunggal untuk kebijakan yang diperlukan untuk memanfaatkan semua fitur yang berbeda dari Mail Manager.

Di halaman fitur Mail Manager, tautan disediakan yang akan membawa Anda ke bagian masing-masing di halaman ini yang berisi kebijakan yang Anda butuhkan untuk memanfaatkan fitur tersebut. Pilih ikon salin kebijakan yang Anda butuhkan dan tempel sesuai petunjuk dalam narasi fitur masing-masing.

Kebijakan berikut memberi Anda izin untuk menggunakan berbagai fitur yang terdapat di Amazon SES Mail Manager melalui kebijakan dan AWS Secrets Manager kebijakan izin sumber daya. Jika Anda baru mengenal kebijakan izin, lihat the section called “Anatomi kebijakan” dan Kebijakan Izin untuk AWS Secrets Manager.

Kebijakan izin untuk titik akhir Ingress

Kedua kebijakan di bagian ini diperlukan untuk membuat titik akhir ingress. Untuk mempelajari cara membuat titik akhir ingress dan tempat menggunakan kebijakan ini, lihat. the section called “Membuat titik akhir ingress (konsol)”

Secrets Manager merahasiakan kebijakan izin sumber daya untuk titik akhir ingress

Kebijakan izin sumber daya rahasia Secrets Manager berikut diperlukan SES untuk mengizinkan akses rahasia menggunakan sumber daya titik akhir ingress.

```
{
    "Version": "2012-10-17",
    "Id": "Id",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
```

```
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "000000000000"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ses:us-east-1:000000000000:mailmanager-
ingress-point/*"
                }
            }
        }
    ]
}
```

KMSkebijakan kunci kunci terkelola pelanggan (CMK) untuk titik akhir ingress

Kebijakan kunci (CMK) kunci terkelola KMS pelanggan berikut diperlukan SES untuk mengizinkan penggunaan kunci Anda saat menggunakan rahasia Anda.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "ses.amazonaws.com"
    },
    "Action": "kms:Decrypt",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
           "kms:ViaService": "secretsmanager.us-east-1.amazonaws.com",
            "aws:SourceAccount": "000000000000"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:aws:ses:us-east-1:000000000000:mailmanager-ingress-
point/*"
        }
    }
}
```

Kebijakan izin untuk SMTP relay

Kedua kebijakan di bagian ini diperlukan untuk membuat SMTP relay. Untuk mempelajari cara membuat SMTP relay dan tempat menggunakan kebijakan ini, lihatthe section called “Membuat SMTP relay (konsol)”.

Secrets Manager rahasia kebijakan izin sumber daya untuk SMTP relay

Kebijakan izin sumber daya rahasia Secrets Manager berikut diperlukan SES untuk memungkinkan mengakses rahasia menggunakan sumber daya SMTP relay.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Principal": {
                "Service": [
                    "ses.amazonaws.com"
                ]
            },
            "Resource": "*",
            "Condition": {
               "StringEquals": {
                   "aws:SourceAccount": "888888888888"
                },
               "ArnLike": {
                   "aws:SourceArn": "arn:aws:ses:us-east-1:888888888888:mailmanager-
smtp-relay/*"
               }
           }
        }
    ]
}
```

KMSkebijakan kunci (CMK) kunci terkelola pelanggan untuk SMTP relay

Kebijakan kunci (CMK) kunci terkelola KMS pelanggan berikut diperlukan SES untuk mengizinkan penggunaan kunci Anda saat menggunakan rahasia Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.us-east-1.amazonaws.com",
                    "aws:SourceAccount": "000000000000"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:ses:us-east-1:000000000000:mailmanager-
smtp-relay/*"
                }
            }
        }
    ]
}
```

Kebijakan izin untuk pengarsipan Email

Dasar Kebijakan IAM identitas pengarsipan

Ini adalah kebijakan IAM identitas untuk mengotorisasi operasi pengarsipan. Kebijakan ini saja mungkin tidak cukup untuk beberapa operasi (lihat Enkripsi pengarsipan KMS CMK dan ekspor Pengarsipan).

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ses:CreateArchive",
                "ses:TagResource"
            ],
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/*"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:RequestTag/key-name": [
                        "value1",
                        "value2"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:ListArchives"
            ],
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:GetArchive",
                "ses:DeleteArchive",
                "ses:UpdateArchive"
            ],
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/MyArchiveID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:ListArchiveSearches"
            ],
```

```
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:GetArchiveSearch",
                "ses:GetArchiveSearchResults",
                "ses:StartArchiveSearch",
                "ses:StopArchiveSearch"
            ],
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/MyArchiveID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:GetArchiveMessage",
                "ses:GetArchiveMessageContent"
            ],
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/MyArchiveID"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:ListArchiveExports"
            ],
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:GetArchiveExport",
                "ses:StartArchiveExport",
                "ses:StopArchiveExport"
            ],
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/MyArchiveID"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:ListTagsForResource",
                "ses:UntagResource"
            ],
            "Resource": [
                "arn:aws:ses:us-east-1:000000000000:mailmanager-archive/MyArchiveID"
            ]
        }
    ]
}
```

Pengarsipan ekspor

Ini adalah kebijakan IAM identitas (selain kebijakan Pengarsipan Dasar di atas) yang diperlukan untuk`StartArchiveExport`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::MyDestinationBucketName"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectTagging",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::MyDestinationBucketName/*"
        }
    ]
}
```

Ini adalah kebijakan untuk ember tujuan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::MyDestinationBucketName"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectTagging",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::MyDestinationBucketName/*"
        }
    ]
}
```

> **Note**
>
> Pengarsipan tidak mendukung kunci kondisi wakil yang membingungkan (aws:SourceArn, aws:SourceAccount, aws: SourceOrg ID, atau aws:SourceOrgPaths). Ini karena pengarsipan email Mail Manager mencegah masalah deputi yang membingungkan dengan menguji apakah identitas panggilan memiliki izin tulis ke bucket tujuan ekspor menggunakan Sesi Akses Teruskan sebelum memulai ekspor yang sebenarnya.

Mengarsipkan enkripsi saat istirahat dengan KMS CMK

Ini adalah enkripsi yang diam dengan kebijakan Kunci Terkelola KMS Pelanggan (CMK) (selain kebijakan Pengarsipan Dasar di atas) yang diperlukan untuk membuat dan bekerja dengan arsip (memanggil Pengarsipan APIs apa pun).

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "kms:DescribeKey",
            "kms:Decrypt",
            "kms:GenerateDataKey"
        ],
        "Resource": "arn:aws:kms:us-west-2:111122223333:key/MyKmsKeyArnID"
    }
}
```

Ini adalah kebijakan KMS utama yang diperlukan untuk pengarsipan email.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:user/MyUserRoleOrGroupName"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "ses.us-east-1.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Kebijakan izin dan kepercayaan untuk menjalankan tindakan aturan

Peran eksekusi SES aturan adalah peran AWS Identity and Access Management (IAM) yang memberikan izin eksekusi aturan untuk mengakses AWS layanan dan sumber daya. Sebelum membuat aturan dalam kumpulan aturan, Anda harus membuat IAM peran dengan kebijakan yang memungkinkan akses ke AWS sumber daya yang diperlukan. SESmengasumsikan peran ini saat menjalankan tindakan aturan. Misalnya, Anda dapat membuat peran eksekusi aturan yang memiliki izin untuk menulis pesan email ke bucket S3 sebagai tindakan aturan yang harus diambil saat kondisi aturan terpenuhi.

Dengan demikian, kebijakan kepercayaan berikut diperlukan selain kebijakan izin individu di bagian ini yang diperlukan untuk menjalankan tindakan aturan Tulis ke S3, Kirim ke kotak pesan, dan Kirim ke internet.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "ses.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
          "StringEquals": {
                   "aws:SourceAccount": "888888888888"
          },
          "ArnLike": {
```

```
            "aws:SourceArn": "arn:aws:ses:us-east-1:888888888888:mailmanager-rule-set/
*"
          }
        }
      }
    ]
  }
```

Kebijakan izin untuk tindakan aturan Menulis ke S3

Kebijakan berikut diperlukan untuk menggunakan tindakan aturan Write to S3 yang mengirimkan email yang diterima ke bucket S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutObject",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::MyDestinationBucketName/*"
            ]
        },
        {
            "Sid": "AllowListBucket",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::MyDestinationBucketName"
            ]
        }
    ]
}
```

Jika Anda menggunakan kunci terkelola AWS KMS pelanggan untuk bucket S3 dengan enkripsi sisi server diaktifkan, Anda harus menambahkan tindakan kebijakan IAM peran,.

`"kms:GenerateDataKey*"` Menggunakan contoh sebelumnya, menambahkan tindakan ini ke kebijakan peran Anda akan muncul sebagai berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowKMSKeyAccess",
            "Effect": "Allow",
            "Action": "kms:GenerateDataKey*",
            "Resource": "arn:aws:kms:us-east-1:888888888888:key/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "kms:ResourceAliases": [
                        "alias/MyKeyAlias"
                    ]
                }
            }
        }
    ]
}
```

Untuk informasi selengkapnya tentang melampirkan kebijakan ke AWS KMS kunci, lihat Menggunakan Kebijakan Utama AWS KMS di Panduan AWS Key Management Service Pengembang.

Kebijakan izin untuk tindakan aturan Kirim ke kotak pesan

Kebijakan berikut diperlukan untuk menggunakan tindakan aturan Kirim ke kotak pesan yang mengirimkan email yang diterima ke akun Amazon WorkMail.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": ["workmail:DeliverToMailbox"],
        "Resource": "arn:aws:workmail:us-
east-1:888888888888:organization/MyWorkMailOrganizationID>"
      }
    ]
  }
```

Kebijakan izin untuk tindakan aturan Kirim ke internet

Kebijakan berikut diperlukan untuk menggunakan tindakan aturan Kirim ke internet yang mengirimkan email yang diterima ke domain eksternal.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": ["ses:SendEmail", "ses:SendRawEmail"],
        "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com"
      }
    ]
  }
```

Mengelola daftar dan langganan Amazon Simple Email Service

Anda dapat mengelola daftar Anda sendiri untuk surat dan langganan serta untuk penindasan email di Amazon. SES Untuk membantu Anda mempertahankan reputasi pengirim, SES menawarkan penindasan tingkat akun dan konfigurasi set-level yang mencegah Anda mengirim ke penerima yang tidak valid dan merusak reputasi pengirim Anda. Sebagai langkah lain terhadap email dan keluhan yang memantul, secara otomatis SES dapat menambahkan tautan berhenti berlangganan ke semua email keluar melalui manajemen berlangganan.

Masing-masing jenis daftar ini dibahas secara rinci di bagian yang tercantum dalam topik Bab ini; Namun, ikhtisar daftar penindasan disajikan di sini untuk memahami perbedaannya serta perubahan kunci dengan manajemen daftar penindasan global. Disarankan agar Anda membaca ikhtisar ini sebelum bekerja dengan salah satu daftar yang dibahas dalam pasal ini.

Ikhtisar daftar penekanan dan mekanisme penggantian penekanan

Fitur penghapusan daftar penindasan global tidak lagi menghadapi pelanggan dan Anda tidak lagi berinteraksi dengannya untuk mengelola penindasan. Daftar penindasan global beroperasi dan dikelola di latar belakang olehSES. Sebagai pelanggan, Anda sekarang telah menyediakan daftar penekanan tingkat akun dan penggantian penekanan set-level konfigurasi yang menawarkan Anda kontrol yang lebih disesuaikan atas cara Anda menangani penekanan email untuk akun Anda sendiri.

Berbagai jenis daftar penindasan, ruang lingkupnya, dan keuntungan apa yang mereka tawarkan dijelaskan di bawah ini.

- Daftar penindasan global — Dimiliki dan dikelola oleh SES untuk melindungi reputasi alamat di kumpulan IP SES bersama.
- Daftar penindasan tingkat akun - Dimiliki dan dikelola oleh pelanggan untuk melindungi reputasi akun mereka - mengesampingkan daftar penindasan global.
  - Penindasan set-level konfigurasi — Mekanisme penggantian untuk memberikan kontrol bersyarat atau halus dari daftar penekanan tingkat akun melalui penggunaan penggantian yang ditentukan dalam set konfigurasi.

Daftar penindasan global adalah satu-satunya jenis daftar penindasan sampai tingkat akun dan penekanan set-level konfigurasi diperkenalkan di konsol Amazon baru dan v2. SES API Daftar

penindasan global dimiliki dan dikelola oleh SES untuk melindungi reputasi. SES Ini diperlukan karena semua SES pelanggan berbagi kumpulan alamat IP yang sama (kecuali mereka telah berdedikasiIPs), penting SES untuk memastikan bahwa pelanggan tidak mengirim spam atau apa pun yang akan berdampak negatif terhadap reputasi alamat IP tersebut di kolam IP SES bersama. Meskipun Anda tidak lagi berinteraksi langsung dengan daftar penindasan global, itu masih beroperasi di latar belakang dan prinsip umum tentang cara kerja daftar penindasan global juga dapat diterapkan untuk menjelaskan prinsip-prinsip keseluruhan tentang bagaimana jenis penindasan lainnya bekerja. Lihat Daftar penindasan SES global Amazon.

> **Note**
>
> Formulir permintaan penghapusan daftar penindasan global tidak lagi ada di SES konsol Amazon karena daftar penekanan tingkat akun telah menggantikannya untuk semua keuntungan yang dijelaskan di bagian ini.

Daftar penindasan tingkat akun diperkenalkan sehingga pelanggan dapat membuat dan mengontrol daftar dan reputasi penindasan mereka sendiri, dengan demikian, daftar penekanan tingkat akun hanya berlaku untuk akun Anda. Antarmuka daftar penekanan tingkat akun di konsol baru menyediakan cara mudah untuk mengelola alamat dalam daftar penekanan tingkat akun Anda, termasuk tindakan massal untuk menambah atau menghapus alamat. Jika alamat ada di daftar penindasan global, tetapi tidak pada daftar penindasan tingkat akun Anda (yang berarti Anda ingin mengirimkannya), dan Anda mengirimkannya, Amazon SES akan tetap mencoba pengiriman, tetapi jika memantul, pantulan akan memengaruhi reputasi Anda sendiri, tetapi tidak ada orang lain yang akan mendapatkan pantulan karena mereka tidak dapat mengirim ke alamat email itu jika mereka tidak menggunakan daftar penindasan tingkat akun mereka sendiri; oleh karena itu, daftar penindasan tingkat akun mengesampingkan daftar penindasan global hanya untuk akun Anda. Lihat Menggunakan daftar SES penindasan tingkat akun Amazon.

Penindasan set-level konfigurasi, meskipun bukan daftar per se, tetapi mekanisme yang memungkinkan Anda mengonfigurasi penyesuaian penekanan dan penggantian ke daftar penekanan tingkat akun Anda melalui penggunaan set konfigurasi yang dibuat khusus untuk skenario pengiriman email yang berbeda. Misalnya, jika daftar penekanan tingkat akun Anda dikonfigurasi untuk alamat pentalan dan keluhan yang akan ditambahkan, tetapi Anda memiliki demografis email tertentu yang ditentukan dalam kumpulan konfigurasi yang hanya Anda minati dengan alamat keluhan yang ditambahkan - Anda akan mencapainya dengan mengaktifkan penggantian penekanan set konfigurasi ini sehingga alamat email ditambahkan ke daftar penekanan tingkat akun Anda hanya untuk keluhan (bukan pantulan dan keluhan seperti disetel di daftar penekanan tingkat akun) dari

email yang dikirim dengan set konfigurasi ini. Dengan penekanan set-level konfigurasi, ada berbagai tingkat penggantian penekanan tingkat akun Anda, termasuk tidak menggunakan penekanan sama sekali. Lihat Menggunakan penekanan tingkat konfigurasi untuk mengganti daftar penekanan tingkat akun.

Daftar penindasan SES global Amazon

Amazon SES mempertahankan daftar penindasan global internal yang beroperasi dan dikelola di latar belakang olehSES. Ketika setiap SES pelanggan mengirim email yang menghasilkan pantulan keras, SES tambahkan alamat email yang menghasilkan pantulan ke daftar penindasan global. Daftar penindasan global bersifat global dalam arti berlaku untuk semua SES pelanggan. Dengan kata lain, jika pelanggan yang berbeda mencoba mengirim email ke alamat yang ada di daftar penindasan global, SES menerima pesan tersebut, tetapi tidak mengirimkannya, karena alamat email ditekan.

Fitur permintaan penghapusan alamat email daftar penindasan global tidak lagi dihadapi pelanggan dan Anda tidak lagi berinteraksi dengannya untuk mengelola penindasan. Untuk mengganti fungsi ini, Amazon SES sekarang menawarkan cara baru bagi Anda untuk mengelola penindasan Anda dengan menyediakan daftar penekanan tingkat akun dan penggantian penekanan set-level konfigurasi yang menawarkan Anda kontrol yang lebih disesuaikan atas cara Anda menangani penekanan email untuk akun Anda sendiri. Untuk informasi selengkapnya, lihat Menggunakan daftar SES penindasan tingkat akun Amazon dan Menggunakan penekanan tingkat konfigurasi untuk mengganti daftar penekanan tingkat akun.

> **Important**
>
> Formulir permintaan penghapusan alamat email daftar penindasan global tidak lagi ada di SES konsol Amazon karena daftar penekanan tingkat akun telah menggantikannya. Untuk mempelajari cara menggunakan daftar penekanan tingkat akun, lihat. Menggunakan daftar SES penindasan tingkat akun Amazon

Pertimbangan daftar penekanan global

Faktor kunci mengenai daftar penindasan global:

- Daftar penindasan global beroperasi dan dikelola di latar belakang oleh SES - Anda tidak dapat berinteraksi dengannya secara langsung; Namun, Anda dapat menggantinya dengan menggunakan daftar penekanan tingkat akun Anda sendiri.

- Daftar penindasan global diaktifkan secara default untuk semua SES akun. Anda tidak dapat menonaktifkannya.
- Karena SES menerapkan daftar penindasan global ke semua pelanggan, Anda tidak dapat menanyakan daftar penindasan global atau menambahkan alamat ke dalamnya secara manual.
- Ketika alamat email menghasilkan pantulan keras, SES tambahkan alamat ke daftar penindasan global untuk waktu yang singkat. Setelah periode waktu berlalu, SES hapus alamat dari daftar. Jika alamat menghasilkan pantulan keras lain, SES tambahkan kembali ke daftar penindasan global untuk jangka waktu yang lebih lama, dan menghapusnya pada akhir periode itu. Jumlah waktu agar alamat tetap di daftar penekanan global meningkat setiap kali alamat menghasilkan pentalan keras. Alamat dapat tetap berada di daftar penekanan global hingga 14 hari.
- Jika Anda mencoba mengirim pesan ke alamat yang ada di daftar penindasan global, SES terima pesan tersebut, tetapi tidak mengirimkannya. SESmenghasilkan pemberitahuan bouncing dengan `bounceType` nilai`Permanent`, dan `bounceSubType` nilai. `Suppressed` Menerima notifikasi pentalan tipe ini adalah satu-satunya cara untuk mengetahui apakah alamat ada di daftar penekanan global. Anda tidak dapat meng-kueri daftar penekanan global.
- SESmenghitung pesan yang Anda kirim ke alamat pada daftar penindasan global terhadap rasio pentalan untuk akun Anda dan menuju kuota pengiriman harian Anda.
- Seperti halnya alamat email yang menghasilkan pentalan keras, Anda harus menghapus alamat yang menyebabkan daftar penekanan terpental dari daftar surat-menyurat Anda kecuali Anda yakin bahwa alamat tersebut valid.
- Pentalan daftar penekanan dihitung terhadap tingkat pentalan akun Anda. Jika rasio pentalan Anda terlalu tinggi, akun Anda mungkin akan ditinjau atau kemampuan akun Anda untuk mengirim email dapat dijeda.

> **Note**
>
> Penting untuk memahami bagaimana daftar SES penindasan saling terkait dan hierarki mereka, lihat Ikhtisar daftar penekanan dan mekanisme penggantian penekanan.

Menggunakan daftar SES penindasan tingkat akun Amazon

Daftar penindasan SES tingkat akun Amazon diperkenalkan sehingga pelanggan dapat membuat dan mengontrol daftar penindasan mereka sendiri dan mengelola reputasi mereka, dengan demikian, daftar penindasan tingkat akun Anda hanya berlaku untuk akun Anda. Antarmuka daftar penekanan

tingkat akun di SES konsol menyediakan cara mudah untuk mengelola alamat dalam daftar penekanan tingkat akun Anda, termasuk tindakan massal untuk menambah atau menghapus alamat.

Daftar penekanan SES tingkat akun Anda berlaku untuk Anda Akun AWS saat ini. Wilayah AWS Anda dapat menambah atau menghapus, secara individu atau massal, alamat dari daftar penekanan tingkat akun Anda dengan menggunakan SES API v2 atau konsol.

> **Note**
>
> Untuk menambah atau menghapus alamat dalam jumlah besar, Anda harus memiliki akses produksi. Untuk mempelajari selengkapnya tentang sandbox, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).

SESPertimbangan daftar penindasan tingkat Akun Amazon

Anda harus mempertimbangkan faktor-faktor berikut ketika Anda menggunakan daftar penekanan tingkat akun Anda:

- Jika Anda mulai menggunakan Amazon SES setelah 25 November 2019, akun Anda menggunakan daftar penindasan tingkat akun secara default untuk pantulan dan keluhan. Jika Anda mulai menggunakan SES sebelum tanggal ini, maka Anda harus mengaktifkan fitur ini dengan menggunakan `PutAccountSuppressionAttributes` operasi di SESAPI.
- Jika Anda mencoba mengirim pesan ke alamat yang ada di daftar penekanan tingkat akun Anda yang memiliki alasan penekanan yang cocok dengan alasan penekanan yang sama yang dipilih untuk pengaturan penekanan tingkat akun Anda, SES menerima pesan tersebut, tetapi tidak mengirimnya—namun, jika tidak cocok, maka akan mengirimkannya. SES Untuk membantu memperjelas hal ini, contoh-contoh berikut disediakan:
  - Anda telah menetapkan pengaturan penekanan tingkat akun Anda dengan alasan penindasan Bounces saja, tidak SES akan mencoba pengiriman alamat dalam daftar penekanan tingkat akun Anda dengan alasan penindasan sebagai Bounce.
  - Anda telah menetapkan pengaturan penekanan tingkat akun Anda dengan alasan penindasan Bounces dan Keluhan, tidak SES akan mencoba pengiriman alamat dalam daftar penekanan tingkat akun Anda dengan alasan penindasan baik Bounce atau Complaint.
  - Anda telah menetapkan pengaturan penekanan tingkat akun Anda dengan alasan penindasan Bounces saja, SES akan mencoba pengiriman alamat dalam daftar penekanan tingkat akun Anda dengan alasan penekanan Keluhan (karena dalam kasus ini, mereka tidak cocok).

- SESTidak menghitung pesan yang Anda kirim ke alamat pada daftar penindasan tingkat akun Anda terhadap Reputasi. BounceRateatau Reputasi. ComplaintRatemetrik di SES namespace AWS/untuk akun Anda. Pesan tersebut dihitung berdasarkan metrik Bounce atau Complaint di namespace/. AWS SES
- Jika alamat ada di daftar penindasan global, tetapi tidak pada daftar penindasan tingkat akun Anda (yang berarti Anda ingin mengirimkannya), dan Anda mengirimkannya, masih SES akan mencoba pengiriman; Namun, jika memantul, itu masih dihitung terhadap rasio pentalan untuk akun Anda dan menuju kuota pengiriman harian Anda.
- SESmenghitung pesan yang Anda kirim ke alamat pada daftar penindasan tingkat akun Anda terhadap kuota pengiriman harian Anda.
- Alamat email pada daftar penindasan tingkat akun Anda tetap ada sampai Anda menghapusnya.
- Jika kemampuan akun Anda untuk mengirim email dijeda, SES secara otomatis menghapus alamat dalam daftar penindasan tingkat akun Anda setelah 90 hari. Jika kemampuan akun Anda untuk mengirim email dipulihkan sebelum periode 90 hari ini berakhir, maka alamat dalam daftar tidak dihapus.
- Gmail tidak memberikan data keluhanSES. Jika penerima menggunakan tombol Spam di klien web Gmail untuk melaporkan pesan yang mereka terima dari Anda sebagai spam, mereka tidak ditambahkan ke daftar penindasan tingkat akun Anda.
- Anda dapat mengaktifkan daftar penekanan tingkat akun jika akun Anda ada di kotak pasir. SES Namun, Anda tidak dapat menggunakan CreateImportJoboperasi PutSuppressedDestinationatau sampai akun Anda dihapus dari kotak pasir. Untuk mempelajari selengkapnya tentang sandbox, lihat Minta akses produksi (Pindah dari SES kotak pasir Amazon).
- Hanya pantulan keras yang ditambahkan ke daftar penekanan tingkat akun Anda. Untuk informasi tentang perbedaan antara pantulan lunak dan keras, lihat. the section called “Setelah Amazon SES mengirim email”
- Saat Anda menggunakan daftar penekanan tingkat akun Anda, SES tambahkan alamat yang menghasilkan pantulan keras ke daftar penindasan global juga.

Mengaktifkan daftar penindasan SES tingkat akun Amazon

Anda dapat menggunakan PutAccountSuppressionAttributesoperasi di Amazon SES API v2 untuk mengaktifkan dan mengatur daftar penekanan tingkat akun Anda. Anda dapat mengonfigurasi pengaturan ini dengan cepat dan mudah menggunakan AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan Pengguna AWS Command Line Interface.

Untuk mengonfigurasi daftar penekanan tingkat akun Anda menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

  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
  ```

  Untuk mengaktifkan daftar penekanan tingkat akun Anda, Anda harus menentukan setidaknya satu alasan untuk parameter tersebut. `suppressed-reasons` Anda dapat menentukan `BOUNCE` atau `COMPLAINT`, atau Anda dapat menentukan keduanya, seperti yang ditunjukkan dalam contoh sebelumnya.

Untuk mengonfigurasi daftar penekanan tingkat akun Anda menggunakan konsol: SES

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Di panel Pengaturan tingkat akun, pilih Edit.
4. Dalam daftar Suppression, centang kotak Diaktifkan.
5. Dalam alasan Penindasan, pilih salah satu alasan mengapa alamat email penerima harus ditambahkan secara otomatis ke daftar penindasan tingkat akun Anda.
6. Pilih Simpan perubahan.

Mengaktifkan daftar penekanan SES tingkat akun Amazon untuk set konfigurasi

Anda juga dapat mengonfigurasi penekanan SES tingkat akun Amazon sehingga hanya berlaku untuk set konfigurasi tertentu. Ketika Anda melakukannya, alamat hanya ditambahkan ke daftar

penekanan jika Anda menentukan set konfigurasi yang ditetapkan ketika Anda mengirim email yang menyebabkan peristiwa pentalan atau aduan.

> **Note**
>
> Prosedur berikut menganggap bahwa Anda telah menginstal AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Untuk mengonfigurasi daftar penekanan tingkat akun Anda untuk set konfigurasi menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

  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
  ```

  Pada contoh sebelumnya, ganti *configSet* dengan nama set konfigurasi yang harus menggunakan daftar penekanan tingkat akun Anda.

Untuk mengonfigurasi daftar penekanan tingkat akun Anda untuk set konfigurasi menggunakan konsol: SES

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Set konfigurasi.

3. Di set Konfigurasi, pilih nama set konfigurasi yang ingin Anda konfigurasikan dengan penekanan khusus.
4. Di panel Opsi daftar Supresi, pilih Edit.
5. Bagian Opsi daftar Supresi menyediakan set keputusan untuk menentukan penekanan khusus yang dimulai dengan opsi untuk menggunakan set konfigurasi ini untuk mengganti penekanan tingkat akun Anda. Peta logika penekanan set-level konfigurasi akan membantu Anda memahami efek dari kombinasi override. Pilihan penggantian multitier ini dapat digabungkan untuk menerapkan tiga tingkat penekanan yang berbeda:

   a. Gunakan penekanan tingkat akun: Jangan mengesampingkan penekanan tingkat akun Anda dan jangan menerapkan penekanan set-level konfigurasi apa pun - pada dasarnya, email apa pun yang dikirim menggunakan set konfigurasi ini hanya akan menggunakan penekanan tingkat akun Anda. Untuk melakukannya:

      - Di Pengaturan daftar Supresi, hapus centang pada kotak Ganti pengaturan tingkat akun.

   b. Jangan gunakan penekanan apa pun: Ganti penekanan tingkat akun Anda tanpa mengaktifkan penekanan set-level konfigurasi apa pun - ini berarti email apa pun yang dikirim menggunakan set konfigurasi ini tidak akan menggunakan penekanan tingkat akun Anda; dengan kata lain, semua penekanan dibatalkan. Untuk melakukannya:

      i. Di Pengaturan daftar Supresi, centang kotak Ganti pengaturan tingkat akun.

      ii. Dalam daftar Suppression, hapus centang pada kotak Diaktifkan.

   c. Gunakan penekanan set-level konfigurasi: Ganti penekanan tingkat akun Anda dengan pengaturan daftar penekanan khusus yang ditentukan dalam set konfigurasi ini - ini berarti email apa pun yang dikirim menggunakan set konfigurasi ini hanya akan menggunakan pengaturan penindasan sendiri dan mengabaikan pengaturan penekanan tingkat akun apa pun. Untuk melakukannya:

      i. Di Pengaturan daftar Supresi, centang kotak Ganti pengaturan tingkat akun.

      ii. Dalam daftar Suppression, centang Diaktifkan.

      iii. Di Tentukan alasannya... , pilih salah satu alasan penindasan untuk konfigurasi ini yang akan digunakan.

6. Pilih Simpan perubahan.

Menambahkan alamat email individual ke daftar SES penindasan tingkat akun Amazon

Anda dapat menambahkan alamat individual ke daftar penekanan SES tingkat akun Amazon Anda dengan menggunakan PutSuppressedDestinationoperasi di v2. SES API Tidak ada batasan jumlah alamat yang dapat Anda tambahkan ke daftar penindasan tingkat akun Anda.

> **Note**
>
> Prosedur berikut menganggap Anda telah memasang AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Untuk menambahkan alamat individual ke daftar penekanan tingkat akun Anda menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

  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
  ```

  Pada contoh sebelumnya, ganti *recipient@example.com* dengan alamat email yang ingin Anda tambahkan ke daftar penekanan tingkat akun Anda, dan *BOUNCE* dengan alasan Anda menambahkan alamat ke daftar penekanan (nilai yang dapat diterima adalah `BOUNCE` dan`COMPLAINT`).

Untuk menambahkan alamat individual ke daftar penekanan tingkat akun Anda menggunakan konsol: SES

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Di panel daftar Supresi, pilih Tambahkan alamat email.
4. Ketik alamat email di bidang Alamat email diikuti dengan memilih alasan di Alasan Penindasan - jika Anda perlu memasukkan lebih banyak alamat, pilih Masukkan alamat lain dan ulangi untuk setiap alamat tambahan.
5. Setelah selesai memasukkan alamat, tinjau entri Anda untuk akurasi. Jika Anda memutuskan salah satu entri Anda tidak boleh menjadi bagian dari kiriman ini, pilih tombol Hapus.
6. Pilih Simpan perubahan untuk menambahkan alamat email yang dimasukkan ke daftar penekanan tingkat akun Anda.

Menambahkan alamat email secara massal ke daftar SES penindasan tingkat akun Amazon Anda

Anda dapat menambahkan alamat secara massal dengan terlebih dahulu mengunggah daftar kontak Anda ke objek Amazon S3 diikuti dengan menggunakan operasi CreateImportJobdi Amazon SES API v2.

> **Note**
>
> - Tidak ada batasan jumlah alamat yang dapat Anda tambahkan ke daftar penekanan tingkat akun Anda, tetapi ada batas penambahan massal 100.000 alamat dalam objek Amazon S3 per panggilan. API
> - Jika sumber data Anda adalah bucket S3, itu harus ada di wilayah yang sama dengan yang Anda impor.

Untuk menambahkan alamat email dalam jumlah besar ke daftar penindasan tingkat akun, selesaikan langkah-langkah berikut.

- Unggah daftar alamat Anda ke objek Amazon S3 dalam salah satu CSV atau JSON format.

CSVcontoh format untuk menambahkan alamat:

recipient1@example.com,*BOUNCE*

recipient2@example.com,*COMPLAINT*

Hanya file yang dibatasi baris baru yang JSON didukung. Dalam format ini, setiap baris adalah JSON objek lengkap yang berisi definisi alamat individual.

JSONcontoh format untuk menambahkan alamat:

{"emailAddress":"*recipient1@example.com*","reason":"*BOUNCE*"}

{"emailAddress":"*recipient2@example.com*","reason":"*COMPLAINT*"}

Dalam contoh sebelumnya, ganti *recipient1@example.com* and *recipient2@example.com* dengan alamat email yang ingin Anda tambahkan ke daftar penekanan tingkat akun Anda. Alasan yang dapat diterima bahwa Anda menambahkan alamat ke daftar penekanan adalah *BOUNCE* dan *COMPLAINT*.

- Berikan SES izin untuk membaca objek Amazon S3.

  Saat diterapkan ke bucket Amazon S3, kebijakan berikut memberikan SES izin untuk membaca bucket tersebut. Untuk informasi selengkapnya tentang melampirkan kebijakan ke bucket Amazon S3, lihat Menggunakan Kebijakan Bucket dan Kebijakan Pengguna di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSESGet",
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::BUCKET-NAME/OBJECT-NAME",
            "Condition": {
                "StringEquals": {
                    "aws:Referer": "AWSACCOUNTID"
                }
```

```
            }
        }
    ]
}
```

- Berikan SES izin untuk menggunakan AWS KMS kunci Anda.

  Jika objek Amazon S3 dienkripsi dengan AWS KMS kunci, Anda harus memberikan SES izin Amazon untuk menggunakan kunci tersebut. AWS KMS SEShanya dapat memperoleh izin dari kunci yang dikelola pelanggan, bukan KMS kunci default. Anda perlu memberikan SES izin untuk menggunakan kunci yang dikelola pelanggan dengan menambahkan pernyataan ke kebijakan kunci.

  Tempelkan pernyataan kebijakan berikut ke dalam kebijakan utama SES untuk mengizinkan penggunaan kunci yang dikelola pelanggan Anda.

```
{
   "Sid": "AllowSESToDecrypt",
   "Effect": "Allow",
   "Principal": {
       "Service":"ses.amazonaws.com"
   },
   "Action": [
       "kms:Decrypt",
   ],
   "Resource": "*"
}
```

- Gunakan CreateImportJoboperasi di SES API v2.

> **Note**
>
> Contoh berikut mengasumsikan bahwa Anda telah menginstal file. AWS CLI Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Di baris perintah, masukkan perintah berikut. Ganti *s3bucket* dengan nama ember Amazon S3 dan *s3object* dengan nama objek Amazon S3.

```
aws sesv2 create-import-job --import-destination
 SuppressionListDestination={SuppressionListImportAction=PUT} --import-data-source
 S3Url=s3://s3bucket/s3object,DataFormat=CSV
```

Untuk menambahkan alamat email secara massal ke daftar penekanan tingkat akun Anda menggunakan konsol: SES

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Dalam tabel daftar Suppression, perluas tombol Tindakan massal dan pilih Tambahkan alamat email secara massal.
4. Dalam spesifikasi tindakan Massal, pilih salah satu (a) Pilih file dari ember S3 atau (b) Impor dari file - prosedur diberikan untuk setiap metode impor:

    a. Pilih file dari bucket S3 - jika file sumber Anda sudah disimpan di bucket Amazon S3:

        i. Jika Anda mengetahui bucket Amazon S3 yang ingin Anda gunakan, masukkan di bidang Amazon URI S3; jika tidak, pilih Jelajahi S3: URI

            A. Di Bucket, pilih nama bucket S3.

            B. Di Objects, pilih nama file lalu pilih Pilih - Anda akan dikembalikan ke spesifikasi tindakan Massal.

            C. (Opsional) Jika Anda ingin dibawa ke konsol Amazon S3 untuk melihat detail tentang objek S3 Anda pilih Lihat.

        ii. Dalam format File, pilih format file yang Anda pilih untuk diimpor dari bucket Amazon S3 Anda.

        iii. Pilih Tambahkan alamat email untuk memulai impor alamat dari file Anda - tabel di bawah tab Tindakan massal ditampilkan.

    b. Impor dari file - jika Anda memiliki file sumber lokal untuk diunggah ke bucket Amazon S3 baru atau yang sudah ada:

        i. Di Impor file sumber, pilih Pilih file.

        ii. Pilih CSV file JSON atau di browser file dan pilih Buka - Anda akan melihat nama, ukuran, dan tanggal file Anda ditampilkan di bawah tombol Pilih file.

        iii. Perluas bucket Amazon S3 dan pilih bucket S3.

- Untuk mengunggah file ke bucket baru, pilih Buat bucket S3, masukkan nama di bidang Nama Bucket, dan pilih Buat bucket.

iv. Pilih Tambahkan alamat email untuk memulai impor alamat dari file Anda - tabel di bawah tab Tindakan massal ditampilkan.

5. Terlepas dari metode impor yang Anda gunakan, ID pekerjaan Anda akan dicantumkan dalam tindakan Massal bersama dengan jenis impor, status, dan tanggal - untuk melihat detail pekerjaan, pilih ID pekerjaan.
6. Pilih tab daftar Suppression dan semua alamat email yang berhasil diimpor ditampilkan dengan alasan penindasan dan tanggal ditambahkan - opsi berikut tersedia:

   a. Pilih alamat email, atau pilih kotak centang yang sesuai dan pilih Lihat laporan untuk melihat detailnya. (Jika itu adalah alamat yang secara otomatis ditambahkan ke daftar penindasan Anda karena pantulan atau keluhan, informasi akan ditampilkan tentang peristiwa umpan balik yang menyebabkannya ditambahkan, termasuk rincian tentang pesan email yang menghasilkan peristiwa pemicu.)

   b. Pilih kotak centang yang sesuai dari satu atau beberapa alamat email yang ingin Anda hapus dari daftar penindasan akun dan pilih Hapus.

Melihat daftar alamat yang ada di daftar SES penindasan tingkat akun Amazon Anda

Anda dapat melihat daftar semua alamat email yang ada di daftar penekanan tingkat akun untuk akun Anda dengan menggunakan ListSuppressedDestinationsoperasi di v2. SES API

> **Note**
>
> Prosedur berikut menganggap Anda telah memasang AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Untuk melihat daftar semua alamat email yang ada di daftar penindasan tingkat akun Anda

- Di baris perintah, masukkan perintah berikut:

```
aws sesv2 list-suppressed-destinations
```

Perintah sebelumnya mengembalikan semua alamat email yang ada di daftar penekanan tingkat akun Anda untuk akun Anda. Output menyerupai contoh berikut:

```
{
    "SuppressedDestinationSummaries": [
        {
            "EmailAddress": "recipient2@example.com",
            "Reason": "COMPLAINT",
            "LastUpdateTime": "2020-04-10T21:03:05Z"
        },
        {
            "EmailAddress": "recipient0@example.com",
            "Reason": "COMPLAINT",
            "LastUpdateTime": "2020-04-10T21:04:26Z"
        },
        {
            "EmailAddress": "recipient1@example.com",
            "Reason": "BOUNCE",
            "LastUpdateTime": "2020-04-10T22:07:59Z"
        }
    ]
}
```

- Catatan - Jika output Anda menyertakan bidang NextToken "" dengan nilai string, ini menunjukkan ada alamat email tambahan pada daftar penindasan untuk akun Anda. Untuk melihat alamat tambahan yang ditekan, keluarkan permintaan lain ke`ListSuppressedDestinations`, dan teruskan nilai string yang dikembalikan dalam `--next-token` parameter seperti ini:

  ```
  aws sesv2 list-suppressed-destinations --next-token string
  ```

  Pada perintah sebelumnya, ganti *`string`* dengan NextToken nilai yang dikembalikan.

  Untuk informasi selengkapnya, lihat Cara mencantumkan lebih dari 1000 alamat email dari daftar penekanan tingkat akun.

Anda dapat menggunakan opsi `StartDate` untuk hanya menampilkan alamat email yang ditambahkan ke daftar setelah tanggal tertentu.

Untuk melihat daftar alamat yang ditambahkan ke daftar penekanan tingkat akun Anda setelah tanggal tertentu

- Di baris perintah, masukkan perintah berikut:

  ```
  aws sesv2 list-suppressed-destinations --start-date 1604394130
  ```

  Pada perintah sebelumnya, ganti *1604394130* dengan stempel waktu Unix dari tanggal mulai.

Anda juga dapat menggunakan opsi `EndDate` untuk hanya menampilkan alamat email yang ditambahkan ke daftar sebelum tanggal tertentu.

Untuk melihat daftar alamat yang ditambahkan ke daftar penindasan tingkat akun Anda sebelum tanggal tertentu

- Di baris perintah, masukkan perintah berikut:

  ```
  aws sesv2 list-suppressed-destinations --end-date 1611126000
  ```

  Pada perintah sebelumnya, ganti *1611126000* dengan stempel waktu Unix dari tanggal akhir.

Di baris perintah Linux, macOS, atau Unix, Anda juga dapat menggunakan utilitas `grep` bawaan untuk mencari alamat atau domain tertentu.

Untuk mencari daftar penindasan tingkat akun Anda untuk alamat tertentu

- Di baris perintah, masukkan perintah berikut:

  ```
  aws sesv2 list-suppressed-destinations | grep -A2 'example.com'
  ```

  Pada perintah sebelumnya, ganti *example.com* dengan string teks (seperti alamat atau domain) yang ingin Anda cari.

Untuk melihat daftar semua alamat email yang ada di daftar penekanan tingkat akun Anda menggunakan konsol: SES

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Di panel daftar Supresi, semua alamat email pada daftar penekanan tingkat akun Anda ditampilkan dengan alasan dan tanggal penindasan yang ditambahkan - opsi berikut tersedia:

    a. Pilih alamat email, atau pilih kotak centang yang sesuai dan pilih Lihat laporan untuk melihat detailnya. (Jika itu adalah alamat yang secara otomatis ditambahkan ke daftar penindasan Anda karena pantulan atau keluhan, informasi akan ditampilkan tentang peristiwa umpan balik yang menyebabkannya ditambahkan, termasuk rincian tentang pesan email yang menghasilkan peristiwa pemicu.)

    b. Anda dapat menyesuaikan tabel daftar penekanan dengan memilih ikon roda gigi - modal akan disajikan di mana Anda dapat menyesuaikan ukuran halaman, bungkus garis, dan kolom untuk dilihat - setelah membuat pilihan, pilih Konfirmasi. Tabel daftar penindasan akan mencerminkan pilihan tampilan Anda.

Menghapus alamat email individual dari daftar SES penindasan tingkat akun Amazon Anda

Jika alamat ada di daftar penindasan untuk akun Anda, tetapi Anda tahu bahwa alamat tersebut tidak boleh ada dalam daftar, Anda dapat menghapusnya dengan menggunakan DeleteSuppressedDestinationoperasi di SES API v2.

> **Note**
>
> Prosedur berikut menganggap Anda telah memasang AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Untuk menghapus alamat individual dari daftar penekanan tingkat akun Anda menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

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
```

Pada contoh sebelumnya, ganti *recipient@example.com* dengan alamat email yang ingin Anda hapus dari daftar penekanan tingkat akun Anda.

Untuk menghapus alamat individual dari daftar penekanan tingkat akun Anda menggunakan konsol: SES

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Hapus alamat email individual baik dengan (a) pemilihan tabel atau (b) entri yang diketik:
   a. Pilih dari tabel: Dalam tabel daftar Suppression, pilih kotak centang yang sesuai dari satu atau beberapa alamat email dan pilih Hapus.
   b. Ketik di bidang:
      i. Dalam tabel daftar Supresi, pilih Hapus alamat email.
      ii. Ketik alamat email di bidang Alamat email - jika Anda perlu memasukkan lebih banyak alamat, pilih Masukkan alamat lain dan ulangi untuk setiap alamat tambahan.
      iii. Setelah selesai memasukkan alamat, tinjau entri Anda untuk akurasi. Jika Anda memutuskan salah satu entri Anda tidak boleh menjadi bagian dari kiriman ini, pilih tombol Hapus.
      iv. Pilih Simpan perubahan untuk menghapus alamat email yang dimasukkan dari daftar penekanan tingkat akun Anda.

Menghapus alamat email secara massal dari daftar SES penindasan tingkat akun Amazon Anda

Anda dapat menghapus alamat secara massal dengan terlebih dahulu mengunggah daftar kontak Anda ke objek Amazon S3 diikuti dengan menggunakan operasi CreateImportJobdi SES API v2.

> **Note**
>
> - Tidak ada batasan jumlah alamat yang dapat Anda hapus dari daftar penekanan tingkat akun, tetapi ada batas penghapusan massal 10.000 alamat di objek Amazon S3 per panggilan. API
> - Jika sumber data Anda adalah bucket S3, itu harus ada di wilayah yang sama dengan yang Anda impor.

Untuk menghapus alamat email dalam jumlah besar dari daftar penekanan tingkat akun, selesaikan langkah-langkah berikut ini.

- Unggah daftar alamat Anda ke objek Amazon S3 dalam salah satu CSV atau JSON format.

  CSVcontoh format untuk menghapus alamat:

  recipient3@example.com

  Hanya file yang dibatasi baris baru yang JSON didukung. Dalam format ini, setiap baris adalah JSON objek lengkap yang berisi definisi alamat individual.

  JSONcontoh format untuk menambahkan alamat:

  `{"emailAddress":"`*recipient3@example.com*`"}`

  Dalam contoh sebelumnya, ganti *recipient3@example.com* dengan alamat email yang ingin Anda hapus dari daftar penekanan tingkat akun Anda.
- Berikan SES izin untuk membaca objek Amazon S3.

  Saat diterapkan ke bucket Amazon S3, kebijakan berikut memberikan SES izin untuk membaca bucket tersebut. Untuk informasi selengkapnya tentang melampirkan kebijakan ke bucket Amazon S3, lihat Menggunakan Kebijakan Bucket dan Kebijakan Pengguna di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSESGet",
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::BUCKET-NAME/OBJECT-NAME",
            "Condition": {
                "StringEquals": {
                    "aws:Referer": "AWSACCOUNTID"
                }
            }
        }
    ]
}
```

- Berikan SES izin untuk menggunakan AWS KMS kunci Anda.

  Jika objek Amazon S3 dienkripsi dengan AWS KMS kunci, Anda harus memberikan SES izin Amazon untuk menggunakan kunci tersebut. AWS KMS SEShanya dapat memperoleh izin dari kunci yang dikelola pelanggan, bukan KMS kunci default. Anda perlu memberikan SES izin untuk menggunakan kunci yang dikelola pelanggan dengan menambahkan pernyataan ke kebijakan kunci.

  Tempelkan pernyataan kebijakan berikut ke dalam kebijakan utama SES untuk mengizinkan penggunaan kunci yang dikelola pelanggan Anda.

```
{
   "Sid": "AllowSESToDecrypt",
   "Effect": "Allow",
   "Principal": {
       "Service":"ses.amazonaws.com"
   },
   "Action": [
       "kms:Decrypt",
   ],
   "Resource": "*"
```

```
}
```

- Gunakan CreateImportJoboperasi di SES API v2.

> **Note**
>
> Contoh berikut mengasumsikan bahwa Anda telah menginstal file. AWS CLI Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Di baris perintah, masukkan perintah berikut. Ganti *s3bucket* dengan nama ember Amazon S3 dan *s3object* dengan nama objek Amazon S3.

```
aws sesv2 create-import-job --import-destination
 SuppressionListDestination={SuppressionListImportAction=DELETE} --import-data-source
 S3Url="s3://s3bucket/s3object",DataFormat=CSV
```

Untuk menghapus alamat email secara massal dari daftar penekanan tingkat akun Anda menggunakan konsol: SES

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Dalam tabel daftar Suppression, perluas tombol Tindakan massal dan pilih Hapus alamat email secara massal.
4. Dalam spesifikasi tindakan Massal, pilih salah satu (a) Pilih file dari ember S3 atau (b) Impor dari file - prosedur diberikan untuk setiap metode impor:

    a. Pilih file dari bucket S3 - jika file sumber Anda sudah disimpan di bucket Amazon S3:

        i. Jika Anda mengetahui bucket Amazon S3 yang ingin Anda gunakan, masukkan di bidang Amazon URI S3; jika tidak, pilih Jelajahi S3: URI

            A. Di Bucket, pilih nama bucket S3.

            B. Di Objects, pilih nama file lalu pilih Pilih - Anda akan dikembalikan ke spesifikasi tindakan Massal.

C. (Opsional) Jika Anda ingin dibawa ke konsol Amazon S3 untuk melihat detail tentang objek S3 Anda pilih Lihat.

ii. Dalam format File, pilih format file yang Anda pilih untuk diimpor dari bucket Amazon S3 Anda.

iii. Pilih Hapus alamat email untuk memulai impor alamat dari file Anda - tabel di bawah tab Tindakan massal ditampilkan.

b. Impor dari file - jika Anda memiliki file sumber lokal untuk diunggah ke bucket Amazon S3 baru atau yang sudah ada:

i. Di Impor file sumber, pilih Pilih file.

ii. Pilih CSV file JSON atau di browser file dan pilih Buka - Anda akan melihat nama, ukuran, dan tanggal file Anda ditampilkan di bawah tombol Pilih file.

iii. Perluas bucket Amazon S3 dan pilih bucket S3.

- Untuk mengunggah file ke bucket baru, pilih Buat bucket S3, masukkan nama di bidang Nama Bucket, dan pilih Buat bucket.

iv. Pilih Hapus alamat email untuk memulai impor alamat dari file Anda - tabel di bawah tab Tindakan massal ditampilkan.

5. Terlepas dari metode impor yang Anda gunakan, ID pekerjaan Anda akan dicantumkan dalam tindakan Massal bersama dengan jenis impor, status, dan tanggal - untuk melihat detail pekerjaan, pilih ID pekerjaan.
6. Pilih tab Daftar Supresi dan semua alamat email yang berhasil diimpor yang telah dihapus dari daftar penindasan Anda tidak akan lagi ditampilkan.

Melihat daftar tugas impor untuk akun

Anda dapat melihat daftar semua alamat email yang ada di daftar penindasan tingkat akun untuk akun Anda dengan menggunakan ListImportJobsoperasi di Amazon v2. SES API

> **Note**
>
> Prosedur berikut menganggap Anda telah memasang AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Untuk melihat daftar semua tugas impor untuk akun

- Di baris perintah, masukkan perintah berikut:

```
aws sesv2 list-import-jobs
```

Perintah sebelumnya mengembalikan semua tugas impor untuk akun. Output menyerupai contoh berikut:

```
{
    "ImportJobs": [
        {
            "CreatedTimestamp": 2020-07-31T06:06:55Z",
            "ImportDestination": {
                "SuppressionListDestination": {
                    "SuppressionListImportAction": "PUT"
                }
            },
            "JobStatus": "COMPLETED",
            "JobId": "755380d7-fbdb-4ed2-a9a3-06866220f5b5"
        },
        {
            "CreatedTimestamp": "2020-07-30T18:45:32Z",
            "ImportDestination": {
                "SuppressionListDestination": {
                    "SuppressionListImportAction": "DELETE"
                }
            },
            "JobStatus": "COMPLETED",
            "JobId": "076683bd-a7ee-4a40-9754-4ad1161ba8b6"
        },
        {
            "CreatedTimestamp": "2020-08-05T16:45:18Z",
            "ImportDestination": {
                "SuppressionListDestination": {
                    "SuppressionListImportAction": "PUT"
                }
            },
            "JobStatus": "COMPLETED",
            "JobId": "6e261869-bd30-4b33-b1f2-9e035a83a395"
        }
    ]
```

```
}
```

Untuk melihat daftar semua pekerjaan impor untuk akun menggunakan SES konsol:

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Di panel daftar Suppression, pilih tab Tindakan massal.
4. Semua pekerjaan impor akan tercantum dalam tabel Tindakan massal bersama dengan jenis impor, status, dan tanggal.
5. Untuk melihat detail pekerjaan, pilih ID pekerjaan dan panel berikut akan ditampilkan:

    a. Status tindakan massal: menunjukkan status keseluruhan pekerjaan, waktu dan tanggal penyelesaiannya, berapa banyak catatan yang diimpor, dan jumlah catatan apa pun yang gagal diimpor dengan sukses.

    b. Detail tindakan massal: menampilkan ID pekerjaan, apakah itu digunakan untuk menambah atau menghapus alamat, apakah format file itu JSON atauCSV, bucket Amazon S3 tempat file massal disimpan, dan waktu serta tanggal tindakan massal dibuat. URI

Mendapatkan informasi tentang tugas impor untuk akun

Anda bisa mendapatkan informasi tentang pekerjaan impor untuk akun dengan menggunakan GetImportJoboperasi di Amazon SES API v2.

> **Note**
>
> Prosedur berikut menganggap Anda telah memasang AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Mendapatkan informasi tentang tugas impor untuk akun

- Di baris perintah, masukkan perintah berikut:

```
aws sesv2 get-import-job --job-id JobId
```

Perintah sebelumnya mengembalikan informasi tentang tugas impor untuk akun. Output menyerupai contoh berikut:

```
{
    "ImportDataSource": {
        "S3Url": "s3://bucket/object",
        "DataFormat": "CSV"
    },
    "ProcessedRecordsCount": 2,
    "FailureInfo": {
        "FailedRecordsS3Url": "s3presignedurl"
    },
    "JobStatus": "COMPLETED",
    "JobId": "jobid",
    "CreatedTimestamp": "2020-08-12T17:05:15Z",
    "FailedRecordsCount": 1,
    "ImportDestination": {
        "SuppressionListDestination": {
            "SuppressionListImportAction": "PUT"
        }
    },
    "CompletedTimestamp": "2020-08-12T17:06:42Z"
}
```

Untuk mendapatkan informasi tentang pekerjaan impor untuk akun menggunakan SES konsol:

1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Di panel daftar Suppression, pilih tab Tindakan massal.
4. Semua pekerjaan impor akan tercantum dalam tabel Tindakan massal bersama dengan jenis impor, status, dan tanggal.
5. Untuk melihat detail pekerjaan, pilih ID pekerjaan dan panel berikut akan ditampilkan:

    a. Status tindakan massal: menunjukkan status keseluruhan pekerjaan, waktu dan tanggal penyelesaiannya, berapa banyak catatan yang diimpor, dan jumlah catatan apa pun yang gagal diimpor dengan sukses.

b. Detail tindakan massal: menampilkan ID pekerjaan, apakah itu digunakan untuk menambah atau menghapus alamat, apakah format file itu JSON atauCSV, bucket Amazon S3 tempat file massal disimpan, dan waktu serta tanggal tindakan massal dibuat. URI

Menonaktifkan daftar penindasan tingkat SES akun Amazon

Anda dapat menggunakan PutAccountSuppressionAttributesoperasi di SES API v2 untuk secara efektif menonaktifkan daftar penekanan tingkat akun Anda dengan menghapus nilai dari atribut. `suppressed-reasons`

> **Note**
>
> Prosedur berikut menganggap Anda telah memasang AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan AWS Command Line Interface Pengguna.

Untuk menonaktifkan daftar penekanan tingkat akun Anda menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

```
aws sesv2 put-account-suppression-attributes --suppressed-reasons
```

Untuk menonaktifkan daftar penekanan tingkat akun Anda menggunakan konsol: SES

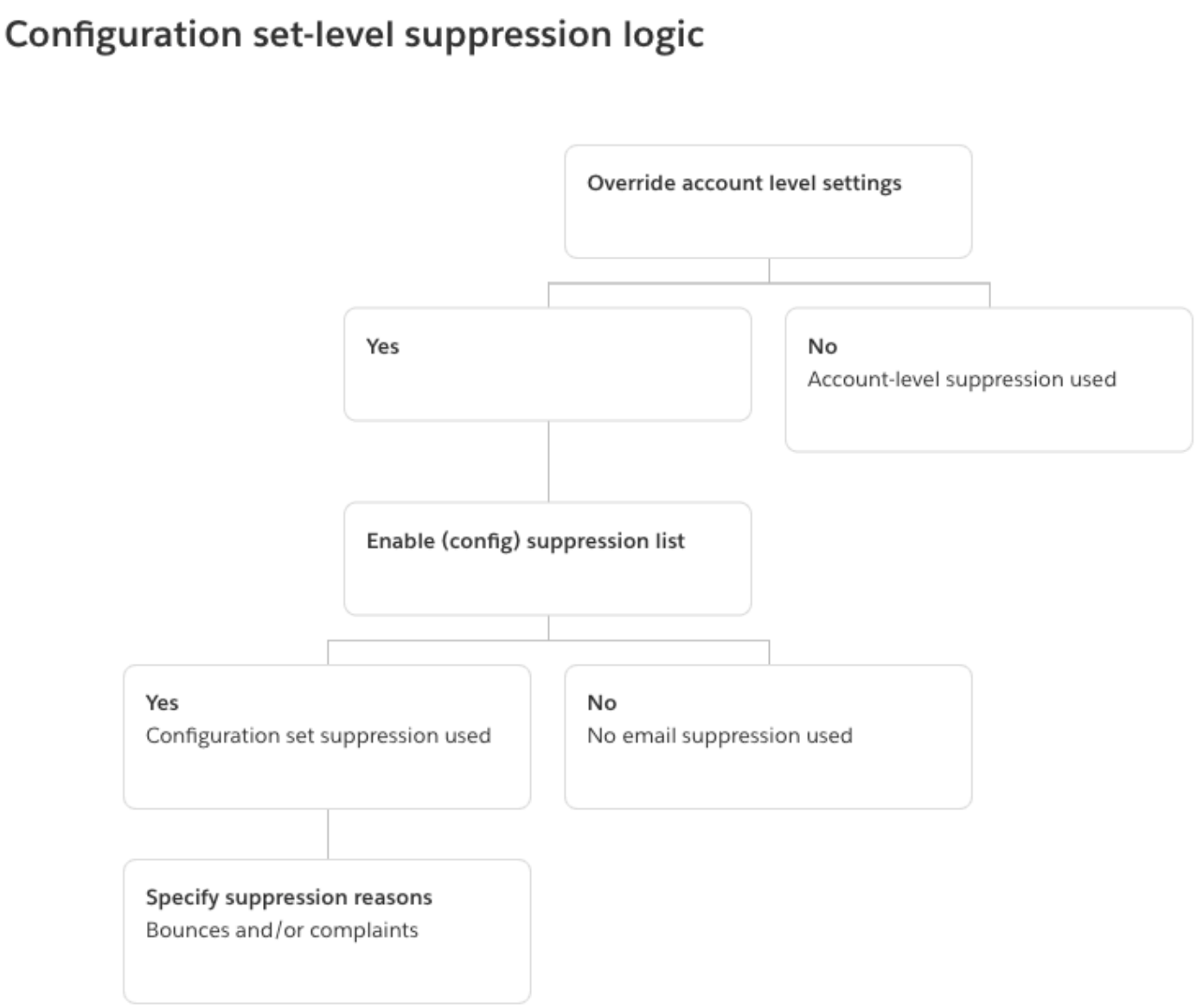
1. Masuk ke AWS Management Console dan buka SES konsol Amazon di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih daftar Suppression.
3. Di panel Pengaturan tingkat akun, pilih Edit.
4. Dalam daftar Suppression, hapus centang pada kotak Diaktifkan.
5. Pilih Simpan perubahan.

Menggunakan penekanan tingkat konfigurasi untuk mengganti daftar penekanan tingkat akun

Sementara daftar penindasan tingkat akun diatur untuk seluruh akun Anda, Anda dapat menyesuaikannya secara terpisah untuk set konfigurasi yang berbeda dengan mengesampingkannya dengan penindasan set-level konfigurasi. Granularitas yang lebih halus ini memungkinkan Anda untuk menggunakan pengaturan penekanan khusus untuk grup pengiriman email yang berbeda yang telah Anda tetapkan ke set konfigurasi mereka sendiri. Misalnya, misalkan daftar penindasan tingkat akun Anda dikonfigurasi untuk alamat bouncing dan keluhan yang akan ditambahkan, tetapi Anda memiliki demografis email tertentu yang didefinisikan dalam rangkaian konfigurasi yang Anda hanya tertarik dengan alamat keluhan yang ditambahkan - Anda akan mencapainya dengan mengaktifkan ini penindasan set konfigurasi menimpa sehingga alamat email ditambahkan ke daftar penindasan tingkat akun Anda hanya untuk keluhan (tidak memantul dan keluhan seperti diatur dalam daftar penindasan tingkat akun Anda) dari email yang dikirim dengan set konfigurasi ini.

Dengan penindasan set-level konfigurasi, ada tingkat yang berbeda untuk mengesampingkan penindasan tingkat akun Anda, termasuk tidak menggunakan penekanan sama sekali. Untuk membantu memahami berbagai tingkat penindasan yang dapat diatur dalam prosedur konsol berikut, berikut peta hubungan model keputusan set pilihan Anda dapat membuat untuk mengaktifkan atau menonaktifkan berbagai tingkat menimpa, yang tergantung pada kombinasi mereka, dapat digunakan untuk menerapkan tiga tingkat penekanan yang berbeda:

- Tidak ada menimpa (default)— Set konfigurasi menggunakan pengaturan daftar penekanan tingkat akun Anda.
- Menimpa pengaturan tingkat akun— ini akan meniadakan pengaturan daftar penindasan tingkat akun; email yang dikirim dengan set konfigurasi ini tidak akan menggunakan pengaturan penindasan sama sekali.
- Mengganti pengaturan tingkat akun dengan penindasan set-level konfigurasi diaktifkan— email yang dikirim dengan set konfigurasi ini hanya akan menggunakan kondisi penindasan yang Anda aktifkan untuk itu (memantul, keluhan, atau memantul dan keluhan) - terlepas dari apa pengaturan daftar penindasan tingkat akun Anda, itu akan menimpa mereka.



Perlu diingat bahwa penindasan set-level konfigurasi bukanlah penindasan yang sebenarnyadaftar, melainkan, ini hanya mekanisme untuk mengganti daftar penindasan tingkat akun Anda dengan pengaturan penekanan khusus yang didefinisikan dalam set konfigurasi - ini berarti setiap email yang dikirim menggunakan set konfigurasi hanya akan menggunakan pengaturan penindasan sendiri dan mengabaikan pengaturan penindasan tingkat akun apa pun. Dengan kata lain, penindasan set-level konfigurasi berinteraksi dengan daftar penindasan tingkat akun Anda hanya dengan mengubah (mengesampingkan) alasan penindasan yang menentukan alamat email apa yang ditambahkan ke daftar penindasan tingkat akun Anda.

Mengaktifkan penekanan tingkat konfigurasi

Untuk mengaktifkan penindasan set-level konfigurasi menggunakan konsol baru Amazon SES:

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, di bawah Konfigurasi, pilih Set konfigurasi.
3. MasukSet konfigurasi, pilih nama konfigurasi yang ingin Anda konfigurasikan dengan penekanan yang disesuaikan.
4. DiPilihan daftar penekananpane, pilihMengedit.
5. ParameterPilihan daftar penekananbagian memberikan keputusan yang ditetapkan untuk menentukan penekanan disesuaikan dimulai dengan opsi untuk menggunakan konfigurasi ini diatur untuk menimpa penindasan tingkat akun Anda. Parameterkonfigurasi set-level penindasan peta logikaakan membantu Anda memahami efek dari kombinasi override. Pilihan multitiered override ini dapat dikombinasikan untuk menerapkan tiga tingkat penekanan yang berbeda:

   a. Gunakan penekanan tingkat akun: Jangan menimpa penindasan tingkat akun Anda dan jangan menerapkan penindasan set-level konfigurasi - pada dasarnya, email yang dikirim menggunakan set konfigurasi ini hanya akan menggunakan penindasan tingkat akun Anda. Untuk melakukannya:

      - MasukPengaturan daftar penekanan, hapus centangMenimpa pengaturan tingkat akunkotak.

   b. Jangan gunakan penindasan apapun: Mengganti penindasan tingkat akun Anda tanpa mengaktifkan penindasan set-level konfigurasi - ini berarti setiap email yang dikirim menggunakan set konfigurasi ini tidak akan menggunakan penindasan tingkat akun Anda; dengan kata lain, semua penindasan dibatalkan. Untuk melakukannya:

      i. MasukPengaturan daftar penekanan, periksaMenimpa pengaturan tingkat akunkotak.

      ii. MasukDaftar penekanan, hapus centangDiaktifkankotak.

   c. Gunakan konfigurasi set-level penindasan: Mengganti daftar penindasan tingkat akun Anda dengan pengaturan penekanan khusus yang didefinisikan dalam set konfigurasi ini - ini berarti setiap email yang dikirim menggunakan set konfigurasi ini hanya akan menggunakan pengaturan penindasan sendiri dan mengabaikan pengaturan penindasan tingkat akun apa pun. Untuk melakukannya:

i. MasukPengaturan daftar penekanan, periksaMenimpa pengaturan tingkat akunkotak.

ii. MasukDaftar penekananpemeriksaanDiaktifkan.

iii. MasukTentukan alasan (s)..., pilih salah satu alasan penindasan untuk konfigurasi ini diatur untuk digunakan.

6. Pilih Simpan perubahan.

Menggunakan pengelolaan daftar

Amazon SES menawarkan kemampuan pengelolaan daftar, yang berarti pelanggan dapat mengelola milis mereka sendiri, yang dikenal sebagai daftar kontak. Daftar kontak adalah daftar yang mengizinkan Anda menyimpan semua kontak yang telah berlangganan topik tertentu atau banyak topik. Kontak adalah pengguna akhir yang menerima email Anda. Topik adalah grup minat, tema, atau label dalam daftar. Daftar dapat memiliki beberapa topik.

Dengan menggunakan `ListContacts`operasi di Amazon SES API v2, Anda dapat mengambil daftar semua kontak Anda yang telah berlangganan topik tertentu, kepada siapa Anda dapat mengirim email menggunakan operasi. `SendEmail`

Untuk informasi tentang pengelolaan langganan, lihat Menggunakan manajemen berlangganan.

Gambaran umum manajemen daftar

Anda harus mempertimbangkan faktor-faktor berikut ketika Anda menggunakan pengelolaan daftar:

- Anda dapat menentukan daftar topik sekaligus membuat daftar.
- Hanya satu daftar kontak yang diizinkan per Akun AWS.
- Daftar dapat memiliki maksimum 20 topik.
- Anda dapat memperbarui daftar kontak yang sudah ada, termasuk menambahkan topik baru ke daftar, menambahkan atau menghapus kontak dari daftar, dan memperbarui preferensi kontak untuk daftar atau topik.
- Anda dapat memperbarui metadata topik, seperti nama tampilan topik atau deskripsi.
- Anda bisa mendapatkan daftar kontak di daftar kontak, kontak yang berlangganan dari topik, kontak yang berhenti berlangganan dari topik, dan kontak yang berhenti berlangganan dari semua topik di daftar.

- Anda dapat mengimpor daftar kontak yang ada ke Amazon SES menggunakan `CreateImportJob`API.
- Amazon SES akan mementalkan email jika dikirim ke kontak yang tidak berlangganan di daftar kontak Anda. Untuk informasi lebih lanjut, lihat Menggunakan manajemen berlangganan.
- Setiap kontak dapat memiliki atribut terkait yang dapat Anda gunakan untuk menyimpan informasi tentang kontak tersebut.

Mengonfigurasi pengelolaan daftar

Anda dapat menggunakan operasi berikut untuk mengonfigurasi kemampuan manajemen daftar. Untuk daftar lengkap kontak dan operasi kontak, lihat Referensi Amazon SES API v2.

Buat daftar kontak

Anda dapat menggunakan `CreateContactList`operasi di Amazon SES API v2 untuk membuat daftar kontak. Anda dapat mengonfigurasi pengaturan ini dengan cepat dan mudah menggunakan AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan Pengguna AWS Command Line Interface.

Untuk membuat daftar kontak dengan menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

  ```
  aws sesv2 create-contact-list --cli-input-json file://CONTACT-LIST-JSON
  ```

  Di perintah sebelumnya, ganti *CONTACT-LIST-JSON dengan jalur ke file JSON* Anda untuk permintaan Anda. `CreateContactList`

  Contoh file JSON `CreateContactList` input untuk permintaan adalah sebagai berikut:

  ```
  {
      "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"
        }
       ]
}
```

Buat kontak

Anda dapat menggunakan `CreateContact`operasi di Amazon SES API v2 untuk membuat daftar kontak. Anda dapat mengonfigurasi pengaturan ini dengan cepat dan mudah menggunakan AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan Pengguna AWS Command Line Interface.

Untuk membuat kontak dengan menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

  ```
  aws sesv2 create-contact --cli-input-json file://CONTACT-JSON
  ```

  Di perintah sebelumnya, ganti *CONTACT-JSON dengan jalur ke file JSON* Anda untuk permintaan Anda. `CreateContact`

Contoh file JSON `CreateContact` input untuk permintaan adalah sebagai berikut:

```
{
    "ContactListName": "ExampleContactListName",
    "EmailAddress": "example@amazon.com",
    "UnsubscribeAll": false,
    "TopicPreferences": [
        {
            "TopicName": "Sports",
            "SubscriptionStatus": "OPT_IN"
        }
    ],
    "AttributesData": "{\"Name\": \"John\", \"Location\": \"Seattle\"}"
}
```

Dalam contoh di atas, `UnsubscribeAll` nilai `false` menunjukkan bahwa kontak belum berhenti berlangganan dari semua topik, dan nilai `true` berarti kontak telah berhenti berlangganan dari semua topik.

`TopicPreferences`mencakup informasi tentang status langganan kontak ke topik. Dalam contoh sebelumnya, kontak telah memilih topik "Olahraga" dan akan menerima semua email ke topik "Olahraga".

`AttributesData`Ini adalah bidang JSON tempat Anda dapat menempatkan metadata apa pun tentang kontak kami. Ini harus menjadi objek JSON valid.

Mengimpor kontak dalam jumlah besar ke daftar kontak Anda

Anda dapat secara manual menambahkan alamat dalam jumlah besar dengan terlebih dahulu mengunggah kontak Anda ke objek Amazon S3 diikuti dengan menggunakan `CreateImportJob`operasi di Amazon SES API v2 atau dengan menggunakan konsol SES. Untuk informasi selengkapnya, lihat Menambahkan alamat email secara massal ke daftar penekanan tingkat akun Anda.

Anda harus membuat daftar kontak sebelum mengimpor kontak Anda.

> **Note**
>
> Anda dapat menambahkan hingga 1 juta kontak ke daftar kontak perImportJob.

Untuk menambahkan kontak dalam jumlah besar ke daftar kontak Anda, selesaikan langkah berikut.

- Unggah kontak Anda ke objek Amazon S3 baik dalam format CSV atau JSON.

  Format CSV

  Baris pertama dari file yang diunggah ke Amazon S3 harus menjadi baris header.

  `topicPreferences`Objek perlu diratakan untuk format CSV. Setiap topik di `topicPreferences` akan memiliki bidang header terpisah.

  Contoh format CSV untuk menambahkan kontak dalam jumlah besar ke daftar kontak:

```
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
```

  Format JSON

  Hanya file JSON yang dibatasi baris baru yang didukung. Di format ini, setiap baris adalah objek JSON lengkap yang berisi informasi satu kontak.

  Contoh format JSON untuk menambahkan kontak dalam jumlah besar ke daftar kontak:

```
{
     "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"
     }
    ]
}
```

Di contoh sebelumnya, ganti *contoh1@amazon.com* dan *contoh2@amazon.com* dengan alamat email yang ingin Anda tambahkan ke daftar kontak. Ganti `attributesData` nilai-nilai dengan nilai-nilai khusus untuk kontak. Selain itu, ganti *Olahraga* dan *Bersepeda* dengan `topicName` yang berlaku untuk kontak Anda. *Yang dapat diterima `topicPreferences` adalah `OPT_IN` dan `OPT_OUT`.*

Atribut berikut didukung ketika mengunggah kontak Anda ke objek Amazon S3 baik dalam format CSV atau JSON:

Atribut	Deskripsi
`emailAddress`	Alamat email kontak. Ini adalah bidang wajib.
`unsubscribeAll`	Status nilai boolean mencatat jika kontak berhenti berlangganan dari semua topik daftar kontak.
`topicPreferences`	Preferensi kontak untuk memilih atau memilih keluar dari topik.
`attributesData`	Data atribut terlampir pada kontak.

- Berikan Amazon SES izin untuk membaca objek Amazon S3.

Saat diterapkan ke bucket Amazon S3, kebijakan tersebut memberikan Amazon SES izin untuk membaca ke bucket tersebut. Untuk informasi selengkapnya tentang melampirkan kebijakan ke bucket Amazon S3, lihat Menggunakan Kebijakan Bucket dan Kebijakan Pengguna dalam Panduan Pengguna Amazon Storage Service.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSESGet",
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::BUCKET-NAME/OBJECT-NAME",
            "Condition": {
                "StringEquals": {
                    "aws:Referer": "AWSACCOUNTID"
                }
            }
        }
    ]
}
```

- Memberi Amazon SES izin untuk menggunakan AWS KMS kunci Anda

  Jika objek Amazon S3 dienkripsi dengan AWS KMS kunci, Anda perlu memberikan Amazon SES izin untuk menggunakan kunci KMS. Amazon SES hanya mendapat izin dari kunci yang dikelola pelanggan, bukan kunci KMS default. Anda harus memberikan Amazon SES izin untuk menggunakan kunci yang dikelola pelanggan dengan menambahkan pernyataan ke kebijakan kunci.

  Tempelkan pernyataan kebijakan berikut ke kebijakan kunci untuk mengizinkan Amazon SES menggunakan kunci yang dikelola pelanggan Anda.

```
{
   "Sid": "AllowSESToDecrypt",
   "Effect": "Allow",
   "Principal": {
       "Service":"ses.amazonaws.com"
```

```
    },
    "Action": [
        "kms:Decrypt",
    ],
    "Resource": "*"
}
```

- Gunakan `CreateImportJob` operasi di Amazon SES API v2.

> **Note**
>
> Contoh berikut menganggap bahwa Anda telah menginstal. AWS CLI Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan Pengguna AWS Command Line Interface.

Di baris perintah, masukkan perintah berikut. Ganti *s3bucket* dengan nama bucket Amazon S3 dan *s3object dengan nama objek* Amazon S3.

```
aws sesv2 create-import-job --import-destination
 ContactListDestination={ContactListName=ExampleContactListName,ContactListImportAction=PUT}
 --import-data-source S3Url="s3://s3bucket/s3object",DataFormat=CSV
```

Panduan manajemen daftar dengan contoh

Panduan berikut memberikan contoh bagaimana Anda dapat menggunakan manajemen daftar untuk mencantumkan kontak Anda, memanfaatkan `ListManagementOptions` untuk menentukan daftar kontak dan nama topik di email Anda, dan cara memasukkan tautan berhenti berlangganan.

1. Daftar kontak dengan menggunakan AWS CLI - Dengan menggunakan `ListContacts` operasi untuk mengambil daftar semua kontak Anda yang telah berlangganan topik tertentu, bersama dengan `SendEmail` operasi, yang mengizinkan Anda untuk mengirimkan mereka email.

   Di baris perintah, masukkan perintah berikut:

   ```
   aws sesv2 list-contacts --cli-input-json file://LIST-CONTACTS-JSON
   ```

   Di perintah sebelumnya, ganti *LIST-CONTACTS-JSON dengan jalur ke file JSON* Anda untuk permintaan Anda. `ListContacts`

Contoh file JSON `ListContacts` input untuk permintaan adalah sebagai berikut:

```
{
    "ContactListName": "ExampleContactListName",
    "Filter": {
        "FilteredStatus": "OPT_IN",
        "TopicFilter": {
            "TopicName": "Cycling",
            "UseDefaultIfPreferenceUnavailable": true
        }
    },
    "PageSize": 50
}
```

`FilteredStatus`Menunjukkan status langganan yang ingin Anda filter, antara `OPT_IN` atau`OPT_OUT`.

`TopicFilter`Ini adalah filter opsional yang menentukan topik mana yang Anda inginkan hasilnya, dan dalam contoh di atas, yaitu "Cycling."

`UseDefaultIfPreferenceUnavailable`dapat memiliki nilai `true` atau`false`. Jika `true`, preferensi default topik akan digunakan jika kontak tidak memiliki preferensi eksplisit untuk topik. Jika `false`, hanya kontak dengan preferensi yang diatur secara eksplisit dipertimbangkan untuk pemfilteran.

2. Kirim email dengan **`ListManagementOptions`** diaktifkan - Setelah mencantumkan kontak dalam daftar Anda menggunakan `ListContacts`operasi di atas, Anda dapat menggunakan `SendEmail`operasi untuk mengirim email ke setiap kontak Anda dengan memanfaatkan `ListManagementOptions`header untuk menentukan daftar kontak dan nama topik Anda.

   Untuk digunakan `ListManagementOptions` dengan `SendEmail` operasi, termasuk `contactListName`dan `topicName`yang email milik (`topicName`adalah opsional):

```
ListManagementOptions:
    String contactListName
    String topicName
```

Jika Anda menyertakan `ListManagementOptions` `SendEmail` permintaan Anda ke alamat email penerima yang tidak ada di daftar kontak Anda, maka kontak akan dibuat di daftar Anda secara otomatis.

Amazon SES akan mementalkan email jika dikirim ke kontak yang tidak berlangganan di daftar kontak Anda, yang berarti Anda tidak perlu memperbarui `SendEmail` permintaan Anda untuk menghindari pengiriman ke kontak yang telah berhenti berlangganan.

3. Tunjukkan lokasi untuk tautan berhenti berlangganan Anda - Saat memanfaatkan `ListManagementOptions`Anda memiliki opsi untuk mengaktifkan Amazon SES untuk menambahkan tautan footer berhenti berlangganan di email Anda menggunakan `{{amazonSESUnsubscribeUrl}}` placeholder untuk menentukan di mana SES perlu memasukkan URL berhenti berlangganan. Penggantian placeholder didukung hanya untuk tipe konten HTML dan TEXT. Anda dapat menyertakan placeholder maksimum dua kali. Jika digunakan lebih dari dua kali, hanya dua kejadian pertama yang diganti. Untuk informasi selengkapnya, lihat Menggunakan manajemen berlangganan.

   Sebagai alternatif, jika Anda menggunakan antarmuka SMTP untuk mengirim email, Anda dapat menggunakan `X-SES-LIST-MANAGEMENT-OPTIONS` header untuk menentukan daftar dan nama topik.

   Untuk menentukan daftar dan nama topik saat mengirim email menggunakan antarmuka SMTP, tambahkan header email berikut pada pesan Anda:

   ```
   X-SES-LIST-MANAGEMENT-OPTIONS: {contactListName}; topic={topicName}
   ```

Menggunakan manajemen berlangganan

Amazon SES menyediakan kemampuan manajemen berlangganan, di mana Amazon SES secara otomatis mengaktifkan tautan berhenti berlangganan di setiap email keluar saat Anda menentukan `contactListName` dan `ListManagementOptions`di `topicName` dalamnya dalam permintaan `SendEmail`operasi.

Jika kontak berhenti berlangganan dari topik atau daftar tertentu, Amazon SES tidak mengizinkan email mengirim ke kontak untuk topik atau daftar di masa mendatang.

> **Note**
>
> - Manajemen langganan Amazon SES mendukung Persyaratan Pengirim Massal sebagaimana diberlakukan oleh banyak penyedia layanan email, lihat Bagian 2 dalam Gambaran Umum Perubahan Pengirim Massal untuk informasi selengkapnya.
> - Manajemen berlangganan tersedia bagi mereka yang menggunakan Easy DKIM di Amazon SES, tapi Amazon SES tidak mungkin menambahkan tautan berhenti berlangganan ke email Anda untuk pengirim yang menandatangani email sendiri sebelum memanggil Amazon SES.

Untuk informasi tentang manajemen daftar dan cara menggunakannya, termasuk mengambil daftar semua kontak Anda yang telah berlangganan topik tertentu, lihat. Menggunakan pengelolaan daftar

Gambaran umum manajemen berlangganan

Anda harus mempertimbangkan faktor-faktor berikut ketika Anda menggunakan manajemen berlangganan:

- Manajemen berlangganan akan dikelola penuh oleh Amazon SES. Ini berarti Amazon SES menerima email berhenti berlangganan dan permintaan dari halaman web berhenti berlangganan lalu memperbarui preferensi kontak dalam daftar Anda. Anda dapat menerima notifikasi berhenti berlangganan menggunakan notifikasi set konfigurasi. Untuk informasi selengkapnya tentang set konfigurasi, lihat Menggunakan set konfigurasi di Amazon SES.
- Anda perlu menentukan daftar kontak saat mengirim email. Manajemen langganan melalui tautan `List-Unsubscribe` header dan `ListManagementOptions` footer akan ditangani sesuai.
- Amazon SES menambahkan dukungan untuk standar `List-Unsubscribe` header, yang akan memungkinkan klien email dan penyedia kotak masuk untuk menampilkan tautan berhenti berlangganan di bagian atas email jika mereka mendukungnya - tidak semua penyedia layanan email mendukung header ini.
- `List-Unsubscribe`header mengikuti perilaku berikut:
  - Jika kontak mengklik tautan berhenti berlangganan di email yang memiliki daftar kontak dan topik yang ditentukan, maka kontak tersebut akan berhenti berlangganan hanya dari topik tertentu tersebut.
  - Jika topik tidak ditentukan, maka kontak akan berhenti berlangganan dari semua topik dalam daftar.

- Kontak akan diarahkan ke halaman arahan berhenti berlangganan saat mereka mengeklik tautan berhenti berlangganan di footer email.
- Halaman arahan berhenti berlangganan akan memberikan opsi kepada kontak untuk memperbarui preferensi mereka, artinya `OPT_IN` atau `OPT_OUT`, untuk semua topik dalam daftar tertentu. Halaman arahan juga memberikan opsi untuk berhenti berlangganan dari semua topik dalam daftar.
- Jika menggunakan `ListManagementOptions`, Anda harus menyertakan `{{amazonSESUnsubscribeUrl}}` placeholder di email Anda untuk menunjukkan di mana Amazon SES perlu memasukkan URL berhenti berlangganan. Anda dapat menyertakan placeholder maksimum dua kali. Jika digunakan lebih dari dua kali, hanya dua kejadian pertama yang diganti.
- Tautan `List-Unsubscribe` header dan `ListManagementOptions` footer ditambahkan hanya jika email dikirim ke satu penerima.
- Untuk email transaksional di mana Anda tidak ingin kontak dapat berhenti berlangganan, Anda dapat menghilangkan `ListManagementOptions`bidang dengan permintaan Anda. `SendEmail`

Pertimbangan header berhenti berlangganan

Manajemen langganan melalui tautan berhenti berlangganan diaktifkan ketika email berisi header berikut:

`List-Unsubscribe`

`List-Unsubscribe-Post`

Saat Anda menggunakan manajemen langganan Amazon SES `ListManagementOptions`, Amazon SES akan mengganti header ini jika ada di email.

Penerima yang berhenti berlangganan dengan mengklik tautan yang dihasilkan oleh header ini akan memiliki pengalaman yang berbeda tergantung pada klien email atau penyedia kotak masuk mereka karena beberapa penyedia tidak mengenali `List-Unsubscribe` dan `List-Unsubscribe-Post` header; email yang dikirim ke penerima menggunakan penyedia tersebut tidak akan melihat tautan Berhenti Berlangganan.

Penerima yang klien emailnya mengenali header ini akan melihat tautan Berhenti Berlangganan dan akan dapat berhenti berlangganan melalui tautan tetapi tidak akan memiliki opsi untuk memilih topik mana mereka berhenti berlangganan, dan hanya akan berhenti berlangganan dari topik tempat email dikirim.

Untuk informasi selengkapnya tentang `List-Unsubscribe` header, lihat RFC 2369, dan untuk `List-Unsubscribe-Post` header, lihat RFC 8058.

> **Note**
>
> Amazon SES mendukung berhenti berlangganan satu klik sesuai dengan Persyaratan Pengirim Massal sebagaimana diberlakukan oleh banyak penyedia layanan email, lihat Menggunakan berhenti berlangganan satu klik dengan Amazon SES untuk informasi selengkapnya.

Menambahkan tautan footer berhenti berlangganan

Anda akan perlu menggunakan placeholder `{{amazonSESUnsubscribeUrl}}` di email yang di templat dan nontemplat untuk menentukan tempat Amazon SES perlu memasukkan URL berhenti berlangganan.

Penggantian placeholder didukung hanya untuk tipe konten HTML dan TEXT.

Anda dapat menyertakan placeholder maksimum dua kali. Jika digunakan lebih dari dua kali, hanya dua kejadian pertama yang diganti.

> **Note**
>
> `{{amazonSESUnsubscribeUrl}}`Placeholder hanya dapat digunakan jika `ListManagementOptions`ditentukan sebagai header saat menggunakan `SendEmail`operasi atau X-SES-LIST-MANAGEMENT-OPTIONS ditentukan sebagai header saat menggunakan antarmuka SMTP. (Jangan bingung dengan `List-Unsubscribe` atau `List-Unsubscribe-Post` header yang tidak bergantung pada `ListManagementOptions` dan dapat digunakan sendiri.)

Memantau aktivitas pengiriman Amazon SES

Amazon SES menyediakan metode untuk memantau aktivitas pengiriman Anda menggunakan peristiwa, metrik, dan statistik. Peristiwa adalah sesuatu yang terjadi terkait dengan aktivitas pengiriman yang telah ditentukan untuk dilacak sebagai metrik. Metrik mewakili kumpulan titik data yang diurutkan waktu yang mewakili nilai dari jenis peristiwa yang dipantau yang menghasilkan statistik. Statistik adalah agregasi data metrik untuk jangka waktu tertentu termasuk hingga saat ini.

Metode pemantauan ini membantu Anda melacak langkah-langkah penting, seperti tingkat pentalan, keluhan, dan penolakan akun Anda. Tingkat pentalan dan keluhan yang terlalu tinggi dapat membahayakan kemampuan Anda untuk mengirim email menggunakan SES. Metode ini juga dapat digunakan untuk mengukur tingkat di mana pelanggan Anda terlibat dengan email yang Anda kirim dengan membantu Anda mengidentifikasi tarif terbuka dan klik secara keseluruhan menggunakan penerbitan acara dan domain khusus yang terkait dengan set konfigurasi - lihat. Mengonfigurasi domain kustom untuk menangani pelacakan buka dan klik

Langkah pertama dalam menyiapkan pemantauan adalah mengidentifikasi jenis peristiwa email yang terkait dengan aktivitas pengiriman Anda yang ingin Anda ukur dan pantau menggunakan SES. Anda dapat memilih jenis acara berikut untuk dipantau di SES:

- Kirim — Permintaan kirim berhasil dan Amazon SES akan mencoba mengirimkan pesan ke server email penerima. (Jika tingkat akun atau penekanan global sedang digunakan, SES masih akan menghitungnya sebagai kirim, tetapi pengiriman ditekan.)
- RenderingFailure— Email tidak dikirim karena masalah rendering template. Tipe peristiwa ini dapat terjadi saat data templat tidak ada, atau jika ada ketidakcocokan antara parameter templat dan data. (Tipe peristiwa ini hanya terjadi ketika Anda mengirim email menggunakan operasi API `SendTemplatedEmail` atau `SendBulkTemplatedEmail`.)
- Tolak — Amazon SES menerima email tersebut, tetapi memutuskan bahwa email tersebut berisi virus dan tidak berusaha mengirimkannya ke server email penerima.
- Pengiriman - Amazon SES berhasil mengirimkan email ke server email penerima.
- Bounce — Pantulan keras yang server email penerima menolak email secara permanen. (Pantulan lunak hanya disertakan ketika SES tidak lagi mencoba mengirimkan email. Umumnya pantulan lunak ini menunjukkan kegagalan pengiriman, meskipun dalam beberapa kasus pantulan lunak dapat dikembalikan bahkan ketika surat berhasil mencapai kotak masuk penerima. Ini biasanya terjadi ketika penerima mengirim balasan out-of-office otomatis. Pelajari lebih lanjut tentang soft bounce di artikel AWS re:Post ini.)

- Keluhan — Email berhasil dikirim ke server email penerima, tetapi penerima menandainya sebagai spam.
- DeliveryDelay— Email tidak dapat dikirim ke server email penerima karena masalah sementara terjadi. Penundaan penyamapaian dapat terjadi, misalnya, saat kotak masuk penerima penuh, atau saat server email penerima mengalami masalah sementara.
- Langganan — Email berhasil dikirimkan, tetapi penerima memperbarui preferensi langganan dengan mengklik `List-Unsubscribe` header email atau `Unsubscribe` tautan di footer.
- Buka — Penerima menerima pesan dan membukanya di klien email mereka.
- Klik — Penerima mengklik satu atau beberapa tautan di email.

Anda dapat memantau peristiwa pengiriman email dengan beberapa cara. Metode yang Anda pilih bergantung pada tipe peristiwa yang ingin Anda pantau, granularitas dan tingkat detail yang ingin Anda pantau, serta lokasi tempat Anda ingin Amazon SES menerbitkan data. Anda harus menggunakan notifikasi umpan balik atau penerbitan peristiwa untuk melacak peristiwa pentalan dan aduan. Anda juga dapat memilih untuk menggunakan beberapa metode pemantauan. Karakteristik masing-masing metode tercantum dalam tabel berikut.

Metode Pemantauan	Peristiwa yang Dapat Anda Pantau	Cara Mengakses Data	Tingkat Detail	Granularitas
Konsol Amazon SES	Kesehatan akun, email yang dikirim, kuota yang digunakan, permintaan kirim yang berhasil, penolakan , pantulan & keluhan (riwayat terbaru dengan reputasi saat ini)	Halaman dasbor akun dalam konsol Amazon SES	Jumlah dan persentase	Di seluruh AWS akun
Konsol Amazon SES	Kesehatan akun, email yang dikirim,	Halaman reputasi metrik	Hanya tingkat yang dihitung	Di seluruh AWS akun

Metode Pemantauan	Peristiwa yang Dapat Anda Pantau	Cara Mengakses Data	Tingkat Detail	Granularitas
	pantulan & keluhan (reputasi saat ini)	dalam konsol Amazon SES		
API Amazon SES	Penyampaian, pentalan, aduan, dan penolakan	Operasi API `GetSendStatistics`	Hanya jumlah	Di seluruh AWS akun

Metode Pemantauan	Peristiwa yang Dapat Anda Pantau	Cara Mengakses Data	Tingkat Detail	Granularitas
CloudWatch Konsol Amazon	Mengirim, mengirimkan, membuka, mengklik, memantul, rasio pentalan, keluhan, tingkat keluhan, penolakan, kegagalan rendering, dan IP yang masuk daftar hitam.	CloudWatch konsol **Note** Beberapa metrik tidak muncul CloudWatch sampai peristiwa terkait terjadi. Misalnya, metrik bouncing tidak muncul CloudWatch hingga setidaknya satu email yang Anda kirim memantul, atau hingga Anda menghasilkan	Hanya jumlah	Di seluruh AWS akun

Metode Pemantauan	Peristiwa yang Dapat Anda Pantau	Cara Mengakses Data	Tingkat Detail	Granularitas
		peristiwa pantulan simulasi dengan menggunakan simulator kotak pesan.		
Notifikasi umpan balik	Penyampaian, pentalan, dan aduan	Notifikasi Amazon SNS (pengiriman, pentalan, dan keluhan) atau email (hanya pantulan dan keluhan). Lihat Menyiapkan notifikasi peristiwa.	Detail tentang setiap peristiwa	Di seluruh AWS akun

Metode Pemantauan	Peristiwa yang Dapat Anda Pantau	Cara Mengakses Data	Tingkat Detail	Granularitas
Penerbitan peristiwa	Pengiriman, penyampaian, pembukaan, pengeklikan, pentalan, aduan, penolakan, dan kegagalan rendering.	Amazon CloudWatch atau Amazon Data Firehose, atau pemberitahuan Amazon SNS —lihat. Pantau pengiriman email menggunakan penerbitan acara (Biaya tambahan berlaku, lihat Harga per metrik untuk CloudWatch.)	Detail tentang setiap peristiwa	Detail (berdasarkan karakteristik email yang dapat ditentukan pengguna)
Penerbitan acara menggunakan domain khusus yang terkait dengan set konfigurasi - info lebih lanjut	Buka dan klik pelacakan.	Amazon CloudWatch atau Amazon Data Firehose, atau dengan pemberitahuan Amazon SNS. (Biaya tambahan berlaku, lihat Harga per metrik untuk CloudWatch.)	Detail pada setiap acara.	Detail (berdasarkan karakteristik email yang dapat ditentukan pengguna)

> **Note**
>
> Metrik yang diukur dengan peristiwa pengiriman email mungkin tidak selaras dengan kuota pengiriman Anda. Perbedaan ini dapat disebabkan oleh pentalan dan penolakan email, atau dengan menggunakan simulator kotak masuk Amazon SES. Untuk mengetahui seberapa dekat kuota pengiriman Anda, lihat Pemantauan kuota pengiriman Anda.

Untuk informasi selengkapnya tentang cara menggunakan setiap metode pemantauan, lihat topik-topik berikut:

- Pemantauan statistik pengiriman Anda menggunakan konsol Amazon SES
- Memantau statistik penggunaan Anda menggunakan API Amazon SES
- Pantau pengiriman email menggunakan penerbitan SES acara Amazon

Pemantauan statistik pengiriman Anda menggunakan konsol Amazon SES

Dari konsol Amazon SESDasbor akun,Metrik reputasi, danPengaturan SMTPhalaman, Anda dapat memantau semua pengiriman email, penggunaan, statistik, pengaturan SMTP, kesehatan akun secara keseluruhan, dan metrik reputasi. Bagian berikut menjelaskan metrik dan statistik yang disediakan di masing-masing halaman konsol ini.

Perlu dicatat bahwa sementara keduanyathe section called “Dasbor akun”danthe section called “Metrik reputasi”halaman konsol berisi metrik bouncing dan keluhan, ada perbedaan halus antara dua set tingkat pentalan dan keluhan ini seperti yang dijelaskan di bawah ini:

- Halaman dasbor akun— berdasarkan rentang tanggal yang dipilih, Anda dapat melihat tingkat bouncing dan keluhan di masa lalu yang menunjukkan perkembangan metrik perubahan menjelang saat ini.
- Halaman metrik reputasi— tingkat pentalan dan keluhan berdasarkan titik data terbaru yang diterima dari penghitungan rata-rata historis Anda secara keseluruhan pada tingkat tinggi (ini tidak boleh disamakan dengan tingkat pantulan/keluhan reguler Anda, yang sesuai dengan peristiwa pantulan/keluhan yang tepat saat terjadi secara real-time seperti yang ditunjukkan padaDasbor akunhalaman).

Sebagai contoh sederhana untuk membandingkan tingkat bouncing atau keluhan antaraMetrik reputasihalaman danDasbor akunhalaman, katakanlah tarifnya 2% kemarin dan 1% sekarang, diMetrik reputasihalaman, Anda hanya akan melihat tingkat saat ini 1%, tetapi padaDasbor akunhalaman, grafik akan memplot perkembangan yang dipetakan yang menunjukkan tingkat 2% untuk kemarin dan 1% untuk hari ini.

Dasbor akun

Anda dapat memantau jumlah email yang dikirim dari akun Anda, serta persentase kuota pengiriman yang telah Anda gunakan, langsung dari konsol SESDasbor akunhalaman dalamPenggunaan email harianpanel. Tingkat pengiriman dan penolakan akun Anda dapat dipantau diStatistik pengirimanpanel, serta faktor kunci lainnya yang terkait dengan pengiriman email Anda di panel berikut:

- Batas pengiriman— berisi kuota berikut yang berlaku untuk mengirim surat melalui SES:
  - Kuota pengiriman harian- Jumlah maksimum email yang dapat Anda kirim dalam periode 24 jam.
  - Laju pengiriman maksimum- Jumlah maksimum email yang dapat Anda kirim dari akun Anda setiap detik.
- Kesehatan akun— status akun SES Anda:
  - `Healthy`- tidak ada masalah terkait reputasi yang saat ini memengaruhi akun Anda.
  - `Under review`- potensi masalah telah diidentifikasi dengan akun SES Anda - akun Anda sedang ditinjau saat Anda bekerja untuk memperbaiki masalah.
  - `Paused`— Kemampuan akun Anda untuk mengirim email saat ini dijeda karena masalah dengan email yang dikirim dari akun Anda. Ketika masalah telah diperbaiki, Anda dapat meminta akun Anda untuk mengirim email dilanjutkan.
- Penggunaan email harian— untuk memeriksa penggunaan harian Anda untuk memastikan Anda tidak mendekati batas pengiriman Anda:
  - Email terkirim- Jumlah email yang dikirim dalam periode 24 jam.
  - Sisa mengirim- Jumlah total email yang tersedia untuk periode 24 jam.
  - Kuota pengiriman yang digunakan- persentase kuota pengiriman harian Anda yang digunakan.
- Statistik pengiriman— terdiri dari grafik yang menunjukkan perkembangan empat metrik penting dalam kumpulan titik data yang diurutkan waktu yang mewakili nilai tipe peristiwa yang dipantau yang menghasilkan statistik untuk rentang tanggal yang dipilih menggunakan periode agregasi1 jam. Anda dapat memilih rentang data dengan nilai awal dari`Last 1 day`kepada`Last 14 days`untuk memfilter grafik di bawah ini:

- Mengirim- jumlah permintaan pengiriman email yang berhasil untuk rentang tanggal yang dipilih.
- Menolak- tingkat rata-rata permintaan kirim yang ditolak oleh SES berdasarkan`Rejects/Sends * 100`untuk rentang tanggal yang dipilih.
- Memantul- tingkat rata-rata yang berasal dari metrik reputasi pengirim historis Anda secara keseluruhan yang menunjukkan perkembangan untuk rentang tanggal yang dipilih.
- Keluhan- tingkat rata-rata yang berasal dari metrik reputasi pengirim historis Anda secara keseluruhan yang menunjukkan perkembangan untuk rentang tanggal yang dipilih.

Masing-masing grafik ini berisiLihat di CloudWatchtombol yang akan membuka metrik masing-masing di Amazon CloudWatch konsol yang memungkinkan data terperinci untuk dilihat, matematika metrik yang disesuaikan dilakukan, dan[pembuatan alarm di CloudWatch](#).

Metrik reputasi

Selain tingkat bouncing dan keluhan,Metrik reputasipage juga menyediakan visibilitas tingkat tinggi lainnya ke dalam faktor-faktor kunci yang mempengaruhi reputasi Anda yang terdiri dari panel berikut:

- Ringkasan— memberikan gambaran kesehatan reputasi Anda.
  - Status- Kesehatan reputasi secara keseluruhan berdasarkan tingkat pentalan dan keluhan historis:
    - `Healthy`- kedua metrik berada dalam level normal.
    - `Under review`— satu atau kedua metrik secara otomatis menyebabkan akun Anda ditinjau.
    - `At risk`- satu atau kedua metrik telah mencapai tingkat yang tidak sehat dan kemampuan akun Anda untuk mengirim email mungkin berisiko.
  - Email terkirim (24 jam terakhir)— jumlah total email yang dikirim dalam periode 24 jam terakhir.
  - Sisa mengirimJumlah total email yang tersedia untuk periode 24 jam.
  - Kuota pengiriman yang digunakan— persentase kuota pengiriman harian Anda yang digunakan.
- Isi tab tingkat akun:
  - Tingkat bouncing
    - Status- menunjukkan kesehatan rasio pentalan Anda menggunakan nilai yang sama seperti yang dijelaskan untuk panel Ringkasan.
    - Tingkat pentalan historis— Persentase email yang mengakibatkan hard bounce yang dihitung dari rata-rata historis Anda secara keseluruhan berdasarkan volume yang mewakili praktik pengiriman Anda yang khas.

    - Tingkat keluhan
        - Status- Menunjukkan kesehatan tingkat keluhan Anda menggunakan nilai yang sama seperti yang dijelaskan untuk panel Ringkasan.
        - Tingkat pentalan historis— Persentase email yang dikirim dari akun Anda yang mengakibatkan penerima melaporkannya sebagai spam yang dihitung dari rata-rata historis Anda secara keseluruhan berdasarkan volume yang mewakili praktik pengiriman Anda yang khas.
- Konfigurasi mengatur isi tab:
    - Reputasi berdasarkan set konfigurasi
        - Set konfigurasi- Memungkinkan Anda mengetik atau memilih set konfigurasi yang memiliki metrik reputasi diaktifkan sehingga Anda dapat melihat ringkasan, bouncing, dan data keluhan berdasarkan email yang dikirim menggunakan set konfigurasi yang dipilih. Panel yang dihasilkan yang muncul setelah memilih set konfigurasi sama seperti yang dijelaskan di atas untuk halaman metrik Reputasi kecuali hanya didasarkan pada email yang dikirim dengan konfigurasi yang dipilih ditetapkan sesuai dengan metrik pengiriman tingkat akun Anda secara keseluruhan.

Pengaturan SMTP

Halaman ini mencantumkan pengaturan SMTP yang diperlukan untuk menggunakan antarmuka SMTP Amazon SES baik melalui SES API atau secara terprogram, dan menyediakan tautan untuk membuat dan mengelola kredensi SMTP Anda:

- Pengaturan SMTP— jika Anda ingin menggunakan bahasa pemrograman yang mendukung SMTP, server email, atau aplikasi untuk terhubung ke antarmuka SMTP Amazon SES, informasi berikut disediakan:
    - Titik akhir SMTP
    - Pelabuhan STARTTLS
    - Keamanan Lapisan Pengangkutan (TLS)
    - Port Pembungkus TLS
    - Tautan otentikasi disediakan untuk pembuatan dan manajemen kredensi SMTP dan IAM

Menggunakan konsol untuk memantau metrik pengiriman dan reputasi

Prosedur berikut akan membantu Anda memulai menjelajahi metrik pengiriman dan reputasi Anda baik menggunakanDasbor akunhalaman untuk metrik berdasarkan riwayat terbaru (hingga 14 hari), atau gunakanMetrik reputasihalaman untuk metrik berdasarkan riwayat keseluruhan Anda hingga saat ini.

Untuk melihat email yang dikirim dan kuota pengiriman yang digunakan

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi, pilih Dasbor akun. Statistik penggunaan Anda ditampilkan dalamPenggunaan email harianbagian.

Untuk melihat jumlah pengiriman, tingkat penolakan, pantulan, dan keluhan

1. Di panel navigasi, pilih Dasbor akun.
2. DiStatistik pengirimanbagian, gunakanRentang tanggaldropdown untuk memilih nilai awal untuk rentang tanggal untuk memfilter empat grafik langsung di bawahStatistik pengirimanbagian.
3. Berdasarkan rentang tanggal yang dipilih, Anda dapat melihat jumlah dan tarif di masa lalu yang menunjukkan perkembangan metrik perubahan yang mengarah ke waktu sekarang.
4. Di salah satu grafik, pilihLihat di CloudWatchtombol untuk membuka metrik masing-masing diAmazon CloudWatchkonsol tempat Anda dapat melihat data terperinci, melakukan matematika metrik yang disesuaikan, danmembuat alarm pemantauan di CloudWatch.

Untuk melihat tingkat pentalan dan keluhan historis

1. Di panel navigasi, pilihMetrik reputasi.
2. DiTingkat bouncingpanel Anda dapat melihat persentase email yang dikirim dari akun Anda yang mengakibatkan hard bounce, dan diTingkat keluhanAnda dapat melihat persentase email yang dikirim dari akun Anda yang mengakibatkan penerima melaporkannya sebagai spam; kedua metrik dihitung dari volume email yang representatif berdasarkan praktik pengiriman yang khas.
3. Di salah satu panel, pilihLihat di CloudWatchtombol untuk membuka metrik masing-masing diAmazon CloudWatchkonsol tempat Anda dapat melihat data terperinci, melakukan matematika metrik yang disesuaikan, danmembuat alarm pemantauan di CloudWatch.

Untuk melihat metrik reputasi berdasarkan set konfigurasi

1. Di panel navigasi, pilihMetrik reputasi.
2. Pada halaman metrik Reputasi, pilihSet konfigurasitab.
3. DiReputasi berdasarkan set konfigurasipanel, klik di dalamSet konfigurasifield dan mulai mengetik untuk, atau pilih, set konfigurasi yang mengaktifkan metrik reputasi.
4. Setelah memilih set konfigurasi, itu akan memuat panel Ringkasan, Pantulan, dan Keluhan yang menampilkan metrik hanya berdasarkan email yang dikirim dengan set konfigurasi yang dipilih.

Memantau statistik penggunaan Anda menggunakan API Amazon SES

API Amazon SES menyediakan operasi `GetSendStatistics`, yang mengembalikan informasi tentang penggunaan layanan Anda. Kami merekomendasikan Anda untuk memeriksa statistik pengiriman secara teratur, sehingga Anda dapat melakukan penyesuaian jika diperlukan.

Ketika Anda memanggil operasi `GetSendStatistics`, Anda menerima daftar titik data yang mewakili aktivitas pengiriman Anda selama dua minggu terakhir. Setiap titik data dalam daftar ini mewakili 15 menit aktivitas dan berisi informasi berikut untuk periode tersebut:

- Jumlah pentalan keras
- Jumlah aduan
- Jumlah percobaan pengiriman (sesuai dengan jumlah email yang telah Anda kirim)
- Jumlah percobaan pengiriman yang ditolak
- Stempel waktu untuk periode analisis

Untuk deskripsi lengkap tentang operasi `GetSendStatistics`, lihat Referensi API Amazon Simple Email Service.

Pada bagian ini, Anda akan menemukan topik berikut:

- the section called “Memanggil operasi API `GetSendStatistics` menggunakan AWS CLI”
- the section called “Memanggil pemrograman operasi `GetSendStatistics`”

Memanggil operasi API **GetSendStatistics** menggunakan AWS CLI

Cara termudah untuk memanggil operasi API `GetSendStatistics` adalah dengan menggunakan AWS Command Line Interface (AWS CLI).

Untuk memanggil operasi API **GetSendStatistics** menggunakan AWS CLI

1. Jika Anda belum melakukannya, instal AWS CLI. Untuk informasi lebih lanjut, lihat "Menginstal AWS Command Line Interface" dalam Panduan Penggunaan AWS Command Line Interface.
2. Jika Anda belum melakukannya, konfigurasikan AWS CLI untuk menggunakan kredensial AWS Anda. Untuk informasi lebih lanjut, lihat "Mengonfigurasi AWS CLI" dalam Panduan Pengguna AWS Command Line Interface.
3. Pada baris perintah, jalankan perintah berikut:

   ```
   aws ses get-send-statistics
   ```

   Jika AWS CLI dikonfigurasi dengan benar, Anda melihat daftar pengiriman statistik dalam format JSON. Setiap objek JSON termasuk statistik pengiriman yang digabungkan untuk jangka waktu 15 menit.

Memanggil pemrograman operasi **GetSendStatistics**

Anda juga dapat memanggil operasi `GetSendStatistics` menggunakan SDK AWS. Bagian ini mencakup contoh kode untuk SDK AWS for Go, PHP, Python, dan Ruby. Pilih salah satu tautan berikut untuk melihat contoh kode untuk bahasa tersebut:

- Contoh kode untukAWS SDK for Go
- Contoh kode untukAWS SDK for PHP
- Contoh kode untukAWS SDK for Python (Boto)
- Contoh kode untukAWS SDK for Ruby

> **Note**
>
> Contoh kode berikut mengasumsikan bahwa Anda telah membuat file kredensial berbagi AWS yang berisi file access key ID AWS Anda, secret access key AWS Anda, dan pilihan Wilayah AWS Anda. Untuk informasi lebih lanjut, lihat File kredensial dan konfigurasi berbagi.

Memanggil **`GetSendStatistics`** menggunakan AWS SDK for Go

```
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)
}
```

Memanggil **GetSendStatistics** menggunakan AWS SDK for PHP

```
<?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");
}

?>
```

Memanggil **GetSendStatistics** menggunakan AWS SDK for Python (Boto)

```
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))
```

Memanggil **GetSendStatistics** menggunakan AWS SDK for Ruby

```
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
```

Pantau pengiriman email menggunakan penerbitan SES acara Amazon

Untuk memungkinkan Anda melacak pengiriman email pada tingkat terperinci, Anda dapat mengatur Amazon SES untuk mempublikasikan peristiwa pengiriman email ke Amazon, Amazon Data Firehose CloudWatch, Amazon Pinpoint, Amazon Simple Notification Service, atau EventBridge Amazon berdasarkan karakteristik yang Anda tentukan.

Anda dapat melacak beberapa tipe peristiwa pengiriman email, yang meliputi pengiriman, penyampaian, pembukaan, pengeklikan, pentalan, aduan, penolakan, kegagalan rendering, dan penundaan penyampaian. Informasi ini dapat berguna untuk tujuan operasional dan analitis. Misalnya, Anda dapat mempublikasikan data pengiriman email ke CloudWatch dan membuat dasbor yang melacak kinerja kampanye email Anda, atau Anda dapat menggunakan Amazon SNS untuk mengirimkan pemberitahuan ketika peristiwa tertentu terjadi.

Cara kerja penerbitan acara dengan set konfigurasi dan tag pesan

Untuk menggunakan penerbitan peristiwa, Anda terlebih dahulu menyiapkan satu atau beberapa set konfigurasi. Sebuah set konfigurasi menentukan tempat untuk memublikasikan peristiwa Anda dan peristiwa yang akan dipublikasikan. Kemudian, setiap kali Anda mengirim email, Anda memberikan nama set konfigurasi dan satu atau beberapa tanda pesan, dalam bentuk pasangan nama/nilai, untuk mengategorikan email. Misalnya, jika Anda mengiklankan buku, maka Anda dapat menamai tanda pesan dengan genre, dan menetapkan nilai fiksi ilmiah atau barat, saat Anda mengirim email untuk kampanye terkait.

Bergantung pada antarmuka pengiriman email yang Anda gunakan, Anda dapat memberikan tag pesan sebagai parameter ke `EmailTags`bidang `SendEmail`APIoperasi atau menambahkan tag pesan ke header `X-SES-MESSAGE-TAGS`email SES -spesifik. Untuk informasi selengkapnya tentang set konfigurasi, lihat Menggunakan set konfigurasi di Amazon SES.

Selain tag pesan yang Anda tentukan, Amazon SES juga menambahkan tag otomatis ke pesan yang Anda kirim. Anda tidak perlu melakukan langkah-langkah tambahan untuk menggunakan tanda otomatis.

Tabel berikut mencantumkan tag otomatis yang diterapkan secara otomatis ke pesan yang Anda kirim menggunakan AmazonSES.

Tag SES Otomatis Amazon

Nama tanda otomatis	Deskripsi
`ses:caller-identity`	IAMIdentitas SES pengguna Amazon yang mengirim email.
`ses:configuration-set`	Nama Set Konfigurasi yang terkait dengan email.
`ses:from-domain`	Domain alamat “Dari”.
`ses:outgoing-ip`	Alamat IP yang SES digunakan Amazon untuk mengirim email.
`ses:source-ip`	Alamat IP yang digunakan pemanggil untuk mengirimkan email.
`ses:source-tls-version`	Versi TLS protokol yang digunakan penelepon untuk mengirim email.

Umpan balik halus untuk kampanye email

`ses:feedback-id-<a or b>`Tag adalah tag pesan opsional yang dapat Anda anggap sebagai tag hibrida atau semi-otomatis—meskipun mirip dengan tag otomatis yang dibahas di bagian sebelumnya, perbedaannya adalah Anda harus menambahkannya secara manual dan menggunakan kunci awalan. `ses:` Anda dapat menggunakan hingga dua tag ini yang didefinisikan sebagai `ses:feedback-id-a` dan`ses:feedback-id-b`.

Saat Anda menentukan tag ini, SES secara otomatis menambahkannya ke `Feedback-ID` header standar yang digunakan dalam menyediakan statistik pengiriman, seperti tingkat keluhan dan spam, sebagai bagian dari loop umpan balik (FBL), lihatLoop umpan balik. `Feedback-ID`Header terdiri dari pengenal, SESInternalID, digunakan oleh SES untuk mengumpulkan informasi keluhan, dan tag statis, Amazon SES, mengidentifikasi SES sebagai platform pengiriman seperti:

```
FeedBackId:feedback-id-a:feedback-id-b:((SESInternalID):(AmazonSES))
```

Tag ID umpan balik opsional ini ditawarkan sebagai cara bagi Anda untuk menghasilkan umpan balik halus, seperti untuk pesan yang Anda kirim sebagai bagian dari kampanye email. Anda dapat

menggunakan `ses:feedback-id-<a or b>` dengan menentukannya sebagai tag pesan di `EmailTags`bidang permintaan `SendEmail`operasi seperti yang ditunjukkan pada contoh berikut:

```
{
  "FromEmailAddress": "noreply@example.com",
  "Destination": {
    "ToAddresses": [
      "customer@example.net"
    ]
  },
  "Content": {
    "Simple": {
      "Subject": {
        "Data": "Hello and welcome"
      },
      "Body": {
        "Text": {
          "Data": "Lorem ipsum dolor sit amet."
        },
        "Html": {
          "Data": "Lorem ipsum dolor sit amet."
        }
      }
    }
  },
  "EmailTags": [
    {
      "Name": "ses:feedback-id-a",
      "Value": "new-members-campaign"
    },
    {
      "Name": "ses:feedback-id-b",
      "Value": "football-campaign"
    }
  ],
  "ConfigurationSetName": "football-club"
}
```

Jika mengirim dalam format mentah, Anda akan menambahkan `ses:feedback-id-<a or b>` sebagai tag pesan ke header SES `X-SES-MESSAGE-TAGS`-specific.

Tag `ses:feedback-id-<a or b>` pesan juga dapat dilacak di Amazon CloudWatch dengan menentukannya sebagai sumber CloudWatch nilai seperti tag pesan lainnya, lihat the section called

"Menambahkan Detail Tujuan CloudWatch Acara" (Biaya tambahan berlaku, lihat Harga per metrik untuk CloudWatch.)

Cara menggunakan penerbitan peristiwa

Bagian berikut berisi informasi yang Anda perlukan untuk mengatur dan menggunakan Amazon SES event publishing.

- Menyiapkan penerbitan peristiwa
- Bekerja dengan data peristiwa

Terminologi penerbitan peristiwa

Daftar berikut mendefinisikan istilah yang terkait dengan penerbitan SES acara Amazon.

Peristiwa pengiriman email

Informasi yang terkait dengan hasil email yang Anda kirimkan ke AmazonSES. Peristiwa pengiriman meliputi hal berikut:

- Kirim — Permintaan kirim berhasil dan Amazon SES akan mencoba mengirimkan pesan ke server email penerima. (Jika tingkat akun atau penekanan global sedang digunakan, masih SES akan menghitungnya sebagai kirim, tetapi pengiriman ditekan.)
- RenderingFailure— Email tidak dikirim karena masalah rendering template. Tipe peristiwa ini dapat terjadi saat data templat tidak ada, atau jika ada ketidakcocokan antara parameter templat dan data. (Jenis peristiwa ini hanya terjadi ketika Anda mengirim email menggunakan `SendTemplatedEmail`atau `SendBulkTemplatedEmail`APIoperasi.)
- Tolak — Amazon SES menerima email tersebut, tetapi memutuskan bahwa email tersebut berisi virus dan tidak berusaha mengirimkannya ke server email penerima.
- Pengiriman - Amazon SES berhasil mengirimkan email ke server email penerima.
- Bounce — Pantulan keras yang server email penerima menolak email secara permanen. (Pantulan lunak hanya disertakan ketika SES tidak lagi mencoba mengirim email. Umumnya pantulan lunak ini menunjukkan kegagalan pengiriman, meskipun dalam beberapa kasus pantulan lunak dapat dikembalikan bahkan ketika surat berhasil mencapai kotak masuk penerima. Ini biasanya terjadi ketika penerima mengirim balasan out-of-office otomatis. Pelajari lebih lanjut tentang soft bounce di artikel AWS re:Post ini.)
- Keluhan — Email berhasil dikirim ke server email penerima, tetapi penerima menandainya sebagai spam.

- DeliveryDelay— Email tidak dapat dikirim ke server email penerima karena masalah sementara terjadi. Penundaan penyamapaian dapat terjadi, misalnya, saat kotak masuk penerima penuh, atau saat server email penerima mengalami masalah sementara.
- Langganan — Email berhasil dikirimkan, tetapi penerima memperbarui preferensi langganan dengan mengklik `List-Unsubscribe` header email atau `Unsubscribe` tautan di footer.
- Buka — Penerima menerima pesan dan membukanya di klien email mereka.
- Klik — Penerima mengklik satu atau beberapa tautan di email.

Set konfigurasi

Seperangkat aturan yang menentukan tujuan tempat Amazon memublikasikan peristiwa pengiriman email, dan jenis peristiwa pengiriman email yang ingin Anda SES publikasikan. Saat Anda mengirim email yang ingin Anda gunakan dengan penerbitan peristiwa, Anda menentukan set konfigurasi untuk dikaitkan dengan email.

Tujuan peristiwa

AWS Layanan tempat Anda mempublikasikan acara pengiriman SES email Amazon. Setiap tujuan peristiwa yang Anda siapkan adalah milik satu, dan hanya satu, set konfigurasi.

Tanda pesan

Pasangan nama/nilai yang Anda gunakan untuk mengategorikan email untuk tujuan penerbitan peristiwa. Contohnya adalah kampanye/buku dan kampanye/pakaian. Saat mengirim email, Anda menentukan tag pesan sebagai parameter API panggilan atau sebagai header email SES khusus Amazon.

Tanda otomatis

Tanda pesan yang secara otomatis disertakan dalam laporan penerbitan peristiwa. Ada tag otomatis untuk nama set konfigurasi, domain alamat “Dari”, alamat IP keluar pemanggil, alamat IP SES keluar Amazon, dan IAM identitas pemanggil.

Menyiapkan penerbitan peristiwa Amazon SES

Bagian ini menjelaskan apa yang perlu Anda lakukan untuk mengonfigurasi Amazon SES untuk menerbitkan peristiwa pengiriman email Anda ke layanan AWS berikut:

- Amazon CloudWatch
- Amazon Data Firehose

- Amazon Pinpoint
- Amazon Simple Notification Service (Amazon SNS)

Langkah-langkah berikut yang diperlukan untuk menyiapkan penerbitan peristiwa tercakup dalam topik di bawah ini:

1. Anda harus membuat set konfigurasi menggunakan konsol Amazon SES atau API.
2. Tambahkan satu atau beberapa tujuan acara (CloudWatch, Firehose, Pinpoint, atau SNS) ke set konfigurasi, dan konfigurasikan parameter unik untuk tujuan acara.
3. Saat Anda mengirimkan email, Anda menentukan konfigurasi yang akan digunakan yang berisi tujuan peristiwa Anda.

Topik di bagian ini

- Langkah 1: Membuat set konfigurasi
- Langkah 2: Tambahkan tujuan peristiwa
- Langkah 3: Tentukan set konfigurasi saat Anda mengirim email

Langkah 1: Membuat set konfigurasi

Anda harus terlebih dahulu memiliki set konfigurasi untuk menyiapkan penerbitan peristiwa. Jika Anda belum memiliki set konfigurasi, atau ingin membuat yang baru, silakan lihat Membuat set konfigurasi di SES

Anda juga dapat membuat set konfigurasi menggunakan operasi CreateConfigurationSet di API Amazon SES V2 atau Amazon SES CLI v2, lihat Buat set konfigurasi. (AWS CLI).

Langkah 2: Tambahkan tujuan peristiwa

Tujuan peristiwa adalah tempat Anda memublikasikan peristiwa Amazon SES. Setiap tujuan peristiwa yang Anda siapkan adalah milik satu, dan hanya satu, set konfigurasi. Saat menyiapkan tujuan acara dengan Amazon SES, Anda memilih tujuan AWS layanan, dan Anda menentukan parameter yang terkait dengan tujuan tersebut.

Saat menyiapkan tujuan acara, Anda dapat memilih untuk mengirim acara ke salah satu AWS layanan berikut:

- Amazon CloudWatch

- Amazon Data Firehose
- Amazon EventBridge
- Amazon Pinpoint
- Amazon Simple Notification Service (Amazon SNS)

Tujuan peristiwa yang Anda pilih bergantung pada tingkat detail terkait peristiwa sesuai keinginan Anda, dan cara menerima informasi peristiwa sesuai keinginan Anda. Jika Anda hanya ingin total berjalan dari setiap jenis acara (misalnya, sehingga Anda dapat mengatur alarm ketika totalnya terlalu tinggi), Anda dapat menggunakannya CloudWatch.

Jika Anda menginginkan catatan peristiwa terperinci yang dapat Anda keluarkan ke layanan lain seperti Amazon OpenSearch Service atau Amazon Redshift untuk analisis, Anda dapat menggunakan Firehose.

Jika Anda ingin menerima notifikasi ketika peristiwa tertentu terjadi, maka Anda dapat menggunakan Amazon SNS.

Bagian ini berisi topik-topik berikut

- Menyiapkan tujuan CloudWatch acara untuk penerbitan acara
- Menyiapkan tujuan acara Firehose Data untuk penerbitan acara Amazon SES
- Siapkan EventBridge tujuan Amazon untuk penerbitan acara
- Siapkan tujuan acara Amazon Pinpoint untuk penerbitan acara
- Siapkan tujuan peristiwa Amazon SNS untuk penerbitan peristiwa

Menyiapkan tujuan CloudWatch acara untuk penerbitan acara

Dengan CloudWatch metrik Amazon, Anda dapat menggunakan tujuan acara untuk mempublikasikan acara pengiriman email Amazon SES. CloudWatch Karena tujuan CloudWatch acara hanya dapat diatur dalam set konfigurasi, Anda harus terlebih dahulu membuat set konfigurasi dan kemudian menambahkan tujuan acara ke set konfigurasi.

Saat menambahkan tujuan CloudWatch acara ke set konfigurasi, Anda harus memilih satu atau beberapa CloudWatch dimensi yang sesuai dengan tag pesan yang Anda gunakan saat mengirim email. Seperti tag pesan, CloudWatch dimensi adalah pasangan nama/nilai yang membantu Anda mengidentifikasi metrik secara unik.

Misalnya, Anda mungkin memiliki tanda pesan dan dimensi yang disebut `campaign` yang Anda gunakan untuk mengidentifikasi kampanye email Anda. Ketika Anda mempublikasikan acara pengiriman email ke CloudWatch, memilih tag dan dimensi pesan Anda penting karena pilihan ini memengaruhi CloudWatch penagihan Anda dan menentukan bagaimana Anda dapat memfilter data peristiwa pengiriman email Anda. CloudWatch

Bagian ini memberikan informasi untuk membantu Anda memilih dimensi, dan kemudian menunjukkan cara menambahkan tujuan CloudWatch acara ke set konfigurasi.

Topik di bagian ini

- Menambahkan Destinasi CloudWatch Acara
- Memilih CloudWatch Dimensi

Menambahkan Destinasi CloudWatch Acara

Prosedur di bagian ini menunjukkan cara menambahkan detail tujuan CloudWatch acara ke set konfigurasi dan mengasumsikan Anda telah menyelesaikan langkah 1 hingga 6 inci. Membuat tujuan acara

Anda juga dapat menggunakan operasi UpdateConfigurationSetEventTujuan di Amazon SES API V2 untuk membuat dan memodifikasi tujuan acara.

Untuk menambahkan detail tujuan CloudWatch acara ke set konfigurasi menggunakan konsol

1. Ini adalah petunjuk terperinci untuk memilih CloudWatch sebagai jenis tujuan acara Anda di Langkah 7 dan mengasumsikan Anda telah menyelesaikan semua langkah sebelumnya. Membuat tujuan acara Setelah memilih jenis CloudWatch Tujuan, memasukkan Nama tujuan, dan mengaktifkan penerbitan Acara, panel CloudWatch dimensi Amazon akan ditampilkan—bidangnya akan dibahas dalam langkah-langkah berikut. (Biaya tambahan berlaku, lihat Harga per metrik untuk CloudWatch.)
2. Untuk Sumber Nilai, tentukan bagaimana Amazon SES akan mendapatkan data yang diteruskan CloudWatch. Sumber nilai berikut tersedia:

    - Tanda Pesan – Amazon SES mengambil nama dan nilai dimensi dari tanda yang Anda tentukan dengan menggunakan header `X-SES-MESSAGE-TAGS` atau parameter API `EmailTags`. Untuk informasi selengkapnya tentang penggunaan tanda pesan, lihat the section called “Langkah 3: Tentukan set konfigurasi Anda saat pengiriman”.

> **Note**
>
> Tanda pesan dapat mencakup angka 0–9, huruf A–Z (huruf besar dan kecil), tanda hubung (-), dan garis bawah (_).

Anda juga dapat menggunakan sumber nilai Tanda Pesan untuk membuat dimensi berdasarkan tanda otomatis Amazon SES. Untuk menggunakan tanda otomatis, ketik nama lengkap dari tanda otomatis tersebut sebagai Nama Dimensi. Misalnya, untuk membuat dimensi berdasarkan tanda otomatis set konfigurasi, gunakan `ses:configuration-set` untuk Nama Dimensi, dan nama set konfigurasi untuk Nilai Default. Untuk daftar lengkap tanda otomatis, lihat Cara kerja penerbitan acara dengan set konfigurasi dan tag pesan.

- Header Email – Amazon SES mengambil nama dan nilai dimensi dari header di dalam email.

> **Note**
>
> Anda tidak dapat menggunakan header email berikut sebagai Nama Dimensi: `Received`, `To`, `From`, `DKIM-Signature`, `CC`, `message-id`, atau `Return-Path`.

- Tanda Tautan – Amazon SES mengambil nama dan nilai dimensi dari tanda yang Anda tentukan di tautan. Untuk informasi selengkapnya tentang menambahkan tanda ke tautan, lihat Dapatkah saya menandai tautan dengan pengenal unik?.

3. Untuk Nama Dimensi, ketik nama dimensi yang ingin Anda lewati CloudWatch.

> **Note**
>
> Nama dimensi hanya dapat berisi huruf ASCII (a-z, A-Z), angka (0-9), garis bawah (_), dan tanda hubung (-). Spasi, karakter beraksen, karakter non-Latin, dan karakter khusus lainnya tidak diperbolehkan.

4. Untuk Nilai default, ketik nilai dimensi.

> **Note**
>
> Nilai dimensi hanya dapat berisi huruf ASCII (a-z, A-Z), angka (0-9), garis bawah (_), tanda hubung (-), pada tanda (@), dan titik (.). Spasi, karakter beraksen, karakter non-Latin, dan karakter khusus lainnya tidak diperbolehkan.

5. Jika Anda ingin menambahkan dimensi lagi, pilih Tambahkan Dimensi. Jika tidak, pilih Selanjutnya.
6. Di layar tinjauan, jika Anda puas dengan cara Anda menentukan tujuan peristiwa, pilih Tambahkan tujuan.

Memilih CloudWatch Dimensi

Saat Anda memilih nama dan nilai untuk digunakan sebagai CloudWatch dimensi, pertimbangkan faktor-faktor berikut:

- Harga per metrik — Anda dapat melihat metrik dasar Amazon SES secara CloudWatch gratis. Namun, ketika Anda mengumpulkan metrik menggunakan penerbitan acara, Anda dikenakan biaya Pemantauan CloudWatch Terperinci. Setiap kombinasi unik dari jenis peristiwa, nama dimensi, dan nilai dimensi menciptakan metrik yang berbeda CloudWatch. Saat Anda menggunakan CloudWatch, Pemantauan Terperinci, Anda dikenakan biaya untuk setiap metrik. Karena alasan ini, Anda mungkin ingin menghindari memilih dimensi yang dapat menghasilkan banyak nilai berbeda. Misalnya, kecuali Anda sangat tertarik untuk melacak peristiwa pengiriman email dengan domain "Dari", Anda mungkin tidak ingin menentukan dimensi untuk tanda otomatis Amazon SES `ses:from-domain` karena dapat menghasilkan banyak nilai berbeda. Untuk informasi selengkapnya, silakan lihat Harga CloudWatch.
- Pemfilteran metrik — Jika metrik memiliki beberapa dimensi, Anda tidak dapat mengakses metrik CloudWatch berdasarkan setiap dimensi secara terpisah. Untuk alasan itu, pikirkan baik-baik sebelum Anda menambahkan lebih dari satu dimensi ke satu tujuan CloudWatch acara. Misalnya, jika Anda menginginkan metrik menurut `campaign` dan menurut kombinasi `campaign` dan `genre`, maka Anda perlu menambahkan dua tujuan peristiwa: satu hanya dengan `campaign` sebagai dimensi, dan satu dengan `campaign` dan `genre` sebagai dimensi.
- Sumber nilai dimensi – Sebagai alternatif untuk menentukan nilai dimensi Anda menggunakan header khusus Amazon SES atau parameter ke API, Anda juga dapat memilih Amazon SES untuk mengambil nilai dimensi dari header pesan MIME Anda sendiri. Anda dapat menggunakan opsi ini jika Anda sudah menggunakan header kustom dan tidak ingin mengubah email atau panggilan

Anda ke API pengiriman email untuk mengumpulkan metrik berdasarkan nilai header Anda. Jika Anda menggunakan header pesan MIME milik Anda sendiri untuk penerbitan peristiwa Amazon SES, maka nama dan nilai header yang Anda gunakan untuk penerbitan peristiwa Amazon SES hanya dapat menyertakan huruf A sampai Z, angka 0 sampai 9, garis bawah (_), tanda at (@), tanda hubung (-), dan titik (.). Jika Anda menentukan nama atau nilai yang berisi karakter lain, panggilan pengiriman email akan tetap berhasil, tetapi metrik acara tidak akan dikirim ke Amazon CloudWatch.

Untuk informasi selengkapnya tentang CloudWatch konsep, lihat CloudWatch Konsep Amazon di Panduan CloudWatch Pengguna Amazon.

Menyiapkan tujuan acara Firehose Data untuk penerbitan acara Amazon SES

Tujuan acara Amazon Data Firehose mewakili entitas yang menerbitkan peristiwa pengiriman email Amazon SES tertentu ke Firehose. Karena tujuan peristiwa Firehose hanya dapat diatur dalam set konfigurasi, Anda harus terlebih dahulu membuat set konfigurasi. Selanjutnya, Anda menambahkan tujuan peristiwa ke set konfigurasi.

Prosedur di bagian ini menunjukkan cara menambahkan detail tujuan acara Firehose ke set konfigurasi dan mengasumsikan Anda telah menyelesaikan langkah 1 hingga 6 inci. Membuat tujuan acara

Anda juga dapat menggunakan operasi UpdateConfigurationSetEventTujuan di tujuan Amazon SES API V2 untuk membuat dan memperbarui tujuan acara.

Untuk menambahkan detail tujuan acara Firehose ke set konfigurasi menggunakan konsol

1. Ini adalah petunjuk terperinci untuk memilih Firehose sebagai jenis tujuan acara Anda di Langkah 7 dan mengasumsikan Anda telah menyelesaikan semua langkah sebelumnya. Membuat tujuan acara Setelah memilih jenis Destinasi Firehose, memasukkan Nama tujuan, dan mengaktifkan penerbitan Acara, panel aliran pengiriman Amazon Data Firehose akan ditampilkan—bidangnya akan dibahas dalam langkah-langkah berikut.
2. Untuk aliran Pengiriman, pilih aliran pengiriman Firehose yang ada, atau pilih Buat aliran baru untuk membuat yang baru menggunakan Firehose console.

   Untuk informasi tentang membuat stream menggunakan Firehose console, lihat Membuat Aliran Pengiriman Amazon Kinesis Firehose di Panduan Pengembang Amazon Data Firehose.

3. Untuk Peran Identity and Access Management (IAM), pilih peran IAM yang izinnya dipublikasikan Amazon SES ke Firehose atas nama Anda. Anda dapat memilih peran yang sudah ada, meminta Amazon SES membuat peran untuk Anda, atau membuat peran Anda sendiri.

   Jika Anda memilih peran yang ada atau membuat peran Anda sendiri, Anda harus mengubah kebijakan peran secara manual untuk memberikan izin peran untuk mengakses aliran pengiriman Firehose, dan memberikan izin Amazon SES untuk mengambil peran tersebut. Untuk kebijakan-kebijakan contoh, lihat Memberikan Izin Amazon SES untuk Mempublikasikan ke Aliran Pengiriman Firehose Anda.
4. Pilih Selanjutnya.
5. Di layar tinjauan, jika Anda puas dengan cara Anda menentukan tujuan peristiwa, pilih Tambahkan tujuan.

Untuk informasi tentang cara menggunakan `UpdateConfigurationSetEventDestination` API untuk menambahkan tujuan peristiwa Firehose, lihat Referensi API Amazon Simple Email Service.

Memberikan Izin Amazon SES untuk Mempublikasikan ke Aliran Pengiriman Firehose Anda

Untuk mengaktifkan Amazon SES memublikasikan catatan ke aliran pengiriman Firehose, Anda harus menggunakan peran AWS Identity and Access Management (IAM) dan melampirkan atau memodifikasi kebijakan izin peran dan kebijakan kepercayaan. Kebijakan izin memungkinkan peran untuk memublikasikan catatan ke aliran pengiriman Firehose Anda, dan kebijakan kepercayaan memungkinkan Amazon SES untuk mengambil peran tersebut.

Bagian ini menyediakan contoh dari kedua kebijakan tersebut. Untuk informasi tentang pelampiran kebijakan untuk IAM role, lihat Mengubah Peran dalam Panduan Pengguna IAM.

Kebijakan Izin

Kebijakan izin berikut memungkinkan peran untuk memublikasikan catatan data ke aliran pengiriman Firehose Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
```

```
        "firehose:PutRecordBatch"
      ],
      "Resource": [
        "arn:aws:firehose:delivery-region:111122223333:deliverystream/delivery-stream-
name"
      ]
    }
  ]
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *wilayah pengiriman* dengan AWS Wilayah tempat Anda membuat aliran pengiriman Firehose.
- Ganti *111122223333* dengan ID akun AWS Anda.
- Ganti *nama aliran pengiriman dengan nama* aliran pengiriman Firehose.

Kebijakan Kepercayaan

Kebijakan kepercayaan berikut memungkinkan Amazon SES untuk mengambil peran.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ses.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "111122223333",
          "AWS:SourceArn": "arn:aws:ses:delivery-region:111122223333:configuration-
set/configuration-set-name"
        }
      }
    }
  ]
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *wilayah pengiriman* dengan AWS Wilayah tempat Anda membuat aliran pengiriman Firehose.
- Ganti *111122223333* dengan ID akun AWS Anda.
- Ganti *configuration-set-name* dengan nama set konfigurasi yang terkait dengan aliran pengiriman Firehose.

Siapkan EventBridge tujuan Amazon untuk penerbitan acara

Tujuan EventBridge acara Amazon memberi tahu Anda tentang peristiwa pengiriman email yang Anda tentukan dalam set konfigurasi. SESmenghasilkan dan mengirim peristiwa pengiriman email yang Anda tentukan saat membuat tujuan acara ke bus acara EventBridge default. Bus acara adalah router yang menerima acara dan dapat mengirimkannya ke beberapa tujuan. Anda dapat mempelajari lebih lanjut tentang mengintegrasikan acara pengiriman email dengan Amazon EventBridge diPemantauan menggunakan EventBridge. Karena tujuan EventBridge acara hanya dapat diatur dalam set konfigurasi, Anda harus membuat set konfigurasi sebelum menambahkan tujuan acara ke set konfigurasi.

Prosedur di bagian ini menunjukkan cara menambahkan detail tujuan EventBridge acara ke set konfigurasi dan mengasumsikan Anda telah menyelesaikan langkah 1 hingga 6 inci. Membuat tujuan acara

Anda juga dapat menggunakan UpdateConfigurationSetEventDestinationoperasi di Amazon SES API V2 untuk membuat dan memodifikasi tujuan acara.

Untuk menambahkan detail tujuan EventBridge acara ke set konfigurasi menggunakan konsol

1. Ini adalah petunjuk terperinci untuk memilih EventBridge sebagai jenis tujuan acara Anda di Langkah 7 dan mengasumsikan Anda telah menyelesaikan semua langkah sebelumnya. Membuat tujuan acara Setelah memilih jenis EventBridge Tujuan Amazon, memasukkan Nama tujuan, dan mengaktifkan penerbitan Acara, panel informasi bus EventBridge acara Amazon akan ditampilkan.
2. Pilih Berikutnya.
3. Di layar tinjauan, jika Anda puas dengan cara Anda menentukan tujuan peristiwa, pilih Tambahkan tujuan. Ini akan membuka halaman ringkasan tujuan acara di mana spanduk sukses akan mengonfirmasi apakah tujuan acara Anda berhasil dibuat atau diubah.

Siapkan tujuan acara Amazon Pinpoint untuk penerbitan acara

Tujuan acara Amazon Pinpoint memberi tahu Anda tentang peristiwa pengiriman email yang Anda tentukan dalam set konfigurasi. Karena tujuan peristiwa Amazon Pinpoint hanya dapat diatur dalam set konfigurasi, Anda harus membuat set konfigurasi sebelum menambahkan tujuan acara ke set konfigurasi.

Prosedur di bagian ini menunjukkan cara menambahkan detail tujuan peristiwa Amazon Pinpoint ke set konfigurasi dan mengasumsikan Anda telah menyelesaikan langkah 1 hingga 6 inci. Membuat tujuan acara

Anda juga dapat menggunakan operasi UpdateConfigurationSetEventTujuan di Amazon SES API V2 untuk membuat dan memodifikasi tujuan acara.

Ada biaya tambahan untuk jenis saluran yang telah Anda konfigurasikan dalam proyek Amazon Pinpoint Anda. Untuk informasi selengkapnya, lihat Harga Amazon Pinpoint.

Untuk menambahkan detail tujuan acara Amazon Pinpoint ke set konfigurasi menggunakan konsol

1. Ini adalah petunjuk terperinci untuk memilih Amazon Pinpoint sebagai jenis tujuan acara Anda di Langkah 7 dan mengasumsikan Anda telah menyelesaikan semua langkah sebelumnya. Membuat tujuan acara

   > **Note**
   >
   > Amazon Pinpoint tidak mendukung jenis acara Penundaan pengiriman atau Langganan.

   Setelah memilih jenis Tujuan Amazon Pinpoint, memasukkan Nama tujuan, dan mengaktifkan penerbitan Acara, panel detail proyek Amazon Pinpoint akan ditampilkan—bidangnya akan dibahas dalam langkah-langkah berikut.

2. Untuk Project, pilih proyek Amazon Pinpoint yang ada, atau pilih Buat proyek baru di Amazon Pinpoint untuk membuat yang baru.

   Untuk informasi tentang membuat proyek, lihat Membuat proyek di Panduan Pengguna Amazon Pinpoint.

3. Pilih Selanjutnya.

4. Di layar tinjauan, jika Anda puas dengan cara Anda menentukan tujuan peristiwa, pilih Tambahkan tujuan. Ini akan membuka halaman ringkasan tujuan acara di mana spanduk sukses akan mengonfirmasi apakah tujuan acara Anda berhasil dibuat atau diubah.

Siapkan tujuan peristiwa Amazon SNS untuk penerbitan peristiwa

Tujuan acara Amazon SNS memberi tahu Anda tentang peristiwa pengiriman email yang Anda tentukan dalam set konfigurasi. Karena tujuan acara Amazon SNS hanya dapat diatur dalam set konfigurasi, Anda harus membuat set konfigurasi sebelum menambahkan tujuan acara ke set konfigurasi.

Prosedur di bagian ini menunjukkan cara menambahkan detail tujuan acara Amazon SNS ke set konfigurasi dan mengasumsikan Anda telah menyelesaikan langkah 1 hingga 6 inci. Membuat tujuan acara

Anda juga dapat menggunakan operasi UpdateConfigurationSetEventTujuan di Amazon SES API V2 untuk membuat dan memodifikasi tujuan acara.

> **Note**
>
> Pemberitahuan umpan balik untuk bouncing, keluhan, dan pengiriman juga dapat diatur melalui Amazon SNS untuk identitas pengiriman terverifikasi Anda. Untuk informasi lebih lanjut, lihat. the section called “Mengonfigurasi notifikasi Amazon SNS”.

Ada biaya tambahan untuk mengirim pesan ke titik akhir yang berlangganan ke topik Amazon SNS Anda. Untuk informasi lebih lanjut, lihat Harga Amazon SNS.

Untuk menambahkan detail tujuan peristiwa Amazon SNS ke set konfigurasi menggunakan konsol

1. Ini adalah petunjuk terperinci untuk memilih Amazon SNS sebagai jenis tujuan acara Anda di Langkah 7 dan mengasumsikan Anda telah menyelesaikan semua langkah sebelumnya. Membuat tujuan acara Setelah memilih jenis Tujuan Amazon SNS, memasukkan Nama tujuan, dan mengaktifkan penerbitan Acara, panel topik Amazon Simple Notification Service (SNS) akan ditampilkan—bidangnya akan dibahas dalam langkah-langkah berikut.
2. Untuk Topik SNS, pilih topik Amazon SNS yang sudah ada, atau pilih Buat topik SNS untuk membuat topik baru.

Untuk informasi selengkapnya, lihat Buat topik dalam Panduan Developer Amazon Simple Notification Service.

> **Important**
>
> Saat Anda membuat topik menggunakan Amazon SNS, untuk Jenis, pilih saja Standar. (SES tidak mendukung topik tipe FIFO.)

3. Pilih Selanjutnya.
4. Di layar tinjauan, jika Anda puas dengan cara Anda menentukan tujuan peristiwa, pilih Tambahkan tujuan. Ini akan membuka halaman ringkasan tujuan acara di mana spanduk sukses akan mengonfirmasi apakah tujuan acara Anda berhasil dibuat atau diubah.
5. Apakah Anda membuat topik SNS baru atau memilih yang sudah ada, Anda sekarang perlu memberikan akses ke SES untuk mempublikasikan pemberitahuan ke topik tersebut. Pada halaman ringkasan tujuan acara dari langkah sebelumnya, pilih Amazon SNS dari kolom Jenis tujuan - ini akan membawa Anda ke daftar Topik di konsol Layanan Pemberitahuan Sederhana Amazon - lakukan langkah-langkah berikut dari konsol Amazon SNS:

   a. Pilih nama topik SNS yang Anda buat atau modifikasi pada langkah sebelumnya.

   b. Pada layar detail topik, pilih Edit.

   c. Untuk memberikan izin SES untuk mempublikasikan pemberitahuan ke topik, pada layar Edit topik di konsol SNS, perluas kebijakan Access dan di editor JSON, tambahkan kebijakan izin berikut:

```
{
  "Version": "2012-10-17",
  "Id": "notification-policy",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ses.amazonaws.com"
      },
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:topic_region:111122223333:topic_name",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "111122223333",
```

```
          "AWS:SourceArn":
 "arn:aws:ses:topic_region:111122223333:configuration-set/configuration-set-
name"
        }
      }
    }
  ]
}
```

Buat perubahan berikut ke contoh kebijakan sebelumnya:

- Ganti *topic_region dengan AWS Wilayah* tempat Anda membuat topik SNS.
- Ganti *111122223333* dengan ID akun Anda. AWS
- Ganti *topic_name dengan nama* topik SNS Anda.
- Ganti *configuration-set-name dengan nama* set konfigurasi Anda yang terkait dengan tujuan acara SNS.

d. Pilih Simpan perubahan.

Langkah 3: Tentukan set konfigurasi saat Anda mengirim email

Setelah Anda buat satu set konfigurasi dan tambahkan tujuan peristiwa, langkah terakhir untuk penerbitan peristiwa adalah dengan mengirim email Anda.

Untuk menerbitkan peristiwa yang terkait dengan email, Anda harus menyediakan nama konfigurasi yang diatur untuk dikaitkan dengan email. Secara opsional, Anda dapat memberikan tanda pesan untuk mengategorikan email.

Anda menyediakan informasi ini ke Amazon SES sebagai parameter untuk API pengiriman email, header email khusus Amazon SES, atau header kustom dalam pesan MIME Anda. Metode yang Anda pilih tergantung pada antarmuka pengiriman email yang Anda gunakan, seperti yang ditunjukkan pada tabel berikut.

Antarmuka Pengiriman Email	Cara Menerbitkan Peristiwa
`SendEmail`	Parameter API:
`SendTemplatedEmail`	Parameter API:

Antarmuka Pengiriman Email	Cara Menerbitkan Peristiwa
`SendBulkTemplatedEmail`	Parameter API:
`SendCustomVerificationEmail`	Parameter API:
`SendRawEmail`	Parameter API, header email khusus Amazon SES, atau header MIME kustom **Important** Jika Anda menentukan tanda pesan menggunakan header dan parameter API, Amazon SES hanya menggunakan tanda pesan yang disediakan oleh parameter API. Amazon SES tidak menggabungkan tanda pesan yang ditentukan oleh parameter dan header API.
Antarmuka SMTP	Header email khusus Amazon SES

Bagian berikut menjelaskan cara menentukan set konfigurasi dan tanda pesan menggunakan header dan menggunakan parameter API.

- Menggunakan Parameter API Amazon SES
- Menggunakan Header Email Khusus Amazon SES
- Menggunakan Header Email Kustom

Note

Anda dapat secara opsional menyertakan tanda pesan di header email Anda. Tanda pesan dapat mencakup angka 0–9, huruf A–Z (huruf besar dan kecil), tanda hubung (-), dan garis bawah (_).

Menggunakan Parameter API Amazon SES

Untuk menggunakan SendEmail, SendTemplatedEmail, SendBulkTemplatedEmail, SendCustomVerificationEmail, atau SendRawEmaildengan penerbitan peristiwa, Anda menentukan set konfigurasi dan tag pesan dengan meneruskan struktur data yang dipanggil ConfigurationSetdan MessageTagke panggilan API.

Untuk informasi selengkapnya tentang cara menggunakan API Amazon SES, lihat Referensi API Amazon Simple Email Service.

Menggunakan Header Email Khusus Amazon SES

Saat Anda menggunakan `SendRawEmail` atau antarmuka SMTP, Anda dapat menentukan set konfigurasi dan tanda pesan dengan menambahkan header khusus Amazon SES ke email. Amazon SES menghapus header sebelum mengirim email. Tabel berikut menunjukkan nama-nama header yang akan digunakan.

Informasi Penerbitan Peristiwa	Header
Set konfigurasi	`X-SES-CONFIGURATION-SET`
Tanda pesan	`X-SES-MESSAGE-TAGS`

Contoh berikut menunjukkan bagaimana header dapat terlihat dalam email mentah yang Anda kirimkan ke Amazon SES.

```
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--
```

Menggunakan Header Email Kustom

Meskipun Anda harus menentukan nama set konfigurasi menggunakan header khusus Amazon SES `X-SES-CONFIGURATION-SET`, Anda dapat menentukan tanda pesan dengan menggunakan header MIME Anda sendiri.

> **Note**
>
> Nama dan nilai header yang Anda gunakan untuk penerbitan peristiwa Amazon SES harus dalam ASCII. Jika Anda menentukan nama atau nilai header non-ASCII untuk penerbitan peristiwa Amazon SES, maka panggilan pengiriman email akan tetap berhasil, tetapi metrik peristiwa tidak akan dipancarkan ke Amazon. CloudWatch

Bekerja dengan data peristiwa Amazon SES

Setelah Anda menyiapkan penerbitan peristiwa dan menentukan set konfigurasi untuk mengirim email, Anda dapat mengambil peristiwa pengiriman email dari tujuan peristiwa yang Anda tentukan saat menyiapkan set konfigurasi yang terkait dengan email.

Bagian ini menjelaskan cara mengambil peristiwa pengiriman email Anda dari Amazon CloudWatch dan Amazon Data Firehose, dan cara menafsirkan data peristiwa yang disediakan oleh Amazon SNS.

- Mengambil data acara Amazon SES dari CloudWatch
- Mengambil data SES peristiwa Amazon dari Firehose
- Menafsirkan data peristiwa Amazon SES dari Amazon SNS

Mengambil data acara Amazon SES dari CloudWatch

Amazon SES dapat mempublikasikan metrik untuk acara pengiriman email Anda ke Amazon CloudWatch. Saat Anda mempublikasikan data peristiwa CloudWatch, data ini menyediakan metrik ini sebagai kumpulan data deret waktu yang diurutkan. Anda dapat menggunakan metrik ini untuk

memantau performa pengiriman email Anda. Misalnya, Anda dapat memantau metrik keluhan dan mengatur CloudWatch alarm untuk memicu ketika metrik melebihi nilai tertentu.

Ada dua tingkat perincian di mana Amazon SES dapat mempublikasikan acara ini ke: CloudWatch

- Di seluruh Anda Akun AWS — Metrik kasar ini, yang sesuai dengan metrik yang Anda pantau menggunakan konsol Amazon SES dan `GetSendStatistics` API, adalah total di seluruh Anda. Akun AWS Amazon SES menerbitkan metrik ini secara otomatis. CloudWatch
- Detail – Metrik ini dikategorikan berdasarkan karakteristik email yang Anda tentukan menggunakan tanda pesan. Untuk memublikasikan metrik ini CloudWatch, Anda harus menyiapkan penerbitan CloudWatch acara dengan tujuan acara dan menentukan set konfigurasi saat mengirim email. Anda juga dapat menentukan tanda pesan atau menggunakan tanda otomatis yang disediakan oleh Amazon SES secara otomatis.

Bagian ini menjelaskan metrik yang tersedia dan cara melihat metrik di. CloudWatch

Metrik yang Tersedia

Anda dapat mempublikasikan metrik pengiriman email Amazon SES berikut ke CloudWatch:

- Kirim — Permintaan kirim berhasil dan Amazon SES akan mencoba mengirimkan pesan ke server email penerima. (Jika tingkat akun atau penekanan global sedang digunakan, SES masih akan menghitungnya sebagai kirim, tetapi pengiriman ditekan.)
- RenderingFailure— Email tidak dikirim karena masalah rendering template. Tipe peristiwa ini dapat terjadi saat data templat tidak ada, atau jika ada ketidakcocokan antara parameter templat dan data. (Tipe peristiwa ini hanya terjadi ketika Anda mengirim email menggunakan operasi API `SendTemplatedEmail` atau `SendBulkTemplatedEmail`.)
- Tolak — Amazon SES menerima email tersebut, tetapi memutuskan bahwa email tersebut berisi virus dan tidak berusaha mengirimkannya ke server email penerima.
- Pengiriman - Amazon SES berhasil mengirimkan email ke server email penerima.
- Bounce — Pantulan keras yang server email penerima menolak email secara permanen. (Pantulan lunak hanya disertakan ketika SES tidak lagi mencoba mengirimkan email. Umumnya pantulan lunak ini menunjukkan kegagalan pengiriman, meskipun dalam beberapa kasus pantulan lunak dapat dikembalikan bahkan ketika surat berhasil mencapai kotak masuk penerima. Ini biasanya terjadi ketika penerima mengirim balasan out-of-office otomatis. Pelajari lebih lanjut tentang soft bounce di artikel AWS re:Post ini.)

- Keluhan — Email berhasil dikirim ke server email penerima, tetapi penerima menandainya sebagai spam.
- DeliveryDelay— Email tidak dapat dikirim ke server email penerima karena masalah sementara terjadi. Penundaan penyamapaian dapat terjadi, misalnya, saat kotak masuk penerima penuh, atau saat server email penerima mengalami masalah sementara.
- Langganan — Email berhasil dikirim, tetapi penerima memperbarui preferensi langganan dengan mengklik `List-Unsubscribe` header email atau `Unsubscribe` tautan di footer.
- Buka — Penerima menerima pesan dan membukanya di klien email mereka.
- Klik — Penerima mengklik satu atau beberapa tautan di email.

Dimensi yang Tersedia

CloudWatch menggunakan nama dimensi yang Anda tentukan saat menambahkan tujuan CloudWatch acara ke set konfigurasi di Amazon SES. Untuk informasi selengkapnya, lihat Menyiapkan tujuan CloudWatch acara untuk penerbitan acara.

Melihat Metrik Amazon SES di Konsol CloudWatch

Prosedur berikut menjelaskan cara melihat metrik penerbitan acara Amazon SES Anda menggunakan CloudWatch konsol.

Untuk melihat metrik menggunakan konsol CloudWatch

1. Masuk ke AWS Management Console dan buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Jika perlu, ubah wilayahnya. Dari bilah navigasi, pilih wilayah tempat AWS sumber daya Anda berada. Untuk informasi selengkapnya, lihat Wilayah dan Titik Akhir.
3. Di panel navigasi, pilih Semua Metrik.
4. Di panel Metrik, pilih SES.
5. Pilih metrik yang ingin Anda lihat. Untuk melihat metrik penerbitan acara berbutir halus, pilih kombinasi dimensi yang Anda tentukan saat menyiapkan tujuan acara. CloudWatch Untuk mempelajari selengkapnya tentang melihat metrik dengan CloudWatch, lihat Menggunakan CloudWatch metrik Amazon.

Untuk melihat metrik menggunakan AWS CLI

- Pada prompt perintah, gunakan perintah berikut:

```
aws cloudwatch list-metrics --namespace "AWS/SES"
```

Mengambil data SES peristiwa Amazon dari Firehose

Amazon SES menerbitkan acara pengiriman email ke Firehose JSON sebagai catatan. Firehose kemudian menerbitkan catatan ke tujuan AWS layanan yang Anda pilih saat mengatur aliran pengiriman di Firehose. Untuk informasi tentang menyiapkan aliran pengiriman Firehose, lihat Membuat Aliran Pengiriman Firehose di Panduan Pengembang Firehose Data Amazon.

Topik di bagian ini:

- Isi data peristiwa yang SES diterbitkan Amazon ke Firehose
- Contoh data peristiwa yang SES diterbitkan Amazon ke Firehose

Isi data peristiwa yang SES diterbitkan Amazon ke Firehose

Amazon SES menerbitkan catatan peristiwa pengiriman email ke Amazon Data Firehose JSON dalam format. Saat menerbitkan acara ke Firehose, Amazon SES mengikuti setiap JSON rekaman dengan karakter baris baru.

Anda dapat menemukan contoh catatan untuk semua tipe notifikasi ini di Contoh data peristiwa yang SES diterbitkan Amazon ke Firehose.

Topik di bagian ini

- Objek tingkat atas JSON
- Objek surat
- Objek pentalan
- Objek aduan
- Objek penyampaian
- Kirim objek
- Tolak objek
- Buka objek
- Klik objek
- Objek Kegagalan Rendering
- DeliveryDelay objek

- Objek berlangganan

Objek tingkat atas JSON

JSONObjek tingkat atas dalam catatan peristiwa pengiriman email berisi bidang berikut.

Nama Bidang	Deskripsi
`eventType`	String yang menjelaskan tipe peristiwa. Nilai yang mungkin: `BounceComplaint` ,`Delivery`,`Send`,,`Reject`,`Open`,`Click`,`Rendering Failure`,`DeliveryDelay` , atau`Subscription` . Jika Anda tidak menyiapkan penerbitan acara, bidang ini diberi nama`notificationType` .
`mail`	JSONObjek yang berisi informasi tentang email yang menghasilkan acara tersebut.
`bounce`	Bidang ini hanya ada jika `eventType` adalah `Bounce`. Bidang ini berisi informasi tentang pentalan.
`complaint`	Bidang ini hanya ada jika `eventType` adalah `Complaint` . Bidang ini berisi informasi tentang aduan.
`delivery`	Bidang ini hanya ada jika `eventType` adalah `Delivery`. Bidang ini berisi informasi tentang penyampaian.
`send`	Bidang ini hanya ada jika `eventType` adalah `Send`.
`reject`	Bidang ini hanya ada jika `eventType` adalah `Reject`. Bidang ini berisi informasi tentang penolakan.

Nama Bidang	Deskripsi
`open`	Bidang ini hanya ada jika `eventType` adalah `Open`. Bidang ini berisi informasi tentang peristiwa pembukaan.
`click`	Bidang ini hanya ada jika `eventType` adalah `Click`. Bidang ini berisi informasi tentang peristiwa pengeklikan.
`failure`	Bidang ini hanya ada jika `eventType` adalah `Rendering Failure` . Bidang ini berisi informasi tentang peristiwa kegagalan rendering.
`deliveryDelay`	Bidang ini hanya ada jika `eventType` adalah `DeliveryDelay` . Bidang tersebut berisi informasi tentang penyampaian email yang tertunda.
`subscription`	Bidang ini hanya ada jika `eventType` adalah `Subscription` . Ini berisi informasi tentang preferensi berlangganan.

Objek surat

Setiap catatan peristiwa pengiriman email berisi informasi tentang email asli di dalam objek `mail`. JSONObjek yang berisi informasi tentang suatu `mail` objek memiliki bidang berikut.

Nama Bidang	Deskripsi
`timestamp`	Tanggal dan waktu, dalam format ISO86 01 (YYYY-MM-:MM: DDThh SS.sz), saat pesan dikirim.
`messageId`	ID unik yang SES ditetapkan Amazon ke pesan. Amazon SES mengembalikan nilai ini kepada Anda saat Anda mengirim pesan.

Nama Bidang	Deskripsi
	Note ID pesan ini ditetapkan oleh AmazonSES. Anda dapat menemukan ID pesan email asli di bidang `headers` dan `commonHeaders` dari objek `mail`.
`source`	Alamat email tempat pesan dikirim (MAILFROM alamat amplop).
`sourceArn`	Nama Sumber Daya Amazon (ARN) dari identitas yang digunakan untuk mengirim email. Dalam hal mengirim otorisasi, itu `sourceArn` adalah identitas yang ARN pemilik identitas mengizinkan pengirim delegasi untuk digunakan untuk mengirim email. Untuk informasi selengkapnya tentang otorisasi pengiriman, lihat Metode autentikasi email.
`sendingAccountId`	ID AWS akun akun yang digunakan untuk mengirim email. Dalam hal otorisasi pengirima n, `sendingAccountId` adalah ID akun pengirim delegasi.
`destination`	Daftar alamat email yang merupakan penerima email asli.
`headersTruncated`	String yang menentukan jika header terpotong atau tidak di dalam notifikasi, yang terjadi jika header lebih besar dari 10 KB. Nilai yang mungkin adalah `true` dan `false`.

Nama Bidang	Deskripsi
`headers`	Daftar header asli email. Setiap header dalam daftar memiliki bidang `name` dan bidang `value`. **Note** ID pesan apa pun dalam `headers` bidang ini berasal dari pesan asli yang Anda kirimkan ke AmazonSES. ID pesan yang SES kemudian ditetapkan Amazon ke pesan ada di `messageId` bidang `mail` objek.
`commonHeaders`	Pemetaan header asli email yang umum digunakan. **Note** ID pesan apa pun dalam `commonHeaders` bidang adalah ID pesan yang SES kemudian ditetapkan Amazon ke pesan di `messageId` bidang `mail` objek.
`tags`	Daftar tag yang terkait dengan email.

Objek pentalan

JSONObjek yang berisi informasi tentang suatu `Bounce` peristiwa akan selalu memiliki bidang berikut.

Nama Bidang	Deskripsi
`bounceType`	Jenis bouncing, seperti yang ditentukan oleh AmazonSES.

Nama Bidang	Deskripsi
`bounceSubType`	Subtipe pantulan, sebagaimana ditentukan oleh Amazon. SES
`bouncedRecipients`	Daftar yang berisi informasi tentang penerima email asli yang terpental.
`timestamp`	Tanggal dan waktu, dalam format ISO86 01 (YYYY-MM-:MM: SS.sz DDThh), ketika mengirim pemberitahuan bouncing. ISP
`feedbackId`	ID unik untuk pentalan.
`reportingMTA`	Nilai `Reporting-MTA` bidang dariDSN. Ini adalah nilai dari Message Transfer Authority (MTA) yang mencoba untuk melakukan pengiriman, relay, atau operasi gateway yang dijelaskan dalamDSN. **Note** Bidang ini hanya muncul jika pemberita huan status pengiriman (DSN) dilampirkan ke bouncing.

Penerima yang terpental

Peristiwa pentalan mungkin berkaitan dengan satu atau beberapa penerima. Bidang `bouncedRecipients` menyimpan daftar objek—satu objek per penerima yang terkait dengan peristiwa pentalan—dan akan selalu berisi bidang berikut.

Nama Bidang	Deskripsi
`emailAddress`	Alamat email penerima. Jika DSN tersedia, ini adalah nilai `Final-Recipient` bidang dariDSN.

Secara opsional, jika a DSN dilampirkan ke pantulan, bidang berikut mungkin juga ada.

Nama Bidang	Deskripsi
`action`	Nilai `Action` bidang dariDSN. Ini menunjukkan tindakan yang dilakukan oleh pelaporan MTA sebagai hasil dari upayanya untuk menyampai kan pesan kepada penerima ini.
`status`	Nilai `Status` bidang dariDSN. Ini adalah kode status bebas-transportasi per penerima yang menunjukkan status penyampaian pesan.
`diagnosticCode`	Kode status yang dikeluarkan oleh pelaporan MTA. Ini adalah nilai `Diagnostic-Code` bidang dariDSN. Bidang ini mungkin tidak ada di DSN (dan karena itu juga tidak ada diJSON).

Tipe pentalan

Setiap peristiwa pentalan akan menjadi salah satu tipe yang ditunjukkan pada tabel berikut.

Sistem penerbitan acara hanya menerbitkan hard bounce dan soft bounce yang tidak akan lagi dicoba lagi oleh Amazon. SES Ketika Anda menerima bounces yang ditandai`Permanent`, Anda harus menghapus alamat email yang sesuai dari milis Anda; Anda tidak akan dapat mengirim kepada mereka di masa depan. `Transient`Pantulan dikirimkan kepada Anda ketika pesan telah memantul beberapa kali, dan Amazon SES telah berhenti mencoba mengirimkannya kembali. Anda mungkin akan berhasil mengirim ulang ke alamat yang awalnya menghasilkan pentalan `Transient` lain kali.

bounceType	bounceSubType	Deskripsi
`Undetermined`	`Undetermined`	Amazon SES tidak dapat menentukan alasan pentalan tertentu.
`Permanent`	`General`	Amazon SES menerima pantulan keras umum. Jika Anda menerima tipe pentalan ini, maka Anda harus menghapus alamat email penerima dari daftar email Anda.

bounceType	bounceSubType	Deskripsi
`Permanent`	`NoEmail`	Amazon SES menerima hard bounce permanen karena alamat email target tidak ada. Jika Anda menerima tipe pentalan ini, maka Anda harus menghapus alamat email penerima dari daftar email Anda.
`Permanent`	`Suppressed`	Amazon SES telah menekan pengiriman ke alamat ini karena memiliki riwayat memantul baru-baru ini sebagai alamat yang tidak valid. Untuk mengganti daftar penindasan global, lihat. Menggunakan daftar SES penindasan tingkat akun Amazon
`Permanent`	`OnAccountSuppressionList`	Amazon SES telah menekan pengiriman ke alamat ini karena ada di daftar penindasan tingkat akun. Ini tidak dihitung terhadap metrik rasio pentalan Anda.
`Transient`	`General`	Amazon SES menerima pantulan umum. Anda mungkin akan berhasil mengirim ke penerima ini lain kali.
`Transient`	`MailboxFull`	Amazon SES menerima bouncing penuh kotak surat. Anda mungkin akan berhasil mengirim ke penerima ini lain kali.
`Transient`	`MessageTooLarge`	Amazon SES menerima pesan pantulan yang terlalu besar. Anda mungkin akan berhasil mengirim ke penerima ini jika Anda mengurangi ukuran pesan.
`Transient`	`ContentRejected`	Amazon SES menerima bouncing konten yang ditolak. Anda mungkin akan berhasil mengirim ke penerima ini jika Anda mengubah konten pesan.

bounceType	bounceSubType	Deskripsi
`Transient`	`AttachmentRejected`	Amazon SES menerima bouncing lampiran yang ditolak. Anda mungkin akan berhasil mengirim ke penerima ini jika Anda menghapus atau mengubah lampiran.

Objek aduan

JSONObjek yang berisi informasi tentang suatu `Complaint` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`complainedRecipients`	Daftar yang berisi informasi tentang penerima yang mungkin telah mengirimkan aduan.
`timestamp`	Tanggal dan waktu, dalam format ISO86 01 (YYYY-MM-:MM: SS.sz DDThh), ketika mengirim pemberitahuan keluhan. ISP
`feedbackId`	ID unik untuk aduan.
`complaintSubType`	Subtipe keluhan, sebagaimana ditentukan oleh AmazonSES.

Selanjutnya, jika laporan umpan balik dilampirkan pada aduan, bidang berikut mungkin ada.

Nama Bidang	Deskripsi
`userAgent`	Nilai bidang `User-Agent` dari laporan umpan balik. Nilai ini menunjukkan nama dan versi sistem yang menghasilkan laporan.
`complaintFeedbackType`	Nilai `Feedback-Type` bidang dari laporan umpan balik yang diterima dariISP. Ini berisi tipe umpan balik.

Nama Bidang	Deskripsi
`arrivalDate`	Nilai `Arrival-Date` atau `Received-Date` bidang dari laporan umpan balik dalam format ISO86 01 (YYYY-MM DDThh -:MM: SS.sz). Bidang ini mungkin tidak ada dalam laporan (dan karena itu juga tidak ada dalamJSON).

Penerima yang diadukan

Bidang `complainedRecipients` berisi daftar penerima yang mungkin telah mengirimkan aduan.

> **Important**
>
> Karena sebagian besar ISPs menyunting alamat email penerima yang mengajukan keluhan dari pemberitahuan keluhan mereka, daftar ini berisi informasi tentang penerima yang mungkin telah mengirim keluhan, berdasarkan penerima pesan asli dan ISP dari mana kami menerima keluhan. Amazon SES melakukan pencarian terhadap pesan asli untuk menentukan daftar penerima ini.

JSONobjek dalam daftar ini berisi bidang berikut.

Nama Bidang	Deskripsi
`emailAddress`	Alamat email penerima.

Tipe aduan

Anda dapat melihat jenis keluhan berikut di `complaintFeedbackType` bidang yang ditetapkan oleh pelaporanISP, menurut situs web Internet Assigned Numbers Authority:

Nama Bidang	Deskripsi
`abuse`	Menunjukkan email yang tidak diminta atau beberapa jenis penyalahgunaan email lainnya.

Nama Bidang	Deskripsi
`auth-failure`	Laporan kegagalan otentikasi email.
`fraud`	Menunjukkan beberapa jenis penipuan atau aktivitas phishing.
`not-spam`	Menunjukkan bahwa entitas yang menyediak an laporan tidak menganggap pesan tersebut sebagai spam. Tindakan ini dapat digunakan untuk memperbaiki pesan yang salah ditandai atau dikategorikan sebagai spam.
`other`	Menunjukkan umpan balik lain yang tidak sesuai dengan tipe terdaftar lainnya.
`virus`	Melaporkan bahwa virus ditemukan dalam pesan asal.

Objek penyampaian

JSONObjek yang berisi informasi tentang suatu `Delivery` peristiwa akan selalu memiliki bidang berikut.

Nama Bidang	Deskripsi
`timestamp`	Tanggal dan waktu Amazon SES mengirimkan email ke server email penerima, dalam format ISO86 01 (YYYY-MM DDThh -:MM: SS.sz).
`processingTimeMillis`	Waktu dalam milidetik antara saat Amazon SES menerima permintaan dari pengirim hingga saat Amazon SES meneruskan pesan ke server email penerima.
`recipients`	Daftar penerima yang dituju yang berlaku untuk peristiwa penyampaian.

Nama Bidang	Deskripsi
`smtpResponse`	Pesan SMTP respons dari remote ISP yang menerima email dari AmazonSES. Pesan ini akan bervariasi menurut email, dengan menerima server email, dan dengan menerimal SP.
`reportingMTA`	Nama host dari server SES email Amazon yang mengirim email.

Kirim objek

JSONObjek yang berisi informasi tentang suatu `send` peristiwa selalu kosong.

Tolak objek

JSONObjek yang berisi informasi tentang suatu `Reject` peristiwa akan selalu memiliki bidang berikut.

Nama Bidang	Deskripsi
`reason`	Alasan email ditolak. Satu-satunya nilai yang mungkin adalah`Bad content`, yang berarti Amazon SES mendeteksi bahwa email tersebut mengandung virus. Ketika pesan ditolak, Amazon SES berhenti memprosesnya, dan tidak mencoba mengirimkannya ke server email penerima.

Buka objek

JSONObjek yang berisi informasi tentang suatu `Open` peristiwa akan selalu berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`ipAddress`	Alamat IP penerima.
`timestamp`	Tanggal dan waktu ketika peristiwa terbuka terjadi dalam format ISO86 01 (YYYY-MM DDThh -:MM: SS.sz).
`userAgent`	Agen pengguna perangkat atau klien email yang digunakan penerima untuk membuka email.

Klik objek

JSONObjek yang berisi informasi tentang suatu `Click` peristiwa akan selalu berisi bidang-bidang berikut.

Nama Bidang	Deskripsi
`ipAddress`	Alamat IP penerima.
`timestamp`	Tanggal dan waktu ketika peristiwa klik terjadi dalam format ISO86 01 (YYYY-MM DDThh -:MM: SS.sz).
`userAgent`	Agen pengguna klien yang digunakan penerima untuk mengeklik tautan di dalam email.
`link`	URLTautan yang diklik penerima.
`linkTags`	Daftar tanda yang ditambahkan ke tautan menggunakan atribut `ses:tags`. Untuk informasi selengkapnya tentang menambahk an tanda ke tautan di email Anda, lihat T5. Dapatkah saya menandai tautan dengan pengenal unik? dalam FAQ metrik pengiriman email Amazon SES.

Objek Kegagalan Rendering

JSONObjek yang berisi informasi tentang suatu `Rendering Failure` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`templateName`	Nama templat yang digunakan untuk mengirim email.
`errorMessage`	Pesan yang menyediakan informasi selengkap nya tentang kegagalan rendering.

DeliveryDelay objek

JSONObjek yang berisi informasi tentang suatu `DeliveryDelay` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`delayType`	Tipe penundaan. Kemungkinan nilai adalah: • InternalFailure— SES Masalah internal Amazon menyebabkan pesan tertunda. • Umum — Kegagalan umum terjadi selama SMTP percakapan. • MailboxFull— Kotak pesan penerima penuh dan tidak dapat menerima pesan tambahan. • SpamDetected— Server email penerima telah mendeteksi sejumlah besar email yang tidak diminta dari akun Anda. • RecipientServerError— Masalah sementara dengan server email penerima mencegah pengiriman pesan. • IPFailureAlamat IP yang mengirim pesan sedang diblokir atau dibatasi oleh penyedia email penerima.

Nama Bidang	Deskripsi
	• TransientCommunicationFailure— Ada kegagalan komunikasi sementara selama SMTP percakapan dengan penyedia email penerima. • BYOIPHostNameLookupUnavailable— Amazon SES tidak dapat mencari DNS nama host untuk alamat IP Anda. Tipe penundaan ini hanya terjadi ketika Anda menggunakan Bawa IP Anda Sendiri. • Belum ditentukan - Amazon SES tidak dapat menentukan alasan keterlambatan pengirima n. • SendingDeferral— Amazon SES telah menganggap pantas untuk menunda pesan secara internal.
`delayedRecipients`	Objek yang berisi informasi tentang penerima email.
`expirationTime`	Tanggal dan waktu ketika Amazon SES akan berhenti mencoba menyampaikan pesan. Nilai ini ditunjukkan dalam format ISO 8601.
`reportingMTA`	Alamat IP dari Message Transfer Agent (MTA) yang melaporkan keterlambatan.
`timestamp`	Tanggal dan waktu ketika penundaan terjadi, ditampilkan dalam format ISO 8601.

Penerima tertunda

Objek `delayedRecipients` berisi nilai-nilai berikut:

Nama Bidang	Deskripsi
`emailAddress`	Alamat email yang mengakibatkan penyampai an pesan tertunda.
`status`	Kode SMTP status yang terkait dengan penundaan pengiriman.
`diagnosticCode`	Kode diagnostik yang disediakan oleh Agen Transfer Pesan penerima (MTA).

Objek berlangganan

JSONObjek yang berisi informasi tentang suatu `Subscription` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`contactList`	Nama daftar kontak berada.
`timestamp`	Tanggal dan waktu, dalam format ISO86 01 (YYYY-MM-:MM: SS.sz DDThh), ketika mengirim pemberitahuan berlangganan. ISP
`source`	Alamat email tempat pesan dikirim (MAILFROM alamat amplop).
`newTopicPreferences`	JSONStruktur data (peta) yang menentukan status langganan semua topik dalam daftar kontak yang menunjukkan status setelah perubahan (kontak berlangganan atau berhenti berlangganan).
`oldTopicPreferences`	JSONStruktur data (peta) yang menentukan status langganan semua topik dalam daftar kontak yang menunjukkan status sebelum perubahan (kontak berlangganan atau berhenti berlangganan).

Preferensi topik baru/lama

`oldTopicPreferences`Objek `newTopicPreferences` dan berisi nilai-nilai berikut.

Nama Bidang	Deskripsi
`unsubscribeAll`	Menentukan apakah kontak berhenti berlangga nan dari semua topik dalam daftar kontak.
`topicSubscriptionStatus`	Menentukan status langganan topik di `topicName` bidang yang menunjukkan apakah saat ini berlangganan untuk menerima pemberitahuan dari SES untuk jenis acara yang ditentukan. Nilai yang mungkin OptIn(berlangg anan) atau OptOut(berhenti berlangganan) di bidang. `subscriptionStatus`
`topicDefaultSubscriptionStatus`	Menentukan status langganan default topik di `topicName` bidang menentukan apakah topik baru yang ditambahkan ke tujuan acara akan berlangganan atau berhenti berlangga nan secara default. Nilai yang mungkin OptIn(berlangganan secara default) atau OptOut(berhenti berlangganan secara default) di bidang. `subscriptionStatus`

Contoh data peristiwa yang SES diterbitkan Amazon ke Firehose

Bagian ini memberikan contoh jenis catatan peristiwa pengiriman email yang SES diterbitkan Amazon ke Firehose.

Topik di bagian ini:

- Rekaman pentalan
- Catatan aduan
- Catatan penyampaian
- Catatan pengiriman
- Tolak catatan

- Buka catatan
- Klik catatan
- Catatan Kegagalan Rendering
- DeliveryDelay rekor
- Catatan berlangganan

> **Note**
>
> Dalam contoh berikut di mana `tag` bidang digunakan, ia menggunakan penerbitan acara melalui set konfigurasi yang SES mendukung penerbitan tag untuk semua jenis acara. Jika menggunakan pemberitahuan umpan balik langsung pada identitas, SES tidak mempublikasikan tag. Baca tentang menambahkan tag saat membuat set konfigurasi atau memodifikasi set konfigurasi.

Rekaman pentalan

Berikut ini adalah contoh catatan `Bounce` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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"
      ]
    }
  }
}
```

Catatan aduan

Berikut ini adalah contoh catatan `Complaint` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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"
      ]
    }
  }
```

```
}
```

Catatan penyampaian

Berikut ini adalah contoh catatan `Delivery` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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"
  }
```

```
}
```

Catatan pengiriman

Berikut ini adalah contoh catatan `Send` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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": {}
}
```

Tolak catatan

Berikut ini adalah contoh catatan `Reject` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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"
  }
}
```

Buka catatan

Berikut ini adalah contoh catatan `Open` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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": [
        "IAM_user_or_role_name"
      ],
```

```
      "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"
  }
}
```

Klik catatan

Berikut ini adalah contoh catatan `Click` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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"
  }
}
```

Catatan Kegagalan Rendering

Berikut ini adalah contoh catatan `Rendering Failure` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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 rekor

Berikut ini adalah contoh catatan `DeliveryDelay` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "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"
    }]
  }
}
```

Catatan berlangganan

Berikut ini adalah contoh catatan `Subscription` acara yang SES diterbitkan Amazon ke Firehose.

```
{
  "eventType": "Subscription",
  "mail": {
    "timestamp": "2022-01-12T01:00:14.340Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLEe4bccb684-777bc8de-afa7-4970-92b0-f515137b1497-000000",
    "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": "EXAMPLEe4bccb684-777bc8de-afa7-4970-92b0-f515137b1497-000000",
      "subject": "Message sent from Amazon SES"
    },
    "tags": {
      "ses:operation": ["SendEmail"],
      "ses:configuration-set": ["ConfigSet"],
      "ses:source-ip": ["192.0.2.0"],
      "ses:from-domain": ["example.com"],
      "ses:caller-identity": ["ses_user"],
      "myCustomTag1": ["myCustomValue1"],
      "myCustomTag2": ["myCustomValue2"]
    }
  },
  "subscription": {
    "contactList": "ContactListName",
    "timestamp": "2022-01-12T01:00:17.910Z",
    "source": "UnsubscribeHeader",
    "newTopicPreferences": {
      "unsubscribeAll": true,
      "topicSubscriptionStatus": [
        {
          "topicName": "ExampleTopicName",
          "subscriptionStatus": "OptOut"
        }
      ]
    },
    "oldTopicPreferences": {
      "unsubscribeAll": false,
      "topicSubscriptionStatus": [
        {
          "topicName": "ExampleTopicName",
          "subscriptionStatus": "OptOut"
        }
      ]
    }
  }
}
```

Menafsirkan data peristiwa Amazon SES dari Amazon SNS

Amazon SES menerbitkan peristiwa pengiriman email ke Amazon Simple Notification Service (Amazon SNS) sebagai catatan JSON. Amazon SNS kemudian menyampaikan notifikasi ke titik akhir yang berlangganan ke topik Amazon SNS yang terkait dengan tujuan peristiwa. Untuk informasi tentang penyiapan topik dan berlangganan di Amazon SNS, lihat Memulai dalam Panduan Developer Amazon Simple Notification Service.

Untuk deskripsi konten catatan dan sebagai contoh catatan, lihat bagian berikut ini.

- Isi catatan peristiwa
- Contoh rekaman peristiwa

Isi data acara yang SES diterbitkan Amazon ke Amazon SNS

Amazon SES menerbitkan catatan peristiwa pengiriman email ke Amazon Simple Notification Service dalam JSON format.

Anda dapat menemukan contoh catatan untuk semua tipe notifikasi ini di Contoh data peristiwa yang diterbitkan Amazon SES ke Amazon SNS.

Topik di bagian ini:

- Objek tingkat atas JSON
- Objek surat
- Objek pentalan
- Objek aduan
- Objek penyampaian
- Kirim objek
- Tolak objek
- Buka objek
- Klik objek
- Objek Kegagalan Rendering
- DeliveryDelay objek
- Objek berlangganan

Objek tingkat atas JSON

JSONObjek tingkat atas dalam catatan peristiwa pengiriman email berisi bidang berikut. Jenis acara menentukan objek lain mana yang hadir.

Nama Bidang	Deskripsi
`eventType`	String yang menjelaskan tipe peristiwa. Nilai yang mungkin: `Bounce` `Complaint` ,`Delivery`,`Send`,,`Reject`,`Open`,`Click`,`Rendering Failure`,`DeliveryDelay` , atau`Subscription` . Jika Anda tidak menyiapkan penerbitan acara, bidang ini diberi nama`notificationType` .
`mail`	JSONObjek yang berisi informasi tentang email yang menghasilkan acara tersebut.
`bounce`	Bidang ini hanya ada jika `eventType` adalah `Bounce`. Bidang ini berisi informasi tentang pentalan.
`complaint`	Bidang ini hanya ada jika `eventType` adalah `Complaint` . Bidang ini berisi informasi tentang aduan.
`delivery`	Bidang ini hanya ada jika `eventType` adalah `Delivery`. Bidang ini berisi informasi tentang penyampaian.
`send`	Bidang ini hanya ada jika `eventType` adalah `Send`.
`reject`	Bidang ini hanya ada jika `eventType` adalah `Reject`. Bidang ini berisi informasi tentang penolakan.

Nama Bidang	Deskripsi
`open`	Bidang ini hanya ada jika `eventType` adalah `Open`. Bidang ini berisi informasi tentang peristiwa pembukaan.
`click`	Bidang ini hanya ada jika `eventType` adalah `Click`. Bidang ini berisi informasi tentang peristiwa pengeklikan.
`failure`	Bidang ini hanya ada jika `eventType` adalah `Rendering Failure` . Bidang ini berisi informasi tentang peristiwa kegagalan rendering.
`deliveryDelay`	Bidang ini hanya ada jika `eventType` adalah `DeliveryDelay` . Bidang tersebut berisi informasi tentang penyampaian email yang tertunda.
`subscription`	Bidang ini hanya ada jika `eventType` adalah `Subscription` . Ini berisi informasi tentang preferensi berlangganan.

Objek surat

Setiap catatan peristiwa pengiriman email berisi informasi tentang email asli di dalam objek `mail`. JSONObjek yang berisi informasi tentang suatu `mail` objek memiliki bidang berikut.

Nama Bidang	Deskripsi
`timestamp`	Tanggal dan waktu, dalam format ISO86 01 (YYYY-MM-:MM: DDThh SS.sz), saat pesan dikirim.
`messageId`	ID unik yang SES ditetapkan Amazon ke pesan. Amazon SES mengembalikan nilai ini kepada Anda saat Anda mengirim pesan.

Nama Bidang	Deskripsi
	Note ID pesan ini ditetapkan oleh AmazonSES. Anda dapat menemukan ID pesan email asli di bidang `headers` dan `commonHeaders` dari objek `mail`.
`source`	Alamat email tempat pesan dikirim (MAILFROM alamat amplop).
`sourceArn`	Nama Sumber Daya Amazon (ARN) dari identitas yang digunakan untuk mengirim email. Dalam hal mengirim otorisasi, itu `sourceArn` adalah identitas yang ARN pemilik identitas mengizinkan pengirim delegasi untuk digunakan untuk mengirim email. Untuk informasi selengkapnya tentang otorisasi pengiriman, lihat Metode autentikasi email.
`sendingAccountId`	ID AWS akun akun yang digunakan untuk mengirim email. Dalam hal otorisasi pengirima n, `sendingAccountId` adalah ID akun pengirim delegasi.
`destination`	Daftar alamat email yang merupakan penerima email asli.
`headersTruncated`	String yang menentukan jika header terpotong atau tidak di dalam notifikasi, yang terjadi jika header lebih besar dari 10 KB. Nilai yang mungkin adalah `true` dan `false`.

Nama Bidang	Deskripsi
`headers`	Daftar header asli email. Setiap header dalam daftar memiliki bidang `name` dan bidang `value`. **Note** ID pesan apa pun dalam `headers` bidang ini berasal dari pesan asli yang Anda kirimkan ke AmazonSES. ID pesan yang SES kemudian ditetapkan Amazon ke pesan ada di `messageId` bidang `mail` objek.
`commonHeaders`	Pemetaan header asli email yang umum digunakan. **Note** Setiap ID pesan dalam `commonHeaders` bidang adalah ID pesan yang SES kemudian ditetapkan Amazon ke pesan di `messageId` bidang `mail` objek.
`tags`	Daftar tag yang terkait dengan email.

Objek pentalan

JSONObjek yang berisi informasi tentang suatu `Bounce` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`bounceType`	Jenis bouncing, seperti yang ditentukan oleh AmazonSES.

Nama Bidang	Deskripsi
`bounceSubType`	Subtipe pantulan, sebagaimana ditentukan oleh Amazon. SES
`bouncedRecipients`	Daftar yang berisi informasi tentang penerima email asli yang terpental.
`timestamp`	Tanggal dan waktu, dalam format ISO86 01 (YYYY-MM-:MM: SS.sz DDThh), ketika mengirim pemberitahuan bouncing. ISP
`feedbackId`	ID unik untuk pentalan.
`reportingMTA`	Nilai `Reporting-MTA` bidang dariDSN. Ini adalah nilai dari Message Transfer Authority (MTA) yang mencoba untuk melakukan pengiriman, relay, atau operasi gateway yang dijelaskan dalamDSN. **Note** Bidang ini hanya muncul jika pemberita huan status pengiriman (DSN) dilampirkan ke bouncing.

Penerima yang terpental

Peristiwa pentalan mungkin berkaitan dengan satu atau beberapa penerima. Bidang `bouncedRecipients` menyimpan daftar objek—satu objek per penerima yang alamat emailnya menghasilkan pentalan—dan berisi bidang berikut.

Nama Bidang	Deskripsi
`emailAddress`	Alamat email penerima. Jika DSN tersedia, ini adalah nilai `Final-Recipient` bidang dariDSN.

Secara opsional, jika a DSN dilampirkan ke pantulan, bidang berikut mungkin juga ada.

Nama Bidang	Deskripsi
`action`	Nilai `Action` bidang dariDSN. Ini menunjukkan tindakan yang dilakukan oleh pelaporan MTA sebagai hasil dari upayanya untuk menyampai kan pesan kepada penerima ini.
`status`	Nilai `Status` bidang dariDSN. Ini adalah kode status bebas-transportasi per penerima yang menunjukkan status penyampaian pesan.
`diagnosticCode`	Kode status yang dikeluarkan oleh pelaporan MTA. Ini adalah nilai `Diagnostic-Code` bidang dariDSN. Bidang ini mungkin tidak ada di DSN (dan karena itu juga tidak ada diJSON).

Tipe pentalan

Setiap peristiwa pentalan adalah salah satu tipe yang ditunjukkan dalam tabel berikut.

Sistem penerbitan acara hanya menerbitkan hard bounce dan soft bounce yang tidak lagi dicoba lagi oleh Amazon. SES Ketika Anda menerima bounces yang ditandai`Permanent`, Anda harus menghapus alamat email yang sesuai dari milis Anda; Anda tidak akan dapat mengirim kepada mereka di masa depan. `Transient`Pantulan dikirimkan kepada Anda ketika pesan telah memantul beberapa kali, dan Amazon SES telah berhenti mencoba mengirimkannya kembali. Anda mungkin akan berhasil mengirim ulang ke alamat yang awalnya menghasilkan pentalan `Transient` lain kali.

bounceType	bounceSubType	Deskripsi
`Undetermined`	`Undetermined`	Amazon SES tidak dapat menentukan alasan pentalan tertentu.
`Permanent`	`General`	Amazon SES menerima pantulan keras umum. Jika Anda menerima tipe pentalan ini, maka Anda harus menghapus alamat email penerima dari daftar email Anda.

bounceType	bounceSubType	Deskripsi
`Permanent`	`NoEmail`	Amazon SES menerima hard bounce permanen karena alamat email target tidak ada. Jika Anda menerima tipe pentalan ini, maka Anda harus menghapus alamat email penerima dari daftar email Anda.
`Permanent`	`Suppressed`	Amazon SES telah menekan pengiriman ke alamat ini karena memiliki riwayat memantul baru-baru ini sebagai alamat yang tidak valid. Untuk mengganti daftar penindasan global, lihat. Menggunakan daftar SES penindasan tingkat akun Amazon
`Permanent`	`OnAccountSuppressionList`	Amazon SES telah menekan pengiriman ke alamat ini karena ada di daftar penindasan tingkat akun. Ini tidak dihitung terhadap metrik rasio pentalan Anda.
`Transient`	`General`	Amazon SES menerima pantulan umum. Anda mungkin akan berhasil mengirim ke penerima ini lain kali.
`Transient`	`MailboxFull`	Amazon SES menerima bouncing penuh kotak surat. Anda mungkin akan berhasil mengirim ke penerima ini lain kali.
`Transient`	`MessageTooLarge`	Amazon SES menerima pesan pantulan yang terlalu besar. Anda mungkin akan berhasil mengirim ke penerima ini jika Anda mengurangi ukuran pesan.
`Transient`	`ContentRejected`	Amazon SES menerima bouncing konten yang ditolak. Anda mungkin akan berhasil mengirim ke penerima ini jika Anda mengubah konten pesan.

bounceType	bounceSubType	Deskripsi
`Transient`	`AttachmentRejected`	Amazon SES menerima bouncing lampiran yang ditolak. Anda mungkin akan berhasil mengirim ke penerima ini jika Anda menghapus atau mengubah lampiran.

Objek aduan

JSONObjek yang berisi informasi tentang suatu `Complaint` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`complainedRecipients`	Daftar yang berisi informasi tentang penerima yang mungkin telah mengirimkan aduan.
`timestamp`	Tanggal dan waktu, dalam format ISO86 01 (YYYY-MM-:MM: SS.sz DDThh), ketika mengirim pemberitahuan keluhan. ISP
`feedbackId`	ID unik untuk aduan.
`complaintSubType`	Subtipe keluhan, sebagaimana ditentukan oleh AmazonSES.

Selanjutnya, jika laporan umpan balik dilampirkan pada aduan, bidang berikut mungkin ada.

Nama Bidang	Deskripsi
`userAgent`	Nilai bidang `User-Agent` dari laporan umpan balik. Nilai ini menunjukkan nama dan versi sistem yang menghasilkan laporan.
`complaintFeedbackType`	Nilai `Feedback-Type` bidang dari laporan umpan balik yang diterima dariISP. Ini berisi tipe umpan balik.

Nama Bidang	Deskripsi
`arrivalDate`	Nilai `Arrival-Date` atau `Received-Date` bidang dari laporan umpan balik dalam format ISO86 01 (YYYY-MM DDThh -:MM: SS.sz). Bidang ini mungkin tidak ada dalam laporan (dan karena itu juga tidak ada dalamJSON).

Penerima yang diadukan

Bidang `complainedRecipients` berisi daftar penerima yang mungkin telah mengirimkan aduan.

> **Important**
>
> Sebagian besar ISPs menyunting alamat email penerima yang mengajukan keluhan. Untuk alasan ini, bidang `complainedRecipients` menyertakan daftar semua orang yang dikirimi email yang alamatnya ada di domain yang mengeluarkan notifikasi aduan.

JSONobjek dalam daftar ini berisi bidang berikut.

Nama Bidang	Deskripsi
`emailAddress`	Alamat email penerima.

Tipe aduan

Anda dapat melihat jenis keluhan berikut di `complaintFeedbackType` bidang yang ditetapkan oleh pelaporanISP, menurut situs web Internet Assigned Numbers Authority:

Nama Bidang	Deskripsi
`abuse`	Menunjukkan email yang tidak diminta atau beberapa jenis penyalahgunaan email lainnya.
`auth-failure`	Laporan kegagalan otentikasi email.

Nama Bidang	Deskripsi
`fraud`	Menunjukkan beberapa jenis penipuan atau aktivitas phishing.
`not-spam`	Menunjukkan bahwa entitas yang menyediak an laporan tidak menganggap pesan tersebut sebagai spam. Tindakan ini dapat digunakan untuk memperbaiki pesan yang salah ditandai atau dikategorikan sebagai spam.
`other`	Menunjukkan umpan balik lain yang tidak sesuai dengan tipe terdaftar lainnya.
`virus`	Melaporkan bahwa virus ditemukan dalam pesan asal.

Subtipe aduan

Nilai bidang `complaintSubType` bisa jadi tidak ada ataupun `0nAccountSuppressionList`. Jika nilainya`0nAccountSuppressionList`, Amazon SES menerima pesan tersebut, tetapi tidak mencoba mengirimkannya karena ada di daftar penindasan tingkat akun.

Objek penyampaian

JSONObjek yang berisi informasi tentang suatu `Delivery` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`timestamp`	Tanggal dan waktu Amazon SES mengirimkan email ke server email penerima, dalam format ISO86 01 (YYYY-MM DDThh -:MM: SS.sz).
`processingTimeMillis`	Waktu dalam milidetik antara saat Amazon SES menerima permintaan dari pengirim hingga saat Amazon SES meneruskan pesan ke server email penerima.

Nama Bidang	Deskripsi
`recipients`	Daftar penerima yang dituju yang berlaku untuk peristiwa penyampaian.
`smtpResponse`	Pesan SMTP respons dari remote ISP yang menerima email dari AmazonSES. Pesan ini akan bervariasi menurut email, dengan menerima server email, dan dengan menerimaI SP.
`reportingMTA`	Nama host dari server SES email Amazon yang mengirim email.

Kirim objek

JSONObjek yang berisi informasi tentang suatu `send` peristiwa selalu kosong.

Tolak objek

JSONObjek yang berisi informasi tentang suatu `Reject` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`reason`	Alasan email ditolak. Satu-satunya nilai yang mungkin adalah`Bad content`, yang berarti Amazon SES mendeteksi bahwa email tersebut mengandung virus. Ketika pesan ditolak, Amazon SES berhenti memprosesnya, dan tidak mencoba mengirimkannya ke server email penerima.

Buka objek

JSONObjek yang berisi informasi tentang suatu `Open` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`ipAddress`	Alamat IP penerima.
`timestamp`	Tanggal dan waktu ketika peristiwa terbuka terjadi dalam format ISO86 01 (YYYY-MM DDThh -:MM: SS.sz).
`userAgent`	Agen pengguna perangkat atau klien email yang digunakan penerima untuk membuka email.

Klik objek

JSONObjek yang berisi informasi tentang suatu `Click` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`ipAddress`	Alamat IP penerima.
`timestamp`	Tanggal dan waktu ketika peristiwa klik terjadi dalam format ISO86 01 (YYYY-MM DDThh -:MM: SS.sz).
`userAgent`	Agen pengguna klien yang digunakan penerima untuk mengeklik tautan di dalam email.
`link`	URLTautan yang diklik penerima.
`linkTags`	Daftar tanda yang ditambahkan ke tautan menggunakan atribut `ses:tags`. Untuk informasi selengkapnya tentang menambahk an tanda ke tautan di email Anda, lihat T5. Dapatkah saya menandai tautan dengan pengenal unik? dalam FAQ metrik pengiriman email Amazon SES.

Objek Kegagalan Rendering

JSONObjek yang berisi informasi tentang suatu `Rendering Failure` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`templateName`	Nama templat yang digunakan untuk mengirim email.
`errorMessage`	Pesan yang menyediakan informasi selengkap nya tentang kegagalan rendering.

DeliveryDelay objek

JSONObjek yang berisi informasi tentang suatu `DeliveryDelay` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`delayType`	Tipe penundaan. Kemungkinan nilai adalah: • InternalFailure— SES Masalah internal Amazon menyebabkan pesan tertunda. • Umum — Kegagalan umum terjadi selama SMTP percakapan. • MailboxFull— Kotak pesan penerima penuh dan tidak dapat menerima pesan tambahan. • SpamDetected— Server email penerima telah mendeteksi sejumlah besar email yang tidak diminta dari akun Anda. • RecipientServerError— Masalah sementara dengan server email penerima mencegah pengiriman pesan. • IPFailureAlamat IP yang mengirim pesan sedang diblokir atau dibatasi oleh penyedia email penerima.

Nama Bidang	Deskripsi
	• TransientCommunicationFailure— Ada kegagalan komunikasi sementara selama SMTP percakapan dengan penyedia email penerima. • BYOIPHostNameLookupUnavailable— Amazon SES tidak dapat mencari DNS nama host untuk alamat IP Anda. Tipe penundaan ini hanya terjadi ketika Anda menggunakan Bawa IP Anda Sendiri. • Belum ditentukan - Amazon SES tidak dapat menentukan alasan keterlambatan pengirima n. • SendingDeferral— Amazon SES telah menganggap pantas untuk menunda pesan secara internal.
`delayedRecipients`	Objek yang berisi informasi tentang penerima email.
`expirationTime`	Tanggal dan waktu ketika Amazon SES akan berhenti mencoba menyampaikan pesan. Nilai ini ditunjukkan dalam format ISO 8601.
`reportingMTA`	Alamat IP dari Message Transfer Agent (MTA) yang melaporkan keterlambatan.
`timestamp`	Tanggal dan waktu ketika penundaan terjadi, ditampilkan dalam format ISO 8601.

Penerima tertunda

Objek `delayedRecipients` berisi nilai-nilai berikut:

Nama Bidang	Deskripsi
`emailAddress`	Alamat email yang mengakibatkan penyampai an pesan tertunda.
`status`	Kode SMTP status yang terkait dengan penundaan pengiriman.
`diagnosticCode`	Kode diagnostik yang disediakan oleh Agen Transfer Pesan penerima (MTA).

Objek berlangganan

JSONObjek yang berisi informasi tentang suatu `Subscription` peristiwa memiliki bidang berikut.

Nama Bidang	Deskripsi
`contactList`	Nama daftar kontak berada.
`timestamp`	Tanggal dan waktu, dalam format ISO86 01 (YYYY-MM-:MM: SS.sz DDThh), ketika mengirim pemberitahuan berlangganan. ISP
`source`	Alamat email tempat pesan dikirim (MAILFROM alamat amplop).
`newTopicPreferences`	JSONStruktur data (peta) yang menentukan status langganan semua topik dalam daftar kontak yang menunjukkan status setelah perubahan (kontak berlangganan atau berhenti berlangganan).
`oldTopicPreferences`	JSONStruktur data (peta) yang menentukan status langganan semua topik dalam daftar kontak yang menunjukkan status sebelum perubahan (kontak berlangganan atau berhenti berlangganan).

Preferensi topik baru/lama

`oldTopicPreferences`Objek `newTopicPreferences` dan berisi nilai-nilai berikut.

Nama Bidang	Deskripsi
`unsubscribeAll`	Menentukan apakah kontak berhenti berlangga nan dari semua topik dalam daftar kontak.
`topicSubscriptionStatus`	Menentukan status langganan topik di `topicName` bidang yang menunjukkan apakah saat ini berlangganan untuk menerima pemberitahuan dari SES untuk jenis acara yang ditentukan. Nilai yang mungkin OptIn(berlangg anan) atau OptOut(berhenti berlangganan) di bidang. `subscriptionStatus`
`topicDefaultSubscriptionStatus`	Menentukan status langganan default topik di `topicName` bidang menentukan apakah topik baru yang ditambahkan ke tujuan acara akan berlangganan atau berhenti berlangga nan secara default. Nilai yang mungkin OptIn(berlangganan secara default) atau OptOut(berhenti berlangganan secara default) di bidang. `subscriptionStatus`

Contoh data peristiwa yang diterbitkan Amazon SES ke Amazon SNS

Bagian ini menyediakan contoh tipe catatan peristiwa pengiriman email yang diterbitkan Amazon SES ke Amazon SNS.

Topik di bagian ini:

- Rekaman pentalan
- Catatan aduan
- Catatan penyampaian
- Catatan pengiriman
- Tolak catatan

- Buka catatan
- Klik catatan
- Catatan Kegagalan Rendering
- DeliveryDelayrekor
- Catatan berlangganan

> **Note**
>
> Dalam contoh berikut di mana `tag` bidang digunakan, ia menggunakan penerbitan acara melalui set konfigurasi yang SES mendukung penerbitan tag untuk semua jenis acara. Jika menggunakan pemberitahuan umpan balik langsung pada identitas, SES tidak mempublikasikan tag. Baca tentang menambahkan tag saat membuat set konfigurasi atau memodifikasi set konfigurasi.

Rekaman pentalan

Berikut ini adalah contoh rekaman peristiwa `Bounce` yang diterbitkan Amazon SES ke Amazon SNS.

```
{
  "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"
      ]
    }
  }
}
```

Catatan aduan

Berikut ini adalah contoh rekaman peristiwa `Complaint` yang diterbitkan Amazon SES ke Amazon SNS.

```
{
  "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"
      ]
```

```
    }
  }
}
```

Catatan penyampaian

Berikut ini adalah contoh rekaman peristiwa `Delivery` yang diterbitkan Amazon SES ke Amazon SNS.

```
{
  "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"
  }
}
```

Catatan pengiriman

Berikut ini adalah contoh rekaman peristiwa Send yang diterbitkan Amazon SES ke Amazon SNS. Beberapa bidang tidak selalu ada. Misalnya, dengan email template, subjek dirender kemudian dan disertakan dalam acara berikutnya.

```
{
  "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": {}
}
```

Tolak catatan

Berikut ini adalah contoh rekaman peristiwa `Reject` yang diterbitkan Amazon SES ke Amazon SNS.

```
{
  "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"
  }
}
```

Buka catatan

Berikut ini adalah contoh rekaman peristiwa 0pen yang diterbitkan Amazon SES ke Amazon SNS.

```
{
  "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": [
        "IAM_user_or_role_name"
      ],
      "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"
  }
}
```

Klik catatan

Berikut ini adalah contoh rekaman peristiwa `Click` yang diterbitkan Amazon SES ke Amazon SNS.

```
{
  "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"
  }
}
```

Catatan Kegagalan Rendering

Berikut ini adalah contoh rekaman peristiwa `Rendering Failure` yang diterbitkan Amazon SES ke Amazon SNS.

```
{
```

```
  "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"
  }
}
```

DeliveryDelayrekor

Berikut ini adalah contoh rekaman peristiwa `DeliveryDelay` yang diterbitkan Amazon SES ke Amazon SNS.

```
{
  "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"
    }]
  }
}
```

Catatan berlangganan

Berikut ini adalah contoh catatan `Subscription` acara yang diterbitkan Amazon SES ke Firehose.

```
{
  "eventType": "Subscription",
  "mail": {
    "timestamp": "2022-01-12T01:00:14.340Z",
    "source": "sender@example.com",
    "sourceArn": "arn:aws:ses:us-east-1:123456789012:identity/sender@example.com",
    "sendingAccountId": "123456789012",
    "messageId": "EXAMPLEe4bccb684-777bc8de-afa7-4970-92b0-f515137b1497-000000",
    "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": "EXAMPLEe4bccb684-777bc8de-afa7-4970-92b0-f515137b1497-000000",
      "subject": "Message sent from Amazon SES"
    },
    "tags": {
      "ses:operation": ["SendEmail"],
      "ses:configuration-set": ["ConfigSet"],
      "ses:source-ip": ["192.0.2.0"],
      "ses:from-domain": ["example.com"],
      "ses:caller-identity": ["ses_user"],
      "myCustomTag1": ["myCustomValue1"],
      "myCustomTag2": ["myCustomValue2"]
    }
  },
  "subscription": {
    "contactList": "ContactListName",
    "timestamp": "2022-01-12T01:00:17.910Z",
    "source": "UnsubscribeHeader",
    "newTopicPreferences": {
      "unsubscribeAll": true,
      "topicSubscriptionStatus": [
        {
          "topicName": "ExampleTopicName",
          "subscriptionStatus": "OptOut"
        }
      ]
    },
    "oldTopicPreferences": {
      "unsubscribeAll": false,
      "topicSubscriptionStatus": [
        {
```

```
          "topicName": "ExampleTopicName",
          "subscriptionStatus": "OptOut"
        }
      ]
    }
  }
}
```

Pemantauan reputasi pengirim Amazon SES Anda

Amazon SES secara aktif melacak beberapa metrik yang dapat menyebabkan reputasi Anda sebagai pengirim rusak, atau yang dapat menyebabkan tingkat pengiriman email Anda menurun. Dua metrik penting yang kami pertimbangkan dalam proses ini adalah tingkat pentalan dan aduan untuk akun Anda. Jika tingkat pentalan atau aduan untuk akun Anda terlalu tinggi, kami akan menempatkan akun Anda dalam peninjauan atau menjeda kemampuan akun Anda untuk mengirim email.

Karena tingkat pentalan dan aduan Anda sangat penting bagi kondisi akun Anda, Amazon SES menyertakan halaman metrik reputasi di konsol Amazon SES yang dapat Anda gunakan untuk melacak metrik ini. Metrik reputasi juga dapat menampilkan informasi tentang faktor-faktor yang tidak terkait dengan pentalan atau aduan yang dapat merusak reputasi pengirim Anda. Misalnya, jika Anda mengirim email ke email jebakan spam yang diketahui, maka Anda akan melihat pesan di dasbor ini.

Bagian ini berisi informasi tentang mengakses metrik reputasi, menginterpretasikan informasi yang dikandungnya, dan menyiapkan sistem untuk secara aktif memberi tahu Anda faktor-faktor yang dapat memengaruhi reputasi pengirim Anda.

Pada bagian ini, Anda akan menemukan topik berikut:

- Menggunakan metrik reputasi untuk melacak tingkat pentalan dan aduan
- Pesan metrik reputasi
- Membuat alarm pemantauan reputasi menggunakan CloudWatch
- Metrik SNDS untuk IP khusus
- Menjeda pengiriman email secara otomatis

Menggunakan metrik reputasi untuk melacak tingkat pentalan dan aduan

Halaman konsol metrik reputasi berisi informasi yang sama yang dilihat oleh tim Amazon SES ketika mereka menentukan kondisi akun individu.

Untuk melihat reputasi metrik

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi pada sisi kiri layar, pilih Metrik reputasi.

Dasbor menampilkan informasi berikut:

- Status akun — Ringkasan dari gabungan kondisi pada tingkat pentalan dan aduan Anda. Nilai yang mungkin termasuk:
  - Sehat – Saat ini tidak ada masalah yang memengaruhi akun Anda.
  - Dalam peninjauan – Akun Anda dalam peninjauan. Jika masalah yang menyebabkan kami menempatkan akun Anda dalam peninjauan tidak terselesaikan pada akhir periode peninjauan, kami mungkin menjeda kemampuan akun Anda untuk mengirim email.
  - Menunggu akhir keputusan peninjauan – Akun Anda dalam peninjauan. Karena sifat masalah yang menyebabkan kami menempatkan akun Anda dalam peninjauan, kami perlu melakukan peninjauan manual terhadap akun Anda sebelum kami mengambil tindakan lebih lanjut.
  - Pengiriman dijeda – Kami telah menjeda kemampuan akun Anda untuk mengirim email. Ketika kemampuan akun Anda untuk mengirim email dijeda, Anda tidak akan dapat mengirim email menggunakan Amazon SES. Anda dapat meminta kami meninjau keputusan ini. Untuk mempelajari selengkapnya tentang meminta peninjauan, lihat FAQ proses peninjauan Pengiriman Amazon SES.
  - Menunggu jeda pengiriman – Akun Anda dalam peninjauan. Masalah yang menyebabkan kami menempatkan akun Anda dalam peninjauan belum teratasi. Dalam situasi ini, kami biasanya menjeda kemampuan akun Anda untuk mengirim email. Namun, karena sifat akun Anda, kami perlu meninjau akun Anda sebelum mengambil tindakan lebih lanjut.
- Tingkat Pentalan – Persentase email yang dikirim dari akun Anda yang menghasilkan pentalan keras. Lihat bagaimana rasio pentalan Anda dihitung.
- Tingkat Aduan – Persentase email yang dikirim dari akun Anda yang mengakibatkan penerima melaporkannya sebagai spam. Lihat bagaimana tingkat keluhan Anda dihitung

> **Note**
>
> Bagian Tingkat Pentalan dan Tingkat Aduan juga menyertakan pesan status untuk metrik masing-masing. Berikut ini adalah daftar pesan status yang mungkin ditampilkan untuk metrik berikut:
>
> - Sehat – Metrik berada dalam level normal.
> - Hampir pulih – Metrik menyebabkan akun Anda ditempatkan dalam peninjauan. Sejak periode peninjauan dimulai, metrik tetap berada di bawah tingkat maksimum.

Jika metrik tetap di bawah tingkat maksimum, status metrik ini akan berubah menjadi Sehat sebelum periode peninjauan berakhir.

- Dalam peninjauan – Metrik menyebabkan akun Anda ditempatkan dalam peninjauan, dan masih berada di atas tingkat maksimum. Jika masalah yang menyebabkan metrik melebihi tingkat maksimum tidak teratasi pada akhir periode peninjauan, kami mungkin menjeda kemampuan akun Anda untuk mengirim email.
- Pengiriman jeda – Metrik ini menyebabkan kami menjeda kemampuan akun Anda untuk mengirim email. Saat kemampuan akun Anda untuk mengirim email dijeda, Anda tidak dapat mengirim email menggunakan Amazon SES. Anda dapat meminta kami meninjau keputusan ini. Untuk mempelajari selengkapnya tentang mengirimkan permintaan peninjauan, lihat FAQ proses peninjauan Pengiriman Amazon SES.
- Menunggu pengiriman jeda – Metrik ini menyebabkan kami menempatkan akun Anda dalam peninjauan. Masalah yang menyebabkan periode peninjauan ini belum teratasi. Masalah ini mungkin membuat kami menjeda kemampuan akun Anda untuk mengirim email. Seorang anggota tim Amazon SES harus meninjau akun Anda sebelum kami mengambil tindakan lebih lanjut.

- Notifikasi Lain – Jika akun Anda mengalami masalah terkait reputasi yang tidak terkait dengan pentalan atau aduan, pesan singkat akan ditampilkan di sini. Untuk informasi lebih lanjut tentang notifikasi yang dapat ditampilkan di area ini, lihat Pesan metrik reputasi.

Pesan metrik reputasi

Halaman konsol metrik Amazon SES Reputation menyediakan metrik penting yang terkait dengan akun Anda. Bagian berikut menjelaskan pesan yang mungkin ditampilkan di dasbor ini, serta memberikan tips dan informasi yang mungkin dapat Anda gunakan untuk menyelesaikan masalah yang berkaitan dengan reputasi pengirim Anda.

Bagian ini memuat informasi tentang jenis notifikasi berikut ini:

- Pesan Status
- Notifikasi Tingkat Pentalan
- Notifikasi Tingkat Aduan
- Notifikasi Organisasi Anti-Spam
- Pemberitahuan Listbombing

- Notifikasi umpan balik langsung
- Notifikasi Daftar Blokir Domain
- Notifikasi Peninjauan Internal
- Notifikasi Penyedia Kotak Surat
- Notifikasi Umpan Balik Penerima
- Notifikasi Akun Terkait
- Notifikasi Jebakan Spam
- Notifikasi Situs Rentan
- Kredensi yang Dikompromikan
- Notifikasi lainnya

Pesan Status

Bila Anda menggunakan halaman konsol metrik reputasi, Anda akan melihat pesan yang menjelaskan status akun Amazon SES Anda. Berikut ini adalah daftar kemungkinan nilai status akun:

- Sehat – Saat ini tidak ada masalah yang memengaruhi akun Anda.
- Dalam peninjauan – Akun Anda dalam peninjauan. Jika masalah yang menyebabkan kami menempatkan akun Anda dalam peninjauan tidak terselesaikan pada akhir periode peninjauan, kami mungkin menjeda kemampuan akun Anda untuk mengirim email.
- Menunggu akhir keputusan peninjauan – Akun Anda dalam peninjauan. Karena sifat masalah yang menyebabkan kami menempatkan akun Anda dalam peninjauan, kami perlu melakukan peninjauan manual terhadap akun Anda sebelum kami mengambil tindakan lebih lanjut.
- Pengiriman dijeda – Kami telah menjeda kemampuan akun Anda untuk mengirim email. Ketika kemampuan akun Anda untuk mengirim email dijeda, Anda tidak akan dapat mengirim email menggunakan Amazon SES. Anda dapat meminta kami meninjau keputusan ini. Untuk mempelajari selengkapnya tentang meminta peninjauan, lihat FAQ proses peninjauan Pengiriman Amazon SES.
- Menunggu jeda pengiriman – Akun Anda dalam peninjauan. Masalah yang menyebabkan kami menempatkan akun Anda dalam peninjauan belum teratasi. Dalam situasi ini, kami biasanya menjeda kemampuan akun Anda untuk mengirim email. Namun, karena sifat akun Anda, kami perlu meninjau akun Anda sebelum mengambil tindakan lebih lanjut.

Selain itu, bagian Tingkat Pentalan dan Tingkat Aduan pada halaman metrik reputasi menampilkan ringkasan status untuk masing-masing metrik. Berikut adalah daftar kemungkinan nilai status metrik:

- Sehat – Metrik berada dalam level normal.
- Hampir pulih – Metrik menyebabkan akun Anda ditempatkan dalam peninjauan. Sejak periode peninjauan dimulai, metrik tetap berada di bawah tingkat maksimum. Jika metrik tetap di bawah tingkat maksimum, status metrik ini akan berubah menjadi Sehat sebelum periode peninjauan berakhir.
- Dalam peninjauan – Metrik menyebabkan akun Anda ditempatkan dalam peninjauan, dan masih berada di atas tingkat maksimum. Jika masalah yang menyebabkan metrik melebihi tingkat maksimum tidak teratasi pada akhir periode peninjauan, kami mungkin menjeda kemampuan akun Anda untuk mengirim email.
- Pengiriman jeda – Metrik ini menyebabkan kami menjeda kemampuan akun Anda untuk mengirim email. Saat kemampuan akun Anda untuk mengirim email dijeda, Anda tidak dapat mengirim email menggunakan Amazon SES. Anda dapat meminta kami meninjau keputusan ini. Untuk mempelajari selengkapnya tentang mengirimkan permintaan peninjauan, lihat FAQ proses peninjauan Pengiriman Amazon SES.
- Menunggu pengiriman jeda – Metrik ini menyebabkan kami menempatkan akun Anda dalam peninjauan. Masalah yang menyebabkan periode peninjauan ini belum teratasi. Masalah ini mungkin membuat kami menjeda kemampuan akun Anda untuk mengirim email. Seorang anggota tim Amazon SES harus meninjau akun Anda sebelum kami mengambil tindakan lebih lanjut.

Notifikasi Tingkat Pentalan

Bagian ini berisi informasi tambahan tentang notifikasi tingkat pentalan yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Anda menerima notifikasi ini karena tingkat pentalan untuk akun Anda terlalu tinggi. Tingkat pentalan didasarkan pada jumlah pentalan keras yang dihasilkan oleh akun Amazon SES Anda. Penyedia email menafsirkan tingkat pentalan tinggi sebagai tanda bahwa pengirim tidak mengelola daftar penerima mereka dengan benar, dan bahwa pengirim mungkin mengirim email yang tidak diminta.

Pentalan keras terjadi ketika email dikirim ke alamat yang tidak ada. Amazon SES tidak memperhitungkan pentalan lunak (yang terjadi ketika alamat penerima sementara tidak dapat menerima pesan) dalam perhitungan ini. Email terpental yang Anda kirim ke alamat dan domain

terverifikasi, serta email yang Anda kirim ke Simulator kotak masuk Amazon SES, juga tidak diperhitungkan dalam perhitungan ini.

Kami menghitung tingkat pentalan Anda berdasarkan volume representatif email. Volume representatif adalah jumlah email yang mewakili praktik pengiriman biasa Anda. Agar adil bagi pengirim volume tinggi dan rendah, volume representatif berbeda untuk setiap akun dan berubah seiring perubahan pola pengiriman akun.

Untuk hasil terbaik, pertahankan tingkat pentalan di bawah 5%. Tingkat pentalan yang lebih tinggi dapat mempengaruhi pengiriman email Anda. Jika tingkat pentalan Anda 5% atau lebih, kami secara otomatis menempatkan akun Anda dalam peninjauan. Jika tingkat pentalan 10% atau lebih besar, kami mungkin menjeda kemampuan akun Anda untuk mengirim email tambahan sampai Anda menyelesaikan masalah yang menyebabkan tingkat pentalan tinggi.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Jika belum melakukannya, lakukan proses untuk menangkap dan mengelola pentalan dan aduan Semua akun Amazon SES perlu melakukan proses ini. Untuk informasi selengkapnya, lihat Metrik keberhasilan program email.

Selanjutnya, tentukan alamat email mana yang mementalkan, dan buat serta terapkan rencana untuk meredam atau menghilangkan pentalan ini. Jika kemampuan akun Anda untuk mengirim email telah dijeda, masuk ke AWS Management Console dan buka AWS Support. Balas kasus yang kami buka atas nama Anda.

Jika akun Anda sedang dalam peninjauan

Pada akhir periode peninjauan, jika tingkat pentalan untuk akun Anda tetap di atas 10%, kami mungkin menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam tanggapan Anda terhadap kasus, jelaskan perubahan yang Anda terapkan. Jika kami setuju bahwa perubahan akan mengurangi tingkat pentalan Anda, kami menyesuaikan perhitungan kami untuk hanya mempertimbangkan pentalan yang diterima setelah perubahan Anda diterapkan.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Ketika Anda menerapkan perubahan yang Anda percaya akan menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Notifikasi Tingkat Aduan

Bagian ini berisi informasi tambahan tentang notifikasi tingkat aduan yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Anda menerima notifikasi ini karena tingkat aduan untuk akun Anda terlalu tinggi. Tingkat aduan didasarkan pada jumlah keluhan yang dihasilkan oleh akun Amazon SES Anda. Penyedia email menafsirkan tingkat aduan yang tinggi sebagai tanda bahwa pengirim tidak mengelola daftar penerima dengan benar, dan pengirim mungkin mengirim email yang tidak diminta.

Aduan terjadi saat penerima mengidentifikasi email yang Anda kirim sebagai spam. Hal ini biasanya terjadi ketika penerima menggunakan tombol Laporkan Spam di klien email mereka. Keluhan yang dihasilkan oleh email yang Anda kirim ke Simulator kotak masuk Amazon SES tidak diperhitungkan dalam perhitungan ini.

Kami menghitung tingkat aduan Anda berdasarkan volume representatif email. Volume representatif adalah jumlah email yang mewakili praktik pengiriman biasa Anda. Agar \adil bagi pengirim volume tinggi dan rendah, volume representatif berbeda untuk setiap akun dan berubah seiring perubahan pola pengiriman akun.

Untuk hasil terbaik, pertahankan tingkat aduan di bawah 0,1%. Tingkat aduan yang lebih tinggi dapat memengaruhi pengiriman email Anda. Jika tingkat aduan Anda 0,1% atau lebih, kami secara otomatis menempatkan akun Anda dalam peninjauan. Jika tingkat aduan Anda 0,5% atau lebih besar, kami mungkin menjeda kemampuan akun Anda untuk mengirim email lain sampai Anda menyelesaikan masalah yang menyebabkan tingginya tingkat aduan.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Jika belum melakukannya, lakukan proses untuk menangkap dan mengelola pentalan dan aduan Semua akun Amazon SES perlu melakukan proses ini. Untuk informasi selengkapnya, lihat Metrik keberhasilan program email.

Selanjutnya, tentukan pesan yang Anda kirim yang menghasilkan aduan, dan terapkan rencana untuk mengurangi aduan ini. Jika kemampuan akun Anda untuk mengirim email telah dijeda, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda

Meskipun Anda harus segera berhenti mengirim ke alamat yang mengadu, penting bagi Anda untuk mengidentifikasi faktor-faktor yang menyebabkan penerima memberikan aduan. Setelah mengidentifikasi faktor-faktor tersebut, sesuaikan perilaku pengiriman email Anda untuk mengatasinya.

Jika akun Anda sedang dalam peninjauan

Pada akhir periode peninjauan, jika tingkat aduan untuk akun Anda tetap di atas 0,5%, kami mungkin menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam tanggapan Anda terhadap kasus, jelaskan perubahan yang Anda terapkan. Jika kami setuju bahwa perubahan akan mengurangi tingkat aduan Anda, kami menyesuaikan perhitungan kami untuk hanya mempertimbangkan aduan yang diterima setelah Anda menerapkan perubahan.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Notifikasi Organisasi Anti-Spam

Bagian ini berisi informasi tambahan tentang notifikasi organisasi anti-spam yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Organisasi anti-spam terkemuka telah melaporkan bahwa beberapa konten yang dikirim dari akun Amazon SES Anda telah ditandai sebagai tidak diminta atau bermasalah oleh sistem mereka.

Kami tidak dapat memberikan informasi tentang pesan tertentu yang menyebabkan organisasi anti-spam menandai konten Anda sebagai bermasalah. Kami tidak dapat memberikan nama organisasi yang mengeluarkan laporan. Biasanya, organisasi anti-spam mempertimbangkan kombinasi faktor-faktor berikut: umpan balik penerima, metrik keterlibatan pesan, percobaan pengiriman ke alamat yang tidak valid, konten yang ditandai oleh temuan filter spam, dan jebakan spam. Ini bukan daftar yang lengkap; faktor lain dapat menyebabkan organisasi ini menandai konten Anda.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Untuk mengatasi masalah ini, Anda perlu menentukan aspek program pengiriman email yang mungkin menyebabkan organisasi anti-spam menandai email Anda sebagai bermasalah. Anda kemudian perlu mengubah program pengiriman Anda untuk mengatasi masalah tersebut.

Jika akun Anda sedang dalam peninjauan

Pada akhir periode peninjauan, jika organisasi anti-spam terus mengidentifikasi email yang dikirim dari akun Anda sebagai bermasalah, kami mungkin menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan detail perubahan yang Anda buat. Saat kami menerima informasi ini, kami akan memperpanjang periode peninjauan untuk memastikan bahwa kami hanya menganalisis notifikasi organisasi anti-spam yang kami terima setelah Anda menerapkan perubahan. Pada akhir periode peninjauan yang diperpanjang ini, akun Anda tidak lagi terdaftar oleh organisasi anti-spam, kami akan menghapus periode peninjauan untuk akun Anda.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Pemberitahuan Listbombing

Bagian ini berisi informasi tambahan tentang notifikasi Listbombing yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Organisasi anti-spam telah mengidentifikasi bahwa proses pengiriman email Anda rentan terhadap “listbombing.” Listbombing adalah bentuk penyalahgunaan di mana penyerang mendaftarkan sejumlah besar alamat email pada formulir berbasis web. Listbombing dapat mengakibatkan gangguan layanan bagi pengguna layanan email yang terkena dampak. Hal ini juga dapat mengakibatkan email Anda diblokir oleh penyedia email.

Organisasi anti-spam menggunakan metode eksklusif untuk mengidentifikasi situs yang rentan terhadap listbombing. Untuk alasan ini, kami tidak dapat memberikan detail tambahan tentang masalah yang menyebabkan organisasi anti-spam mengidentifikasi proses pengiriman email Anda sebagai masalah. Kami juga tidak dapat membagikan nama organisasi yang mengidentifikasi masalah.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Anda harus memeriksa semua formulir pendaftaran berbasis web Anda untuk memastikan bahwa mereka tidak rentan terhadap penyalahgunaan semacam ini. Setiap formulir harus menyertakan CAPTCHA untuk mencegah skrip otomatis mengirimkan permintaan berlangganan. Selain itu, ketika pengguna baru mendaftar untuk produk atau layanan Anda, kirimkan email kepada mereka untuk mengonfirmasi bahwa mereka melakukannya, pada kenyataannya, bermaksud untuk mendaftar. Jangan mengirim email tambahan kepada pelanggan kecuali mereka secara eksplisit ikut serta dalam komunikasi Anda.

Akhirnya, Anda harus melakukan “izin lulus” pada daftar email Anda. Dalam izin izin, Anda mengirim email ke semua pelanggan Anda menanyakan apakah mereka masih ingin menerima email dari Anda. Hanya kirim email ke pelanggan yang memverifikasi bahwa mereka ingin terus menerima email dari Anda.

Jika akun Anda sedang dalam peninjauan

Pada akhir periode peninjauan, jika organisasi anti-spam terus mengidentifikasi email yang dikirim dari akun Anda sebagai bermasalah, kami mungkin menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan detail perubahan yang Anda buat. Saat kami menerima informasi ini, kami akan memperpanjang periode peninjauan untuk memastikan bahwa kami hanya menganalisis notifikasi organisasi anti-spam yang kami terima setelah Anda menerapkan perubahan. Pada akhir periode peninjauan yang diperpanjang ini, akun Anda tidak lagi terdaftar oleh organisasi anti-spam, kami akan menghapus periode peninjauan untuk akun Anda.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Notifikasi umpan balik langsung

Bagian ini berisi informasi tambahan tentang notifikasi umpan balik langsung yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Sejumlah besar pengguna telah menghubungi Amazon SES secara langsung untuk melaporkan pesan yang mereka terima dari alamat atau domain yang terkait dengan akun Amazon SES Anda. Jenis umpan balik ini tidak terlihat dalam aduan yang dilaporkan oleh penyedia kotak surat secara langsung, dan tidak disertakan dalam metrik pentalan dan aduan yang ditampilkan di halaman metrik reputasi.

Untuk melindungi privasi pengguna yang melaporkan masalah ini, kami tidak dapat memberikan alamat email mereka.

Penerima dapat mengadu ke Amazon SES saat mereka menerima pesan yang tidak mereka daftarkan untuk diterima, saat mereka tidak menerima jenis email yang mereka harapkan, saat mereka tidak menganggap email yang mereka terima berguna atau menarik, saat mereka tidak menyadari bahwa pesan tersebut adalah sesuatu yang mereka daftarkan, atau saat mereka

menerima terlalu banyak pesan. Daftar ini tidak lengkap; faktor yang relevan dalam kasus Anda tergantung pada program pengiriman email tertentu.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Kami menyarankan Anda untuk menerapkan strategi opt-in ganda, seperti yang dijelaskan dalam Membangun dan mempertahankan daftar Anda, untuk memperoleh alamat baru, dan Anda hanya mengirim email ke alamat yang menyelesaikan proses opt-in ganda.

Selain itu, Anda harus menghapus daftar alamat yang akhir-akhir ini belum berinteraksi dengan email Anda. Anda dapat menggunakan pelacakan buka dan klik, seperti yang dijelaskan di Memantau aktivitas pengiriman Amazon SES, untuk menentukan pengguna yang melihat dan berinteraksi dengan konten yang Anda kirim.

Jika akun Anda sedang dalam peninjauan

Pada akhir periode peninjauan, jika Amazon SES terus menerima sejumlah besar aduan langsung tentang pesan yang dikirim dari akun Anda, kami mungkin menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah ini mencegah masalah terjadi lagi di masa mendatang. Jika kami setuju bahwa perubahan yang telah Anda buat mengatasi masalah tersebut dengan tepat, kami membatalkan periode peninjauan pada akun Anda.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Notifikasi Daftar Blokir Domain

Bagian ini berisi informasi tambahan tentang notifikasi daftar blokir domain yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Email yang dikirim dari akun Amazon SES Anda berisi referensi ke domain yang telah terdaftar di Daftar Blokir Domain terkemuka. Domain pada daftar ini biasanya terkait dengan penyalahgunaan atau perilaku berbahaya. Domain yang dimaksud bisa jadi atau mungkin bukan domain tempat Anda mengirim email. Pesan yang menyertakan referensi atau tautan ke domain di daftar blokir, atau yang menyertakan citra yang dihosting di domain tersebut, mungkin juga ditandai.

Kami tidak dapat memberikan nama domain yang menyebabkan pesan Anda ditandai, atau untuk mengidentifikasi email yang ditandai dengan cara ini.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Pertama, buat daftar semua domain yang direferensikan dalam email yang Anda kirim melalui Amazon SES. Selanjutnya, gunakan Alat Pencarian Domain Spamhaus untuk menentukan domain di email Anda yang ada di daftar blokir domain. Lebih dari satu domain yang direferensikan dalam email yang Anda kirim mungkin ada di daftar blokir ini.

Daftar Blokir Domain Spamhaus tidak berafiliasi dengan Amazon SES atau AWS. Kami tidak menjamin keakuratan domain dalam daftar ini. Daftar Blokir Domain Spamhaus dan Alat Pencarian Domain dimiliki, dioperasikan, dan dikelola oleh Spamhaus Project.

Jika akun Anda sedang dalam peninjauan

Kami mencari referensi ke domain yang telah digunakan untuk tujuan berbahaya di email yang Anda kirim selama periode peninjauan. Jika email Anda masih berisi sejumlah besar referensi yang signifikan ke domain tersebut, kami dapat menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan detail perubahan yang Anda buat. Ketika kami menerima informasi ini, kami memperpanjang periode peninjauan untuk memastikan bahwa kami hanya menganalisis jumlah domain yang diblokir yang ada di email Anda setelah Anda menerapkan perubahan. Pada akhir

periode peninjauan yang diperpanjang ini, jika jumlah notifikasi daftar blokir domain telah dikurangi atau dihilangkan, dan kami percaya bahwa Anda telah mengambil langkah-langkah untuk mencegah masalah ini terjadi lagi di masa mendatang, kami membatalkan periode peninjauan untuk akun Anda.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Notifikasi Peninjauan Internal

Bagian ini berisi informasi tambahan tentang notifikasi peninjauan internal yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Tinjauan menyeluruh atas akun Anda mengidentifikasi beberapa karakteristik yang dapat menyebabkan penyedia kotak surat atau penerima mengidentifikasi pesan Anda sebagai spam.

Untuk melindungi proses deteksi penyalahgunaan, kami tidak dapat mengungkapkan faktor-faktor spesifik yang menyebabkan akun Anda ditandai dengan cara ini.

Faktor umum yang menyebabkan penentuan ini meliputi:

- Pesan yang ditandai oleh sistem anti-spam komersial.
- Konten pesan yang menyiratkan penerima belum meminta email secara eksplisit.
- Ketidaksesuaian antara pengirim pesan dan pencitraan merek dalam isi email.
- Konten yang tidak memperjelas pengirimnya.
- Mengirim pesan yang berhubungan dengan konten yang terkait dengan email yang tidak diminta.
- Memformat pola yang terkait dengan email yang tidak diminta.
- Mengirim dari atau membuat referensi ke domain dengan reputasi buruk.

Ini bukan daftar lengkap. Alasan khusus untuk notifikasi ini mungkin merupakan kombinasi dari salah satu faktor ini, atau alasannya mungkin sesuatu yang tidak tercantum.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Saran berikut dapat membantu mengurangi tingkat kepelikan masalah:

- Pastikan bahwa satu-satunya penerima yang Anda hubungi adalah mereka yang secara eksplisit diminta untuk menerima email dari Anda.
- Jangan pernah membeli, menyewa, atau meminjam daftar penerima email.
- Jangan mencoba menyembunyikan identitas atau tujuan komunikasi Anda dalam pesan yang Anda kirim.
- Buat daftar semua domain yang dirujuk dalam email yang Anda kirim melalui Amazon SES, dan kemudian gunakan alat Spamhaus Domain Lookup di https://www.spamhaus.org/lookup/ untuk menentukan salah satu domain tersebut terdapat dalam Daftar Blokir Domain Spamhaus.
- Pastikan bahwa Anda mengikuti praktik terbaik industri saat mendesain email Anda.

Daftar ini tidak lengkap, namun akan membantu Anda mengidentifikasi beberapa faktor paling umum yang mungkin menyebabkan email Anda ditandai.

Daftar Blokir Domain Spamhaus tidak berafiliasi dengan Amazon SES atau AWS. Kami tidak menjamin keakuratan domain dalam daftar ini. Daftar Blokir Domain Spamhaus dan Alat Pencarian Domain dimiliki, dioperasikan, dan dikelola oleh Spamhaus Project.

Jika akun Anda sedang dalam peninjauan, atau jika kemampuan akun Anda untuk mengirim email dijeda

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah ini mencegah masalah terjadi lagi di masa mendatang. Jika kami setuju bahwa perubahan yang telah Anda buat untuk mengatasi masalah telah tepat, kami membatalkan periode peninjauan atau menghapus jeda pengiriman dari akun Anda.

Jika kami menghapus periode peninjauan atau jeda pengiriman dari akun Anda, dan kami mengamati masalah yang sama di lain waktu, kami mungkin menempatkan akun Anda dalam peninjauan atau menjeda kemampuan Anda untuk mengirim email lagi. Dalam kasus ekstrem, atau jika kami

mengamati instans berulang dari masalah yang sama, kami mungkin menangguhkan kemampuan akun Anda untuk mengirim email secara permanen.

Lihat FAQ proses peninjauan Pengiriman Amazon SES untuk informasi lebih lanjut tentang tindakan yang harus dilakukan jika akun Anda sedang dalam peninjauan, atau kemampuan mengirim email akun Anda dijeda.

Notifikasi Penyedia Kotak Surat

Bagian ini berisi informasi tambahan tentang notifikasi penyedia kotak surat yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Penyedia kotak surat utama telah melaporkan kepada kami bahwa email yang tidak diminta atau berbahaya dikirim dari alamat atau domain yang terkait dengan akun Amazon SES Anda.

Kami tidak dapat membagikan identitas organisasi yang mengeluarkan laporan ini. Selain itu, kami tidak memiliki informasi tentang faktor-faktor tertentu yang menyebabkan penyedia kotak surat mengeluarkan laporan. Biasanya, penyedia kotak surat membuat penentuan semacam ini berdasarkan umpan balik pelanggan, metrik keterlibatan pelanggan, upaya pengiriman ke alamat yang tidak valid, dan konten yang ditandai oleh filter spam. Daftar ini tidak lengkap; mungkin ada faktor lain yang menyebabkan penyedia kotak surat menandai konten Anda.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Untuk mengatasi masalah ini, Anda perlu menentukan aspek program pengiriman email yang mungkin telah menyebabkan penyedia kotak surat untuk menandai surat Anda sebagai bermasalah. Anda kemudian harus mengubah program pengiriman Anda untuk mengatasi masalah tersebut.

Jika akun Anda sedang dalam peninjauan

Pada akhir periode peninjauan, jika penyedia kotak surat terus mengidentifikasi email yang dikirim dari akun Anda sebagai bermasalah, kami dapat menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan detail perubahan yang Anda buat. Saat kami menerima informasi ini, kami akan memperpanjang periode peninjauan untuk memastikan bahwa kami hanya menganalisis jumlah notifikasi penyedia kotak surat yang kami terima setelah Anda menerapkan perubahan. Pada akhir

periode peninjauan yang diperpanjang ini, jika penyedia kotak surat tidak lagi melaporkan akun Anda sebagai bermasalah, kami dapat menghapus peninjauan dari akun Anda.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Notifikasi Umpan Balik Penerima

Bagian ini berisi informasi tambahan tentang notifikasi umpan balik penerima yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Penyedia kotak surat utama telah melaporkan kepada kami bahwa sejumlah besar pengguna mereka melaporkan email yang dikirim dari akun Amazon SES Anda sebagai tidak diminta. Jenis umpan balik ini tidak terlihat dalam keluhan yang dilaporkan oleh penyedia kotak surat secara langsung, dan tidak disertakan dalam notifikasi pentalan dan aduan Amazon SES.

Sejumlah besar aduan dapat berdampak negatif pada semua pengguna Amazon SES. Untuk melindungi reputasi Anda dan pelanggan Amazon SES lainnya, kami mengambil tindakan segera ketika akun menerima sejumlah aduan.

Kami tidak dapat memberikan daftar alamat email spesifik yang melaporkan email Anda sebagai tidak diminta. Selain itu, kami tidak dapat membagikan nama penyedia kotak surat yang telah melaporkan masalah ini kepada kami.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Untuk mengatasi masalah ini, Anda perlu menentukan aspek program pengiriman email yang mungkin menyebabkan penerima Anda mengeluarkan aduan terhadap pesan email yang mereka terima dari Anda. Setelah Anda mengidentifikasi faktor-faktor ini, ubah praktik pengiriman email Anda untuk memperbaikinya.

Untuk mendapatkan alamat baru, kami menyarankan Anda menerapkan strategi penyertaan ganda, seperti yang dijelaskan di Membangun dan mempertahankan daftar Anda. Kami menyarankan Anda hanya mengirim email ke alamat yang telah menyelesaikan proses penyertaan ganda.

Selain itu, Anda harus menghapus daftar alamat yang akhir-akhir ini belum berinteraksi dengan email Anda. Anda dapat menggunakan pelacakan buka dan klik, seperti yang dijelaskan di Memantau aktivitas pengiriman Amazon SES, untuk menentukan pengguna yang melihat dan berinteraksi dengan konten yang Anda kirim.

Jika akun Anda sedang dalam peninjauan

Pada akhir periode peninjauan, jika penyedia kotak surat terus melaporkan sejumlah besar aduan, kami dapat menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan detail perubahan yang Anda buat. Saat kami menerima informasi ini, kami memperpanjang periode peninjauan untuk memastikan bahwa kami hanya menganalisis jumlah aduan penyedia kotak surat yang kami terima setelah Anda menerapkan perubahan. Pada akhir periode peninjauan yang diperpanjang ini, jika jumlah aduan penyedia kotak surat telah dikurangi atau dihilangkan, kami dapat menghapus peninjauan dari akun Anda.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Notifikasi Akun Terkait

Bagian ini berisi informasi tambahan tentang notifikasi akun terkait yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Kami telah mendeteksi masalah serius yang terkait dengan email yang dikirim dari akun Amazon SES lain. Kami percaya bahwa akun yang bermasalah terkait dengan Akun AWS Anda, jadi kami telah mengambil tindakan untuk menghindari masalah serupa.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Saat kami menjeda kemampuan akun untuk mengirim email, kami selalu mengirimkan informasi tentang alasan jeda pengiriman kepada pemilik akun tersebut. Lihat email yang kami kirimkan kepada pemilik akun terkait untuk informasi lebih lanjut.

Anda harus mengatasi masalah dengan akun terkait terlebih dahulu. Setelah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah ini mencegah masalah terjadi lagi di masa mendatang. Jika kami setuju bahwa perubahan yang telah Anda buat untuk mengatasi masalah telah tepat, kami membatalkan periode peninjauan atau menghapus jeda pengiriman dari akun Anda.

Notifikasi Jebakan Spam

Bagian ini berisi informasi tambahan tentang notifikasi jebakan spam yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Organisasi anti-spam pihak ketiga telah melaporkan kepada kami bahwa alamat jebakan spam mereka baru-baru ini menerima email dari alamat terverifikasi atau domain yang terkait dengan akun Amazon SES Anda.

Jebakan spam adalah alamat email aktif yang digunakan secara eksklusif untuk memikat email yang tidak diminta (spam). Sejumlah besar laporan jebakan spam dapat memiliki dampak negatif pada semua pengguna Amazon SES. Untuk melindungi reputasi Anda dan reputasi pelanggan Amazon SES lainnya, kami segera mengambil tindakan saat akun mengirimkan volume email tertentu ke alamat jebakan spam.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Kami tidak dapat mengungkapkan alamat email yang terkait dengan jebakan spam yang Anda temui. Alamat ini dijaga ketat oleh organisasi yang memilikinya, dan setelah alamat diketahui, mereka menjadi tidak berguna.

Mengirim email ke alamat jebakan spam biasanya menunjukkan bahwa ada masalah dengan cara Anda memperoleh alamat email pelanggan Anda. Misalnya, daftar alamat email yang dibeli dapat berisi alamat jebakan spam, itulah sebabnya pengiriman ke daftar yang dibeli atau disewa dilarang oleh persyaratan layanan Amazon SES. Untuk mendapatkan alamat baru, kami menyarankan Anda menerapkan strategi penyertaan ganda, seperti yang dijelaskan di Membangun dan mempertahankan daftar Anda. Kami menyarankan Anda hanya mengirim email ke alamat yang telah menyelesaikan proses penyertaan ganda.

Selain itu, Anda harus menghapus daftar alamat yang akhir-akhir ini belum berinteraksi dengan email Anda. Anda dapat menggunakan pelacakan buka dan klik, seperti yang dijelaskan di Memantau aktivitas pengiriman Amazon SES, untuk menentukan pengguna yang melihat dan berinteraksi dengan konten yang Anda kirim.

Jika akun Anda sedang dalam peninjauan

Pada akhir periode peninjauan, jika pesan masih dikirim ke alamat jebakan spam dari akun Anda, kami mungkin menjeda kemampuan akun Anda untuk mengirim email hingga Anda menyelesaikan masalah tersebut.

Jika Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan detail perubahan yang Anda buat. Ketika kami menerima informasi ini, kami memperpanjang periode peninjauan untuk memastikan bahwa kami hanya menganalisis jumlah laporan jebakan spam yang kami terima setelah Anda menerapkan perubahan. Pada akhir periode peninjauan yang diperpanjang ini, jika jumlah laporan jebakan spam telah dikurangi atau dihilangkan, kami dapat menghapus peninjauan dari akun Anda.

Jika kemampuan akun Anda untuk mengirim email dijeda

Anda dapat meminta agar kami mempertimbangkan kembali keputusan ini. Untuk informasi selengkapnya, lihat FAQ proses peninjauan Pengiriman Amazon SES.

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan

detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Notifikasi Situs Rentan

Bagian ini berisi informasi tambahan tentang notifikasi situs rentan yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Tinjauan komprehensif menemukan bahwa ada pesan yang dikirim dari akun Anda yang menurut kami tidak Anda maksudkan untuk dikirim. Pesan ini sangat mungkin ditandai sebagai spam oleh penyedia kotak surat dan penerima.

Dalam situasi ini, pihak ke tiga sering kali menyalahgunakan fitur situs web Anda untuk mengirim email yang tidak diinginkan. Misalnya, jika situs web Anda berisi “email ke teman”, “kontak kami”, “undang teman”, atau fitur serupa, pihak ke tiga dapat menggunakan fitur tersebut untuk mengirim email yang tidak diminta.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Pertama, identifikasi fitur dari situs web atau aplikasi Anda yang mengizinkan pihak ke tiga untuk mengirim email menggunakan Amazon SES tanpa sepengetahuan Anda. Dalam kasus Pusat Dukungan, Anda dapat meminta contoh pesan yang kami yakini dikirim dengan cara ini.

Selanjutnya, ubah aplikasi atau situs web Anda untuk mencegah pengiriman yang tidak diminta. Misalnya, tambahkan CAPTCHA, batasi laju pengiriman email, hapus kemampuan pengguna untuk mengirimkan konten tertentu, mengharuskan pengguna masuk untuk mengirim email, dan menghapus kemampuan aplikasi untuk menghasilkan beberapa notifikasi secara bersamaan.

Jika akun Anda sedang dalam peninjauan, atau jika kemampuan akun Anda untuk mengirim email dijeda

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Jika kami menghapus periode peninjauan atau mengirim jeda dari akun Anda, dan kami menemukan masalah yang sama di kemudian hari, kami mungkin menempatkan akun Anda dalam peninjauan atau menjeda kemampuan Anda untuk mengirim email lagi. Jika kami mengamati masalah ekstrim atau instans berulang dari masalah yang sama, kami mungkin menangguhkan kemampuan akun Anda untuk mengirim email secara permanen.

Lihat FAQ proses peninjauan Pengiriman Amazon SES untuk informasi lebih lanjut tentang tindakan yang harus dilakukan jika akun Anda sedang dalam peninjauan, atau kemampuan mengirim email akun Anda dijeda.

Kredensi yang Dikompromikan

Bagian ini berisi informasi tambahan tentang notifikasi situs kredensi yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Tinjauan komprehensif menemukan bahwa ada pesan yang dikirim dari akun Anda yang menurut kami tidak Anda maksudkan untuk dikirim. Pesan ini sangat mungkin ditandai sebagai spam oleh penyedia kotak surat dan penerima.

Beberapa penyebab umum adalah kunci akses IAM yang dikompromikan, kata sandi SMTP yang dikompromikan, atau kerentanan keamanan lainnya.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Anda harus melakukan tinjauan keamanan komprehensif mekanisme pemanfaatan SES Anda. Pastikan Anda telah memutar kata sandi yang berlaku atau SMTP dan bahwa Anda telah menghapus pengguna atau sumber daya yang tidak sah dari akun Anda. Pastikan Anda tidak menyimpan informasi sensitif seperti kata sandi atau kunci akses di situs web atau repositori pihak ketiga. Sekarang disarankan agar Anda tidak menggunakan kunci akses IAM untuk pengguna, dan tidak pernah untuk pengguna root. Jika Anda masih menggunakannya, Anda harus memigrasinya ke mekanisme yang memberikan kredensi sementara seperti membuat penggunaAWS IAM Identity Center.

Jika akun Anda sedang dalam peninjauan, atau jika kemampuan akun Anda untuk mengirim email dijeda

Bila Anda telah menerapkan perubahan yang Anda yakini dapat menyelesaikan masalah, masuk ke Konsol AWS dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Sertakan

detail tindakan yang telah Anda ambil untuk menyelesaikan masalah ini, serta detail rencana Anda untuk memastikan bahwa masalah ini tidak terjadi lagi. Setelah menerima permintaan Anda, kami meninjau informasi yang Anda beri dan mengubah status akun Anda jika diperlukan.

Jika kami menghapus periode peninjauan atau mengirim jeda dari akun Anda, dan kami menemukan masalah yang sama di kemudian hari, kami mungkin menempatkan akun Anda dalam peninjauan atau menjeda kemampuan Anda untuk mengirim email lagi. Jika kami mengamati masalah ekstrim atau instans berulang dari masalah yang sama, kami mungkin menangguhkan kemampuan akun Anda untuk mengirim email secara permanen.

Lihat FAQ proses peninjauan Pengiriman Amazon SES untuk informasi lebih lanjut tentang tindakan yang harus dilakukan jika akun Anda sedang dalam peninjauan, atau kemampuan mengirim email akun Anda dijeda.

Notifikasi lainnya

Bagian ini berisi informasi tambahan tentang notifikasi lain yang ditampilkan di halaman metrik reputasi Amazon SES.

Alasan Anda menerima notifikasi ini

Peninjauan otomatis atau oleh manusia telah mengidentifikasi masalah yang tidak tercantum di bagian sebelumnya dari dokumen ini.

Hal yang dapat Anda lakukan untuk menyelesaikan masalah

Lihat kasus Pusat Dukungan yang kami buka atas nama Anda untuk detail tentang masalah tertentu. Untuk mengakses Pusat Dukungan, masuk ke AWS Management Console dan kemudian pilih Pusat Dukungan. Dalam tanggapan Anda terhadap kasus, jelaskan perubahan yang Anda terapkan. Bergantung pada situasi spesifik Anda dan sifat masalah yang kami temukan, kami mungkin mengakhiri periode peninjauan atau memulihkan kemampuan akun Anda untuk mengirim email.

Membuat alarm pemantauan reputasi menggunakan CloudWatch

Amazon SES secara otomatis menerbitkan serangkaian metrik yang terkait reputasi ke Amazon. CloudWatch Anda dapat menggunakan metrik ini untuk membuat alarm yang memberi tahu Anda ketika tingkat pentalan atau aduan mencapai tingkat yang dapat memengaruhi kemampuan akun Anda untuk mengirim email.

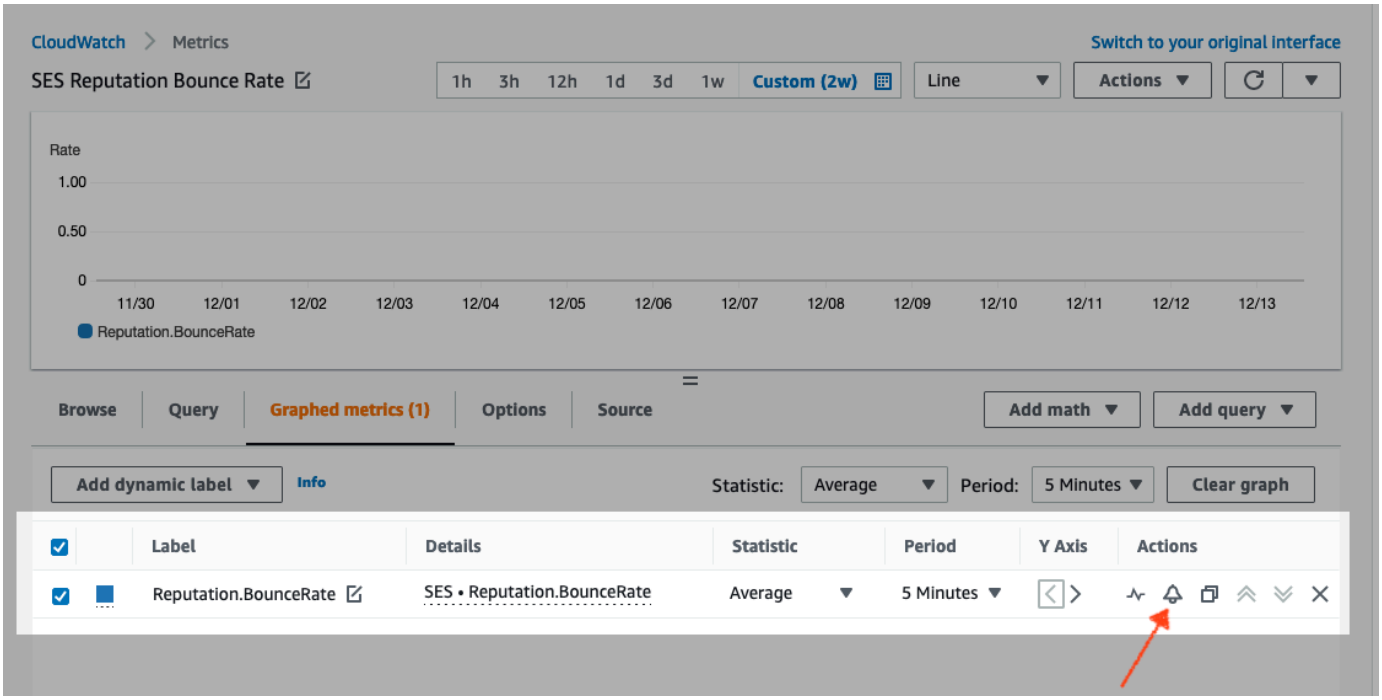
> **Note**
>
> CloudWatch Bagian dari prosedur di bagian ini dimaksudkan untuk hanya menyajikan langkah-langkah inti untuk mengatur CloudWatch alarm untuk memantau reputasi pengirim SES Anda. Mereka tidak mengeksplorasi konfigurasi lanjutan mengenai pengaturan opsional untuk CloudWatch alarm. Untuk informasi selengkapnya tentang mengonfigurasi CloudWatch alarm, lihat Menggunakan CloudWatch alarm Amazon di Panduan Pengguna Amazon CloudWatch .

Prasyarat

- Membuat topik Amazon SNS, lalu berlangganan menggunakan titik akhir pilihan Anda (seperti email atau SMS). Untuk informasi selengkapnya, lihat Membuat topik Amazon SNS dan Berlangganan topik Amazon SNS di Panduan Developer Amazon Simple Notification Service.
- Jika Anda belum pernah mengirim email di Wilayah saat ini, Anda mungkin tidak melihat namespace SES. Untuk memastikan bahwa Anda memiliki metrik, kirim email pengujian ke simulator kotak surat.

Untuk membuat CloudWatch alarm untuk memantau reputasi pengiriman

1. Masuk ke AWS Management Console dan buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di panel navigasi pada sisi kiri layar, pilih Metrik reputasi.
3. Pada halaman Metrik reputasi di bawah tab Tingkat akun, baik di panel Bounce rate atau Complaint rate, pilih View in CloudWatch - ini akan membuka CloudWatch konsol dengan metrik yang Anda pilih.
4. Di bawah tab Metrik grafik, pada baris metrik yang Anda pilih, untuk contoh ini, Reputasi. BounceRate, pilih ikon bel alarm di kolom Tindakan (lihat gambar di bawah) - ini akan membuka halaman Tentukan metrik dan kondisi.



5. Gulir ke bawah ke panel Kondisi, dan pilih Statis di bidang Jenis ambang batas.

   a. Dalam ***metrik*** Kapanpun adalah... bidang, pilih Greater/Equal.

   b. Dalam dari... bidang, tentukan nilai yang seharusnya dapat CloudWatch menyebabkan alarm.

      - Jika Anda membuat alarm untuk memantau rasio pentalan Anda, perhatikan bahwa Amazon SES merekomendasikan agar Anda mempertahankan rasio pentalan di bawah 5%. Jika rasio pentalan untuk akun Anda lebih besar dari 10%, kami akan menjeda kemampuan akun Anda untuk mengirim email. Karena alasan ini, Anda harus mengonfigurasi CloudWatch untuk mengirimkan notifikasi bila rasio pentalan untuk akun Anda lebih besar dari atau sama dengan 0,05 (5%).
      - Jika Anda membuat alarm untuk memantau tingkat aduan Anda, perhatikan bahwa Amazon SES merekomendasikan Anda untuk mempertahankan tingkat aduan di bawah 0,1%. Jika tingkat aduan untuk akun Anda lebih besar dari 0,5%, kami akan menjeda kemampuan akun Anda untuk mengirim email. Karena alasan ini, Anda harus mengonfigurasi CloudWatch untuk mengirimkan notifikasi bila tingkat aduan pada akun Anda lebih besar dari atau sama dengan 0,001 (0,1%).

   c. Perluas Konfigurasi tambahan dan pilih Perlakukan status data yang hilang menjadi abaikan (pertahankan status alarm) di bidang Perawatan data yang hilang.

   d. Pilih Selanjutnya.

6. Pada panel Konfigurasi tindakan, pilih Dalam Alarm di bidang Pemicu status alarm.
   a. Pilih Pilih topik SNS yang sudah ada di bidang Pilih topik SNS.
   b. Pilih topik yang Anda buat dan berlangganan di prasyarat di Kirim notifikasi ke... kotak pencarian.
   c. Pilih Selanjutnya.
7. Di panel Tambahkan nama dan deskripsi, masukkan nama dan penjelasan untuk alarm, lalu pilih Berikutnya.
8. Pada panel Pratinjau dan buat, konfirmasikan pengaturan Anda, dan jika puas, pilih Buat alarm. Jika ada sesuatu yang ingin Anda ubah, pilih tombol Sebelumnya untuk setiap bagian yang ingin Anda kembalikan dan edit.

Metrik SNDS untuk IP khusus

Anda dapat melihat data Smart Network Data Services (SNDS) untuk alamat IP khusus yang disewakan di setiap Wilayah AWS tempat Anda menggunakan Amazon SES. Data SNDS ini tersedia melalui konsol Amazon CloudWatch.

SNDS adalah program Outlook yang memungkinkan pemilik IP membantu mencegah spam dalam ruang IP mereka. Amazon SES menyediakan data penting ini bagi mereka yang menyewa IP khusus. Data SNDS memberikan wawasan tentang perilaku pengiriman email IP dan memanggil area yang menjadi perhatian untuk reputasi pengirim Anda.

> **Note**
>
> Apabila merujuk ke Outlook, hal ini mencakup semua domain yang mereka lacak. Misalnya, hal ini dapat mencakup Hotmail.com, Outlook.com, dan Live.com.

Untuk melihat data SNDS untuk alamat IP khusus Anda

1. Masuk ke Amazon CloudWatch konsol dihttps://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, perluasMetrikdan memiilihSemua metrik.

   (Arah diberikan untuk yang baru CloudWatch antarmuka konsol.)
3. Di bawahJelajahitab diMetrikkontainer, pilihWilayah AWS, lalu pilihSES.
4. PilihMetrik IPyang akan menampilkan semua IP khusus yang dilacak oleh SNDS.

(Catatan: jika tidak ada alamat IP khusus yang terkait dengan akun Anda di wilayah yang dipilih,Metrik IPtidak akan muncul dalam CloudWatch konsol.)

5. Lihat semua IP khusus yang dilacak oleh SNDS dalam daftar ini, atau pilih alamat IP individual untuk melihat metriknya saja.

Metrik berikut disediakan untuk setiap alamat IP khusus dan ditentukan oleh Outlook. Untuk informasi lebih lanjut, lihat FAQ SNDS Outlook.

> **Note**
>
> Metrik ini mewakili periode aktivitas yang menyediakan data terbaru sekali sehari. Metrik juga memiliki stempel waktu yang sesuai, yang mencerminkan periode 24 jam.

- SNDS.RCPTCommands - Ini adalah jumlah perintah RCPT yang diterima oleh SNDS untuk alamat IP tertentu selama periode aktivitas. Perintah RCPT adalah bagian dari protokol SMTP yang digunakan untuk mengirim surat, yang menentukan alamat penerima yang Anda coba kirimkan email.
- SNDS.DATACommands - Jumlah perintah DATA yang diterima oleh SNDS untuk alamat IP tertentu selama periode aktivitas. Perintah DATA adalah bagian dari protokol SMTP yang digunakan untuk mengirim surat, khususnya bagian yang benar-benar mentransmisikan pesan ke penerima yang dimaksudkan sebelumnya (s).
- SNDS.MessageRecipients - Jumlah penerima pada pesan yang diterima oleh SNDS untuk alamat IP tertentu selama periode aktivitas.
- SNDS.SpamRate - Menampilkan hasil gabungan dari pemfilteran spam yang diterapkan ke semua pesan yang dikirim oleh alamat IP selama periode aktivitas tertentu.
  - SEBUAH SpamRate 0 berarti alamat IP memiliki spam kurang dari 10%.
  - SEBUAH SpamRate 0,5 berarti bahwa 10% dan 90% spam dihasilkan dari alamat IP.
  - SEBUAH SpamRate 1 berarti 90% atau lebih spam dihasilkan dari alamat IP.
- SNDS.ComplaintRate - Merupakan sebagian kecil dari waktu diterimanya pesan dari IP yang diadukan oleh pengguna Outlook selama periode aktivitas.
  - SEBUAH ComplaintRate 1 berarti tingkat aduannya sebesar 100%.
  - SEBUAH ComplaintRate 0,05 berarti tingkat aduannya sebesar 5%.
  - SEBUAH ComplaintRate 0 berarti tingkat kurang dari 0,1%.

- SNDS.TrapHits - Menampilkan jumlah pesan yang dikirim ke "akun perangkap." Akun perangkap merupakan akun yang dikelola oleh Outlook yang tidak meminta email apa pun. Jadi, setiap pesan yang dikirim ke akun perangkap kemungkinan besar adalah spam.

Pertanyaan terkait pemecahan masalah

Q1. Mengapa data tidak terisi setiap hari? Salah satu skenario berikut dapat diterapkan:

- Data SNDS bergantung pada program SNDS Outlook.
- Terdapat ambang minimum dari email SNDS yang perlu diterima untuk menghitung nilai. Data mungkin tidak tersedia ketika volume email pada IP sedang rendah.

Q2. Mengapa metrik SNDS.SpamRate dan SNDS.ComplaintRate berubah, dan apa yang harus saya lakukan jika nilai berubah menjadi nilai 1?

Hal ini merupakan indikator bahwa sesuatu dalam perilaku pengiriman Anda telah memicu respons negatif dari program Outlook SNDS. Dalam hal ini, Anda ingin memeriksa Penyedia Layanan Internet (ISP) lainnya beserta jumlah keterlibatan Anda guna memastikan bahwa hal tersebut bukan masalah global. Jika hal tersebut adalah masalah global, Anda mungkin melihat masalah dengan beberapa ISP, yang akan menyarankan masalah daftar, konten, distribusi, atau izin. Jika hal tersebut spesifik untuk Outlook, tinjau cara terbaik untuk mengirimkan ke Outlook.

Q3. Tindakan apa yang akan diambil oleh AWS Support jika SNDS.SpamRate saya berubah dari nilai 0 (atau 0,5) ke 1?

AWS tidak memiliki kontrol atas SNDS, dan oleh karenanya, tidak berpengaruh terhadap SNDS. Semua permintaan mitigasi perlu diajukan langsung dengan Outlook melalui Formulir permintaan dukungan yang baru.

Menjeda pengiriman email secara otomatis

Untuk melindungi reputasi pengirim Anda, Anda dapat menjeda pengiriman email sementara waktu untuk pesan yang dikirim menggunakan set konfigurasi tertentu, atau untuk semua pesan yang dikirim dari akun Amazon SES Anda di Wilayah AWS spesifik.

Dengan menggunakan Amazon CloudWatch dan Lambda, Anda dapat membuat solusi yang secara otomatis menjeda pengiriman email ketika metrik reputasi Anda (seperti rasio pentalan atau tingkat aduan) melebihi ambang batas tertentu. Topik ini berisi prosedur untuk menyiapkan solusi ini.

Topik di bagian ini:

- Secara otomatis menjeda pengiriman email untuk seluruh akun Amazon SES
- Menjeda pengiriman email secara otomatis untuk satu set konfigurasi

Secara otomatis menjeda pengiriman email untuk seluruh akun Amazon SES

Prosedur di bagian ini menjelaskan langkah-langkah untuk mengatur Amazon SES, Amazon SNS, Amazon CloudWatch, dan AWS Lambda untuk secara otomatis menjeda pengiriman email untuk akun Amazon SES Anda dalam satu Wilayah. AWS Jika Anda mengirim email dari beberapa wilayah, ulangi prosedur di bagian ini untuk setiap wilayah tempat Anda ingin menerapkan solusi ini.

Topik di bagian ini:

- Bagian 1: Buat IAM role
- Bagian 2: Buat fungsi Lambda
- Bagian 3: Aktifkan Ulang Pengiriman Email untuk Akun Anda
- Bagian 4: Buat Topik dan Langganan Amazon SNS
- Bagian 5: Buat CloudWatch Alarm
- Bagian 6: Uji solusinya

Bagian 1: Buat IAM role

Langkah pertama mengonfigurasi penjedaan pengiriman email secara otomatis adalah untuk membuat IAM role yang dapat mengeksekusi operasi API `UpdateAccountSendingEnabled`.

Buat IAM role

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Pada halaman Pilih entitas tepercaya, pilih AWS layanan untuk jenis entitas Tepercaya.
5. Di bawah Kasus penggunaan, pilih Lambda, lalu pilih Berikutnya.
6. Pada halaman Tambahkan izin, pilih kebijakan berikut:

- AWSLambdaBasicExecutionRole
- Amazonses FullAccess

> **Tip**
>
> Gunakan kotak pencarian di bawah Kebijakan izin untuk menemukan kebijakan ini dengan cepat, tetapi perhatikan bahwa setelah mencari dan memilih kebijakan pertama, Anda harus memilih Hapus filter sebelum mencari dan memilih kebijakan kedua.

Lalu pilih Selanjutnya.

7. Pada halaman Nama, tinjau, dan buat, di bawah Detail peran, masukkan nama yang berarti untuk kebijakan di bidang Nama peran.
8. Verifikasi bahwa dua kebijakan yang Anda pilih tercantum dalam tabel ringkasan kebijakan izin, lalu pilih Buat peran.

Bagian 2: Buat fungsi Lambda

Setelah membuat IAM role, Anda dapat membuat fungsi Lambda yang dapat menghentikan pengiriman email untuk akun Anda.

Untuk membuat fungsi Lambda

1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Gunakan pemilih wilayah untuk memilih wilayah tempat Anda ingin men-deploy fungsi Lambda ini.

> **Note**
>
> Fungsi ini hanya menjeda pengiriman email di AWS Wilayah yang Anda pilih pada langkah ini. Jika Anda mengirim email dari beberapa wilayah, ulangi prosedur di bagian ini untuk setiap wilayah tempat Anda ingin menjeda pengiriman email secara otomatis.

3. Pilih Buat fungsi.
4. Di bawah Buat fungsi, pilih Penulis dari scratch.

5. Di bawah Informasi dasar, selesaikan langkah-langkah berikut:

   - Untuk nama Fungsi, ketik nama untuk fungsi Lambda.
   - Untuk Runtime, pilih Node.js 18x (atau versi yang saat ini ditawarkan dalam daftar pilih).
   - Untuk Arsitektur, pertahankan default yang telah dipilih sebelumnya, x86_64.
   - Di bawah Izin, perluas Ubah peran eksekusi default dan pilih Gunakan peran yang ada.
   - Klik di dalam kotak daftar peran yang ada, dan pilih peran IAM yang Anda buat. the section called “Bagian 1: Buat IAM role”

   Lalu pilih Buat fungsi.

6. Di bawah Sumber kode, di editor kode, tempel kode berikut:

```
'use strict';


const { SES } = require("@aws-sdk/client-ses")

// Create a new SES object.

var ses = new 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);
        }
    });
};
```

   Kemudian pilih Deploy.

7. Pilih Uji. Jika jendela Configure test event muncul, ketikkan nama di bidang Nama acara, lalu pilih Simpan.
8. Perluas kotak drop Test dan pilih nama acara yang baru saja Anda buat, lalu pilih Test.
9. Tab Hasil Eksekusi akan muncul - tepat di bawahnya dan ke kanan, pastikan `Status: Succeeded` itu ditampilkan. Jika fungsi gagal dijalankan, lakukan hal berikut:
   - Verifikasikan bahwa IAM role yang Anda buat di the section called “Bagian 1: Buat IAM role” berisi kebijakan yang benar.
   - Verifikasi bahwa kode dalam fungsi Lambda tidak berisi kesalahan. Editor kode Lambda secara otomatis menyoroti kesalahan sintaksis dan potensi masalah lainnya.

Bagian 3: Aktifkan Ulang Pengiriman Email untuk Akun Anda

Efek samping dari pengujian fungsi Lambda di the section called “Bagian 2: Buat fungsi Lambda” adalah bahwa pengiriman email untuk akun Amazon SES Anda dijeda. Dalam kebanyakan kasus, Anda tidak ingin menjeda pengiriman untuk akun Anda sampai CloudWatch alarm dipicu.

Prosedur di bagian ini mengaktifkan kembali pengiriman email untuk akun Amazon SES Anda. Untuk menyelesaikan prosedur ini, Anda harus menginstal dan mengonfigurasi AWS Command Line Interface. Untuk informasi lebih lanjut, lihat Panduan Pengguna AWS Command Line Interface.

Untuk mengaktifkan kembali pengiriman email

1. Pada baris perintah, ketik perintah berikut ini untuk mengaktifkan kembali pengiriman email untuk akun Anda. Ganti *`sending_region`* dengan nama Wilayah tempat Anda ingin mengaktifkan kembali pengiriman email.

   ```
   aws ses update-account-sending-enabled --enabled --region sending_region
   ```

2. Pada baris perintah, ketik perintah berikut ini untuk memeriksa status pengiriman email untuk akun Anda:

   ```
   aws ses get-account-sending-enabled --region sending_region
   ```

   Jika Anda melihat output berikut ini, maka Anda telah berhasil mengaktifkan kembali pengiriman email untuk akun Anda:

   ```
   {
   ```

```
    "Enabled": true
}
```

Bagian 4: Buat Topik dan Langganan Amazon SNS

CloudWatch Untuk menjalankan fungsi Lambda Anda saat alarm dipicu, Anda harus terlebih dahulu membuat topik Amazon SNS dan berlangganan fungsi Lambda ke sana.

Untuk membuat topik Amazon SNS dan berlangganan fungsi Lambda ke sana

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Buat topik dengan mengikuti langkah-langkah di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.
   - Tipe harus Standar (bukan FIFO).
3. Berlangganan topik dengan mengikuti langkah-langkah di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.
   a. Untuk Protocol (Protokol) pilih AWS Lambda.
   b. Untuk Titik akhir, pilih fungsi Lambda yang Anda buat di the section called “Bagian 2: Buat fungsi Lambda”.

Bagian 5: Buat CloudWatch Alarm

Bagian ini berisi prosedur untuk membuat alarm CloudWatch yang dipicu ketika metrik mencapai ambang tertentu. Ketika alarm dipicu, alarm tersebut akan menyampaikan notifikasi ke topik Amazon SNS yang Anda buat di the section called “Bagian 4: Buat Topik dan Langganan Amazon SNS”, yang kemudian mengeksekusi fungsi Lambda yang Anda buat di the section called “Bagian 2: Buat fungsi Lambda”.

Untuk membuat CloudWatch alarm

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Gunakan pemilih wilayah untuk memilih wilayah tempat Anda ingin menjeda pengiriman e-mail secara otomatis.
3. Di panel navigasi, pilih Alarm.
4. Pilih Buat Alarm.

5. Pada jendela Buat Alarm, di bawah Metrik SES pilih Metrik akun.
6. Di bawah Nama Metrik, pilih salah satu opsi berikut:

   - Reputasi. BounceRate— Pilih metrik ini jika Anda ingin menjeda pengiriman email untuk akun Anda ketika rasio pentalan keras keseluruhan untuk akun Anda melewati ambang batas yang Anda tentukan.
   - Reputasi. ComplaintRate— Pilih metrik ini jika Anda ingin menjeda pengiriman email untuk akun Anda ketika tingkat keluhan keseluruhan untuk akun Anda melewati ambang batas yang Anda tentukan.

   Pilih Selanjutnya.
7. Selesaikan langkah-langkah berikut:

   - Di bawah Ambang Batas Alarm, untuk Nama, ketikkan nama untuk alarm.
   - Di Bawah Kapanpun: Reputasi. BounceRateatau Kapan pun: Reputasi. ComplaintRate, tentukan ambang batas yang menyebabkan alarm terpicu.

     > **Note**
     >
     > Akun Anda secara otomatis ditempatkan dalam peninjauan jika rasio pentalan Anda melebihi 5%, atau jika tingkat keluhan Anda melebihi 0,1%. Saat Anda menentukan rasio pentalan atau keluhan yang menyebabkan CloudWatch alarm dipicu, kami sarankan Anda menggunakan nilai yang berada di bawah tarif ini untuk mencegah akun Anda ditempatkan dalam peninjauan.

   - Di bawah Tindakan, untuk Setiap kali alarm ini, pilih Status adalah ALARM. Untuk Kirim notifikasi ke, pilih topik Amazon SNS yang Anda buat di the section called “Bagian 4: Buat Topik dan Langganan Amazon SNS”.

   Pilih Buat Alarm.

Bagian 6: Uji solusinya

Anda sekarang dapat menguji alarm guna memastikan bahwa alarm tersebut menjalankan fungsi Lambda ketika memasuki status `ALARM`. Anda dapat menggunakan operasi API `SetAlarmState` untuk mengubah status alarm sementara waktu.

Prosedur di bagian ini bersifat opsional, namun kami merekomendasikan kepada Anda untuk menyelesaikannya guna memastikan bahwa seluruh solusi sudah dikonfigurasi dengan benar.

1. Pada baris perintah, ketik perintah berikut ini guna memeriksa status pengiriman email untuk akun Anda. Ganti *wilayah* dengan nama Wilayah.

   ```
   aws ses get-account-sending-enabled --region region
   ```

   Jika pengiriman diaktifkan untuk akun Anda, Anda dapat melihat output berikut:

   ```
   {
       "Enabled": true
   }
   ```

2. Pada baris perintah, ketik perintah berikut ini untuk mengubah status alarm sementara waktu menjadi ALARM: aws cloudwatch set-alarm-state --alarm-name ***MyAlarm*** --state-value ALARM --state-reason "Testing execution of Lambda function" --region ***region***

   Ganti *MyAlarm*di perintah sebelumnya dengan nama alarm yang Anda buatthe section called “Bagian 5: Buat CloudWatch Alarm”, dan ganti *wilayah dengan Wilayah* tempat Anda ingin menjeda pengiriman email secara otomatis.

   > **Note**
   >
   > Ketika Anda menjalankan perintah ini, status alarm beralih dari OK ke ALARM dan kembali lagi ke OK dalam beberapa detik. Anda dapat melihat perubahan status ini pada tab Riwayat alarm di CloudWatch konsol, atau dengan menggunakan DescribeAlarmHistoryoperasi.

3. Pada baris perintah, ketik perintah berikut ini guna memeriksa status pengiriman email untuk akun Anda.

   ```
   aws ses get-account-sending-enabled --region region
   ```

   Jika fungsi Lambda berhasil dilaksanakan, Anda akan melihat output berikut:

   ```
   {
       "Enabled": false
   ```

```
}
```

4. Selesaikan langkah-langkah pada the section called “Bagian 3: Aktifkan Ulang Pengiriman Email untuk Akun Anda” guna mengaktifkan kembali pengiriman email untuk akun Anda.

Menjeda pengiriman email secara otomatis untuk satu set konfigurasi

Anda dapat mengonfigurasi Amazon SES untuk mengekspor reputasi metrik yang spesifik untuk email yang dikirim menggunakan konfigurasi tertentu yang ditetapkan ke Amazon CloudWatch. Anda kemudian dapat menggunakan metrik ini untuk membuat CloudWatch alarm yang khusus untuk rangkaian konfigurasi ini. Ketika alarm ini melebihi ambang batas tertentu, Anda dapat secara otomatis menjeda pengiriman email yang menggunakan set konfigurasi tertentu, tanpa mempengaruhi kemampuan pengiriman email secara keseluruhan dari akun Amazon SES Anda.

> **Note**
>
> Solusi yang dijelaskan di bagian ini menjeda pengiriman email untuk konfigurasi tertentu yang ditetapkan dalam satu Wilayah AWS. Jika Anda mengirim email dari beberapa wilayah, ulangi prosedur di bagian ini untuk setiap wilayah tempat Anda ingin menerapkan solusi ini.

Topik di bagian ini:

- Bagian 1: Aktifkan Reporting Reputasi untuk Set Konfigurasi
- Bagian 2: Buat IAM Role
- Bagian 3: Buat fungsi Lambda
- Bagian 4: Aktifkan Ulang Pengiriman Email untuk Set Konfigurasi
- Bagian 5: Membuat Topik Amazon SNS
- Bagian 6: Buat CloudWatch Alarm
- Bagian 7: Pengujian solusi

Bagian 1: Aktifkan Reporting Reputasi untuk Set Konfigurasi

Sebelum Anda dapat mengonfigurasi Amazon SES untuk secara otomatis menjeda pengiriman email untuk satu set konfigurasi, Anda harus terlebih dahulu mengaktifkan ekspor reputasi metrik untuk set konfigurasi.

Untuk mengaktifkan ekspor metrik dari pentalan dan aduan untuk set konfigurasi, selesaikan langkah-langkah di the section called “Lihat dan ekspor metrik reputasi”.

Bagian 2: Buat IAM Role

Langkah pertama dalam mengonfigurasi penjedaan pengiriman email secara otomatis adalah untuk membuat IAM role yang dapat mengeksekusi operasi API `UpdateConfigurationSetSendingEnabled`.

Buat IAM role

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Di bawah Pilih tipe entitas tepercaya, pilih Layanan AWS.
5. Di bawah Pilih layanan yang akan menggunakan peran ini, pilih Lambda. Pilih Berikutnya: Izin.
6. Pada halaman Lampirkan kebijakan izin, pilih kebijakan berikut:

   - AWS LambdaBasicExecutionRole
   - AmazonSESFullAccess

   > **Tip**
   >
   > Gunakan kotak pencarian di bagian atas daftar kebijakan untuk menemukan kebijakan ini dengan cepat.

   Pilih Berikutnya: Peninjauan.
7. Di halaman Tinjau, untuk Nama, ketikkan nama untuk peran tersebut. Pilih Buat peran.

Bagian 3: Buat fungsi Lambda

Setelah Anda membuat IAM role, Anda dapat membuat fungsi Lambda yang menjeda pengiriman email untuk set konfigurasi.

Untuk membuat fungsi Lambda

1. Buka konsol AWS Lambda tersebut di https://console.aws.amazon.com/lambda/.
2. Gunakan pemilih wilayah untuk memilih wilayah tempat Anda ingin men-deploy fungsi Lambda ini.

   > **Note**
   >
   > Fungsi ini hanya berhenti mengirim email untuk set konfigurasi di Wilayah AWS yang Anda pilih pada langkah ini. Jika Anda mengirim email dari beberapa wilayah, ulangi prosedur di bagian ini untuk setiap wilayah tempat Anda ingin menjeda pengiriman email secara otomatis.

3. Pilih Buat fungsi.
4. Di bawah Buat fungsi, pilih Penulis dari scratch.
5. Di bawah Penulis dari scratch, selesaikan langkah berikut:

   - Untuk Nama, ketikkan nama untuk fungsi Lambda.
   - UntukWaktu pengoperasian, pilihNode.js 14x(atau versi yang saat ini ditawarkan dalam daftar pilih).
   - Untuk Peran, pilih Pilih peran yang sudah ada.
   - Untuk Peran yang sudah ada, pilih IAM role yang Anda buat di the section called “Bagian 2: Buat IAM Role”.

   Pilih Buat fungsi.
6. Di bawah Kode fungsi, di kode editor, tempelkan kode fungsi berikut:

```
'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);
        }
    });
};
```

Ganti *ConfigSet* pada kode sebelumnya dengan nama set konfigurasi. Pilih Simpan.

7. Pilih Uji. Jika jendela Konfigurasikan peristiwa uji muncul, ketik nama di kolom Nama peristiwa, lalu pilih Buat.
8. Pastikan bahwa bilah notifikasi di bagian atas halaman menyampaikan `Execution result: succeeded`. Jika fungsi gagal dijalankan, lakukan hal berikut:

   - Verifikasikan bahwa IAM role yang Anda buat di the section called “Bagian 2: Buat IAM Role” berisi kebijakan yang benar.
   - Verifikasi bahwa kode dalam fungsi Lambda tidak berisi kesalahan. Editor kode Lambda secara otomatis menyoroti kesalahan sintaksis dan potensi masalah lainnya.

Bagian 4: Aktifkan Ulang Pengiriman Email untuk Set Konfigurasi

Efek samping dari pengujian fungsi Lambda di the section called “Bagian 3: Buat fungsi Lambda” adalah pengiriman email untuk set konfigurasi dijeda. Dalam kebanyakan kasus, Anda tidak ingin menjeda pengiriman untuk rangkaian konfigurasi sampai CloudWatch alarm dipicu.

Prosedur di bagian ini mengaktifkan kembali pengiriman email untuk set konfigurasi Anda. Untuk menyelesaikan prosedur ini, Anda harus menginstal dan mengonfigurasi AWS Command Line Interface. Untuk informasi lebih lanjut, lihat Panduan Pengguna AWS Command Line Interface.

Untuk mengaktifkan kembali pengiriman email

1. Pada baris perintah, ketik perintah berikut guna mengaktifkan kembali pengiriman email untuk set konfigurasi:

   ```
   aws ses update-configuration-set-sending-enabled \
   --configuration-set-name ConfigSet \
   --enabled
   ```

   Di perintah sebelumnya, ganti*ConfigSet*dengan nama rangkaian konfigurasi yang ingin Anda jeda pengiriman email.

2. Pada baris perintah, ketik perintah berikut ini guna memastikan bahwa pengiriman email diaktifkan:

   ```
   aws ses describe-configuration-set \
   --configuration-set-name ConfigSet \
   --configuration-set-attribute-names reputationOptions
   ```

   Perintah menghasilkan output yang mirip dengan contoh berikut ini:

   ```
   {
       "ConfigurationSet": {
           "Name": "ConfigSet"
       },
       "ReputationOptions": {
           "ReputationMetricsEnabled": true,
           "SendingEnabled": true
       }
   }
   ```

   Jika nilai dari `SendingEnabled` adalah `true`, maka pengiriman email untuk rangkaian konfigurasi berhasil diaktifkan kembali.

Bagian 5: Membuat Topik Amazon SNS

Untuk CloudWatch untuk menjalankan fungsi Lambda ketika alarm dipicu, Anda harus terlebih dahulu membuat topik Amazon SNS dan berlangganan fungsi Lambda untuk Cambda.

Untuk membuat topik Amazon SNS

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Gunakan pemilih wilayah untuk memilih wilayah tempat Anda ingin menjeda pengiriman e-mail secara otomatis.
3. Di panel navigasi, pilih Topik.
4. Pilih Buat topik baru.
5. Pada jendela Buat topik baru, untuk Nama topik, ketikkan nama untuk topik tersebut. Secara opsional, Anda dapat mengetik nama yang lebih deskriptif di kolom Nama tampilan.

   Pilih Buat topik.
6. Dalam daftar topik, periksa kotak yang berada di samping topik yang Anda buat pada langkah sebelumnya. Pada menu Tindakan, pilih Berlangganan topik.
7. Pada jendela Buat langganan, buat pilihan berikut:

   - Untuk Protokol, pilih AWS Lambda.
   - Untuk Titik akhir, pilih fungsi Lambda yang Anda buat di the section called “Bagian 3: Buat fungsi Lambda”.
   - Untuk Versi atau alias, pilih default.
8. Pilih Buat langganan.

Bagian 6: Buat CloudWatch Alarm

Bagian ini berisi prosedur untuk membuat alarm di CloudWatch yang dipicu ketika metrik mencapai ambang batas tertentu. Ketika alarm dipicu, alarm tersebut akan menyampaikan notifikasi ke topik Amazon SNS yang Anda buat di the section called “Bagian 5: Membuat Topik Amazon SNS”, yang kemudian mengeksekusi fungsi Lambda yang Anda buat di the section called “Bagian 3: Buat fungsi Lambda”.

Untuk membuat CloudWatch alarm

1. Buka CloudWatch konsol dihttps://console.aws.amazon.com/cloudwatch/.
2. Gunakan pemilih wilayah untuk memilih wilayah tempat Anda ingin menjeda pengiriman e-mail secara otomatis.
3. Di panel navigasi di sebelah kiri, pilih Pengguna.
4. Pilih Buat Alarm.

5. Pada jendela Buat Alarm, di bawah Metrik SES, pilih Metrik Set Konfigurasi.
6. Di kolom ses:configuration-set, temukan rangkaian konfigurasi yang ingin Anda buat untuk sebuah alarm. Di bawah Nama Metrik, pilih salah satu opsi berikut:

   - Reputation.BounceRate — silahkan Pilih metrik ini jika Anda ingin menjeda pengiriman email untuk konfigurasi yang ditetapkan bila keseluruhan rasio pentalan keras untuk rangkaian konfigurasi melewati ambang batas yang Anda tentukan.
   - Reputation.ComplaintRate — Pilih metrik ini jika Anda ingin menjeda pengiriman email untuk konfigurasi yang ditetapkan bila keseluruhan tingkat aduan untuk rangkaian konfigurasi melewati ambang batas yang Anda tentukan.

   Pilih Selanjutnya.
7. Selesaikan langkah-langkah berikut:

   - Di bawah Ambang Batas Alarm, untuk Nama, ketikkan nama untuk alarm.
   - Di bawahKapan pun: reputasi.BouncerateatauKapan pun: reputasi.complaintrate, tentukan ambang yang menyebabkan alarm dipicu.

     > **Note**
     >
     > Jika tingkat pentalan keseluruhan untuk akun Amazon SES Anda melebihi 10%, atau jika tingkat aduan keseluruhan untuk akun Amazon SES Anda melebihi 0,5%, maka akun Amazon SES Anda secara otomatis akan ditempatkan di bawah peninjauan. Bila Anda menentukan tingkat bouncing atau keluhan yang menyebabkan CloudWatch alarm untuk dipicu, sebaiknya gunakan nilai yang jauh di bawah tarif ini guna mencegah akun Anda ditempatkan dalam peninjauan.

   - Di bawah Tindakan, untuk Setiap kali alarm ini, pilih Status adalah ALARM. Untuk Kirim notifikasi ke, pilih topik Amazon SNS yang Anda buat di the section called “Bagian 5: Membuat Topik Amazon SNS”.

   Pilih Buat Alarm.

Bagian 7: Pengujian solusi

Anda sekarang dapat menguji alarm guna memastikan bahwa alarm tersebut menjalankan fungsi Lambda ketika memasuki status `ALARM`. Anda dapat menggunakan`SetAlarmState`operasi di CloudWatch API untuk mengubah status alarm sementara waktu.

Prosedur di bagian ini adalah opsional, tetapi kami menyarankan agar Anda menyelesaikannya untuk memverifikasi bahwa seluruh solusi sudah dikonfigurasi dengan benar.

Untuk menguji solusi tersebut

1. Pada baris perintah, ketik perintah berikut ini guna memeriksa status pengiriman email untuk set konfigurasi:

   ```
   aws ses describe-configuration-set --configuration-set-name ConfigSet
   ```

   Jika pengiriman diaktifkan untuk set konfigurasi, Anda dapat melihat output berikut:

   ```
   {
       "ConfigurationSet": {
           "Name": "ConfigSet"
       },
       "ReputationOptions": {
           "ReputationMetricsEnabled": true,
           "SendingEnabled": true
       }
   }
   ```

   Jika nilai dari `SendingEnabled` adalah `true`, maka pengiriman email saat ini diaktifkan untuk set konfigurasi.
2. Pada baris perintah, ketik perintah berikut ini untuk mengubah status alarm sementara waktu menjadi `ALARM`:

   ```
   aws cloudwatch set-alarm-state \
   --alarm-name MyAlarm \
   --state-value ALARM \
   --state-reason "Testing execution of Lambda function"
   ```

   Ganti *MyAlarm* pada perintah sebelumnya dengan nama alarm yang Anda buat di the section called “Bagian 6: Buat CloudWatch Alarm”.

> **Note**
>
> Ketika Anda menjalankan perintah ini, status alarm beralih dari `OK` ke `ALARM` dan kembali lagi ke `OK` dalam beberapa detik. Anda dapat melihat perubahan status ini pada alarmRiwayattab di CloudWatch konsol, atau dengan menggunakanDescribeAlarmHistoryoperasi.

3. Pada baris perintah, ketik perintah berikut ini guna memeriksa status pengiriman email untuk set konfigurasi:

```
aws ses describe-configuration-set \
--configuration-set-name ConfigSet
```

Jika fungsi Lambda berhasil dijalankan, Anda dapat melihat output yang menyerupai contoh berikut:

```
{
    "ConfigurationSet": {
        "Name": "ConfigSet"
    },
    "ReputationOptions": {
        "ReputationMetricsEnabled": true,
        "SendingEnabled": false
    }
}
```

Jika nilai dari `SendingEnabled` adalah `false`, maka pengiriman email untuk set konfigurasi dinonaktifkan, yang menunjukkan bahwa fungsi Lambda berhasil dijalankan.

4. Selesaikan langkah-langkah pada the section called “Bagian 4: Aktifkan Ulang Pengiriman Email untuk Set Konfigurasi” guna mengaktifkan kembali pengiriman email untuk set konfigurasi.

Memantau SES peristiwa menggunakan Amazon EventBridge

EventBridge adalah layanan tanpa server yang menggunakan peristiwa untuk menghubungkan komponen aplikasi bersama-sama, sehingga memudahkan Anda untuk membangun aplikasi berbasis peristiwa yang dapat diskalakan. Arsitektur berbasis peristiwa adalah gaya membangun sistem perangkat lunak yang digabungkan secara longgar yang bekerja sama dengan memancarkan dan menanggapi peristiwa. Peristiwa adalah pesan JSON berformat yang biasanya mewakili perubahan dalam sumber daya atau lingkungan, atau acara manajemen lainnya.

SESFitur tertentu akan menghasilkan dan mengirim acara yang Anda tentukan saat membuat tujuan acara ke bus acara EventBridge default. Bus acara adalah router yang menerima acara dan mengirimkannya ke nol atau lebih tujuan, atau target. Aturan yang Anda kaitkan dengan bus acara mengevaluasi acara saat mereka tiba. Setiap aturan memeriksa apakah suatu peristiwa cocok dengan pola aturan. Jika acara tidak cocok, EventBridge kirimkan acara ke target yang ditentukan.

SESmengirimkan peristiwa ke EventBridge saat fitur memiliki perubahan status atau pembaruan status. Anda dapat menggunakan EventBridge aturan untuk merutekan peristiwa ke target yang ditentukan. Acara-acara ini akan disampaikan dengan upaya terbaik, dan mereka mungkin dikirim rusak.

Topik

- SESacara
- SESreferensi skema acara
- Menggunakan EventBridge dengan SES acara
- EventBridge Sumber daya tambahan

SESacara

Peristiwa berikut dihasilkan oleh SES fitur dan dikirim ke bus acara default di EventBridge. Untuk informasi selengkapnya, termasuk data detail untuk setiap jenis acara, lihat???.

Acara penasihat Manajer Pengiriman Virtual

Jenis peristiwa	Deskripsi
Status Rekomendasi Penasihat Terbuka	Acara yang dihasilkan setiap kali rekomendasi baru dibuka di penasihat Virtual Deliverability Manager.
Status Rekomendasi Penasihat Diselesaikan	Peristiwa yang dihasilkan setiap kali rekomendasi diselesaikan di penasihat Virtual Deliverability Manager.

SESacara pengiriman email

Jenis peristiwa	Deskripsi
Email Terpental	Sebuah hard bounce bahwa server email penerima secara permanen menolak email. (Pantulan lunak hanya disertaka n ketika SES gagal mengirimkan email setelah mencoba lagi untuk jangka waktu tertentu.)
Email diklik	Penerima mengklik satu atau beberapa tautan di email.
Email Keluhan Diterima	Email berhasil dikirim ke server email penerima, tetapi penerima menandainya sebagai spam.
Email Terkirim	SESberhasil mengirimkan email ke server email penerima.
Pengiriman Email Tertunda	Email tidak dapat dikirim ke server email penerima karena masalah sementara terjadi. Penundaan penyamapaian dapat terjadi, misalnya, saat kotak masuk penerima penuh, atau saat server email penerima mengalami masalah sementara.
Email Dibuka	Penerima menerima pesan dan membukanya di klien email mereka.
Email Ditolak	SESmenerima email, tetapi memutuskan bahwa itu berisi virus dan tidak berusaha mengirimkannya ke server email penerima.
Perenderan Email Gagal	Email tidak dikirim karena masalah rendering template. Tipe peristiwa ini dapat terjadi saat data templat tidak ada, atau jika ada ketidakcocokan antara parameter templat dan data.

Jenis peristiwa	Deskripsi
	(Jenis peristiwa ini hanya terjadi ketika Anda mengirim email menggunakan `SendTemplatedEmail` atau `SendBulkTemplatedEmail` APIoperasi.)
Email Terkirim	Permintaan kirim berhasil dan SES akan mencoba mengirimkan pesan ke server email penerima. (Jika tingkat akun atau penekanan global sedang digunakan, masih SES akan menghitungnya sebagai kirim, tetapi pengiriman ditekan.)
Email Berlangganan	Email berhasil dikirim, tetapi penerima memperbarui preferensi langganan dengan mengklik `List-Unsubscribe` header email atau `Unsubscribe` tautan di footer.

SESreferensi skema acara

Semua peristiwa dari AWS layanan memiliki seperangkat bidang umum yang berisi metadata tentang acara tersebut, seperti AWS layanan yang merupakan sumber acara, waktu acara dibuat, akun dan wilayah tempat acara berlangsung, dan lainnya. Untuk definisi bidang umum ini, lihat Referensi struktur acara di Panduan EventBridge Pengguna.

Selain itu, setiap acara memiliki `detail` bidang yang berisi data khusus untuk peristiwa tertentu. Referensi di bawah ini mendefinisikan bidang detail untuk berbagai SES acara.

Saat menggunakan EventBridge untuk memilih dan mengelola SES acara, penting untuk mengingat hal berikut:

- `source`Bidang untuk semua acara dari SES diatur ke`aws.ses`.
- `detail-type`Bidang menentukan jenis acara. Lihat tabel jenis acara dithe section called “SESacara”.
- `detail`Bidang berisi data yang spesifik untuk peristiwa tertentu.

  Untuk beberapa jenis acara, seperti untuk Virtual Deliverability Manager, bidang detail adalah string data yang agak sederhana yang diisi dari serangkaian nilai statis yang terbatas. Sebaliknya, bidang detail untuk peristiwa pengiriman email lebih kompleks karena dapat terdiri dari banyak sub-bidang detail yang merupakan kombinasi dari nilai statis dan dinamis seperti stempel waktu ketika email dikirim, alamat penerima, dan banyak atribut email lainnya.

Topik

- Skema status penasihat Manajer Pengiriman Virtual
- SESskema status pengiriman email

Skema status penasihat Manajer Pengiriman Virtual

Referensi skema berikut mendefinisikan bidang khusus untuk peristiwa status penasihat Virtual Deliverability Manager.

Definisi untuk bidang umum yang muncul di semua skema acara (sepertiversion,, idaccount, dan lainnya) dapat ditemukan dalam referensi struktur Acara di Panduan EventBridge Pengguna. detail-typeBidang source dan disertakan dalam referensi di bawah ini karena berisi nilai SES - spesifik untuk SES acara.

`source`

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk SES acara, nilai ini adalahaws.ses.

`detail-type`

Mengidentifikasi jenis acara.

Nilai untuk bidang ini tercantum dalam tabel acara penasihat Virtual Deliverability Manager di. the section called “SESacara”

`detail`

JSONObjek yang berisi informasi tentang acara tersebut. Layanan yang menghasilkan acara menentukan konten bidang ini.

Nilai untuk bidang ini dapat berupa:

- `DKIM verification is not enabled.`
- `DKIM verification has failed.`
- `DKIM signing key length is below 2048 bits.`
- `DMARC configuration was not found.`
- `DMARC configuration could not be parsed.`
- `DKIM record was not found.`

- `DKIM record is not aligned.`
- `MAIL FROM record is not aligned.`
- `SPF record was not found.`
- `SPF record for Amazon SES was not found.`
- `SPF all qualifier is missing.`
- `An SPF configuration issue was found.`
- `BIMI record not found or configured without default selector.`
- `BIMI has malformed TXT record.`

Example Contoh: Acara status penasihat Manajer Pengiriman Virtual

Berikut ini adalah contoh acara status penasihat Virtual Deliverability Manager untuk jenis acara. `Advisor Recommendation Status Open` Nilai detail peristiwa dalam contoh ini adalah`SPF record was not found.`.

```
{
 "version": "0",
  "id": "abcd9999-ef33-0123-90ab-abcdef666666",
  "detail-type": "Advisor Recommendation Status Open",
  "source": "aws.ses",
  "account": "012345678901",
  "time": "2023-11-15T17:00:59Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ses:us-east-1:012345678901:identity/vdm.events-publishing.cajun.syster-
games.example.com"
  ],
  "detail": { "version": "1.0.0", "data": "SPF record was not found." }
}
```

SESskema status pengiriman email

Referensi skema berikut mendefinisikan bidang khusus untuk peristiwa status pengiriman SES email.

Definisi untuk bidang umum yang muncul di semua skema acara (sepertiversion,, idaccount, dan lainnya) dapat ditemukan dalam referensi struktur Acara di Panduan EventBridge Pengguna. `detail-type`Bidang `source` dan disertakan dalam referensi di bawah ini karena berisi nilai SES - spesifik untuk SES acara.

`source`

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk SES acara, nilai ini adalah`aws.ses`.

`detail-type`

Mengidentifikasi jenis acara.

Nilai untuk bidang ini tercantum dalam tabel peristiwa pengiriman SES email dithe section called “SESacara”.

`detail`

JSONObjek yang berisi informasi tentang acara tersebut. Layanan yang menghasilkan acara menentukan konten bidang ini.

Semua nilai yang mungkin untuk bidang ini tidak dapat dicantumkan di sini karena terdiri dari nilai statis dan dinamis yang dihasilkan oleh setiap email unik yang dikirim pada saat tertentu. Namun, sebuah contoh disediakan untuk memberi Anda gambaran tentang jenis data yang dapat berisi bidang ini. Contoh data detail untuk semua jenis peristiwa pengiriman email dapat ditemukan menggunakan EventBridge Sandbox, lihatTentukan contoh peristiwa di EventBridge.

Contoh data detail yang dihasilkan untuk acara pengiriman SES email`Email Rendering Failed`:


```
...,
  "detail": {
    "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"
    }
  }
```

Example Contoh: Acara status pengiriman email

Berikut ini adalah contoh acara status pengiriman email lengkap untuk jenis acara`Email Rendering Failed`. Nilai peristiwa detail dalam contoh ini adalah kombinasi dari nilai statis dan dinamis berdasarkan peristiwa pengiriman email untuk email tertentu.

```
{
  "version": "0",
  "id": "12a18625-3328-fafd-2809-a5e16004f112",
  "detail-type": "Email Rendering Failed",
  "source": "aws.ses",
  "account": "123456789012",
  "time": "2023-07-17T16:48:05Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ses:us-east-1:123456789012:identity/example.com"],
  "detail": {
    "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"
    }
  }
}
```

Menggunakan EventBridge dengan SES acara

Secara default, SES mengirimkan acara ke bus acara EventBridge default. Anda dapat membuat aturan pada bus acara default untuk mengidentifikasi peristiwa tertentu untuk dikirim EventBridge ke satu atau beberapa target yang ditentukan. Setiap aturan berisi pola acara yang EventBridge digunakan untuk mencocokkan acara saat mereka tiba di bus acara. Jika suatu peristiwa cocok dengan pola acara untuk aturan tertentu, EventBridge kirimkan acara ke target yang ditentukan dalam aturan.

Dalam EventBridge, mendefinisikan pola acara biasanya merupakan bagian dari proses yang lebih besar untuk membuat aturan baru atau mengedit yang sudah ada. Untuk mempelajari cara membuat EventBridge aturan, lihat Membuat EventBridge aturan Amazon yang bereaksi terhadap peristiwa di Panduan EventBridge Pengguna.

Dengan menggunakan fitur Sandbox di EventBridge, Anda dapat dengan cepat menentukan pola peristiwa dan menggunakan contoh peristiwa untuk mengonfirmasi pola cocok dengan peristiwa yang diinginkan, tanpa harus terlebih dahulu membuat atau mengedit aturan. Untuk petunjuk mendetail tentang penggunaan Kotak Pasir, lihat Menguji pola peristiwa menggunakan EventBridge Kotak Pasir di EventBridge Panduan Pengguna.

Tentukan SES contoh peristiwa di Kotak EventBridge Pasir

Anda dapat memilih contoh peristiwa untuk SES acara untuk menggunakannya dalam menguji pola acara yang Anda buat.

Untuk menentukan SES contoh peristiwa di Kotak EventBridge Pasir

1. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Di panel navigasi, pilih Sumber daya pengembang, lalu pilih Sandbox, dan pada halaman Sandbox pilih tab Pola acara.
3. Untuk sumber Acara, pilih AWS acara atau acara EventBridge mitra.
4. Di bagian Contoh peristiwa, untuk Contoh jenis acara, pilih AWS acara.
5. Untuk contoh peristiwa, gulir ke bawah ke SESdan kemudian pilih SES acara yang diinginkan.

   EventBridge menampilkan contoh peristiwa, bersama dengan semua data detailnya, untuk jenis acara.

Anda kemudian dapat menggunakan acara ini untuk menguji pola acara yang Anda buat di bagian Pola acara, atau menggunakannya sebagai dasar untuk membuat kejadian sampel Anda sendiri untuk pengujian pola yang tercakup dalam bagian berikut.

Membuat dan menguji pola acara untuk SES acara

Setelah Anda memilih contoh peristiwa, seperti yang dijelaskan di bagian sebelumnya, Anda dapat membuat pola acara dan menggunakan contoh peristiwa untuk memastikannya cocok dengan peristiwa yang diinginkan.

Untuk membuat dan menguji pola acara yang cocok dengan SES peristiwa di Kotak EventBridge Pasir

1. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Di panel navigasi, pilih Sumber daya pengembang, lalu pilih Sandbox, dan pada halaman Sandbox pilih tab Pola acara.
3. Untuk sumber acara, pilih AWS acara atau acara EventBridge mitra, dan pilih contoh acara yang ingin Anda uji seperti yang dijelaskan di bagian sebelumnya.
4. Gulir ke bawah ke metode Creation, dan pilih Use pattern form.
5. Di bagian Pola acara, untuk Sumber acara pilih AWS layanan.
6. Di bawah AWS layanan, pilih SES.
7. Untuk jenis Acara, pilih jenis SES acara yang ingin Anda cocokkan.

   EventBridge menampilkan pola acara minimum, terdiri dari `source` dan `detail-type` bidang, yang cocok dengan acara yang dipilihSES.

   Dalam dua contoh, pola acara pertama cocok dengan semua `Advisor Recommendation Status Resolved` peristiwa, dan yang kedua, semua `Email Bounced` peristiwa:

```
{
  "source": ["aws.ses"],
  "detail-type": ["Advisor Recommendation Status Resolved"]
}
```

```
{
  "source": ["aws.ses"],
```

```
  "detail-type": ["Email Bounced"]
}
```

8. Untuk membuat perubahan pada pola acara, pilih Edit pola dan buat perubahan Anda di JSON editor.

   Anda juga dapat mencocokkan nilai dalam satu atau beberapa bidang data detail. Ini termasuk menentukan beberapa nilai yang mungkin untuk nilai bidang.

   Dalam contoh berikut, bidang detail ditambahkan ke pola peristiwa minimum yang dihasilkan dengan nilai `data` bidang yang ditentukan untuk menemukan semua acara penasihat Virtual Deliverability Manager dengan nilai detail yang sama: `DKIM record was not found`

```
{
  "source": ["aws.ses"],
  "detail-type": ["Advisor Recommendation Status Resolved"],
  "detail": {
    "data": ["DKIM record was not found."]
  }
}
```

   Dalam contoh ini, sub-bidang detail ditambahkan untuk melaporkan peristiwa yang dihasilkan oleh semua email yang dikirim dari noreply@example.com pada 2024-08-05 yang memantul. (Pencocokan awalan sedang digunakan di sini sebagai bagian dari pemfilteran Konten.):

```
{
  "source": ["aws.ses"],
  "detail-type": ["Email Bounced"],
  "detail": {
    "mail": {
      "timestamp": [{
        "prefix": "2024-08-05"
      }],
      "source": ["noreply@example.com"]
    }
  }
}
```

   Penting bagi Anda untuk membaca pola Peristiwa di Panduan EventBridge Pengguna —ini menjelaskan bahwa nilai pola peristiwa yang Anda masukkan di JSON editor harus dikelilingi

oleh tanda kurung siku `[...]` karena dianggap sebagai array. Ini dan informasi lebih lanjut tentang cara membangun pola acara lanjutan juga disediakan.

9. Untuk menguji apakah pola acara Anda cocok dengan peristiwa sampel yang Anda tentukan di panel Peristiwa sampel di atas, pilih Pola uji. Jika cocok, spanduk hijau di bagian bawah JSON editor akan ditampilkan, “Contoh acara cocok dengan pola acara”.
10. Untuk memecahkan masalah kesalahan setelah memilih Pola uji:

    - Jika ada kesalahan JSON terkait, pesan akan menunjukkan alasannya, seperti, “Pola acara tidak valid. Alasan: “data” harus berupa objek atau array pada baris: 5, kolom: 14". Untuk mengatasinya, lampirkan nilai pada baris 5 dengan tanda kurung siku`[...]`.
    - Jika ada perbedaan antara nilai dalam peristiwa Sampel dan pola Peristiwa Anda, pesannya adalah, “Contoh peristiwa tidak cocok dengan pola acara”. Ini berarti bahwa satu atau lebih nilai yang ingin Anda uji berbeda dari nilai contoh yang dihasilkan oleh generator peristiwa Sampel. Untuk memperbaiki ini, lanjutkan dengan langkah-langkah yang tersisa.
11. Untuk mengubah nilai sampel dalam peristiwa Sampel agar berhasil menguji pola Peristiwa Anda, di panel acara Contoh, pilih Salin di bawah JSON editor.
12. Pilih tombol radio di sebelah Enter my own for Contoh jenis acara di atas editor.
13. Tempelkan peristiwa sampel ke JSON editor, dan untuk bidang apa pun yang Anda gunakan dalam pola acara Anda, ganti nilai bidang yang sama agar sesuai dengan nilai yang Anda tentukan dalam pola acara Anda.
14. Gulir kembali ke bawah ke panel pola Peristiwa dan pilih Pola uji lagi. Jika semua nilai dimasukkan dengan benar dan cocok, spanduk hijau di bagian bawah JSON editor akan ditampilkan, “Contoh acara cocok dengan pola acara”.

EventBridge Sumber daya tambahan

Lihat topik berikut di Panduan EventBridge Pengguna Amazon untuk informasi selengkapnya tentang cara menggunakan EventBridge untuk memproses dan mengelola acara.

- Untuk informasi terperinci tentang cara kerja bus acara, lihat Bus EventBridge acara Amazon.
- Untuk informasi tentang struktur acara, lihat Acara
- Untuk informasi tentang membuat pola acara untuk EventBridge digunakan saat mencocokkan peristiwa dengan aturan, lihat Pola acara
- Untuk informasi tentang membuat aturan untuk menentukan EventBridge proses peristiwa, lihat Aturan

- Untuk informasi tentang menentukan layanan atau tujuan lain yang EventBridge mengirimkan peristiwa yang cocok, lihat Target

Contoh kode untuk Amazon SES menggunakanAWS SDK

Contoh kode berikut ini menunjukkan cara menggunakan Amazon SES denganAWS perangkat pengembangan perangkat lunak (SDK).

Untuk daftar lengkap panduan pengembangAWS SDK dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- Contoh kode untuk Amazon SES menggunakan AWS SDKs
  - Contoh dasar untuk Amazon SES menggunakan AWS SDKs
    - Tindakan untuk Amazon SES menggunakan AWS SDKs
      - Gunakan CreateReceiptFilter dengan AWS SDK atau CLI
      - Gunakan CreateReceiptRule dengan AWS SDK atau CLI
      - Gunakan CreateReceiptRuleSet dengan AWS SDK atau CLI
      - Gunakan CreateTemplate dengan AWS SDK atau CLI
      - Gunakan DeleteIdentity dengan AWS SDK atau CLI
      - Gunakan DeleteReceiptFilter dengan AWS SDK atau CLI
      - Gunakan DeleteReceiptRule dengan AWS SDK atau CLI
      - Gunakan DeleteReceiptRuleSet dengan AWS SDK atau CLI
      - Gunakan DeleteTemplate dengan AWS SDK atau CLI
      - Gunakan DescribeReceiptRuleSet dengan AWS SDK atau CLI
      - Gunakan GetIdentityVerificationAttributes dengan AWS SDK atau CLI
      - Gunakan GetSendQuota dengan AWS SDK atau CLI
      - Gunakan GetSendStatistics dengan AWS SDK atau CLI
      - Gunakan GetTemplate dengan AWS SDK atau CLI
      - Gunakan ListIdentities dengan AWS SDK atau CLI
      - Gunakan ListReceiptFilters dengan AWS SDK atau CLI
      - Gunakan ListTemplates dengan AWS SDK atau CLI
      - Gunakan SendBulkTemplatedEmail dengan AWS SDK atau CLI
      - Gunakan SendEmail dengan AWS SDK atau CLI

        - Gunakan SendRawEmail dengan AWS SDK atau CLI
        - Gunakan SendTemplatedEmail dengan AWS SDK atau CLI
        - Gunakan UpdateTemplate dengan AWS SDK atau CLI
        - Gunakan VerifyDomainIdentity dengan AWS SDK atau CLI
        - Gunakan VerifyEmailIdentity dengan AWS SDK atau CLI
    - Skenario untuk Amazon SES menggunakan AWS SDKs
        - Membangun aplikasi streaming Amazon Transcribe
        - Salin SES email Amazon dan identitas domain dari satu AWS Wilayah ke Wilayah lain menggunakan AWS SDK
        - Membuat aplikasi web untuk melacak data DynamoDB
        - Buat pelacak item Amazon Redshift
        - Buat pelacak butir kerja Aurora Nirserver
        - Mendeteksi PPE dalam gambar dengan Amazon Rekognition menggunakan AWS SDK
        - Mendeteksi objek dalam gambar dengan Amazon Rekognition menggunakan AWS SDK
        - Mendeteksi orang dan objek dalam video dengan Amazon Rekognition menggunakan AWS SDK
        - Menghasilkan kredensi untuk terhubung ke titik akhir Amazon SES SMTP
        - Menggunakan Step Functions untuk menginvokasi fungsi Lambda
        - Verifikasi identitas email dan kirim pesan dengan Amazon SES menggunakan AWS SDK
- Contoh kode untuk Amazon SES API v2 menggunakan AWS SDKs
    - Contoh dasar untuk Amazon SES API v2 menggunakan AWS SDKs
        - Tindakan untuk Amazon SES API v2 menggunakan AWS SDKs
            - Gunakan CreateContact dengan AWS SDK atau CLI
            - Gunakan CreateContactList dengan AWS SDK atau CLI
            - Gunakan CreateEmailIdentity dengan AWS SDK atau CLI
            - Gunakan CreateEmailTemplate dengan AWS SDK atau CLI
            - Gunakan DeleteContactList dengan AWS SDK atau CLI
            - Gunakan DeleteEmailIdentity dengan AWS SDK atau CLI
            - Gunakan DeleteEmailTemplate dengan AWS SDK atau CLI
            - Gunakan GetEmailIdentity dengan AWS SDK atau CLI
            - Gunakan ListContactLists dengan AWS SDK atau CLI

        - Gunakan ListContacts dengan AWS SDK atau CLI
        - Gunakan SendEmail dengan AWS SDK atau CLI
- Skenario untuk Amazon SES API v2 menggunakan AWS SDKs
    - Alur kerja Newsletter Amazon SES API v2 lengkap menggunakan AWS SDK

Contoh kode untuk Amazon SES menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan Amazon SES dengan kit pengembangan AWS perangkat lunak (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Contoh kode

- Contoh dasar untuk Amazon SES menggunakan AWS SDKs
    - Tindakan untuk Amazon SES menggunakan AWS SDKs
        - Gunakan CreateReceiptFilter dengan AWS SDK atau CLI
        - Gunakan CreateReceiptRule dengan AWS SDK atau CLI
        - Gunakan CreateReceiptRuleSet dengan AWS SDK atau CLI
        - Gunakan CreateTemplate dengan AWS SDK atau CLI
        - Gunakan DeleteIdentity dengan AWS SDK atau CLI
        - Gunakan DeleteReceiptFilter dengan AWS SDK atau CLI
        - Gunakan DeleteReceiptRule dengan AWS SDK atau CLI
        - Gunakan DeleteReceiptRuleSet dengan AWS SDK atau CLI
        - Gunakan DeleteTemplate dengan AWS SDK atau CLI
        - Gunakan DescribeReceiptRuleSet dengan AWS SDK atau CLI

    - Gunakan GetIdentityVerificationAttributes dengan AWS SDK atau CLI
    - Gunakan GetSendQuota dengan AWS SDK atau CLI
    - Gunakan GetSendStatistics dengan AWS SDK atau CLI
    - Gunakan GetTemplate dengan AWS SDK atau CLI
    - Gunakan ListIdentities dengan AWS SDK atau CLI
    - Gunakan ListReceiptFilters dengan AWS SDK atau CLI
    - Gunakan ListTemplates dengan AWS SDK atau CLI
    - Gunakan SendBulkTemplatedEmail dengan AWS SDK atau CLI
    - Gunakan SendEmail dengan AWS SDK atau CLI
    - Gunakan SendRawEmail dengan AWS SDK atau CLI
    - Gunakan SendTemplatedEmail dengan AWS SDK atau CLI
    - Gunakan UpdateTemplate dengan AWS SDK atau CLI
    - Gunakan VerifyDomainIdentity dengan AWS SDK atau CLI
    - Gunakan VerifyEmailIdentity dengan AWS SDK atau CLI
- Skenario untuk Amazon SES menggunakan AWS SDKs
    - Membangun aplikasi streaming Amazon Transcribe
    - Salin SES email Amazon dan identitas domain dari satu AWS Wilayah ke Wilayah lain menggunakan AWS SDK
    - Membuat aplikasi web untuk melacak data DynamoDB
    - Buat pelacak item Amazon Redshift
    - Buat pelacak butir kerja Aurora Nirserver
    - Mendeteksi PPE dalam gambar dengan Amazon Rekognition menggunakan AWS SDK
    - Mendeteksi objek dalam gambar dengan Amazon Rekognition menggunakan AWS SDK
    - Mendeteksi orang dan objek dalam video dengan Amazon Rekognition menggunakan AWS SDK
    - Menghasilkan kredensi untuk terhubung ke titik akhir Amazon SES SMTP
    - Menggunakan Step Functions untuk menginvokasi fungsi Lambda
    - Verifikasi identitas email dan kirim pesan dengan Amazon SES menggunakan AWS SDK

Contoh dasar untuk Amazon SES menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar Amazon Simple Email Service dengan AWS SDKs.

Contoh

- Tindakan untuk Amazon SES menggunakan AWS SDKs
  - Gunakan CreateReceiptFilter dengan AWS SDK atau CLI
  - Gunakan CreateReceiptRule dengan AWS SDK atau CLI
  - Gunakan CreateReceiptRuleSet dengan AWS SDK atau CLI
  - Gunakan CreateTemplate dengan AWS SDK atau CLI
  - Gunakan DeleteIdentity dengan AWS SDK atau CLI
  - Gunakan DeleteReceiptFilter dengan AWS SDK atau CLI
  - Gunakan DeleteReceiptRule dengan AWS SDK atau CLI
  - Gunakan DeleteReceiptRuleSet dengan AWS SDK atau CLI
  - Gunakan DeleteTemplate dengan AWS SDK atau CLI
  - Gunakan DescribeReceiptRuleSet dengan AWS SDK atau CLI
  - Gunakan GetIdentityVerificationAttributes dengan AWS SDK atau CLI
  - Gunakan GetSendQuota dengan AWS SDK atau CLI
  - Gunakan GetSendStatistics dengan AWS SDK atau CLI
  - Gunakan GetTemplate dengan AWS SDK atau CLI
  - Gunakan ListIdentities dengan AWS SDK atau CLI
  - Gunakan ListReceiptFilters dengan AWS SDK atau CLI
  - Gunakan ListTemplates dengan AWS SDK atau CLI
  - Gunakan SendBulkTemplatedEmail dengan AWS SDK atau CLI
  - Gunakan SendEmail dengan AWS SDK atau CLI
  - Gunakan SendRawEmail dengan AWS SDK atau CLI
  - Gunakan SendTemplatedEmail dengan AWS SDK atau CLI
  - Gunakan UpdateTemplate dengan AWS SDK atau CLI
  - Gunakan VerifyDomainIdentity dengan AWS SDK atau CLI
  - Gunakan VerifyEmailIdentity dengan AWS SDK atau CLI

Tindakan untuk Amazon SES menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara melakukan SES tindakan Amazon individual dengan AWS SDKs. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Kutipan ini menyebut Amazon SES API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Anda dapat melihat tindakan dalam konteks diSkenario untuk Amazon SES menggunakan AWS SDKs .

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat APIReferensi Layanan Email Sederhana Amazon.

Contoh

- Gunakan CreateReceiptFilter dengan AWS SDK atau CLI
- Gunakan CreateReceiptRule dengan AWS SDK atau CLI
- Gunakan CreateReceiptRuleSet dengan AWS SDK atau CLI
- Gunakan CreateTemplate dengan AWS SDK atau CLI
- Gunakan DeleteIdentity dengan AWS SDK atau CLI
- Gunakan DeleteReceiptFilter dengan AWS SDK atau CLI
- Gunakan DeleteReceiptRule dengan AWS SDK atau CLI
- Gunakan DeleteReceiptRuleSet dengan AWS SDK atau CLI
- Gunakan DeleteTemplate dengan AWS SDK atau CLI
- Gunakan DescribeReceiptRuleSet dengan AWS SDK atau CLI
- Gunakan GetIdentityVerificationAttributes dengan AWS SDK atau CLI
- Gunakan GetSendQuota dengan AWS SDK atau CLI
- Gunakan GetSendStatistics dengan AWS SDK atau CLI
- Gunakan GetTemplate dengan AWS SDK atau CLI
- Gunakan ListIdentities dengan AWS SDK atau CLI
- Gunakan ListReceiptFilters dengan AWS SDK atau CLI
- Gunakan ListTemplates dengan AWS SDK atau CLI
- Gunakan SendBulkTemplatedEmail dengan AWS SDK atau CLI

- Gunakan SendEmail dengan AWS SDK atau CLI
- Gunakan SendRawEmail dengan AWS SDK atau CLI
- Gunakan SendTemplatedEmail dengan AWS SDK atau CLI
- Gunakan UpdateTemplate dengan AWS SDK atau CLI
- Gunakan VerifyDomainIdentity dengan AWS SDK atau CLI
- Gunakan VerifyEmailIdentity dengan AWS SDK atau CLI

Gunakan **`CreateReceiptFilter`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateReceiptFilter`.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Create an Amazon Simple Email Service (Amazon SES) receipt filter..
/*!
  \param receiptFilterName: The name for the receipt filter.
  \param cidr: IP address or IP address range in Classless Inter-Domain Routing
 (CIDR) notation.
  \param policy: Block or allow enum of type ReceiptFilterPolicy.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::createReceiptFilter(const Aws::String &receiptFilterName,
                                      const Aws::String &cidr,
                                      Aws::SES::Model::ReceiptFilterPolicy
 policy,
                                      const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);
    Aws::SES::Model::CreateReceiptFilterRequest createReceiptFilterRequest;
    Aws::SES::Model::ReceiptFilter receiptFilter;
```

```
    Aws::SES::Model::ReceiptIpFilter receiptIpFilter;
    receiptIpFilter.SetCidr(cidr);
    receiptIpFilter.SetPolicy(policy);
    receiptFilter.SetName(receiptFilterName);
    receiptFilter.SetIpFilter(receiptIpFilter);
    createReceiptFilterRequest.SetFilter(receiptFilter);
    Aws::SES::Model::CreateReceiptFilterOutcome createReceiptFilterOutcome =
 sesClient.CreateReceiptFilter(
            createReceiptFilterRequest);
    if (createReceiptFilterOutcome.IsSuccess()) {
        std::cout << "Successfully created receipt filter." << std::endl;
    }
    else {
        std::cerr << "Error creating receipt filter: " <<
                  createReceiptFilterOutcome.GetError().GetMessage() <<
 std::endl;
    }

    return createReceiptFilterOutcome.IsSuccess();
}
```

- Untuk API detailnya, lihat CreateReceiptFilterdi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import {
  CreateReceiptFilterCommand,
  ReceiptFilterPolicy,
} from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";

const createCreateReceiptFilterCommand = ({ policy, ipOrRange, name }) => {
```

```
  return new CreateReceiptFilterCommand({
    Filter: {
      IpFilter: {
        Cidr: ipOrRange, // string, either a single IP address (10.0.0.1) or an
 IP address range in CIDR notation (10.0.0.1/24)).
        Policy: policy, // enum ReceiptFilterPolicy, email traffic from the
 filtered addressesOptions.
      },
      /*
        The name of the IP address filter. Only ASCII letters, numbers,
 underscores, or dashes.
        Must be less than 64 characters and start and end with a letter or
 number.
       */
      Name: name,
    },
  });
};

const FILTER_NAME = getUniqueName("ReceiptFilter");

const run = async () => {
  const createReceiptFilterCommand = createCreateReceiptFilterCommand({
    policy: ReceiptFilterPolicy.Allow,
    ipOrRange: "10.0.0.1",
    name: FILTER_NAME,
  });

  try {
    return await sesClient.send(createReceiptFilterCommand);
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MessageRejected") {
      /** @type { import('@aws-sdk/client-ses').MessageRejected} */
      const messageRejectedError = caught;
      return messageRejectedError;
    }
    throw caught;
  }
};
```

- Untuk API detailnya, lihat CreateReceiptFilterdi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesReceiptHandler:
    """Encapsulates Amazon SES receipt handling functions."""

    def __init__(self, ses_client, s3_resource):
        """
        :param ses_client: A Boto3 Amazon SES client.
        :param s3_resource: A Boto3 Amazon S3 resource.
        """
        self.ses_client = ses_client
        self.s3_resource = s3_resource


    def create_receipt_filter(self, filter_name, ip_address_or_range, allow):
        """
        Creates a filter that allows or blocks incoming mail from an IP address
 or
        range.

        :param filter_name: The name to give the filter.
        :param ip_address_or_range: The IP address or range to block or allow.
        :param allow: When True, incoming mail is allowed from the specified IP
                      address or range; otherwise, it is blocked.
        """
        try:
            policy = "Allow" if allow else "Block"
            self.ses_client.create_receipt_filter(
                Filter={
                    "Name": filter_name,
                    "IpFilter": {"Cidr": ip_address_or_range, "Policy": policy},
                }
            )
            logger.info(
```

```
                "Created receipt filter %s to %s IP of %s.",
                filter_name,
                policy,
                ip_address_or_range,
            )
        except ClientError:
            logger.exception("Couldn't create receipt filter %s.", filter_name)
            raise
```

- Untuk API detailnya, lihat CreateReceiptFilter AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`CreateReceiptRule`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateReceiptRule`.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Create an Amazon Simple Email Service (Amazon SES) receipt rule.
/*!
  \param receiptRuleName: The name for the receipt rule.
  \param s3BucketName: The name of the S3 bucket for incoming mail.
  \param s3ObjectKeyPrefix: The prefix for the objects in the S3 bucket.
  \param ruleSetName: The name of the rule set where the receipt rule is added.
  \param recipients: Aws::Vector of recipients.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
```

```
bool AwsDoc::SES::createReceiptRule(const Aws::String &receiptRuleName,
                                    const Aws::String &s3BucketName,
                                    const Aws::String &s3ObjectKeyPrefix,
                                    const Aws::String &ruleSetName,
                                    const Aws::Vector<Aws::String> &recipients,
                                    const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::CreateReceiptRuleRequest createReceiptRuleRequest;

    Aws::SES::Model::S3Action s3Action;
    s3Action.SetBucketName(s3BucketName);
    s3Action.SetObjectKeyPrefix(s3ObjectKeyPrefix);

    Aws::SES::Model::ReceiptAction receiptAction;
    receiptAction.SetS3Action(s3Action);

    Aws::SES::Model::ReceiptRule receiptRule;
    receiptRule.SetName(receiptRuleName);
    receiptRule.WithRecipients(recipients);

    Aws::Vector<Aws::SES::Model::ReceiptAction> receiptActionList;
    receiptActionList.emplace_back(receiptAction);
    receiptRule.SetActions(receiptActionList);

    createReceiptRuleRequest.SetRuleSetName(ruleSetName);
    createReceiptRuleRequest.SetRule(receiptRule);

    auto outcome = sesClient.CreateReceiptRule(createReceiptRuleRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully created receipt rule." << std::endl;
    }
    else {
        std::cerr << "Error creating receipt rule. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat CreateReceiptRuledi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { CreateReceiptRuleCommand, TlsPolicy } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";

const RULE_SET_NAME = getUniqueName("RuleSetName");
const RULE_NAME = getUniqueName("RuleName");
const S3_BUCKET_NAME = getUniqueName("S3BucketName");

const createS3ReceiptRuleCommand = ({
  bucketName,
  emailAddresses,
  name,
  ruleSet,
}) => {
  return new CreateReceiptRuleCommand({
    Rule: {
      Actions: [
        {
          S3Action: {
            BucketName: bucketName,
            ObjectKeyPrefix: "email",
          },
        },
      ],
      Recipients: emailAddresses,
      Enabled: true,
      Name: name,
      ScanEnabled: false,
```

```
      TlsPolicy: TlsPolicy.Optional,
    },
    RuleSetName: ruleSet, // Required
  });
};

const run = async () => {
  const s3ReceiptRuleCommand = createS3ReceiptRuleCommand({
    bucketName: S3_BUCKET_NAME,
    emailAddresses: ["email@example.com"],
    name: RULE_NAME,
    ruleSet: RULE_SET_NAME,
  });

  try {
    return await sesClient.send(s3ReceiptRuleCommand);
  } catch (err) {
    console.log("Failed to create S3 receipt rule.", err);
    throw err;
  }
};
```

- Untuk API detailnya, lihat CreateReceiptRuledi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Buat bucket Amazon S3 tempat Amazon SES dapat menempatkan salinan email masuk dan membuat aturan yang menyalin email masuk ke bucket untuk daftar penerima tertentu.

```
class SesReceiptHandler:
    """Encapsulates Amazon SES receipt handling functions."""

    def __init__(self, ses_client, s3_resource):
```

```
        """
        :param ses_client: A Boto3 Amazon SES client.
        :param s3_resource: A Boto3 Amazon S3 resource.
        """
        self.ses_client = ses_client
        self.s3_resource = s3_resource


    def create_bucket_for_copy(self, bucket_name):
        """
        Creates a bucket that can receive copies of emails from Amazon SES. This
        includes adding a policy to the bucket that grants Amazon SES permission
        to put objects in the bucket.

        :param bucket_name: The name of the bucket to create.
        :return: The newly created bucket.
        """
        allow_ses_put_policy = {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "AllowSESPut",
                    "Effect": "Allow",
                    "Principal": {"Service": "ses.amazonaws.com"},
                    "Action": "s3:PutObject",
                    "Resource": f"arn:aws:s3:::{bucket_name}/*",
                }
            ],
        }
        bucket = None
        try:
            bucket = self.s3_resource.create_bucket(
                Bucket=bucket_name,
                CreateBucketConfiguration={
                    "LocationConstraint":
self.s3_resource.meta.client.meta.region_name
                },
            )
            bucket.wait_until_exists()
            bucket.Policy().put(Policy=json.dumps(allow_ses_put_policy))
            logger.info("Created bucket %s to receive copies of emails.",
bucket_name)
        except ClientError:
```

```
            logger.exception("Couldn't create bucket to receive copies of
emails.")
            if bucket is not None:
                bucket.delete()
            raise
        else:
            return bucket


    def create_s3_copy_rule(
        self, rule_set_name, rule_name, recipients, bucket_name, prefix
    ):
        """
        Creates a rule so that all emails received by the specified recipients
are
        copied to an Amazon S3 bucket.

        :param rule_set_name: The name of a previously created rule set to
contain
                              this rule.
        :param rule_name: The name to give the rule.
        :param recipients: When an email is received by one of these recipients,
it
                           is copied to the Amazon S3 bucket.
        :param bucket_name: The name of the bucket to receive email copies. This
                            bucket must allow Amazon SES to put objects into it.
        :param prefix: An object key prefix to give the emails copied to the
bucket.
        """
        try:
            self.ses_client.create_receipt_rule(
                RuleSetName=rule_set_name,
                Rule={
                    "Name": rule_name,
                    "Enabled": True,
                    "Recipients": recipients,
                    "Actions": [
                        {
                            "S3Action": {
                                "BucketName": bucket_name,
                                "ObjectKeyPrefix": prefix,
                            }
                        }
                    ],
```

```
                },
            )
            logger.info(
                "Created rule %s to copy mail received by %s to bucket %s.",
                rule_name,
                recipients,
                bucket_name,
            )
        except ClientError:
            logger.exception("Couldn't create rule %s.", rule_name)
            raise
```

- Untuk API detailnya, lihat CreateReceiptRule AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`CreateReceiptRuleSet`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateReceiptRuleSet`.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Create an Amazon Simple Email Service (Amazon SES) receipt rule set.
/*!
  \param ruleSetName: The name of the rule set.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::createReceiptRuleSet(const Aws::String &ruleSetName,
```

```
                                        const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::CreateReceiptRuleSetRequest createReceiptRuleSetRequest;

    createReceiptRuleSetRequest.SetRuleSetName(ruleSetName);

    Aws::SES::Model::CreateReceiptRuleSetOutcome outcome =
 sesClient.CreateReceiptRuleSet(
            createReceiptRuleSetRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully created receipt rule set." << std::endl;
    }
    else {
        std::cerr << "Error creating receipt rule set. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat CreateReceiptRuleSetdi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { CreateReceiptRuleSetCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";

const RULE_SET_NAME = getUniqueName("RuleSetName");
```

```
const createCreateReceiptRuleSetCommand = (ruleSetName) => {
  return new CreateReceiptRuleSetCommand({ RuleSetName: ruleSetName });
};

const run = async () => {
  const createReceiptRuleSetCommand =
    createCreateReceiptRuleSetCommand(RULE_SET_NAME);

  try {
    return await sesClient.send(createReceiptRuleSetCommand);
  } catch (err) {
    console.log("Failed to create receipt rule set", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat CreateReceiptRuleSetdi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesReceiptHandler:
    """Encapsulates Amazon SES receipt handling functions."""

    def __init__(self, ses_client, s3_resource):
        """
        :param ses_client: A Boto3 Amazon SES client.
        :param s3_resource: A Boto3 Amazon S3 resource.
        """
        self.ses_client = ses_client
        self.s3_resource = s3_resource
```

```
    def create_receipt_rule_set(self, rule_set_name):
        """
        Creates an empty rule set. Rule sets contain individual rules and can be
        used to organize rules.

        :param rule_set_name: The name to give the rule set.
        """
        try:
            self.ses_client.create_receipt_rule_set(RuleSetName=rule_set_name)
            logger.info("Created receipt rule set %s.", rule_set_name)
        except ClientError:
            logger.exception("Couldn't create receipt rule set %s.",
rule_set_name)
            raise
```

- Untuk API detailnya, lihat CreateReceiptRuleSet AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`CreateTemplate`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateTemplate`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Create an email template.
   /// </summary>
   /// <param name="name">Name of the template.</param>
   /// <param name="subject">Email subject.</param>
   /// <param name="text">Email body text.</param>
   /// <param name="html">Email HTML body text.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> CreateEmailTemplateAsync(string name, string subject,
string text,
       string html)
   {
       var success = false;
       try
       {
           var response = await _amazonSimpleEmailService.CreateTemplateAsync(
               new CreateTemplateRequest
               {
                   Template = new Template
                   {
                       TemplateName = name,
                       SubjectPart = subject,
                       TextPart = text,
                       HtmlPart = html
                   }
               });
           success = response.HttpStatusCode == HttpStatusCode.OK;
       }
       catch (Exception ex)
       {
           Console.WriteLine("CreateEmailTemplateAsync failed with exception: "
+ ex.Message);
       }

       return success;
   }
```

- Untuk API detailnya, lihat CreateTemplatedi AWS SDK for .NET APIReferensi.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Create an Amazon Simple Email Service (Amazon SES) template.
/*!
  \param templateName: The name of the template.
  \param htmlPart: The HTML body of the email.
  \param subjectPart: The subject line of the email.
  \param textPart: The plain text version of the email.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::createTemplate(const Aws::String &templateName,
                                 const Aws::String &htmlPart,
                                 const Aws::String &subjectPart,
                                 const Aws::String &textPart,
                                 const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::CreateTemplateRequest createTemplateRequest;
    Aws::SES::Model::Template aTemplate;

    aTemplate.SetTemplateName(templateName);
    aTemplate.SetHtmlPart(htmlPart);
    aTemplate.SetSubjectPart(subjectPart);
    aTemplate.SetTextPart(textPart);

    createTemplateRequest.SetTemplate(aTemplate);

    Aws::SES::Model::CreateTemplateOutcome outcome = sesClient.CreateTemplate(
            createTemplateRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully created template." << templateName << "."
```

```
                  << std::endl;
    }
    else {
        std::cerr << "Error creating template. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat CreateTemplatedi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { CreateTemplateCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";

const TEMPLATE_NAME = getUniqueName("TestTemplateName");

const createCreateTemplateCommand = () => {
  return new CreateTemplateCommand({
    /**
     * The template feature in Amazon SES is based on the Handlebars template
 system.
     */
    Template: {
      /**
       * The name of an existing template in Amazon SES.
       */
```

```
      TemplateName: TEMPLATE_NAME,
      HtmlPart: `
        <h1>Hello, {{contact.firstName}}!</h1>
        <p>
        Did you know Amazon has a mascot named Peccy?
        </p>
      `,
      SubjectPart: "Amazon Tip",
    },
  });
};

const run = async () => {
  const createTemplateCommand = createCreateTemplateCommand();

  try {
    return await sesClient.send(createTemplateCommand);
  } catch (err) {
    console.log("Failed to create template.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat CreateTemplatedi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesTemplate:
    """Encapsulates Amazon SES template functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
```

```
        """
        self.ses_client = ses_client
        self.template = None
        self.template_tags = set()

    def _extract_tags(self, subject, text, html):
        """
        Extracts tags from a template as a set of unique values.

        :param subject: The subject of the email.
        :param text: The text version of the email.
        :param html: The html version of the email.
        """
        self.template_tags = set(re.findall(TEMPLATE_REGEX, subject + text +
html))
        logger.info("Extracted template tags: %s", self.template_tags)


    def create_template(self, name, subject, text, html):
        """
        Creates an email template.

        :param name: The name of the template.
        :param subject: The subject of the email.
        :param text: The plain text version of the email.
        :param html: The HTML version of the email.
        """
        try:
            template = {
                "TemplateName": name,
                "SubjectPart": subject,
                "TextPart": text,
                "HtmlPart": html,
            }
            self.ses_client.create_template(Template=template)
            logger.info("Created template %s.", name)
            self.template = template
            self._extract_tags(subject, text, html)
        except ClientError:
            logger.exception("Couldn't create template %s.", name)
            raise
```

- Untuk API detailnya, lihat CreateTemplate AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteIdentity`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteIdentity`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <summary>
/// Delete an email identity.
/// </summary>
/// <param name="identityEmail">The identity email to delete.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteIdentityAsync(string identityEmail)
{
    var success = false;
    try
    {
        var response = await _amazonSimpleEmailService.DeleteIdentityAsync(
            new DeleteIdentityRequest
            {
                Identity = identityEmail
            });
```

```
            success = response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (Exception ex)
        {
            Console.WriteLine("DeleteIdentityAsync failed with exception: " +
 ex.Message);
        }

        return success;
    }
```

- Untuk API detailnya, lihat DeleteIdentitydi AWS SDK for .NET APIReferensi.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Delete the specified identity (an email address or a domain).
/*!
  \param identity: The identity to delete.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::deleteIdentity(const Aws::String &identity,
                                 const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::DeleteIdentityRequest deleteIdentityRequest;

    deleteIdentityRequest.SetIdentity(identity);

    Aws::SES::Model::DeleteIdentityOutcome outcome = sesClient.DeleteIdentity(
            deleteIdentityRequest);
```

```
    if (outcome.IsSuccess()) {
        std::cout << "Successfully deleted identity." << std::endl;
    }
    else {
        std::cerr << "Error deleting identity. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat DeleteIdentitydi AWS SDK for C++ APIReferensi.

CLI

AWS CLI

Untuk menghapus identitas

Contoh berikut menggunakan `delete-identity` perintah untuk menghapus identitas dari daftar identitas yang diverifikasi dengan AmazonSES:

```
aws ses delete-identity --identity user@example.com
```

Untuk informasi selengkapnya tentang identitas terverifikasi, lihat Memverifikasi Alamat Email dan Domain SES di Amazon di Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat DeleteIdentitydi Referensi AWS CLI Perintah.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { DeleteIdentityCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";

const IDENTITY_EMAIL = "fake@example.com";

const createDeleteIdentityCommand = (identityName) => {
  return new DeleteIdentityCommand({
    Identity: identityName,
  });
};

const run = async () => {
  const deleteIdentityCommand = createDeleteIdentityCommand(IDENTITY_EMAIL);

  try {
    return await sesClient.send(deleteIdentityCommand);
  } catch (err) {
    console.log("Failed to delete identity.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat DeleteIdentitydi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesIdentity:
    """Encapsulates Amazon SES identity functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
```

```
        """
        self.ses_client = ses_client


    def delete_identity(self, identity):
        """
        Deletes an identity.

        :param identity: The identity to remove.
        """
        try:
            self.ses_client.delete_identity(Identity=identity)
            logger.info("Deleted identity %s.", identity)
        except ClientError:
            logger.exception("Couldn't delete identity %s.", identity)
            raise
```

- Untuk API detailnya, lihat DeleteIdentity AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteReceiptFilter`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteReceiptFilter`.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Delete an Amazon Simple Email Service (Amazon SES) receipt filter.
/*!
```

```
  \param receiptFilterName: The name for the receipt filter.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::deleteReceiptFilter(const Aws::String &receiptFilterName,
                                      const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::DeleteReceiptFilterRequest deleteReceiptFilterRequest;

    deleteReceiptFilterRequest.SetFilterName(receiptFilterName);

    Aws::SES::Model::DeleteReceiptFilterOutcome outcome =
 sesClient.DeleteReceiptFilter(
            deleteReceiptFilterRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully deleted receipt filter." << std::endl;
    }
    else {
        std::cerr << "Error deleting receipt filter. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat DeleteReceiptFilterdi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { DeleteReceiptFilterCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";

const RECEIPT_FILTER_NAME = getUniqueName("ReceiptFilterName");

const createDeleteReceiptFilterCommand = (filterName) => {
  return new DeleteReceiptFilterCommand({ FilterName: filterName });
};

const run = async () => {
  const deleteReceiptFilterCommand =
    createDeleteReceiptFilterCommand(RECEIPT_FILTER_NAME);

  try {
    return await sesClient.send(deleteReceiptFilterCommand);
  } catch (err) {
    console.log("Error deleting receipt filter.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat DeleteReceiptFilterdi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesReceiptHandler:
    """Encapsulates Amazon SES receipt handling functions."""

    def __init__(self, ses_client, s3_resource):
        """
        :param ses_client: A Boto3 Amazon SES client.
```

```
        :param s3_resource: A Boto3 Amazon S3 resource.
        """
        self.ses_client = ses_client
        self.s3_resource = s3_resource


    def delete_receipt_filter(self, filter_name):
        """
        Deletes a receipt filter.

        :param filter_name: The name of the filter to delete.
        """
        try:
            self.ses_client.delete_receipt_filter(FilterName=filter_name)
            logger.info("Deleted receipt filter %s.", filter_name)
        except ClientError:
            logger.exception("Couldn't delete receipt filter %s.", filter_name)
            raise
```

- Untuk API detailnya, lihat DeleteReceiptFilter AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteReceiptRule`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteReceiptRule`.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Delete an Amazon Simple Email Service (Amazon SES) receipt rule.
/*!
  \param receiptRuleName: The name for the receipt rule.
  \param receiptRuleSetName: The name for the receipt rule set.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::deleteReceiptRule(const Aws::String &receiptRuleName,
                                    const Aws::String &receiptRuleSetName,
                                    const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::DeleteReceiptRuleRequest deleteReceiptRuleRequest;

    deleteReceiptRuleRequest.SetRuleName(receiptRuleName);
    deleteReceiptRuleRequest.SetRuleSetName(receiptRuleSetName);

    Aws::SES::Model::DeleteReceiptRuleOutcome outcome =
 sesClient.DeleteReceiptRule(
            deleteReceiptRuleRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully deleted receipt rule." << std::endl;
    }
    else {
        std::cout << "Error deleting receipt rule. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat DeleteReceiptRuledi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { DeleteReceiptRuleCommand } from "@aws-sdk/client-ses";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { sesClient } from "./libs/sesClient.js";

const RULE_NAME = getUniqueName("RuleName");
const RULE_SET_NAME = getUniqueName("RuleSetName");

const createDeleteReceiptRuleCommand = () => {
  return new DeleteReceiptRuleCommand({
    RuleName: RULE_NAME,
    RuleSetName: RULE_SET_NAME,
  });
};

const run = async () => {
  const deleteReceiptRuleCommand = createDeleteReceiptRuleCommand();
  try {
    return await sesClient.send(deleteReceiptRuleCommand);
  } catch (err) {
    console.log("Failed to delete receipt rule.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat DeleteReceiptRuledi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesReceiptHandler:
    """Encapsulates Amazon SES receipt handling functions."""

    def __init__(self, ses_client, s3_resource):
        """
        :param ses_client: A Boto3 Amazon SES client.
        :param s3_resource: A Boto3 Amazon S3 resource.
        """
        self.ses_client = ses_client
        self.s3_resource = s3_resource


    def delete_receipt_rule(self, rule_set_name, rule_name):
        """
        Deletes a rule.

        :param rule_set_name: The rule set that contains the rule to delete.
        :param rule_name: The rule to delete.
        """
        try:
            self.ses_client.delete_receipt_rule(
                RuleSetName=rule_set_name, RuleName=rule_name
            )
            logger.info("Removed rule %s from rule set %s.", rule_name,
 rule_set_name)
        except ClientError:
            logger.exception(
                "Couldn't remove rule %s from rule set %s.", rule_name,
 rule_set_name
            )
            raise
```

- Untuk API detailnya, lihat DeleteReceiptRule AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteReceiptRuleSet`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteReceiptRuleSet`.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Delete an Amazon Simple Email Service (Amazon SES) receipt rule set.
/*!
  \param receiptRuleSetName: The name for the receipt rule set.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::deleteReceiptRuleSet(const Aws::String &receiptRuleSetName,
                                       const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::DeleteReceiptRuleSetRequest deleteReceiptRuleSetRequest;

    deleteReceiptRuleSetRequest.SetRuleSetName(receiptRuleSetName);

    Aws::SES::Model::DeleteReceiptRuleSetOutcome outcome =
 sesClient.DeleteReceiptRuleSet(
            deleteReceiptRuleSetRequest);
```

```
    if (outcome.IsSuccess()) {
        std::cout << "Successfully deleted receipt rule set." << std::endl;
    }

    else {
        std::cerr << "Error deleting receipt rule set. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat DeleteReceiptRuleSetdi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { DeleteReceiptRuleSetCommand } from "@aws-sdk/client-ses";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { sesClient } from "./libs/sesClient.js";

const RULE_SET_NAME = getUniqueName("RuleSetName");

const createDeleteReceiptRuleSetCommand = () => {
  return new DeleteReceiptRuleSetCommand({ RuleSetName: RULE_SET_NAME });
};

const run = async () => {
  const deleteReceiptRuleSetCommand = createDeleteReceiptRuleSetCommand();

  try {
    return await sesClient.send(deleteReceiptRuleSetCommand);
  } catch (err) {
```

```
    console.log("Failed to delete receipt rule set.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat DeleteReceiptRuleSetdi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesReceiptHandler:
    """Encapsulates Amazon SES receipt handling functions."""

    def __init__(self, ses_client, s3_resource):
        """
        :param ses_client: A Boto3 Amazon SES client.
        :param s3_resource: A Boto3 Amazon S3 resource.
        """
        self.ses_client = ses_client
        self.s3_resource = s3_resource


    def delete_receipt_rule_set(self, rule_set_name):
        """
        Deletes a rule set. When a rule set is deleted, all of the rules it
 contains
        are also deleted.

        :param rule_set_name: The name of the rule set to delete.
        """
        try:
            self.ses_client.delete_receipt_rule_set(RuleSetName=rule_set_name)
            logger.info("Deleted rule set %s.", rule_set_name)
        except ClientError:
```

```
            logger.exception("Couldn't delete rule set %s.", rule_set_name)
            raise
```

- Untuk API detailnya, lihat DeleteReceiptRuleSet AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteTemplate`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteTemplate`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <summary>
/// Delete an email template.
/// </summary>
/// <param name="templateName">Name of the template.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteEmailTemplateAsync(string templateName)
{
    var success = false;
```

```
        try
        {
            var response = await _amazonSimpleEmailService.DeleteTemplateAsync(
                new DeleteTemplateRequest
                {
                    TemplateName = templateName
                });
            success = response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (Exception ex)
        {
            Console.WriteLine("DeleteEmailTemplateAsync failed with exception: "
 + ex.Message);
        }

        return success;
    }
```

- Untuk API detailnya, lihat DeleteTemplatedi AWS SDK for .NET APIReferensi.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Delete an Amazon Simple Email Service (Amazon SES) template.
/*!
  \param templateName: The name for the template.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::deleteTemplate(const Aws::String &templateName,
                                 const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);
```

```
    Aws::SES::Model::DeleteTemplateRequest deleteTemplateRequest;

    deleteTemplateRequest.SetTemplateName(templateName);

    Aws::SES::Model::DeleteTemplateOutcome outcome = sesClient.DeleteTemplate(
            deleteTemplateRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully deleted template." << std::endl;
    }
    else {
        std::cerr << "Error deleting template. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat DeleteTemplatedi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { DeleteTemplateCommand } from "@aws-sdk/client-ses";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { sesClient } from "./libs/sesClient.js";

const TEMPLATE_NAME = getUniqueName("TemplateName");

const createDeleteTemplateCommand = (templateName) =>
  new DeleteTemplateCommand({ TemplateName: templateName });
```

```
const run = async () => {
  const deleteTemplateCommand = createDeleteTemplateCommand(TEMPLATE_NAME);

  try {
    return await sesClient.send(deleteTemplateCommand);
  } catch (err) {
    console.log("Failed to delete template.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat DeleteTemplatedi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesTemplate:
    """Encapsulates Amazon SES template functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client
        self.template = None
        self.template_tags = set()

    def _extract_tags(self, subject, text, html):
        """
        Extracts tags from a template as a set of unique values.

        :param subject: The subject of the email.
        :param text: The text version of the email.
```

```
        :param html: The html version of the email.
        """
        self.template_tags = set(re.findall(TEMPLATE_REGEX, subject + text +
html))
        logger.info("Extracted template tags: %s", self.template_tags)


    def delete_template(self):
        """
        Deletes an email template.
        """
        try:

self.ses_client.delete_template(TemplateName=self.template["TemplateName"])
            logger.info("Deleted template %s.", self.template["TemplateName"])
            self.template = None
            self.template_tags = None
        except ClientError:
            logger.exception(
                "Couldn't delete template %s.", self.template["TemplateName"]
            )
            raise
```

- Untuk API detailnya, lihat DeleteTemplate AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DescribeReceiptRuleSet`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DescribeReceiptRuleSet`.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesReceiptHandler:
    """Encapsulates Amazon SES receipt handling functions."""

    def __init__(self, ses_client, s3_resource):
        """
        :param ses_client: A Boto3 Amazon SES client.
        :param s3_resource: A Boto3 Amazon S3 resource.
        """
        self.ses_client = ses_client
        self.s3_resource = s3_resource


    def describe_receipt_rule_set(self, rule_set_name):
        """
        Gets data about a rule set.

        :param rule_set_name: The name of the rule set to retrieve.
        :return: Data about the rule set.
        """
        try:
            response = self.ses_client.describe_receipt_rule_set(
                RuleSetName=rule_set_name
            )
            logger.info("Got data for rule set %s.", rule_set_name)
        except ClientError:
            logger.exception("Couldn't get data for rule set %s.", rule_set_name)
            raise
        else:
            return response
```

- Untuk API detailnya, lihat DescribeReceiptRuleSet AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`GetIdentityVerificationAttributes`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetIdentityVerificationAttributes`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Get identity verification status for an email.
    /// </summary>
    /// <returns>The verification status of the email.</returns>
    public async Task<VerificationStatus> GetIdentityStatusAsync(string email)
    {
        var result = VerificationStatus.TemporaryFailure;
        try
        {
            var response =
                await
_amazonSimpleEmailService.GetIdentityVerificationAttributesAsync(
                    new GetIdentityVerificationAttributesRequest
                    {
```

```
                        Identities = new List<string> { email }
                    });

            if (response.VerificationAttributes.ContainsKey(email))
                result =
response.VerificationAttributes[email].VerificationStatus;
        }
        catch (Exception ex)
        {
            Console.WriteLine("GetIdentityStatusAsync failed with exception: " +
ex.Message);
        }

        return result;
    }
```

- Untuk API detailnya, lihat GetIdentityVerificationAttributesdi AWS SDK for .NET APIReferensi.

CLI

AWS CLI

Untuk mendapatkan status SES verifikasi Amazon untuk daftar identitas

Contoh berikut menggunakan `get-identity-verification-attributes` perintah untuk mengambil status SES verifikasi Amazon untuk daftar identitas:

```
aws ses get-identity-verification-attributes --
identities "user1@example.com" "user2@example.com"
```

Output:

```
{
    "VerificationAttributes": {
        "user1@example.com": {
            "VerificationStatus": "Success"
        },
        "user2@example.com": {
            "VerificationStatus": "Pending"
```

```
        }
    }
}
```

Jika Anda memanggil perintah ini dengan identitas yang belum pernah Anda kirimkan untuk verifikasi, identitas itu tidak akan muncul di output.

Untuk informasi selengkapnya tentang identitas terverifikasi, lihat Memverifikasi Alamat Email dan Domain SES di Amazon di Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat GetIdentityVerificationAttributesdi Referensi AWS CLI Perintah.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesIdentity:
    """Encapsulates Amazon SES identity functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client


    def get_identity_status(self, identity):
        """
        Gets the status of an identity. This can be used to discover whether
        an identity has been successfully verified.

        :param identity: The identity to query.
        :return: The status of the identity.
        """
        try:
            response = self.ses_client.get_identity_verification_attributes(
```

```
                Identities=[identity]
            )
            status = response["VerificationAttributes"].get(
                identity, {"VerificationStatus": "NotFound"}
            )["VerificationStatus"]
            logger.info("Got status of %s for %s.", status, identity)
        except ClientError:
            logger.exception("Couldn't get status for %s.", identity)
            raise
        else:
            return status
```

- Untuk API detailnya, lihat GetIdentityVerificationAttributes AWSSDKReferensi Python (Boto3). API

Ruby

SDKuntuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-ses' # v2: require 'aws-sdk'

# Create client in us-west-2 region
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
client = Aws::SES::Client.new(region: 'us-west-2')

# Get up to 1000 identities
ids = client.list_identities({
                               identity_type: 'EmailAddress'
                             })

ids.identities.each do |email|
  attrs = client.get_identity_verification_attributes({
```

```
                                                    identities: [email]
                                                })

  status = attrs.verification_attributes[email].verification_status

  # Display email addresses that have been verified
  puts email if status == 'Success'
end
```

- Untuk API detailnya, lihat GetIdentityVerificationAttributesdi AWS SDK for Ruby APIReferensi.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`GetSendQuota`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetSendQuota`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Get information on the current account's send quota.
    /// </summary>
    /// <returns>The send quota response data.</returns>
    public async Task<GetSendQuotaResponse> GetSendQuotaAsync()
    {
        var result = new GetSendQuotaResponse();
        try
        {
```

```
            var response = await _amazonSimpleEmailService.GetSendQuotaAsync(
                new GetSendQuotaRequest());
            result = response;
        }
        catch (Exception ex)
        {
            Console.WriteLine("GetSendQuotaAsync failed with exception: " +
 ex.Message);
        }

        return result;
    }
```

- Untuk API detailnya, lihat GetSendQuotadi AWS SDK for .NET APIReferensi.

CLI

AWS CLI

Untuk mendapatkan batas SES pengiriman Amazon Anda

Contoh berikut menggunakan `get-send-quota` perintah untuk mengembalikan batas SES pengiriman Amazon Anda:

```
aws ses get-send-quota
```

Output:

```
{
   "Max24HourSend": 200.0,
   "SentLast24Hours": 1.0,
   "MaxSendRate": 1.0
}
```

Max24 HourSend adalah kuota pengiriman Anda, yaitu jumlah email maksimum yang dapat Anda kirim dalam jangka waktu 24 jam. Kuota pengiriman mencerminkan periode waktu bergulir. Setiap kali Anda mencoba mengirim email, Amazon SES memeriksa berapa banyak email yang Anda kirim dalam 24 jam sebelumnya. Selama jumlah email yang Anda kirim kurang dari kuota Anda, permintaan kirim Anda akan diterima dan email Anda akan terkirim.

SentLast24Hours adalah jumlah email yang telah Anda kirim dalam 24 jam sebelumnya.

MaxSendRate adalah jumlah maksimum email yang dapat Anda kirim per detik.

Perhatikan bahwa batas pengiriman didasarkan pada penerima, bukan pada pesan. Misalnya, email yang memiliki 10 penerima dihitung sebagai 10 terhadap kuota pengiriman Anda.

Untuk informasi selengkapnya, lihat Mengelola Batas SES Pengiriman Amazon Anda di Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat GetSendQuotadi Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Perintah ini mengembalikan batas pengiriman pengguna saat ini.

```
Get-SESSendQuota
```

- Untuk API detailnya, lihat GetSendQuotadi AWS Tools for PowerShell Referensi Cmdlet.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`GetSendStatistics`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetSendStatistics`.

CLI

AWS CLI

Untuk mendapatkan statistik SES pengiriman Amazon Anda

Contoh berikut menggunakan `get-send-statistics` perintah untuk mengembalikan statistik SES pengiriman Amazon Anda

```
aws ses get-send-statistics
```

Output:

```
{
   "SendDataPoints": [
       {
           "Complaints": 0,
           "Timestamp": "2013-06-12T19:32:00Z",
           "DeliveryAttempts": 2,
           "Bounces": 0,
           "Rejects": 0
       },
       {
           "Complaints": 0,
           "Timestamp": "2013-06-12T00:47:00Z",
           "DeliveryAttempts": 1,
           "Bounces": 0,
           "Rejects": 0
       }
   ]
}
```

Hasilnya adalah daftar titik data, yang mewakili dua minggu terakhir aktivitas pengiriman. Setiap titik data dalam daftar berisi statistik untuk interval 15 menit.

Dalam contoh ini, hanya ada dua titik data karena satu-satunya email yang dikirim pengguna dalam dua minggu terakhir jatuh dalam dua interval 15 menit.

Untuk informasi selengkapnya, lihat Memantau Statistik SES Penggunaan Amazon Anda di Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat GetSendStatisticsdi Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell

Contoh 1: Perintah ini mengembalikan statistik pengiriman pengguna. Hasilnya adalah daftar titik data, yang mewakili dua minggu terakhir aktivitas pengiriman. Setiap titik data dalam daftar berisi statistik untuk interval 15 menit.

```
Get-SESSendStatistic
```

- Untuk API detailnya, lihat GetSendStatisticsdi AWS Tools for PowerShell Referensi Cmdlet.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`GetTemplate`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetTemplate`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Get a template's attributes.
/*!
  \param templateName: The name for the template.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::getTemplate(const Aws::String &templateName,
                              const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::GetTemplateRequest getTemplateRequest;

    getTemplateRequest.SetTemplateName(templateName);
```

```
    Aws::SES::Model::GetTemplateOutcome outcome = sesClient.GetTemplate(
            getTemplateRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully got template." << std::endl;
    }

    else {
        std::cerr << "Error getting template. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat GetTemplatedi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { GetTemplateCommand } from "@aws-sdk/client-ses";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { sesClient } from "./libs/sesClient.js";

const TEMPLATE_NAME = getUniqueName("TemplateName");

const createGetTemplateCommand = (templateName) =>
  new GetTemplateCommand({ TemplateName: templateName });

const run = async () => {
  const getTemplateCommand = createGetTemplateCommand(TEMPLATE_NAME);
```

```
  try {
    return await sesClient.send(getTemplateCommand);
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MessageRejected") {
      /** @type { import('@aws-sdk/client-ses').MessageRejected} */
      const messageRejectedError = caught;
      return messageRejectedError;
    }
    throw caught;
  }
};
```

- Untuk API detailnya, lihat GetTemplatedi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesTemplate:
    """Encapsulates Amazon SES template functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client
        self.template = None
        self.template_tags = set()

    def _extract_tags(self, subject, text, html):
        """
        Extracts tags from a template as a set of unique values.

        :param subject: The subject of the email.
        :param text: The text version of the email.
```

```
        :param html: The html version of the email.
        """
        self.template_tags = set(re.findall(TEMPLATE_REGEX, subject + text +
html))
        logger.info("Extracted template tags: %s", self.template_tags)


    def get_template(self, name):
        """
        Gets a previously created email template.

        :param name: The name of the template to retrieve.
        :return: The retrieved email template.
        """
        try:
            response = self.ses_client.get_template(TemplateName=name)
            self.template = response["Template"]
            logger.info("Got template %s.", name)
            self._extract_tags(
                self.template["SubjectPart"],
                self.template["TextPart"],
                self.template["HtmlPart"],
            )
        except ClientError:
            logger.exception("Couldn't get template %s.", name)
            raise
        else:
            return self.template
```

- Untuk API detailnya, lihat GetTemplate AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`ListIdentities`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListIdentities`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Salin identitas email dan domain di seluruh Wilayah
- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Get the identities of a specified type for the current account.
   /// </summary>
   /// <param name="identityType">IdentityType to list.</param>
   /// <returns>The list of identities.</returns>
   public async Task<List<string>> ListIdentitiesAsync(IdentityType
identityType)
   {
       var result = new List<string>();
       try
       {
           var response = await _amazonSimpleEmailService.ListIdentitiesAsync(
               new ListIdentitiesRequest
               {
                   IdentityType = identityType
               });
           result = response.Identities;
       }
       catch (Exception ex)
       {
           Console.WriteLine("ListIdentitiesAsync failed with exception: " +
ex.Message);
       }

       return result;
   }
```

- Untuk API detailnya, lihat ListIdentitiesdi AWS SDK for .NET APIReferensi.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! List the identities associated with this account.
/*!
  \param identityType: The identity type enum. "NOT_SET" is a valid option.
  \param identities; A vector to receive the retrieved identities.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::listIdentities(Aws::SES::Model::IdentityType identityType,
                                 Aws::Vector<Aws::String> &identities,
                                 const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::ListIdentitiesRequest listIdentitiesRequest;

    if (identityType != Aws::SES::Model::IdentityType::NOT_SET) {
        listIdentitiesRequest.SetIdentityType(identityType);
    }

    Aws::String nextToken; // Used for paginated results.
    do {
        if (!nextToken.empty()) {
            listIdentitiesRequest.SetNextToken(nextToken);
        }
        Aws::SES::Model::ListIdentitiesOutcome outcome =
 sesClient.ListIdentities(
                listIdentitiesRequest);
```

```
        if (outcome.IsSuccess()) {
            const auto &retrievedIdentities =
 outcome.GetResult().GetIdentities();
            if (!retrievedIdentities.empty()) {
                identities.insert(identities.cend(),
 retrievedIdentities.cbegin(),
                                  retrievedIdentities.cend());
            }
            nextToken = outcome.GetResult().GetNextToken();
        }
        else {
            std::cout << "Error listing identities. " <<
 outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!nextToken.empty());

    return true;
}
```

- Untuk API detailnya, lihat ListIdentitiesdi AWS SDK for C++ APIReferensi.

CLI

AWS CLI

Untuk mencantumkan semua identitas (alamat email dan domain) untuk akun tertentu AWS

Contoh berikut menggunakan `list-identities` perintah untuk mencantumkan semua identitas yang telah dikirimkan untuk verifikasi dengan AmazonSES:

```
aws ses list-identities
```

Output:

```
{
    "Identities": [
      "user@example.com",
      "example.com"
```

```
    ]
}
```

Daftar yang dikembalikan berisi semua identitas terlepas dari status verifikasi (verifikasi, verifikasi tertunda, kegagalan, dll.).

Dalam contoh ini, alamat email dan domain dikembalikan karena kami tidak menentukan parameter tipe identitas.

Untuk informasi selengkapnya tentang verifikasi, lihat Memverifikasi Alamat Email dan Domain di Amazon SES di Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat ListIdentitiesdi Referensi AWS CLI Perintah.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ses.SesClient;
import software.amazon.awssdk.services.ses.model.ListIdentitiesResponse;
import software.amazon.awssdk.services.ses.model.SesException;
import java.io.IOException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListIdentities {
```

```
    public static void main(String[] args) throws IOException {
        Region region = Region.US_WEST_2;
        SesClient client = SesClient.builder()
                .region(region)
                .build();

        listSESIdentities(client);
    }

    public static void listSESIdentities(SesClient client) {
        try {
            ListIdentitiesResponse identitiesResponse = client.listIdentities();
            List<String> identities = identitiesResponse.identities();
            for (String identity : identities) {
                System.out.println("The identity is " + identity);
            }

        } catch (SesException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Untuk API detailnya, lihat ListIdentitiesdi AWS SDK for Java 2.x APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { ListIdentitiesCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";

const createListIdentitiesCommand = () =>
  new ListIdentitiesCommand({ IdentityType: "EmailAddress", MaxItems: 10 });
```

```
const run = async () => {
  const listIdentitiesCommand = createListIdentitiesCommand();

  try {
    return await sesClient.send(listIdentitiesCommand);
  } catch (err) {
    console.log("Failed to list identities.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat ListIdentitiesdi AWS SDK for JavaScript APIReferensi.

PowerShell

Alat untuk PowerShell

Contoh 1: Perintah ini mengembalikan daftar yang berisi semua identitas (alamat email dan domain) untuk AWS Akun tertentu, terlepas dari status verifikasi.

```
Get-SESIdentity
```

- Untuk API detailnya, lihat ListIdentitiesdi AWS Tools for PowerShell Referensi Cmdlet.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesIdentity:
    """Encapsulates Amazon SES identity functions."""

    def __init__(self, ses_client):
```

```
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client


    def list_identities(self, identity_type, max_items):
        """
        Gets the identities of the specified type for the current account.

        :param identity_type: The type of identity to retrieve, such as
EmailAddress.
        :param max_items: The maximum number of identities to retrieve.
        :return: The list of retrieved identities.
        """
        try:
            response = self.ses_client.list_identities(
                IdentityType=identity_type, MaxItems=max_items
            )
            identities = response["Identities"]
            logger.info("Got %s identities for the current account.",
len(identities))
        except ClientError:
            logger.exception("Couldn't list identities for the current account.")
            raise
        else:
            return identities
```

- Untuk API detailnya, lihat ListIdentities AWSSDKReferensi Python (Boto3). API

Ruby

SDKuntuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-ses' # v2: require 'aws-sdk'

# Create client in us-west-2 region
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
client = Aws::SES::Client.new(region: 'us-west-2')

# Get up to 1000 identities
ids = client.list_identities({
                               identity_type: 'EmailAddress'
                             })

ids.identities.each do |email|
  attrs = client.get_identity_verification_attributes({
                                                        identities: [email]
                                                      })

  status = attrs.verification_attributes[email].verification_status

  # Display email addresses that have been verified
  puts email if status == 'Success'
end
```

- Untuk API detailnya, lihat ListIdentitiesdi AWS SDK for Ruby APIReferensi.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`ListReceiptFilters`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListReceiptFilters`.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! List the receipt filters associated with this account.
/*!
  \param filters; A vector of "ReceiptFilter" to receive the retrieved filters.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool
AwsDoc::SES::listReceiptFilters(Aws::Vector<Aws::SES::Model::ReceiptFilter>
 &filters,
                                const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);
    Aws::SES::Model::ListReceiptFiltersRequest listReceiptFiltersRequest;

    Aws::SES::Model::ListReceiptFiltersOutcome outcome =
 sesClient.ListReceiptFilters(
            listReceiptFiltersRequest);
    if (outcome.IsSuccess()) {
        auto &retrievedFilters = outcome.GetResult().GetFilters();
        if (!retrievedFilters.empty()) {
            filters.insert(filters.cend(), retrievedFilters.cbegin(),
                           retrievedFilters.cend());
        }
    }
    else {
        std::cerr << "Error retrieving IP address filters: "
                  << outcome.GetError().GetMessage() << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat ListReceiptFiltersdi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { ListReceiptFiltersCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";

const createListReceiptFiltersCommand = () => new ListReceiptFiltersCommand({});

const run = async () => {
  const listReceiptFiltersCommand = createListReceiptFiltersCommand();

  return await sesClient.send(listReceiptFiltersCommand);
};
```

- Untuk API detailnya, lihat ListReceiptFiltersdi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesReceiptHandler:
    """Encapsulates Amazon SES receipt handling functions."""
```

```
    def __init__(self, ses_client, s3_resource):
        """
        :param ses_client: A Boto3 Amazon SES client.
        :param s3_resource: A Boto3 Amazon S3 resource.
        """
        self.ses_client = ses_client
        self.s3_resource = s3_resource


    def list_receipt_filters(self):
        """
        Gets the list of receipt filters for the current account.

        :return: The list of receipt filters.
        """
        try:
            response = self.ses_client.list_receipt_filters()
            filters = response["Filters"]
            logger.info("Got %s receipt filters.", len(filters))
        except ClientError:
            logger.exception("Couldn't get receipt filters.")
            raise
        else:
            return filters
```

- Untuk API detailnya, lihat ListReceiptFilters AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`ListTemplates`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListTemplates`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// List email templates for the current account.
    /// </summary>
    /// <returns>A list of template metadata.</returns>
    public async Task<List<TemplateMetadata>> ListEmailTemplatesAsync()
    {
        var result = new List<TemplateMetadata>();
        try
        {
            var response = await _amazonSimpleEmailService.ListTemplatesAsync(
                new ListTemplatesRequest());
            result = response.TemplatesMetadata;
        }
        catch (Exception ex)
        {
            Console.WriteLine("ListEmailTemplatesAsync failed with exception: " +
ex.Message);
        }

        return result;
    }
```

- Untuk API detailnya, lihat ListTemplatesdi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.sesv2.SesV2Client;
import software.amazon.awssdk.services.sesv2.model.ListEmailTemplatesRequest;
import software.amazon.awssdk.services.sesv2.model.ListEmailTemplatesResponse;
import software.amazon.awssdk.services.sesv2.model.SesV2Exception;

public class ListTemplates {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        SesV2Client sesv2Client = SesV2Client.builder()
                .region(region)
                .build();

        listAllTemplates(sesv2Client);
    }

    public static void listAllTemplates(SesV2Client sesv2Client) {
        try {
            ListEmailTemplatesRequest templatesRequest =
 ListEmailTemplatesRequest.builder()
                    .pageSize(1)
                    .build();

            ListEmailTemplatesResponse response =
 sesv2Client.listEmailTemplates(templatesRequest);
            response.templatesMetadata()
                    .forEach(template -> System.out.println("Template name: " +
 template.templateName()));

        } catch (SesV2Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
    }
}
```

- Untuk API detailnya, lihat ListTemplatesdi AWS SDK for Java 2.x APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { ListTemplatesCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";

const createListTemplatesCommand = (maxItems) =>
  new ListTemplatesCommand({ MaxItems: maxItems });

const run = async () => {
  const listTemplatesCommand = createListTemplatesCommand(10);

  try {
    return await sesClient.send(listTemplatesCommand);
  } catch (err) {
    console.log("Failed to list templates.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat ListTemplatesdi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesTemplate:
    """Encapsulates Amazon SES template functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client
        self.template = None
        self.template_tags = set()

    def _extract_tags(self, subject, text, html):
        """
        Extracts tags from a template as a set of unique values.

        :param subject: The subject of the email.
        :param text: The text version of the email.
        :param html: The html version of the email.
        """
        self.template_tags = set(re.findall(TEMPLATE_REGEX, subject + text +
 html))
        logger.info("Extracted template tags: %s", self.template_tags)


    def list_templates(self):
        """
        Gets a list of all email templates for the current account.

        :return: The list of retrieved email templates.
        """
        try:
            response = self.ses_client.list_templates()
```

```
            templates = response["TemplatesMetadata"]
            logger.info("Got %s templates.", len(templates))
        except ClientError:
            logger.exception("Couldn't get templates.")
            raise
        else:
            return templates
```

- Untuk API detailnya, lihat ListTemplates AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`SendBulkTemplatedEmail`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`SendBulkTemplatedEmail`.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { SendBulkTemplatedEmailCommand } from "@aws-sdk/client-ses";
import {
  getUniqueName,
  postfix,
} from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { sesClient } from "./libs/sesClient.js";

/**
 * Replace this with the name of an existing template.
 */
const TEMPLATE_NAME = getUniqueName("ReminderTemplate");
```

```
/**
 * Replace these with existing verified emails.
 */
const VERIFIED_EMAIL_1 = postfix(getUniqueName("Bilbo"), "@example.com");
const VERIFIED_EMAIL_2 = postfix(getUniqueName("Frodo"), "@example.com");

const USERS = [
  { firstName: "Bilbo", emailAddress: VERIFIED_EMAIL_1 },
  { firstName: "Frodo", emailAddress: VERIFIED_EMAIL_2 },
];

/**
 *
 * @param { { emailAddress: string, firstName: string }[] } users
 * @param { string } templateName the name of an existing template in SES
 * @returns { SendBulkTemplatedEmailCommand }
 */
const createBulkReminderEmailCommand = (users, templateName) => {
  return new SendBulkTemplatedEmailCommand({
    /**
     * Each 'Destination' uses a corresponding set of replacement data. We can
 map each user
     * to a 'Destination' and provide user specific replacement data to create
 personalized emails.
     *
     * Here's an example of how a template would be replaced with user data:
     * Template: <h1>Hello {{name}},</h1><p>Don't forget about the party gifts!</
p>
     * Destination 1: <h1>Hello Bilbo,</h1><p>Don't forget about the party gifts!
</p>
     * Destination 2: <h1>Hello Frodo,</h1><p>Don't forget about the party gifts!
</p>
     */
    Destinations: users.map((user) => ({
      Destination: { ToAddresses: [user.emailAddress] },
      ReplacementTemplateData: JSON.stringify({ name: user.firstName }),
    })),
    DefaultTemplateData: JSON.stringify({ name: "Shireling" }),
    Source: VERIFIED_EMAIL_1,
    Template: templateName,
  });
};
```

```
const run = async () => {
  const sendBulkTemplateEmailCommand = createBulkReminderEmailCommand(
    USERS,
    TEMPLATE_NAME,
  );
  try {
    return await sesClient.send(sendBulkTemplateEmailCommand);
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MessageRejected") {
      /** @type { import('@aws-sdk/client-ses').MessageRejected} */
      const messageRejectedError = caught;
      return messageRejectedError;
    }
    throw caught;
  }
};
```

- Untuk API detailnya, lihat SendBulkTemplatedEmaildi AWS SDK for JavaScript APIReferensi.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`SendEmail`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`SendEmail`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <summary>
///  Send an email by using Amazon SES.
/// </summary>
/// <param name="toAddresses">List of recipients.</param>
/// <param name="ccAddresses">List of cc recipients.</param>
/// <param name="bccAddresses">List of bcc recipients.</param>
/// <param name="bodyHtml">Body of the email in HTML.</param>
/// <param name="bodyText">Body of the email in plain text.</param>
/// <param name="subject">Subject line of the email.</param>
/// <param name="senderAddress">From address.</param>
/// <returns>The messageId of the email.</returns>
public async Task<string> SendEmailAsync(List<string> toAddresses,
    List<string> ccAddresses, List<string> bccAddresses,
    string bodyHtml, string bodyText, string subject, string senderAddress)
{
    var messageId = "";
    try
    {
        var response = await _amazonSimpleEmailService.SendEmailAsync(
            new SendEmailRequest
            {
                Destination = new Destination
                {
                    BccAddresses = bccAddresses,
                    CcAddresses = ccAddresses,
                    ToAddresses = toAddresses
                },
                Message = new Message
                {
                    Body = new Body
                    {
```

```
                            Html = new Content
                            {
                                Charset = "UTF-8",
                                Data = bodyHtml
                            },
                            Text = new Content
                            {
                                Charset = "UTF-8",
                                Data = bodyText
                            }
                        },
                        Subject = new Content
                        {
                            Charset = "UTF-8",
                            Data = subject
                        }
                    },
                    Source = senderAddress
                });
            messageId = response.MessageId;
        }
        catch (Exception ex)
        {
            Console.WriteLine("SendEmailAsync failed with exception: " +
ex.Message);
        }

        return messageId;
    }
```

- Untuk API detailnya, lihat SendEmaildi AWS SDK for .NET APIReferensi.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Send an email to a list of recipients.
/*!
  \param recipients; Vector of recipient email addresses.
  \param subject: Email subject.
  \param htmlBody: Email body as HTML. At least one body data is required.
  \param textBody: Email body as plain text. At least one body data is required.
  \param senderEmailAddress: Email address of sender. Ignored if empty string.
  \param ccAddresses: Vector of cc addresses. Ignored if empty.
  \param replyToAddress: Reply to email address. Ignored if empty string.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::sendEmail(const Aws::Vector<Aws::String> &recipients,
                            const Aws::String &subject,
                            const Aws::String &htmlBody,
                            const Aws::String &textBody,
                            const Aws::String &senderEmailAddress,
                            const Aws::Vector<Aws::String> &ccAddresses,
                            const Aws::String &replyToAddress,
                            const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::Destination destination;
    if (!ccAddresses.empty()) {
        destination.WithCcAddresses(ccAddresses);
    }
    if (!recipients.empty()) {
        destination.WithToAddresses(recipients);
    }

    Aws::SES::Model::Body message_body;
    if (!htmlBody.empty()) {
        message_body.SetHtml(

 Aws::SES::Model::Content().WithCharset("UTF-8").WithData(htmlBody));
    }

    if (!textBody.empty()) {
        message_body.SetText(

 Aws::SES::Model::Content().WithCharset("UTF-8").WithData(textBody));
    }
```

```
    Aws::SES::Model::Message message;
    message.SetBody(message_body);
    message.SetSubject(
            Aws::SES::Model::Content().WithCharset("UTF-8").WithData(subject));

    Aws::SES::Model::SendEmailRequest sendEmailRequest;
    sendEmailRequest.SetDestination(destination);
    sendEmailRequest.SetMessage(message);
    if (!senderEmailAddress.empty()) {
        sendEmailRequest.SetSource(senderEmailAddress);
    }
    if (!replyToAddress.empty()) {
        sendEmailRequest.AddReplyToAddresses(replyToAddress);
    }

    auto outcome = sesClient.SendEmail(sendEmailRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully sent message with ID "
                  << outcome.GetResult().GetMessageId()
                  << "." << std::endl;
    }
    else {
        std::cerr << "Error sending message. " << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat SendEmaildi AWS SDK for C++ APIReferensi.

CLI

AWS CLI

Untuk mengirim email yang diformat menggunakan Amazon SES

Contoh berikut menggunakan `send-email` perintah untuk mengirim email yang diformat:

```
aws ses send-email --from sender@example.com --destination file://
destination.json --message file://message.json
```

Output:

```
{
   "MessageId": "EXAMPLEf3a5efcd1-51adec81-d2a4-4e3f-9fe2-5d85c1b23783-000000"
}
```

Tujuan dan pesan adalah struktur JSON data yang disimpan dalam file.json di direktori saat ini. File-file ini adalah sebagai berikut:

`destination.json`:

```
{
  "ToAddresses":  ["recipient1@example.com", "recipient2@example.com"],
  "CcAddresses":  ["recipient3@example.com"],
  "BccAddresses": []
}
```

`message.json`:

```
{
   "Subject": {
       "Data": "Test email sent using the AWS CLI",
       "Charset": "UTF-8"
   },
   "Body": {
       "Text": {
           "Data": "This is the message body in text format.",
           "Charset": "UTF-8"
       },
       "Html": {
           "Data": "This message body contains HTML formatting. It can, for
 example, contain links like this one: <a class=\"ulink\" href=\"http://
docs.aws.amazon.com/ses/latest/DeveloperGuide\" target=\"_blank\">Amazon SES
 Developer Guide</a>.",
           "Charset": "UTF-8"
       }
   }
}
```

Ganti alamat email pengirim dan penerima dengan yang ingin Anda gunakan. Perhatikan bahwa alamat email pengirim harus diverifikasi dengan AmazonSES. Sampai Anda diberikan akses produksi ke AmazonSES, Anda juga harus memverifikasi alamat email setiap penerima kecuali penerima adalah simulator SES kotak surat Amazon. Untuk informasi selengkapnya tentang verifikasi, lihat Memverifikasi Alamat Email dan Domain di Amazon SES di Panduan Pengembang Layanan Email Sederhana Amazon.

ID Pesan dalam output menunjukkan bahwa panggilan ke kirim email berhasil.

Jika Anda tidak menerima email, centang kotak Sampah Anda.

Untuk informasi selengkapnya tentang mengirim email yang diformat, lihat Mengirim Email Berformat Menggunakan Amazon SES API di Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat SendEmaildi Referensi AWS CLI Perintah.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ses.SesClient;
import software.amazon.awssdk.services.ses.model.Content;
import software.amazon.awssdk.services.ses.model.Destination;
import software.amazon.awssdk.services.ses.model.Message;
import software.amazon.awssdk.services.ses.model.Body;
import software.amazon.awssdk.services.ses.model.SendEmailRequest;
import software.amazon.awssdk.services.ses.model.SesException;

import javax.mail.MessagingException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessageEmailRequest {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <sender> <recipient> <subject>\s

                Where:
                    sender - An email address that represents the sender.\s
                    recipient -  An email address that represents the recipient.
\s
                    subject - The  subject line.\s
                """;

        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 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, bodyHTML);
            client.close();
            System.out.println("Done");
```

```
        } catch (MessagingException e) {
            e.getStackTrace();
        }
    }

    public static void send(SesClient client,
            String sender,
            String recipient,
            String subject,
            String bodyHTML) throws MessagingException {

        Destination destination = Destination.builder()
                .toAddresses(recipient)
                .build();

        Content content = Content.builder()
                .data(bodyHTML)
                .build();

        Content sub = Content.builder()
                .data(subject)
                .build();

        Body body = Body.builder()
                .html(content)
                .build();

        Message msg = Message.builder()
                .subject(sub)
                .body(body)
                .build();

        SendEmailRequest emailRequest = SendEmailRequest.builder()
                .destination(destination)
                .message(msg)
                .source(sender)
                .build();

        try {
            System.out.println("Attempting to send an email through Amazon SES "
+ "using the AWS SDK for Java...");
            client.sendEmail(emailRequest);

        } catch (SesException e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.ses.SesClient;
import javax.activation.DataHandler;
import javax.activation.DataSource;
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 javax.mail.util.ByteArrayDataSource;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.file.Files;
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;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class SendMessageAttachment {
    public static void main(String[] args) throws IOException {
        final String usage = """

                Usage:
```

```
                    <sender> <recipient> <subject> <fileLocation>\s

                Where:
                    sender - An email address that represents the sender.\s
                    recipient -  An email address that represents the recipient.
\s
                    subject - The  subject line.\s
                    fileLocation - The location of a Microsoft Excel file to use
 as an attachment (C:/AWS/customers.xls).\s
                """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String sender = args[0];
        String recipient = args[1];
        String subject = args[2];
        String fileLocation = args[3];

        // The email body for recipients with non-HTML email clients.
        String bodyText = "Hello,\r\n" + "Please see the attached file for a list
 "
                + "of customers to contact.";

        // The HTML body of the email.
        String bodyHTML = "<html>" + "<head></head>" + "<body>" + "<h1>Hello!</
h1>"
                + "<p>Please see the attached file for a " + "list of customers
 to contact.</p>" + "</body>"
                + "</html>";

        Region region = Region.US_WEST_2;
        SesClient client = SesClient.builder()
                .region(region)
                .build();

        try {
            sendemailAttachment(client, sender, recipient, subject, bodyText,
 bodyHTML, fileLocation);
            client.close();
            System.out.println("Done");
```

```
        } catch (IOException | MessagingException e) {
            e.getStackTrace();
        }
    }

    public static void sendemailAttachment(SesClient client,
            String sender,
            String recipient,
            String subject,
            String bodyText,
            String bodyHTML,
            String fileLocation) throws AddressException, MessagingException,
IOException {

        java.io.File theFile = new java.io.File(fileLocation);
        byte[] fileContent = Files.readAllBytes(theFile.toPath());

        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 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);
        msg.addBodyPart(wrap);

        // Define the attachment.
        MimeBodyPart att = new MimeBodyPart();
        DataSource fds = new ByteArrayDataSource(fileContent,
                "application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet");
        att.setDataHandler(new DataHandler(fds));

        String reportName = "WorkReport.xls";
        att.setFileName(reportName);

        // Add the attachment to the message.
        msg.addBodyPart(att);

        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);
        }
        System.out.println("Email sent using SesClient with attachment");
    }
}
```

- Untuk API detailnya, lihat SendEmaildi AWS SDK for Java 2.x APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { SendEmailCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";

const createSendEmailCommand = (toAddress, fromAddress) => {
  return new SendEmailCommand({
    Destination: {
      /* required */
      CcAddresses: [
        /* more items */
      ],
      ToAddresses: [
        toAddress,
        /* more To-email addresses */
      ],
    },
    Message: {
      /* required */
```

```
      Body: {
        /* required */
        Html: {
          Charset: "UTF-8",
          Data: "HTML_FORMAT_BODY",
        },
        Text: {
          Charset: "UTF-8",
          Data: "TEXT_FORMAT_BODY",
        },
      },
      Subject: {
        Charset: "UTF-8",
        Data: "EMAIL_SUBJECT",
      },
    },
    Source: fromAddress,
    ReplyToAddresses: [
      /* more items */
    ],
  });
};

const run = async () => {
  const sendEmailCommand = createSendEmailCommand(
    "recipient@example.com",
    "sender@example.com",
  );

  try {
    return await sesClient.send(sendEmailCommand);
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MessageRejected") {
      /** @type { import('@aws-sdk/client-ses').MessageRejected} */
      const messageRejectedError = caught;
      return messageRejectedError;
    }
    throw caught;
  }
};
```

- Untuk API detailnya, lihat SendEmaildi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesMailSender:
    """Encapsulates functions to send emails with Amazon SES."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client


    def send_email(self, source, destination, subject, text, html,
 reply_tos=None):
        """
        Sends an email.

        Note: If your account is in the Amazon SES  sandbox, the source and
        destination email accounts must both be verified.

        :param source: The source email account.
        :param destination: The destination email account.
        :param subject: The subject of the email.
        :param text: The plain text version of the body of the email.
        :param html: The HTML version of the body of the email.
        :param reply_tos: Email accounts that will receive a reply if the
 recipient
                          replies to the message.
        :return: The ID of the message, assigned by Amazon SES.
        """
        send_args = {
            "Source": source,
            "Destination": destination.to_service_format(),
            "Message": {
```

```
                "Subject": {"Data": subject},
                "Body": {"Text": {"Data": text}, "Html": {"Data": html}},
            },
        }
        if reply_tos is not None:
            send_args["ReplyToAddresses"] = reply_tos
        try:
            response = self.ses_client.send_email(**send_args)
            message_id = response["MessageId"]
            logger.info(
                "Sent mail %s from %s to %s.", message_id, source,
 destination.tos
            )
        except ClientError:
            logger.exception(
                "Couldn't send mail from %s to %s.", source, destination.tos
            )
            raise
        else:
            return message_id
```

- Untuk API detailnya, lihat SendEmail AWSSDKReferensi Python (Boto3). API

Ruby

SDKuntuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-ses' # v2: 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. To use a configuration
# set, uncomment the next line and line 74.
#   configsetname = "ConfigSet"

# 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 client in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: 'us-west-2')

# Try to send the email.
begin
  # Provide the contents of the email.
  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
    # Uncomment the following line to use a configuration set.
    # configuration_set_name: configsetname,
  )

  puts "Email sent to #{recipient}"

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => e
  puts "Email not sent. Error message: #{e}"
end
```

- Untuk API detailnya, lihat SendEmaildi AWS SDK for Ruby APIReferensi.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`SendRawEmail`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`SendRawEmail`.

CLI

AWS CLI

Untuk mengirim email mentah menggunakan Amazon SES

Contoh berikut menggunakan `send-raw-email` perintah untuk mengirim email dengan TXT lampiran:

```
aws ses send-raw-email --raw-message file://message.json
```

Output:

```
{
   "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"
}
```

Pesan mentah adalah struktur JSON data yang disimpan dalam file bernama `message.json` di direktori saat ini. Ini berisi yang berikut:

```
{
   "Data": "From: sender@example.com\nTo: recipient@example.com\nSubject:
 Test email sent using the AWS CLI (contains an attachment)\nMIME-Version:
 1.0\nContent-type: Multipart/Mixed; boundary=\"NextPart\"\n\n--NextPart
\nContent-Type: text/plain\n\nThis is the message body.\n\n--NextPart\nContent-
Type: text/plain;\nContent-Disposition: attachment; filename=\"attachment.txt\"\n
\nThis is the text in the attachment.\n\n--NextPart--"
}
```

Seperti yang Anda lihat, “Data” adalah satu string panjang yang berisi seluruh konten email mentah dalam MIME format, termasuk lampiran yang disebut attachment.txt.

Ganti sender@example.com dan recipient@example.com dengan alamat yang ingin Anda gunakan. Perhatikan bahwa alamat email pengirim harus diverifikasi dengan AmazonSES. Sampai Anda diberikan akses produksi ke AmazonSES, Anda juga harus memverifikasi alamat email penerima kecuali penerima adalah simulator SES kotak surat Amazon. Untuk informasi selengkapnya tentang verifikasi, lihat Memverifikasi Alamat Email dan Domain di Amazon SES di Panduan Pengembang Layanan Email Sederhana Amazon.

ID Pesan dalam output menunjukkan bahwa panggilan ke send-raw-email berhasil.

Jika Anda tidak menerima email, centang kotak Sampah Anda.

Untuk informasi selengkapnya tentang mengirim email mentah, lihat Mengirim Email Mentah Menggunakan Amazon SES API di Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat SendRawEmaildi Referensi AWS CLI Perintah.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Gunakan nodemailer untuk mengirim email dengan lampiran.

```
import sesClientModule from "@aws-sdk/client-ses";
/**
 * nodemailer wraps the SES SDK and calls SendRawEmail. Use this for more
 advanced
 * functionality like adding attachments to your email.
 *
 * https://nodemailer.com/transports/ses/
 */
import nodemailer from "nodemailer";

/**
 * @param {string} from An Amazon SES verified email address.
 * @param {*} to An Amazon SES verified email address.
 */
export const sendEmailWithAttachments = (
  from = "from@example.com",
  to = "to@example.com",
) => {
  const ses = new sesClientModule.SESClient({});
  const transporter = nodemailer.createTransport({
    SES: { ses, aws: sesClientModule },
  });

  return new Promise((resolve, reject) => {
    transporter.sendMail(
      {
        from,
        to,
        subject: "Hello World",
        text: "Greetings from Amazon SES!",
        attachments: [{ content: "Hello World!", filename: "hello.txt" }],
```

```
      },
      (err, info) => {
        if (err) {
          reject(err);
        } else {
          resolve(info);
        }
      },
    );
  });
};
```

- Untuk API detailnya, lihat SendRawEmaildi AWS SDK for JavaScript APIReferensi.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`SendTemplatedEmail`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`SendTemplatedEmail`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <summary>
```

```
    /// Send an email using a template.
    /// </summary>
    /// <param name="sender">Address of the sender.</param>
    /// <param name="recipients">Addresses of the recipients.</param>
    /// <param name="templateName">Name of the email template.</param>
    /// <param name="templateDataObject">Data for the email template.</param>
    /// <returns>The messageId of the email.</returns>
    public async Task<string> SendTemplateEmailAsync(string sender, List<string>
recipients,
        string templateName, object templateDataObject)
    {
        var messageId = "";
        try
        {
            // Template data should be serialized JSON from either a class or a
dynamic object.
            var templateData = JsonSerializer.Serialize(templateDataObject);

            var response = await
_amazonSimpleEmailService.SendTemplatedEmailAsync(
                new SendTemplatedEmailRequest
                {
                    Source = sender,
                    Destination = new Destination
                    {
                        ToAddresses = recipients
                    },
                    Template = templateName,
                    TemplateData = templateData
                });
            messageId = response.MessageId;
        }
        catch (Exception ex)
        {
            Console.WriteLine("SendTemplateEmailAsync failed with exception: " +
ex.Message);
        }

        return messageId;
    }
```

- Untuk API detailnya, lihat SendTemplatedEmaildi AWS SDK for .NET APIReferensi.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Send a templated email to a list of recipients.
/*!
  \param recipients; Vector of recipient email addresses.
  \param templateName: The name of the template to use.
  \param templateData: Map of key-value pairs for replacing text in template.
  \param senderEmailAddress: Email address of sender. Ignored if empty string.
  \param ccAddresses: Vector of cc addresses. Ignored if empty.
  \param replyToAddress: Reply to email address. Ignored if empty string.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::sendTemplatedEmail(const Aws::Vector<Aws::String> &recipients,
                                     const Aws::String &templateName,
                                     const Aws::Map<Aws::String, Aws::String>
 &templateData,
                                     const Aws::String &senderEmailAddress,
                                     const Aws::Vector<Aws::String> &ccAddresses,
                                     const Aws::String &replyToAddress,
                                     const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::Destination destination;
    if (!ccAddresses.empty()) {
        destination.WithCcAddresses(ccAddresses);
    }
    if (!recipients.empty()) {
        destination.WithToAddresses(recipients);
    }

    Aws::SES::Model::SendTemplatedEmailRequest sendTemplatedEmailRequest;
    sendTemplatedEmailRequest.SetDestination(destination);
```

```
    sendTemplatedEmailRequest.SetTemplate(templateName);

    std::ostringstream templateDataStream;
    templateDataStream << "{";
    size_t dataCount = 0;
    for (auto &pair: templateData) {
        templateDataStream << "\"" << pair.first << "\":\"" << pair.second <<
 "\"";
        dataCount++;
        if (dataCount < templateData.size()) {
            templateDataStream << ",";
        }
    }
    templateDataStream << "}";

    sendTemplatedEmailRequest.SetTemplateData(templateDataStream.str());

    if (!senderEmailAddress.empty()) {
        sendTemplatedEmailRequest.SetSource(senderEmailAddress);
    }
    if (!replyToAddress.empty()) {
        sendTemplatedEmailRequest.AddReplyToAddresses(replyToAddress);
    }

    auto outcome = sesClient.SendTemplatedEmail(sendTemplatedEmailRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully sent templated message with ID "
                  << outcome.GetResult().GetMessageId()
                  << "." << std::endl;
    }
    else {
        std::cerr << "Error sending templated message. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat SendTemplatedEmaildi AWS SDK for C++ APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.sesv2.model.Destination;
import software.amazon.awssdk.services.sesv2.model.EmailContent;
import software.amazon.awssdk.services.sesv2.model.SendEmailRequest;
import software.amazon.awssdk.services.sesv2.model.SesV2Exception;
import software.amazon.awssdk.services.sesv2.SesV2Client;
import software.amazon.awssdk.services.sesv2.model.Template;

/**
 * Before running this AWS SDK for Java (v2) example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * Also, make sure that you create a template. See the following documentation
 * topic:
 *
 * https://docs.aws.amazon.com/ses/latest/dg/send-personalized-email-api.html
 */

public class SendEmailTemplate {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <template> <sender> <recipient>\s

                Where:
                    template - The name of the email template.
```

```
                    sender - An email address that represents the sender.\s
                    recipient - An email address that represents the recipient.\s
                """";

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String templateName = args[0];
        String sender = args[1];
        String recipient = args[2];
        Region region = Region.US_EAST_1;
        SesV2Client sesv2Client = SesV2Client.builder()
                .region(region)
                .build();

        send(sesv2Client, sender, recipient, templateName);
    }

    public static void send(SesV2Client client, String sender, String recipient,
String templateName) {
        Destination destination = Destination.builder()
                .toAddresses(recipient)
                .build();

        /*
         * Specify both name and favorite animal (favoriteanimal) in your code
when
         * defining the Template object.
         * If you don't specify all the variables in the template, Amazon SES
doesn't
         * send the email.
         */
        Template myTemplate = Template.builder()
                .templateName(templateName)
                .templateData("{\n" +
                        "  \"name\": \"Jason\"\n," +
                        "  \"favoriteanimal\": \"Cat\"\n" +
                        "}")
                .build();

        EmailContent emailContent = EmailContent.builder()
                .template(myTemplate)
```

```
                .build();

        SendEmailRequest emailRequest = SendEmailRequest.builder()
                .destination(destination)
                .content(emailContent)
                .fromEmailAddress(sender)
                .build();

        try {
            System.out.println("Attempting to send an email based on a template
 using the AWS SDK for Java (v2)...");
            client.sendEmail(emailRequest);
            System.out.println("email based on a template was sent");

        } catch (SesV2Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Untuk API detailnya, lihat SendTemplatedEmaildi AWS SDK for Java 2.x APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { SendTemplatedEmailCommand } from "@aws-sdk/client-ses";
import {
  getUniqueName,
  postfix,
} from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { sesClient } from "./libs/sesClient.js";

/**
```

```
 * Replace this with the name of an existing template.
 */
const TEMPLATE_NAME = getUniqueName("ReminderTemplate");

/**
 * Replace these with existing verified emails.
 */
const VERIFIED_EMAIL = postfix(getUniqueName("Bilbo"), "@example.com");

const USER = { firstName: "Bilbo", emailAddress: VERIFIED_EMAIL };

/**
 *
 * @param { { emailAddress: string, firstName: string } } user
 * @param { string } templateName - The name of an existing template in Amazon
 SES.
 * @returns { SendTemplatedEmailCommand }
 */
const createReminderEmailCommand = (user, templateName) => {
  return new SendTemplatedEmailCommand({
    /**
     * Here's an example of how a template would be replaced with user data:
     * Template: <h1>Hello {{contact.firstName}},</h1><p>Don't forget about the
 party gifts!</p>
     * Destination: <h1>Hello Bilbo,</h1><p>Don't forget about the party gifts!</
p>
     */
    Destination: { ToAddresses: [user.emailAddress] },
    TemplateData: JSON.stringify({ contact: { firstName: user.firstName } }),
    Source: VERIFIED_EMAIL,
    Template: templateName,
  });
};

const run = async () => {
  const sendReminderEmailCommand = createReminderEmailCommand(
    USER,
    TEMPLATE_NAME,
  );
  try {
    return await sesClient.send(sendReminderEmailCommand);
  } catch (caught) {
    if (caught instanceof Error && caught.name === "MessageRejected") {
      /** @type { import('@aws-sdk/client-ses').MessageRejected} */
```

```
      const messageRejectedError = caught;
      return messageRejectedError;
    }
    throw caught;
  }
};
```

- Untuk API detailnya, lihat SendTemplatedEmaildi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesMailSender:
    """Encapsulates functions to send emails with Amazon SES."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client


    def send_templated_email(
        self, source, destination, template_name, template_data, reply_tos=None
    ):
        """
        Sends an email based on a template. A template contains replaceable tags
        each enclosed in two curly braces, such as {{name}}. The template data
 passed
        in this function contains key-value pairs that define the values to
 insert
        in place of the template tags.

        Note: If your account is in the Amazon SES  sandbox, the source and
```

```
        destination email accounts must both be verified.

        :param source: The source email account.
        :param destination: The destination email account.
        :param template_name: The name of a previously created template.
        :param template_data: JSON-formatted key-value pairs of replacement
values
                              that are inserted in the template before it is
sent.
        :return: The ID of the message, assigned by Amazon SES.
        """
        send_args = {
            "Source": source,
            "Destination": destination.to_service_format(),
            "Template": template_name,
            "TemplateData": json.dumps(template_data),
        }
        if reply_tos is not None:
            send_args["ReplyToAddresses"] = reply_tos
        try:
            response = self.ses_client.send_templated_email(**send_args)
            message_id = response["MessageId"]
            logger.info(
                "Sent templated mail %s from %s to %s.",
                message_id,
                source,
                destination.tos,
            )
        except ClientError:
            logger.exception(
                "Couldn't send templated mail from %s to %s.", source,
destination.tos
            )
            raise
        else:
            return message_id
```

- Untuk API detailnya, lihat SendTemplatedEmail AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`UpdateTemplate`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`UpdateTemplate`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Verifikasi identitas email dan kirim pesan

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Update an Amazon Simple Email Service (Amazon SES) template.
/*!
  \param templateName: The name of the template.
  \param htmlPart: The HTML body of the email.
  \param subjectPart: The subject line of the email.
  \param textPart: The plain text version of the email.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::updateTemplate(const Aws::String &templateName,
                                 const Aws::String &htmlPart,
                                 const Aws::String &subjectPart,
                                 const Aws::String &textPart,
                                 const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::Template templateValues;
```

```
    templateValues.SetTemplateName(templateName);
    templateValues.SetSubjectPart(subjectPart);
    templateValues.SetHtmlPart(htmlPart);
    templateValues.SetTextPart(textPart);

    Aws::SES::Model::UpdateTemplateRequest updateTemplateRequest;
    updateTemplateRequest.SetTemplate(templateValues);

    Aws::SES::Model::UpdateTemplateOutcome outcome =
 sesClient.UpdateTemplate(updateTemplateRequest);

    if (outcome.IsSuccess()) {
        std::cout << "Successfully updated template." << std::endl;
    } else {
        std::cerr << "Error updating template. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat UpdateTemplatedi AWS SDK for C++ APIReferensi.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { UpdateTemplateCommand } from "@aws-sdk/client-ses";
import { getUniqueName } from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { sesClient } from "./libs/sesClient.js";

const TEMPLATE_NAME = getUniqueName("TemplateName");
const HTML_PART = "<h1>Hello, World!</h1>";
```

```
const createUpdateTemplateCommand = () => {
  return new UpdateTemplateCommand({
    Template: {
      TemplateName: TEMPLATE_NAME,
      HtmlPart: HTML_PART,
      SubjectPart: "Example",
      TextPart: "Updated template text.",
    },
  });
};

const run = async () => {
  const updateTemplateCommand = createUpdateTemplateCommand();

  try {
    return await sesClient.send(updateTemplateCommand);
  } catch (err) {
    console.log("Failed to update template.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat UpdateTemplatedi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesTemplate:
    """Encapsulates Amazon SES template functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
```

```
        """
        self.ses_client = ses_client
        self.template = None
        self.template_tags = set()

    def _extract_tags(self, subject, text, html):
        """
        Extracts tags from a template as a set of unique values.

        :param subject: The subject of the email.
        :param text: The text version of the email.
        :param html: The html version of the email.
        """
        self.template_tags = set(re.findall(TEMPLATE_REGEX, subject + text +
html))
        logger.info("Extracted template tags: %s", self.template_tags)


    def update_template(self, name, subject, text, html):
        """
        Updates a previously created email template.

        :param name: The name of the template.
        :param subject: The subject of the email.
        :param text: The plain text version of the email.
        :param html: The HTML version of the email.
        """
        try:
            template = {
                "TemplateName": name,
                "SubjectPart": subject,
                "TextPart": text,
                "HtmlPart": html,
            }
            self.ses_client.update_template(Template=template)
            logger.info("Updated template %s.", name)
            self.template = template
            self._extract_tags(subject, text, html)
        except ClientError:
            logger.exception("Couldn't update template %s.", name)
            raise
```

- Untuk API detailnya, lihat UpdateTemplate AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`VerifyDomainIdentity`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`VerifyDomainIdentity`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Salin identitas email dan domain di seluruh Wilayah
- Verifikasi identitas email dan kirim pesan

CLI

AWS CLI

Untuk memverifikasi domain dengan Amazon SES

Contoh berikut menggunakan `verify-domain-identity` perintah untuk memverifikasi domain:

```
aws ses verify-domain-identity --domain example.com
```

Output:

```
{
   "VerificationToken": "eoEmxw+YaYhb3h3iVJHuXMJXqeu1q1/wwmvjuEXAMPLE"
}
```

Untuk menyelesaikan verifikasi domain, Anda harus menambahkan TXT catatan dengan token verifikasi yang dikembalikan ke DNS pengaturan domain Anda. Untuk informasi selengkapnya, lihat Memverifikasi Domain di Amazon SES dalam Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat VerifyDomainIdentitydi Referensi AWS CLI Perintah.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import { VerifyDomainIdentityCommand } from "@aws-sdk/client-ses";
import {
  getUniqueName,
  postfix,
} from "@aws-doc-sdk-examples/lib/utils/util-string.js";
import { sesClient } from "./libs/sesClient.js";

/**
 * You must have access to the domain's DNS settings to complete the
 * domain verification process.
 */
const DOMAIN_NAME = postfix(getUniqueName("Domain"), ".example.com");

const createVerifyDomainIdentityCommand = () => {
  return new VerifyDomainIdentityCommand({ Domain: DOMAIN_NAME });
};

const run = async () => {
  const VerifyDomainIdentityCommand = createVerifyDomainIdentityCommand();

  try {
    return await sesClient.send(VerifyDomainIdentityCommand);
  } catch (err) {
    console.log("Failed to verify domain.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat VerifyDomainIdentitydi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesIdentity:
    """Encapsulates Amazon SES identity functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client


    def verify_domain_identity(self, domain_name):
        """
        Starts verification of a domain identity. To complete verification, you
 must
        create a TXT record with a specific format through your DNS provider.

        For more information, see *Verifying a domain with Amazon SES* in the
        Amazon SES documentation:
            https://docs.aws.amazon.com/ses/latest/DeveloperGuide/verify-domain-
procedure.html

        :param domain_name: The name of the domain to verify.
        :return: The token to include in the TXT record with your DNS provider.
        """
        try:
            response = self.ses_client.verify_domain_identity(Domain=domain_name)
            token = response["VerificationToken"]
            logger.info("Got domain verification token for %s.", domain_name)
        except ClientError:
            logger.exception("Couldn't verify domain %s.", domain_name)
```

```
            raise
        else:
            return token
```

- Untuk API detailnya, lihat VerifyDomainIdentity AWSSDKReferensi Python (Boto3). API

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`VerifyEmailIdentity`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`VerifyEmailIdentity`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Salin identitas email dan domain di seluruh Wilayah
- Verifikasi identitas email dan kirim pesan

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
/// <summary>
/// Starts verification of an email identity. This request sends an email
/// from Amazon SES to the specified email address. To complete
/// verification, follow the instructions in the email.
/// </summary>
/// <param name="recipientEmailAddress">Email address to verify.</param>
/// <returns>True if successful.</returns>
```

```
    public async Task<bool> VerifyEmailIdentityAsync(string
recipientEmailAddress)
    {
        var success = false;
        try
        {
            var response = await
_amazonSimpleEmailService.VerifyEmailIdentityAsync(
                new VerifyEmailIdentityRequest
                {
                    EmailAddress = recipientEmailAddress
                });

            success = response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (Exception ex)
        {
            Console.WriteLine("VerifyEmailIdentityAsync failed with exception: "
+ ex.Message);
        }

        return success;
    }
```

- Untuk API detailnya, lihat VerifyEmailIdentitydi AWS SDK for .NET APIReferensi.

C++

SDKuntuk C ++

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
//! Add an email address to the list of identities associated with this account
 and
//! initiate verification.
/*!
```

```
  \param emailAddress; The email address to add.
  \param clientConfiguration: AWS client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::SES::verifyEmailIdentity(const Aws::String &emailAddress,
                          const Aws::Client::ClientConfiguration
 &clientConfiguration)
{
    Aws::SES::SESClient sesClient(clientConfiguration);

    Aws::SES::Model::VerifyEmailIdentityRequest verifyEmailIdentityRequest;

    verifyEmailIdentityRequest.SetEmailAddress(emailAddress);

    Aws::SES::Model::VerifyEmailIdentityOutcome outcome =
 sesClient.VerifyEmailIdentity(verifyEmailIdentityRequest);

    if (outcome.IsSuccess())
    {
        std::cout << "Email verification initiated." << std::endl;
    }

    else
    {
        std::cerr << "Error initiating email verification. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Untuk API detailnya, lihat VerifyEmailIdentitydi AWS SDK for C++ APIReferensi.

CLI

AWS CLI

Untuk memverifikasi alamat email dengan Amazon SES

Contoh berikut menggunakan `verify-email-identity` perintah untuk memverifikasi alamat email:

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

Sebelum Anda dapat mengirim email menggunakan AmazonSES, Anda harus memverifikasi alamat atau domain tempat Anda mengirim email untuk membuktikan bahwa Anda memilikinya. Jika Anda belum memiliki akses produksi, Anda juga perlu memverifikasi alamat email apa pun yang Anda kirimi email kecuali alamat email yang disediakan oleh simulator SES kotak surat Amazon.

Setelah verify-email-identity dipanggil, alamat email akan menerima email verifikasi. Pengguna harus mengklik tautan di email untuk menyelesaikan proses verifikasi.

Untuk informasi selengkapnya, lihat Memverifikasi Alamat Email di Amazon SES dalam Panduan Pengembang Layanan Email Sederhana Amazon.

- Untuk API detailnya, lihat VerifyEmailIdentitydi Referensi AWS CLI Perintah.

JavaScript

SDKuntuk JavaScript (v3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
// Import required AWS SDK clients and commands for Node.js
import { VerifyEmailIdentityCommand } from "@aws-sdk/client-ses";
import { sesClient } from "./libs/sesClient.js";

const EMAIL_ADDRESS = "name@example.com";

const createVerifyEmailIdentityCommand = (emailAddress) => {
  return new VerifyEmailIdentityCommand({ EmailAddress: emailAddress });
};

const run = async () => {
  const verifyEmailIdentityCommand =
    createVerifyEmailIdentityCommand(EMAIL_ADDRESS);
  try {
    return await sesClient.send(verifyEmailIdentityCommand);
```

```
  } catch (err) {
    console.log("Failed to verify email identity.", err);
    return err;
  }
};
```

- Untuk API detailnya, lihat VerifyEmailIdentitydi AWS SDK for JavaScript APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
class SesIdentity:
    """Encapsulates Amazon SES identity functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client


    def verify_email_identity(self, email_address):
        """
        Starts verification of an email identity. This function causes an email
        to be sent to the specified email address from Amazon SES. To complete
        verification, follow the instructions in the email.

        :param email_address: The email address to verify.
        """
        try:
            self.ses_client.verify_email_identity(EmailAddress=email_address)
            logger.info("Started verification of %s.", email_address)
        except ClientError:
            logger.exception("Couldn't start verification of %s.", email_address)
```

```
            raise
```

- Untuk API detailnya, lihat VerifyEmailIdentity AWSSDKReferensi Python (Boto3). API

Ruby

SDKuntuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-ses' # v2: require 'aws-sdk'

# Replace recipient@example.com with a "To" address.
recipient = 'recipient@example.com'

# Create a new SES resource in the us-west-2 region.
# Replace us-west-2 with the AWS Region you're using for Amazon SES.
ses = Aws::SES::Client.new(region: 'us-west-2')

# Try to verify email address.
begin
  ses.verify_email_identity({
                              email_address: recipient
                            })

  puts "Email sent to #{recipient}"

# If something goes wrong, display an error message.
rescue Aws::SES::Errors::ServiceError => e
  puts "Email not sent. Error message: #{e}"
end
```

- Untuk API detailnya, lihat VerifyEmailIdentitydi AWS SDK for Ruby APIReferensi.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Skenario untuk Amazon SES menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menerapkan skenario umum di Amazon SES dengan AWS SDKs. Skenario ini menunjukkan kepada Anda cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi di Amazon SES atau digabungkan dengan yang lain Layanan AWS. Setiap skenario menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode.

Skenario menargetkan tingkat pengalaman menengah untuk membantu Anda memahami tindakan layanan dalam konteks.

Contoh

- Membangun aplikasi streaming Amazon Transcribe
- Salin SES email Amazon dan identitas domain dari satu AWS Wilayah ke Wilayah lain menggunakan AWS SDK
- Membuat aplikasi web untuk melacak data DynamoDB
- Buat pelacak item Amazon Redshift
- Buat pelacak butir kerja Aurora Nirserver
- Mendeteksi PPE dalam gambar dengan Amazon Rekognition menggunakan AWS SDK
- Mendeteksi objek dalam gambar dengan Amazon Rekognition menggunakan AWS SDK
- Mendeteksi orang dan objek dalam video dengan Amazon Rekognition menggunakan AWS SDK
- Menghasilkan kredensi untuk terhubung ke titik akhir Amazon SES SMTP
- Menggunakan Step Functions untuk menginvokasi fungsi Lambda
- Verifikasi identitas email dan kirim pesan dengan Amazon SES menggunakan AWS SDK

Membangun aplikasi streaming Amazon Transcribe

Contoh kode berikut menunjukkan cara membuat aplikasi yang merekam, mentranskripsikan, dan menerjemahkan audio langsung secara real-time, dan mengirim email hasilnya.

JavaScript

SDKuntuk JavaScript (v3)

Menunjukkan cara menggunakan Amazon Transcribe untuk membuat aplikasi yang merekam, mentranskripsikan, dan menerjemahkan audio langsung secara real-time, dan mengirim email hasilnya menggunakan Amazon Simple Email Service (Amazon). SES

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Comprehend
- Amazon SES
- Amazon Transcribe
- Amazon Translate

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Salin SES email Amazon dan identitas domain dari satu AWS Wilayah ke Wilayah lain menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menyalin SES email Amazon dan identitas domain dari satu AWS Wilayah ke Wilayah lainnya. Ketika identitas domain dikelola oleh Route 53, catatan verifikasi akan disalin ke domain untuk Wilayah tujuan.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
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()
```

- Untuk API detailnya, lihat topik berikut AWS SDKuntuk Referensi Python (Boto3). API
  - ListIdentities
  - SetIdentityNotificationTopic
  - VerifyDomainDkim
  - VerifyDomainIdentity
  - VerifyEmailIdentity

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Membuat aplikasi web untuk melacak data DynamoDB

Contoh kode berikut menunjukkan cara membuat aplikasi web yang melacak item kerja dalam tabel Amazon DynamoDB dan menggunakan Amazon Simple Email Service (SESAmazon) untuk mengirim laporan.

.NET

AWS SDK for .NET

Menunjukkan cara menggunakan Amazon DynamoDB. NETAPIuntuk membuat aplikasi web dinamis yang melacak data kerja DynamoDB.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- DynamoDB
- Amazon SES

Java

SDKuntuk Java 2.x

Menunjukkan cara menggunakan Amazon API DynamoDB untuk membuat aplikasi web dinamis yang melacak data kerja DynamoDB.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- DynamoDB
- Amazon SES

Kotlin

SDKuntuk Kotlin

Menunjukkan cara menggunakan Amazon API DynamoDB untuk membuat aplikasi web dinamis yang melacak data kerja DynamoDB.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- DynamoDB

- Amazon SES

Python

SDKuntuk Python (Boto3)

Menunjukkan cara menggunakan AWS SDK for Python (Boto3) untuk membuat REST layanan yang melacak item kerja di Amazon DynamoDB dan laporan email dengan menggunakan Amazon Simple Email Service (Amazon). SES Contoh ini menggunakan kerangka web Flask untuk menangani HTTP routing dan terintegrasi dengan halaman web React untuk menyajikan aplikasi web yang berfungsi penuh.

- Bangun REST layanan Flask yang terintegrasi dengan. Layanan AWS
- Baca, tulis, dan perbarui item kerja yang disimpan dalam tabel DynamoDB.
- Gunakan Amazon SES untuk mengirim laporan email tentang item pekerjaan.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkap di Repositori Contoh AWS Kode di. GitHub

Layanan yang digunakan dalam contoh ini

- DynamoDB
- Amazon SES

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Buat pelacak item Amazon Redshift

Contoh kode berikut menunjukkan cara membuat aplikasi web yang melacak dan melaporkan item pekerjaan menggunakan database Amazon Redshift.

Java

SDKuntuk Java 2.x

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan item pekerjaan yang disimpan dalam database Amazon Redshift.

Untuk kode sumber lengkap dan petunjuk tentang cara menyiapkan Spring REST API yang menanyakan data Amazon Redshift dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. GitHub

Layanan yang digunakan dalam contoh ini

- Amazon Redshift
- Amazon SES

Kotlin

SDKuntuk Kotlin

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan item pekerjaan yang disimpan dalam database Amazon Redshift.

Untuk kode sumber lengkap dan petunjuk tentang cara menyiapkan Spring REST API yang menanyakan data Amazon Redshift dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. GitHub

Layanan yang digunakan dalam contoh ini

- Amazon Redshift
- Amazon SES

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Buat pelacak butir kerja Aurora Nirserver

Contoh kode berikut menunjukkan cara membuat aplikasi web yang melacak item pekerjaan dalam database Amazon Aurora Tanpa Server dan menggunakan Amazon Simple Email Service (AmazonSES) untuk mengirim laporan.

.NET

AWS SDK for .NET

Menunjukkan cara menggunakan AWS SDK for .NET untuk membuat aplikasi web yang melacak item pekerjaan dalam database Amazon Aurora dan laporan email dengan

menggunakan Amazon Simple Email Service (AmazonSES). Contoh ini menggunakan front end yang dibangun dengan React.js untuk berinteraksi dengan fileRESTful. NETbackend.

- Integrasikan aplikasi web React dengan AWS layanan.
- Cantumkan, tambahkan, perbarui, dan hapus butir di tabel Aurora.
- Kirim laporan email tentang item pekerjaan yang difilter menggunakan AmazonSES.
- Menyebarkan dan mengelola sumber daya contoh dengan AWS CloudFormation skrip yang disertakan.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan RDS Data Amazon
- Amazon SES

C++

SDKuntuk C ++

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan butir kerja yang tersimpan dalam basis data Amazon Aurora Nirserver.

Untuk kode sumber lengkap dan instruksi tentang cara menyiapkan C++ REST API yang menanyakan data Amazon Aurora Tanpa Server dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. GitHub

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan RDS Data Amazon
- Amazon SES

Java

SDKuntuk Java 2.x

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan item pekerjaan yang disimpan dalam RDS database Amazon.

Untuk kode sumber lengkap dan petunjuk tentang cara menyiapkan Spring REST API yang menanyakan data Amazon Aurora Tanpa Server dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. GitHub

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan contoh yang menggunakan JDBCAPI, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan RDS Data Amazon
- Amazon SES

JavaScript

SDKuntuk JavaScript (v3)

Menunjukkan cara menggunakan AWS SDK for JavaScript (v3) untuk membuat aplikasi web yang melacak item pekerjaan dalam database Amazon Aurora dan laporan email dengan menggunakan Amazon Simple Email Service (AmazonSES). Contoh ini menggunakan sisi depan yang dibangun dengan React.js untuk berinteraksi dengan backend Express Node.js.

- Integrasikan aplikasi web React.js dengan Layanan AWS.
- Cantumkan, tambahkan, dan perbarui butir di tabel Aurora.
- Kirim laporan email item pekerjaan yang difilter menggunakan AmazonSES.
- Menyebarkan dan mengelola sumber daya contoh dengan AWS CloudFormation skrip yang disertakan.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan RDS Data Amazon
- Amazon SES

Kotlin

SDKuntuk Kotlin

Menunjukkan cara membuat aplikasi web yang melacak dan melaporkan item pekerjaan yang disimpan dalam RDS database Amazon.

Untuk kode sumber lengkap dan petunjuk tentang cara menyiapkan Spring REST API yang menanyakan data Amazon Aurora Tanpa Server dan untuk digunakan oleh aplikasi React, lihat contoh lengkapnya di. GitHub

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan RDS Data Amazon
- Amazon SES

PHP

SDKuntuk PHP

Menunjukkan cara menggunakan AWS SDK for PHP untuk membuat aplikasi web yang melacak item pekerjaan dalam RDS database Amazon dan laporan email dengan menggunakan Amazon Simple Email Service (AmazonSES). Contoh ini menggunakan front end yang dibangun dengan React.js untuk berinteraksi dengan RESTful PHP backend.

- Integrasikan aplikasi web React.js dengan AWS layanan.
- Buat daftar, tambahkan, perbarui, dan hapus item di RDS tabel Amazon.
- Kirim laporan email tentang item pekerjaan yang difilter menggunakan AmazonSES.

- Menyebarkan dan mengelola sumber daya contoh dengan AWS CloudFormation skrip yang disertakan.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan RDS Data Amazon
- Amazon SES

Python

SDKuntuk Python (Boto3)

Menunjukkan cara menggunakan AWS SDK for Python (Boto3) untuk membuat REST layanan yang melacak item pekerjaan di database Amazon Aurora Tanpa Server dan laporan email dengan menggunakan Amazon Simple Email Service (Amazon). SES Contoh ini menggunakan kerangka web Flask untuk menangani HTTP routing dan terintegrasi dengan halaman web React untuk menyajikan aplikasi web yang berfungsi penuh.

- Bangun REST layanan Flask yang terintegrasi dengan. Layanan AWS
- Baca, tulis, dan perbarui butir kerja yang tersimpan dalam basis data Aurora Nirserver.
- Buat AWS Secrets Manager rahasia yang berisi kredensi database dan gunakan untuk mengautentikasi panggilan ke database.
- Gunakan Amazon SES untuk mengirim laporan email tentang item pekerjaan.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Aurora
- Amazon RDS
- Layanan RDS Data Amazon

- Amazon SES

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Mendeteksi PPE dalam gambar dengan Amazon Rekognition menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membuat aplikasi yang menggunakan Amazon Rekognition untuk mendeteksi Personal Protective Equipment PPE () dalam gambar.

Java

SDKuntuk Java 2.x

Menunjukkan cara membuat AWS Lambda fungsi yang mendeteksi gambar dengan Alat Pelindung Diri.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Mendeteksi objek dalam gambar dengan Amazon Rekognition menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membuat aplikasi yang menggunakan Amazon Rekognition untuk mendeteksi objek berdasarkan kategori dalam gambar.

.NET

AWS SDK for .NET

Menunjukkan cara menggunakan Amazon Rekognition. NETAPIuntuk membuat aplikasi yang menggunakan Amazon Rekognition untuk mengidentifikasi objek berdasarkan kategori dalam gambar yang terletak di bucket Amazon Simple Storage Service (Amazon S3). Aplikasi ini mengirimkan pemberitahuan email kepada admin dengan hasilnya menggunakan Amazon Simple Email Service (AmazonSES).

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Rekognition
- Amazon S3
- Amazon SES

Java

SDKuntuk Java 2.x

Menunjukkan cara menggunakan Amazon Rekognition API Java untuk membuat aplikasi yang menggunakan Amazon Rekognition untuk mengidentifikasi objek berdasarkan kategori dalam gambar yang terletak di bucket Amazon Simple Storage Service (Amazon S3). Aplikasi ini mengirimkan pemberitahuan email kepada admin dengan hasilnya menggunakan Amazon Simple Email Service (AmazonSES).

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDKuntuk JavaScript (v3)

Menunjukkan cara menggunakan Amazon Rekognition dengan membuat aplikasi AWS SDK for JavaScript yang menggunakan Amazon Rekognition untuk mengidentifikasi objek berdasarkan kategori dalam gambar yang terletak di bucket Amazon Simple Storage Service (Amazon S3). Aplikasi ini mengirimkan pemberitahuan email kepada admin dengan hasilnya menggunakan Amazon Simple Email Service (AmazonSES).

Pelajari cara:

- Membuat pengguna yang tidak diautentikasi menggunakan Amazon Cognito.
- Menganalisi gambar untuk objek menggunakan Amazon Rekognition.
- Verifikasi alamat email untuk AmazonSES.
- Kirim pemberitahuan email menggunakan AmazonSES.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Rekognition
- Amazon S3
- Amazon SES

Kotlin

SDKuntuk Kotlin

Menunjukkan cara menggunakan Amazon Rekognition API Kotlin untuk membuat aplikasi yang menggunakan Amazon Rekognition untuk mengidentifikasi objek berdasarkan kategori dalam gambar yang terletak di bucket Amazon Simple Storage Service (Amazon S3). Aplikasi ini mengirimkan pemberitahuan email kepada admin dengan hasilnya menggunakan Amazon Simple Email Service (AmazonSES).

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Rekognition
- Amazon S3
- Amazon SES

Python

SDKuntuk Python (Boto3)

Menunjukkan cara menggunakan AWS SDK for Python (Boto3) untuk membuat aplikasi web yang memungkinkan Anda melakukan hal berikut:

- Mengunggah foto ke bucket Amazon Simple Storage Service (Amazon S3).
- Menggunakan Amazon Rekognition untuk menganalisis dan memberi label pada foto.
- Gunakan Amazon Simple Email Service (AmazonSES) untuk mengirim laporan email tentang analisis gambar.

Contoh ini berisi dua komponen utama: halaman web yang ditulis di dalamnya JavaScript yang dibangun dengan React, dan REST layanan yang ditulis dengan Python yang dibangun dengan Flask-. RESTful

Anda dapat menggunakan halaman web React untuk:

- Menampilkan daftar gambar yang disimpan di bucket S3 Anda.
- Mengunggah gambar dari komputer ke bucket S3.
- Menampilkan gambar dan label yang mengidentifikasi item yang terdeteksi dalam gambar.
- Mendapatkan laporan semua gambar di bucket S3 Anda dan mengirimkan email laporan tersebut.

Halaman web memanggil REST layanan. Layanan mengirimkan permintaan ke AWS untuk melakukan tindakan berikut:

- Mendapatkan dan memfilter daftar gambar dalam bucket S3 Anda.
- Merngunggah foto ke bucket S3 Anda.
- Menggunakan Amazon Rekognition untuk menganalisis foto individual dan mendapatkan daftar label yang mengidentifikasi item yang terdeteksi dalam foto.
- Analisis semua foto di bucket S3 Anda dan gunakan Amazon SES untuk mengirim laporan melalui email.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Rekognition
- Amazon S3
- Amazon SES

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Mendeteksi orang dan objek dalam video dengan Amazon Rekognition menggunakan AWS SDK

Contoh kode berikut menunjukkan cara mendeteksi orang dan objek dalam video dengan Amazon Rekognition.

Java

SDKuntuk Java 2.x

Menunjukkan cara menggunakan Amazon Rekognition API Java untuk membuat aplikasi untuk mendeteksi wajah dan objek dalam video yang terletak di bucket Amazon Simple Storage Service (Amazon S3). Aplikasi ini mengirimkan pemberitahuan email kepada admin dengan hasilnya menggunakan Amazon Simple Email Service (AmazonSES).

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- Amazon Rekognition
- Amazon S3
- Amazon SES

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Menghasilkan kredensi untuk terhubung ke titik akhir Amazon SES SMTP

Contoh kode berikut menunjukkan cara menghasilkan kredensi untuk terhubung ke titik akhir Amazon SESSMTP.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
#!/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)
    "eu-south-1",  # Europe (Milan)
    "eu-north-1",  # Europe (Stockholm)
    "sa-east-1",  # South America (Sao Paulo)
    "us-gov-west-1",  # AWS GovCloud (US)
    "us-gov-east-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 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()
```

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Menggunakan Step Functions untuk menginvokasi fungsi Lambda

Contoh kode berikut menunjukkan cara membuat mesin AWS Step Functions status yang memanggil AWS Lambda fungsi secara berurutan.

Java

SDKuntuk Java 2.x

Menunjukkan cara membuat alur kerja AWS tanpa server dengan menggunakan AWS Step Functions dan. AWS SDK for Java 2.x Setiap langkah alur kerja diimplementasikan menggunakan AWS Lambda fungsi.

Untuk kode sumber lengkap dan instruksi tentang cara mengatur dan menjalankan, lihat contoh lengkapnya di GitHub.

Layanan yang digunakan dalam contoh ini

- DynamoDB
- Lambda
- Amazon SES
- Step Functions

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Verifikasi identitas email dan kirim pesan dengan Amazon SES menggunakan AWS SDK

Contoh kode berikut ini menunjukkan cara:

- Tambahkan dan verifikasi alamat email dengan AmazonSES.
- Kirim pesan email standar.
- Buat template dan kirim pesan email template.

- Kirim pesan dengan menggunakan SES SMTP server Amazon.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Verifikasi alamat email dengan Amazon SES dan kirim pesan.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon Simple Email Service (Amazon SES) email demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    ses_client = boto3.client("ses")
    ses_identity = SesIdentity(ses_client)
    ses_mail_sender = SesMailSender(ses_client)
    ses_template = SesTemplate(ses_client)
    email = input("Enter an email address to send mail with Amazon SES: ")
    status = ses_identity.get_identity_status(email)
    verified = status == "Success"
    if not verified:
        answer = input(
            f"The address '{email}' is not verified with Amazon SES. Unless your
 "
            f"Amazon SES account is out of sandbox, you can send mail only from "
            f"and to verified accounts. Do you want to verify this account for
 use "
            f"with Amazon SES? If yes, the address will receive a verification "
            f"email (y/n): "
        )
        if answer.lower() == "y":
            ses_identity.verify_email_identity(email)
            print(f"Follow the steps in the email to {email} to complete
 verification.")
```

```
            print("Waiting for verification...")
            try:
                ses_identity.wait_until_identity_exists(email)
                print(f"Identity verified for {email}.")
                verified = True
            except WaiterError:
                print(
                    f"Verification timeout exceeded. You must complete the "
                    f"steps in the email sent to {email} to verify the address."
                )

    if verified:
        test_message_text = "Hello from the Amazon SES mail demo!"
        test_message_html = "<p>Hello!</p><p>From the <b>Amazon SES</b> mail
demo!</p>"

        print(f"Sending mail from {email} to {email}.")
        ses_mail_sender.send_email(
            email,
            SesDestination([email]),
            "Amazon SES demo",
            test_message_text,
            test_message_html,
        )
        input("Mail sent. Check your inbox and press Enter to continue.")

        template = {
            "name": "doc-example-template",
            "subject": "Example of an email template.",
            "text": "This is what {{name}} will {{action}} if {{name}} can't
display "
            "HTML.",
            "html": "<p><i>This</i> is what {{name}} will {{action}} if {{name}}
"
            "<b>can</b> display HTML.</p>",
        }
        print("Creating a template and sending a templated email.")
        ses_template.create_template(**template)
        template_data = {"name": email.split("@")[0], "action": "read"}
        if ses_template.verify_tags(template_data):
            ses_mail_sender.send_templated_email(
                email, SesDestination([email]), ses_template.name(),
template_data
            )
```

```
            input("Mail sent. Check your inbox and press Enter to continue.")

        print("Sending mail through the Amazon SES SMTP server.")
        boto3_session = boto3.Session()
        region = boto3_session.region_name
        credentials = boto3_session.get_credentials()
        port = 587
        smtp_server = f"email-smtp.{region}.amazonaws.com"
        password = calculate_key(credentials.secret_key, region)
        message = """
Subject: Hi there

This message is sent from the Amazon SES SMTP mail demo."""
        context = ssl.create_default_context()
        with smtplib.SMTP(smtp_server, port) as server:
            server.starttls(context=context)
            server.login(credentials.access_key, password)
            server.sendmail(email, email, message)
        print("Mail sent. Check your inbox!")

    if ses_template.template is not None:
        print("Deleting demo template.")
        ses_template.delete_template()
    if verified:
        answer = input(f"Do you want to remove {email} from Amazon SES (y/n)? ")
        if answer.lower() == "y":
            ses_identity.delete_identity(email)
    print("Thanks for watching!")
    print("-" * 88)
```

Buat fungsi untuk membungkus tindakan SES identitas Amazon.

```
class SesIdentity:
    """Encapsulates Amazon SES identity functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client
```

```
    def verify_domain_identity(self, domain_name):
        """
        Starts verification of a domain identity. To complete verification, you
 must
        create a TXT record with a specific format through your DNS provider.

        For more information, see *Verifying a domain with Amazon SES* in the
        Amazon SES documentation:
            https://docs.aws.amazon.com/ses/latest/DeveloperGuide/verify-domain-
procedure.html

        :param domain_name: The name of the domain to verify.
        :return: The token to include in the TXT record with your DNS provider.
        """
        try:
            response = self.ses_client.verify_domain_identity(Domain=domain_name)
            token = response["VerificationToken"]
            logger.info("Got domain verification token for %s.", domain_name)
        except ClientError:
            logger.exception("Couldn't verify domain %s.", domain_name)
            raise
        else:
            return token


    def verify_email_identity(self, email_address):
        """
        Starts verification of an email identity. This function causes an email
        to be sent to the specified email address from Amazon SES. To complete
        verification, follow the instructions in the email.

        :param email_address: The email address to verify.
        """
        try:
            self.ses_client.verify_email_identity(EmailAddress=email_address)
            logger.info("Started verification of %s.", email_address)
        except ClientError:
            logger.exception("Couldn't start verification of %s.", email_address)
            raise


    def wait_until_identity_exists(self, identity):
```

```
        """
        Waits until an identity exists. The waiter polls Amazon SES until the
        identity has been successfully verified or until it exceeds its maximum
time.

        :param identity: The identity to wait for.
        """
        try:
            waiter = self.ses_client.get_waiter("identity_exists")
            logger.info("Waiting until %s exists.", identity)
            waiter.wait(Identities=[identity])
        except WaiterError:
            logger.error("Waiting for identity %s failed or timed out.",
identity)
            raise


    def get_identity_status(self, identity):
        """
        Gets the status of an identity. This can be used to discover whether
        an identity has been successfully verified.

        :param identity: The identity to query.
        :return: The status of the identity.
        """
        try:
            response = self.ses_client.get_identity_verification_attributes(
                Identities=[identity]
            )
            status = response["VerificationAttributes"].get(
                identity, {"VerificationStatus": "NotFound"}
            )["VerificationStatus"]
            logger.info("Got status of %s for %s.", status, identity)
        except ClientError:
            logger.exception("Couldn't get status for %s.", identity)
            raise
        else:
            return status


    def delete_identity(self, identity):
        """
        Deletes an identity.
```

```
        :param identity: The identity to remove.
        """
        try:
            self.ses_client.delete_identity(Identity=identity)
            logger.info("Deleted identity %s.", identity)
        except ClientError:
            logger.exception("Couldn't delete identity %s.", identity)
            raise


    def list_identities(self, identity_type, max_items):
        """
        Gets the identities of the specified type for the current account.

        :param identity_type: The type of identity to retrieve, such as
 EmailAddress.
        :param max_items: The maximum number of identities to retrieve.
        :return: The list of retrieved identities.
        """
        try:
            response = self.ses_client.list_identities(
                IdentityType=identity_type, MaxItems=max_items
            )
            identities = response["Identities"]
            logger.info("Got %s identities for the current account.",
 len(identities))
        except ClientError:
            logger.exception("Couldn't list identities for the current account.")
            raise
        else:
            return identities
```

Buat fungsi untuk membungkus tindakan SES template Amazon.

```
class SesTemplate:
    """Encapsulates Amazon SES template functions."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
```

```
        """
        self.ses_client = ses_client
        self.template = None
        self.template_tags = set()

    def _extract_tags(self, subject, text, html):
        """
        Extracts tags from a template as a set of unique values.

        :param subject: The subject of the email.
        :param text: The text version of the email.
        :param html: The html version of the email.
        """
        self.template_tags = set(re.findall(TEMPLATE_REGEX, subject + text +
html))
        logger.info("Extracted template tags: %s", self.template_tags)


    def create_template(self, name, subject, text, html):
        """
        Creates an email template.

        :param name: The name of the template.
        :param subject: The subject of the email.
        :param text: The plain text version of the email.
        :param html: The HTML version of the email.
        """
        try:
            template = {
                "TemplateName": name,
                "SubjectPart": subject,
                "TextPart": text,
                "HtmlPart": html,
            }
            self.ses_client.create_template(Template=template)
            logger.info("Created template %s.", name)
            self.template = template
            self._extract_tags(subject, text, html)
        except ClientError:
            logger.exception("Couldn't create template %s.", name)
            raise


    def delete_template(self):
```

```
        """
        Deletes an email template.
        """
        try:

self.ses_client.delete_template(TemplateName=self.template["TemplateName"])
            logger.info("Deleted template %s.", self.template["TemplateName"])
            self.template = None
            self.template_tags = None
        except ClientError:
            logger.exception(
                "Couldn't delete template %s.", self.template["TemplateName"]
            )
            raise


    def get_template(self, name):
        """
        Gets a previously created email template.

        :param name: The name of the template to retrieve.
        :return: The retrieved email template.
        """
        try:
            response = self.ses_client.get_template(TemplateName=name)
            self.template = response["Template"]
            logger.info("Got template %s.", name)
            self._extract_tags(
                self.template["SubjectPart"],
                self.template["TextPart"],
                self.template["HtmlPart"],
            )
        except ClientError:
            logger.exception("Couldn't get template %s.", name)
            raise
        else:
            return self.template


    def list_templates(self):
        """
        Gets a list of all email templates for the current account.

        :return: The list of retrieved email templates.
```

```
        """
        try:
            response = self.ses_client.list_templates()
            templates = response["TemplatesMetadata"]
            logger.info("Got %s templates.", len(templates))
        except ClientError:
            logger.exception("Couldn't get templates.")
            raise
        else:
            return templates


    def update_template(self, name, subject, text, html):
        """
        Updates a previously created email template.

        :param name: The name of the template.
        :param subject: The subject of the email.
        :param text: The plain text version of the email.
        :param html: The HTML version of the email.
        """
        try:
            template = {
                "TemplateName": name,
                "SubjectPart": subject,
                "TextPart": text,
                "HtmlPart": html,
            }
            self.ses_client.update_template(Template=template)
            logger.info("Updated template %s.", name)
            self.template = template
            self._extract_tags(subject, text, html)
        except ClientError:
            logger.exception("Couldn't update template %s.", name)
            raise
```

Buat fungsi untuk membungkus tindakan SES email Amazon.

```
class SesDestination:
    """Contains data about an email destination."""
```

```
    def __init__(self, tos, ccs=None, bccs=None):
        """
        :param tos: The list of recipients on the 'To:' line.
        :param ccs: The list of recipients on the 'CC:' line.
        :param bccs: The list of recipients on the 'BCC:' line.
        """
        self.tos = tos
        self.ccs = ccs
        self.bccs = bccs

    def to_service_format(self):
        """
        :return: The destination data in the format expected by Amazon SES.
        """
        svc_format = {"ToAddresses": self.tos}
        if self.ccs is not None:
            svc_format["CcAddresses"] = self.ccs
        if self.bccs is not None:
            svc_format["BccAddresses"] = self.bccs
        return svc_format



class SesMailSender:
    """Encapsulates functions to send emails with Amazon SES."""

    def __init__(self, ses_client):
        """
        :param ses_client: A Boto3 Amazon SES client.
        """
        self.ses_client = ses_client


    def send_email(self, source, destination, subject, text, html,
 reply_tos=None):
        """
        Sends an email.

        Note: If your account is in the Amazon SES  sandbox, the source and
        destination email accounts must both be verified.

        :param source: The source email account.
        :param destination: The destination email account.
```

```
        :param subject: The subject of the email.
        :param text: The plain text version of the body of the email.
        :param html: The HTML version of the body of the email.
        :param reply_tos: Email accounts that will receive a reply if the
recipient
                          replies to the message.
        :return: The ID of the message, assigned by Amazon SES.
        """
        send_args = {
            "Source": source,
            "Destination": destination.to_service_format(),
            "Message": {
                "Subject": {"Data": subject},
                "Body": {"Text": {"Data": text}, "Html": {"Data": html}},
            },
        }
        if reply_tos is not None:
            send_args["ReplyToAddresses"] = reply_tos
        try:
            response = self.ses_client.send_email(**send_args)
            message_id = response["MessageId"]
            logger.info(
                "Sent mail %s from %s to %s.", message_id, source,
destination.tos
            )
        except ClientError:
            logger.exception(
                "Couldn't send mail from %s to %s.", source, destination.tos
            )
            raise
        else:
            return message_id


    def send_templated_email(
        self, source, destination, template_name, template_data, reply_tos=None
    ):
        """
        Sends an email based on a template. A template contains replaceable tags
        each enclosed in two curly braces, such as {{name}}. The template data
passed
        in this function contains key-value pairs that define the values to
insert
        in place of the template tags.
```

```
        Note: If your account is in the Amazon SES  sandbox, the source and
        destination email accounts must both be verified.

        :param source: The source email account.
        :param destination: The destination email account.
        :param template_name: The name of a previously created template.
        :param template_data: JSON-formatted key-value pairs of replacement
values
                              that are inserted in the template before it is
sent.
        :return: The ID of the message, assigned by Amazon SES.
        """
        send_args = {
            "Source": source,
            "Destination": destination.to_service_format(),
            "Template": template_name,
            "TemplateData": json.dumps(template_data),
        }
        if reply_tos is not None:
            send_args["ReplyToAddresses"] = reply_tos
        try:
            response = self.ses_client.send_templated_email(**send_args)
            message_id = response["MessageId"]
            logger.info(
                "Sent templated mail %s from %s to %s.",
                message_id,
                source,
                destination.tos,
            )
        except ClientError:
            logger.exception(
                "Couldn't send templated mail from %s to %s.", source,
destination.tos
            )
            raise
        else:
            return message_id
```

- Untuk API detailnya, lihat topik berikut AWS SDKuntuk Referensi Python (Boto3). API

- CreateTemplate
- DeleteIdentity
- DeleteTemplate
- GetIdentityVerificationAttributes
- GetTemplate
- ListIdentities
- ListTemplates
- SendEmail
- SendTemplatedEmail
- UpdateTemplate
- VerifyDomainIdentity
- VerifyEmailIdentity

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Contoh kode untuk Amazon SES API v2 menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan Amazon SES API v2 dengan kit pengembangan AWS perangkat lunak (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Contoh kode

- Contoh dasar untuk Amazon SES API v2 menggunakan AWS SDKs
  - Tindakan untuk Amazon SES API v2 menggunakan AWS SDKs
    - Gunakan CreateContact dengan AWS SDK atau CLI
    - Gunakan CreateContactList dengan AWS SDK atau CLI
    - Gunakan CreateEmailIdentity dengan AWS SDK atau CLI
    - Gunakan CreateEmailTemplate dengan AWS SDK atau CLI
    - Gunakan DeleteContactList dengan AWS SDK atau CLI
    - Gunakan DeleteEmailIdentity dengan AWS SDK atau CLI
    - Gunakan DeleteEmailTemplate dengan AWS SDK atau CLI
    - Gunakan GetEmailIdentity dengan AWS SDK atau CLI
    - Gunakan ListContactLists dengan AWS SDK atau CLI
    - Gunakan ListContacts dengan AWS SDK atau CLI
    - Gunakan SendEmail dengan AWS SDK atau CLI
- Skenario untuk Amazon SES API v2 menggunakan AWS SDKs
  - Alur kerja Newsletter Amazon SES API v2 lengkap menggunakan AWS SDK

Contoh dasar untuk Amazon SES API v2 menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar Amazon Simple Email Service API v2 dengan AWS SDKs.

Contoh

- Tindakan untuk Amazon SES API v2 menggunakan AWS SDKs
  - Gunakan CreateContact dengan AWS SDK atau CLI
  - Gunakan CreateContactList dengan AWS SDK atau CLI
  - Gunakan CreateEmailIdentity dengan AWS SDK atau CLI
  - Gunakan CreateEmailTemplate dengan AWS SDK atau CLI
  - Gunakan DeleteContactList dengan AWS SDK atau CLI
  - Gunakan DeleteEmailIdentity dengan AWS SDK atau CLI
  - Gunakan DeleteEmailTemplate dengan AWS SDK atau CLI
  - Gunakan GetEmailIdentity dengan AWS SDK atau CLI
  - Gunakan ListContactLists dengan AWS SDK atau CLI

- Gunakan ListContacts dengan AWS SDK atau CLI
- Gunakan SendEmail dengan AWS SDK atau CLI

Tindakan untuk Amazon SES API v2 menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara melakukan tindakan Amazon SES API v2 individual dengan AWS SDKs. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Kutipan ini menyebut Amazon SES API v2 API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Anda dapat melihat tindakan dalam konteks diSkenario untuk Amazon SES API v2 menggunakan AWS SDKs .

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat APIReferensi Amazon Simple Email Service API v2.

Contoh

- Gunakan CreateContact dengan AWS SDK atau CLI
- Gunakan CreateContactList dengan AWS SDK atau CLI
- Gunakan CreateEmailIdentity dengan AWS SDK atau CLI
- Gunakan CreateEmailTemplate dengan AWS SDK atau CLI
- Gunakan DeleteContactList dengan AWS SDK atau CLI
- Gunakan DeleteEmailIdentity dengan AWS SDK atau CLI
- Gunakan DeleteEmailTemplate dengan AWS SDK atau CLI
- Gunakan GetEmailIdentity dengan AWS SDK atau CLI
- Gunakan ListContactLists dengan AWS SDK atau CLI
- Gunakan ListContacts dengan AWS SDK atau CLI
- Gunakan SendEmail dengan AWS SDK atau CLI

Gunakan **`CreateContact`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateContact`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Alur kerja buletin

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Creates a contact and adds it to the specified contact list.
    /// </summary>
    /// <param name="emailAddress">The email address of the contact.</param>
    /// <param name="contactListName">The name of the contact list.</param>
    /// <returns>The response from the CreateContact operation.</returns>
    public async Task<bool> CreateContactAsync(string emailAddress, string
contactListName)
    {
        var request = new CreateContactRequest
        {
            EmailAddress = emailAddress,
            ContactListName = contactListName
        };

        try
        {
            var response = await _sesClient.CreateContactAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AlreadyExistsException ex)
        {
            Console.WriteLine($"Contact with email address {emailAddress} already
exists in the contact list {contactListName}.");
            Console.WriteLine(ex.Message);
            return true;
        }
        catch (NotFoundException ex)
        {
```

```
            Console.WriteLine($"The contact list {contactListName} does not
exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while creating the contact:
{ex.Message}");
        }
        return false;
    }
```

- Untuk API detailnya, lihat CreateContactdi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
      try {
        // Create a new contact with the provided email address in the
        CreateContactRequest contactRequest = CreateContactRequest.builder()
            .contactListName(CONTACT_LIST_NAME)
            .emailAddress(emailAddress)
            .build();

        sesClient.createContact(contactRequest);
        contacts.add(emailAddress);

        System.out.println("Contact created: " + emailAddress);
```

```
        // Send a welcome email to the new contact
        String welcomeHtml = Files.readString(Paths.get("resources/
coupon_newsletter/welcome.html"));
        String welcomeText = Files.readString(Paths.get("resources/
coupon_newsletter/welcome.txt"));

        SendEmailRequest welcomeEmailRequest = SendEmailRequest.builder()
            .fromEmailAddress(this.verifiedEmail)
            .destination(Destination.builder().toAddresses(emailAddress).build())
            .content(EmailContent.builder()
                .simple(
                    Message.builder()
                        .subject(Content.builder().data("Welcome to the Weekly
 Coupons Newsletter").build())
                        .body(Body.builder()
                            .text(Content.builder().data(welcomeText).build())
                            .html(Content.builder().data(welcomeHtml).build())
                            .build())
                        .build())
                .build())
            .build();
        SendEmailResponse welcomeEmailResponse =
 sesClient.sendEmail(welcomeEmailRequest);
        System.out.println("Welcome email sent: " +
 welcomeEmailResponse.messageId());
      } catch (AlreadyExistsException e) {
        // If the contact already exists, skip this step for that contact and
 proceed
        // with the next contact
        System.out.println("Contact already exists, skipping creation...");
      } catch (Exception e) {
        System.err.println("Error occurred while processing email address " +
 emailAddress + ": " + e.getMessage());
        throw e;
      }
    }
```

- Untuk API detailnya, lihat CreateContactdi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
        self.ses_client = ses_client
        self.sleep = sleep


            try:
                # Create a new contact
                self.ses_client.create_contact(
                    ContactListName=CONTACT_LIST_NAME, EmailAddress=email
```

```
                )
                print(f"Contact with email '{email}' created successfully.")

                # Send the welcome email
                self.ses_client.send_email(
                    FromEmailAddress=self.verified_email,
                    Destination={"ToAddresses": [email]},
                    Content={
                        "Simple": {
                            "Subject": {
                                "Data": "Welcome to the Weekly Coupons
Newsletter"
                            },
                            "Body": {
                                "Text": {"Data": welcome_text},
                                "Html": {"Data": welcome_html},
                            },
                        }
                    },
                )
                print(f"Welcome email sent to '{email}'.")
                if self.sleep:
                    # 1 email per second in sandbox mode, remove in production.
                    sleep(1.1)
            except ClientError as e:
                # If the contact already exists, skip and proceed
                if e.response["Error"]["Code"] == "AlreadyExistsException":
                    print(f"Contact with email '{email}' already exists.
Skipping...")
                else:
                    raise e
```

- Untuk API detailnya, lihat CreateContact AWSSDKReferensi Python (Boto3). API

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn add_contact(client: &Client, list: &str, email: &str) -> Result<(),
 Error> {
    client
        .create_contact()
        .contact_list_name(list)
        .email_address(email)
        .send()
        .await?;

    println!("Created contact");

    Ok(())
}
```

- Untuk API detailnya, lihat CreateContact AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`CreateContactList`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateContactList`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Alur kerja buletin

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Creates a contact list with the specified name.
    /// </summary>
    /// <param name="contactListName">The name of the contact list.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> CreateContactListAsync(string contactListName)
    {
        var request = new CreateContactListRequest
        {
            ContactListName = contactListName
        };

        try
        {
            var response = await _sesClient.CreateContactListAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AlreadyExistsException ex)
        {
            Console.WriteLine($"Contact list with name {contactListName} already
exists.");
            Console.WriteLine(ex.Message);
            return true;
        }
        catch (LimitExceededException ex)
        {
            Console.WriteLine("The limit for contact lists has been exceeded.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
```

```
            Console.WriteLine("Too many requests were made. Please try again
 later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while creating the contact
 list: {ex.Message}");
        }
        return false;
    }
```

- Untuk API detailnya, lihat CreateContactListdi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    try {
      // 2. Create a contact list
      String contactListName = CONTACT_LIST_NAME;
      CreateContactListRequest createContactListRequest =
 CreateContactListRequest.builder()
          .contactListName(contactListName)
          .build();
      sesClient.createContactList(createContactListRequest);
      System.out.println("Contact list created: " + contactListName);
    } catch (AlreadyExistsException e) {
      System.out.println("Contact list already exists, skipping creation: weekly-
coupons-newsletter");
    } catch (LimitExceededException e) {
      System.err.println("Limit for contact lists has been exceeded.");
      throw e;
    } catch (SesV2Exception e) {
      System.err.println("Error creating contact list: " + e.getMessage());
```

```
            throw e;
        }
```

- Untuk API detailnya, lihat CreateContactListdi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
        self.ses_client = ses_client
        self.sleep = sleep
```

```
        try:

 self.ses_client.create_contact_list(ContactListName=CONTACT_LIST_NAME)
            print(f"Contact list '{CONTACT_LIST_NAME}' created successfully.")
        except ClientError as e:
            # If the contact list already exists, skip and proceed
            if e.response["Error"]["Code"] == "AlreadyExistsException":
                print(f"Contact list '{CONTACT_LIST_NAME}' already exists.")
            else:
                raise e
```

- Untuk API detailnya, lihat CreateContactList AWSSDKReferensi Python (Boto3). API

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn make_list(client: &Client, contact_list: &str) -> Result<(), Error> {
    client
        .create_contact_list()
        .contact_list_name(contact_list)
        .send()
        .await?;

    println!("Created contact list.");

    Ok(())
}
```

- Untuk API detailnya, lihat CreateContactList AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`CreateEmailIdentity`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateEmailIdentity`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Alur kerja buletin

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Creates an email identity (email address or domain) and starts the
verification process.
   /// </summary>
   /// <param name="emailIdentity">The email address or domain to create and
verify.</param>
   /// <returns>The response from the CreateEmailIdentity operation.</returns>
   public async Task<CreateEmailIdentityResponse>
CreateEmailIdentityAsync(string emailIdentity)
   {
       var request = new CreateEmailIdentityRequest
       {
           EmailIdentity = emailIdentity
       };

       try
       {
           var response = await _sesClient.CreateEmailIdentityAsync(request);
```

```
            return response;
        }
        catch (AlreadyExistsException ex)
        {
            Console.WriteLine($"Email identity {emailIdentity} already exists.");
            Console.WriteLine(ex.Message);
            throw;
        }
        catch (ConcurrentModificationException ex)
        {
            Console.WriteLine($"The email identity {emailIdentity} is being
modified by another operation or thread.");
            Console.WriteLine(ex.Message);
            throw;
        }
        catch (LimitExceededException ex)
        {
            Console.WriteLine("The limit for email identities has been
exceeded.");
            Console.WriteLine(ex.Message);
            throw;
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The email identity {emailIdentity} does not
exist.");
            Console.WriteLine(ex.Message);
            throw;
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while creating the email
identity: {ex.Message}");
            throw;
        }
    }
```

- Untuk API detailnya, lihat CreateEmailIdentitydi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
   try {
     CreateEmailIdentityRequest createEmailIdentityRequest =
CreateEmailIdentityRequest.builder()
         .emailIdentity(verifiedEmail)
         .build();
     sesClient.createEmailIdentity(createEmailIdentityRequest);
     System.out.println("Email identity created: " + verifiedEmail);
   } catch (AlreadyExistsException e) {
     System.out.println("Email identity already exists, skipping creation: " +
verifiedEmail);
   } catch (NotFoundException e) {
     System.err.println("The provided email address is not verified: " +
verifiedEmail);
     throw e;
   } catch (LimitExceededException e) {
     System.err
         .println("You have reached the limit for email identities. Please
remove some identities and try again.");
     throw e;
   } catch (SesV2Exception e) {
     System.err.println("Error creating email identity: " + e.getMessage());
     throw e;
   }
```

- Untuk API detailnya, lihat CreateEmailIdentitydi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
        self.ses_client = ses_client
        self.sleep = sleep


        try:

 self.ses_client.create_email_identity(EmailIdentity=self.verified_email)
```

```
            print(f"Email identity '{self.verified_email}' created
successfully.")
        except ClientError as e:
            # If the email identity already exists, skip and proceed
            if e.response["Error"]["Code"] == "AlreadyExistsException":
                print(f"Email identity '{self.verified_email}' already exists.")
            else:
                raise e
```

- Untuk API detailnya, lihat CreateEmailIdentity AWSSDKReferensi Python (Boto3). API

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
        match self
            .client
            .create_email_identity()
            .email_identity(self.verified_email.clone())
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Email identity created
successfully.")?,
            Err(e) => match e.into_service_error() {
                CreateEmailIdentityError::AlreadyExistsException(_) => {
                    writeln!(
                        self.stdout,
                        "Email identity already exists, skipping creation."
                    )?;
                }
                e => return Err(anyhow!("Error creating email identity: {}", e)),
            },
        }
```

- Untuk API detailnya, lihat CreateEmailIdentity AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`CreateEmailTemplate`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`CreateEmailTemplate`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Alur kerja buletin

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Creates an email template with the specified content.
    /// </summary>
    /// <param name="templateName">The name of the email template.</param>
    /// <param name="subject">The subject of the email template.</param>
    /// <param name="htmlContent">The HTML content of the email template.</param>
    /// <param name="textContent">The text content of the email template.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> CreateEmailTemplateAsync(string templateName, string
subject, string htmlContent, string textContent)
    {
        var request = new CreateEmailTemplateRequest
        {
```

```
            TemplateName = templateName,
            TemplateContent = new EmailTemplateContent
            {
                Subject = subject,
                Html = htmlContent,
                Text = textContent
            }
        };

        try
        {
            var response = await _sesClient.CreateEmailTemplateAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AlreadyExistsException ex)
        {
            Console.WriteLine($"Email template with name {templateName} already
exists.");
            Console.WriteLine(ex.Message);
        }
        catch (LimitExceededException ex)
        {
            Console.WriteLine("The limit for email templates has been
exceeded.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while creating the email
template: {ex.Message}");
        }

        return false;
    }
```

- Untuk API detailnya, lihat CreateEmailTemplatedi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    try {
      // Create an email template named "weekly-coupons"
      String newsletterHtml = loadFile("resources/coupon_newsletter/coupon-
newsletter.html");
      String newsletterText = loadFile("resources/coupon_newsletter/coupon-
newsletter.txt");

      CreateEmailTemplateRequest templateRequest =
 CreateEmailTemplateRequest.builder()
          .templateName(TEMPLATE_NAME)
          .templateContent(EmailTemplateContent.builder()
              .subject("Weekly Coupons Newsletter")
              .html(newsletterHtml)
              .text(newsletterText)
              .build())
          .build();

      sesClient.createEmailTemplate(templateRequest);

      System.out.println("Email template created: " + TEMPLATE_NAME);
    } catch (AlreadyExistsException e) {
      // If the template already exists, skip this step and proceed with the next
      // operation
      System.out.println("Email template already exists, skipping creation...");
    } catch (LimitExceededException e) {
      // If the limit for email templates is exceeded, fail the workflow and
 inform
      // the user
      System.err.println("You have reached the limit for email templates. Please
 remove some templates and try again.");
      throw e;
    } catch (Exception e) {
```

```
      System.err.println("Error occurred while creating email template: " +
 e.getMessage());
      throw e;
    }
```

- Untuk API detailnya, lihat CreateEmailTemplatedi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
```

```
        self.ses_client = ses_client
        self.sleep = sleep


        try:
            template_content = {
                "Subject": "Weekly Coupons Newsletter",
                "Html": load_file_content("coupon-newsletter.html"),
                "Text": load_file_content("coupon-newsletter.txt"),
            }
            self.ses_client.create_email_template(
                TemplateName=TEMPLATE_NAME, TemplateContent=template_content
            )
            print(f"Email template '{TEMPLATE_NAME}' created successfully.")
        except ClientError as e:
            # If the template already exists, skip and proceed
            if e.response["Error"]["Code"] == "AlreadyExistsException":
                print(f"Email template '{TEMPLATE_NAME}' already exists.")
            else:
                raise e
```

- Untuk API detailnya, lihat CreateEmailTemplate AWSSDKReferensi Python (Boto3). API

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
        let template_html =
            std::fs::read_to_string("../resources/newsletter/coupon-
newsletter.html")
                .unwrap_or_else(|_| "Missing coupon-
newsletter.html".to_string());
        let template_text =
            std::fs::read_to_string("../resources/newsletter/coupon-
newsletter.txt")
```

```
                .unwrap_or_else(|_| "Missing coupon-newsletter.txt".to_string());

        // Create the email template
        let template_content = EmailTemplateContent::builder()
            .subject("Weekly Coupons Newsletter")
            .html(template_html)
            .text(template_text)
            .build();

        match self
            .client
            .create_email_template()
            .template_name(TEMPLATE_NAME)
            .template_content(template_content)
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Email template created
successfully.")?,
            Err(e) => match e.into_service_error() {
                CreateEmailTemplateError::AlreadyExistsException(_) => {
                    writeln!(
                        self.stdout,
                        "Email template already exists, skipping creation."
                    )?;
                }
                e => return Err(anyhow!("Error creating email template: {}", e)),
            },
        }
```

- Untuk API detailnya, lihat CreateEmailTemplate AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteContactList`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteContactList`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Alur kerja buletin

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Deletes a contact list and all contacts within it.
    /// </summary>
    /// <param name="contactListName">The name of the contact list to delete.</
param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteContactListAsync(string contactListName)
    {
        var request = new DeleteContactListRequest
        {
            ContactListName = contactListName
        };

        try
        {
            var response = await _sesClient.DeleteContactListAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (ConcurrentModificationException ex)
        {
            Console.WriteLine($"The contact list {contactListName} is being
 modified by another operation or thread.");
            Console.WriteLine(ex.Message);
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The contact list {contactListName} does not
 exist.");
            Console.WriteLine(ex.Message);
        }
```

```
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while deleting the contact
list: {ex.Message}");
        }

        return false;
    }
```

- Untuk API detailnya, lihat DeleteContactListdi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    try {
      // Delete the contact list
      DeleteContactListRequest deleteContactListRequest =
DeleteContactListRequest.builder()
          .contactListName(CONTACT_LIST_NAME)
          .build();

      sesClient.deleteContactList(deleteContactListRequest);

      System.out.println("Contact list deleted: " + CONTACT_LIST_NAME);
    } catch (NotFoundException e) {
      // If the contact list does not exist, log the error and proceed
      System.out.println("Contact list not found. Skipping deletion...");
    } catch (Exception e) {
```

```
      System.err.println("Error occurred while deleting the contact list: " +
 e.getMessage());
      e.printStackTrace();
    }
```

- Untuk API detailnya, lihat DeleteContactListdi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
```

```
        self.ses_client = ses_client
        self.sleep = sleep


        try:

self.ses_client.delete_contact_list(ContactListName=CONTACT_LIST_NAME)
            print(f"Contact list '{CONTACT_LIST_NAME}' deleted successfully.")
        except ClientError as e:
            # If the contact list doesn't exist, skip and proceed
            if e.response["Error"]["Code"] == "NotFoundException":
                print(f"Contact list '{CONTACT_LIST_NAME}' does not exist.")
            else:
                print(e)
```

- Untuk API detailnya, lihat DeleteContactList AWSSDKReferensi Python (Boto3). API

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
        match self
            .client
            .delete_contact_list()
            .contact_list_name(CONTACT_LIST_NAME)
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Contact list deleted
successfully.")?,
            Err(e) => return Err(anyhow!("Error deleting contact list: {e}")),
        }
```

- Untuk API detailnya, lihat DeleteContactList AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteEmailIdentity`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteEmailIdentity`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Alur kerja buletin

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Deletes an email identity (email address or domain).
    /// </summary>
    /// <param name="emailIdentity">The email address or domain to delete.</
param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteEmailIdentityAsync(string emailIdentity)
    {
        var request = new DeleteEmailIdentityRequest
        {
            EmailIdentity = emailIdentity
        };

        try
        {
```

```
            var response = await _sesClient.DeleteEmailIdentityAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (ConcurrentModificationException ex)
        {
            Console.WriteLine($"The email identity {emailIdentity} is being
modified by another operation or thread.");
            Console.WriteLine(ex.Message);
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The email identity {emailIdentity} does not
exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while deleting the email
identity: {ex.Message}");
        }

        return false;
    }
```

- Untuk API detailnya, lihat DeleteEmailIdentitydi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
      try {
        // Delete the email identity
        DeleteEmailIdentityRequest deleteIdentityRequest =
DeleteEmailIdentityRequest.builder()
            .emailIdentity(this.verifiedEmail)
            .build();

        sesClient.deleteEmailIdentity(deleteIdentityRequest);

        System.out.println("Email identity deleted: " + this.verifiedEmail);
      } catch (NotFoundException e) {
        // If the email identity does not exist, log the error and proceed
        System.out.println("Email identity not found. Skipping deletion...");
      } catch (Exception e) {
        System.err.println("Error occurred while deleting the email identity: " +
e.getMessage());
        e.printStackTrace();
      }
    } else {
      System.out.println("Skipping email identity deletion.");
    }
```

- Untuk API detailnya, lihat DeleteEmailIdentitydi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
```

```
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
        self.ses_client = ses_client
        self.sleep = sleep


            try:

 self.ses_client.delete_email_identity(EmailIdentity=self.verified_email)
                print(f"Email identity '{self.verified_email}' deleted
 successfully.")
            except ClientError as e:
                # If the email identity doesn't exist, skip and proceed
                if e.response["Error"]["Code"] == "NotFoundException":
                    print(f"Email identity '{self.verified_email}' does not
 exist.")
                else:
                    print(e)
```

- Untuk API detailnya, lihat DeleteEmailIdentity AWSSDKReferensi Python (Boto3). API

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
            match self
                .client
                .delete_email_identity()
                .email_identity(self.verified_email.clone())
                .send()
                .await
            {
                Ok(_) => writeln!(self.stdout, "Email identity deleted
successfully.")?,
                Err(e) => {
                    return Err(anyhow!("Error deleting email identity: {}", e));
                }
            }
```

- Untuk API detailnya, lihat DeleteEmailIdentity AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`DeleteEmailTemplate`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`DeleteEmailTemplate`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Alur kerja buletin

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Deletes an email template.
    /// </summary>
    /// <param name="templateName">The name of the email template to delete.</
param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteEmailTemplateAsync(string templateName)
    {
        var request = new DeleteEmailTemplateRequest
        {
            TemplateName = templateName
        };

        try
        {
            var response = await _sesClient.DeleteEmailTemplateAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The email template {templateName} does not
 exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
 later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
```

```
            Console.WriteLine($"An error occurred while deleting the email
template: {ex.Message}");
        }

        return false;
    }
```

- Untuk API detailnya, lihat DeleteEmailTemplatedi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    try {
      // Delete the template
      DeleteEmailTemplateRequest deleteTemplateRequest =
DeleteEmailTemplateRequest.builder()
          .templateName(TEMPLATE_NAME)
          .build();

      sesClient.deleteEmailTemplate(deleteTemplateRequest);

      System.out.println("Email template deleted: " + TEMPLATE_NAME);
    } catch (NotFoundException e) {
      // If the email template does not exist, log the error and proceed
      System.out.println("Email template not found. Skipping deletion...");
    } catch (Exception e) {
      System.err.println("Error occurred while deleting the email template: " +
e.getMessage());
      e.printStackTrace();
    }
```

- Untuk API detailnya, lihat DeleteEmailTemplatedi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
        self.ses_client = ses_client
        self.sleep = sleep


        try:
            self.ses_client.delete_email_template(TemplateName=TEMPLATE_NAME)
            print(f"Email template '{TEMPLATE_NAME}' deleted successfully.")
        except ClientError as e:
```

```
            # If the email template doesn't exist, skip and proceed
            if e.response["Error"]["Code"] == "NotFoundException":
                print(f"Email template '{TEMPLATE_NAME}' does not exist.")
            else:
                print(e)
```

- Untuk API detailnya, lihat DeleteEmailTemplate AWSSDKReferensi Python (Boto3). API

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
        match self
            .client
            .delete_email_template()
            .template_name(TEMPLATE_NAME)
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Email template deleted
successfully.")?,
            Err(e) => {
                return Err(anyhow!("Error deleting email template: {e}"));
            }
        }
```

- Untuk API detailnya, lihat DeleteEmailTemplate AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`GetEmailIdentity`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`GetEmailIdentity`.

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Menentukan apakah alamat email telah diverifikasi.

```
async fn is_verified(client: &Client, email: &str) -> Result<(), Error> {
    let resp = client
        .get_email_identity()
        .email_identity(email)
        .send()
        .await?;

    if resp.verified_for_sending_status() {
        println!("The address is verified");
    } else {
        println!("The address is not verified");
    }

    Ok(())
}
```

- Untuk API detailnya, lihat GetEmailIdentity AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`ListContactLists`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListContactLists`.

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn show_lists(client: &Client) -> Result<(), Error> {
    let resp = client.list_contact_lists().send().await?;

    println!("Contact lists:");

    for list in resp.contact_lists() {
        println!("  {}", list.contact_list_name().unwrap_or_default());
    }

    Ok(())
}
```

- Untuk API detailnya, lihat ListContactLists AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`ListContacts`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`ListContacts`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- Alur kerja buletin

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    /// <summary>
    /// Lists the contacts in the specified contact list.
    /// </summary>
    /// <param name="contactListName">The name of the contact list.</param>
    /// <returns>The list of contacts response from the ListContacts operation.</
returns>
    public async Task<List<Contact>> ListContactsAsync(string contactListName)
    {
        var request = new ListContactsRequest
        {
            ContactListName = contactListName
        };

        try
        {
            var response = await _sesClient.ListContactsAsync(request);
            return response.Contacts;
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The contact list {contactListName} does not
 exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
 later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
```

```
            Console.WriteLine($"An error occurred while listing the contacts:
{ex.Message}");
        }

        return new List<Contact>();
    }
```

- Untuk API detailnya, lihat ListContactsdi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
     ListContactsRequest contactListRequest = ListContactsRequest.builder()
         .contactListName(CONTACT_LIST_NAME)
         .build();

     List<String> contactEmails;
     try {
       ListContactsResponse contactListResponse =
sesClient.listContacts(contactListRequest);

       contactEmails = contactListResponse.contacts().stream()
           .map(Contact::emailAddress)
           .toList();
     } catch (Exception e) {
       // TODO: Remove when listContacts's GET body issue is resolved.
       contactEmails = this.contacts;
     }
```

- Untuk API detailnya, lihat ListContactsdi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
        self.ses_client = ses_client
        self.sleep = sleep


        try:
            contacts_response = self.ses_client.list_contacts(
                ContactListName=CONTACT_LIST_NAME
            )
```

```
        except ClientError as e:
            if e.response["Error"]["Code"] == "NotFoundException":
                print(f"Contact list '{CONTACT_LIST_NAME}' does not exist.")
                return
            else:
                raise e
```

- Untuk API detailnya, lihat ListContacts AWSSDKReferensi Python (Boto3). API

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
async fn show_contacts(client: &Client, list: &str) -> Result<(), Error> {
    let resp = client
        .list_contacts()
        .contact_list_name(list)
        .send()
        .await?;

    println!("Contacts:");

    for contact in resp.contacts() {
        println!("  {}", contact.email_address().unwrap_or_default());
    }

    Ok(())
}
```

- Untuk API detailnya, lihat ListContacts AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Gunakan **`SendEmail`** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan`SendEmail`.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
   /// <summary>
   /// Sends an email with the specified content and options.
   /// </summary>
   /// <param name="fromEmailAddress">The email address to send the email
from.</param>
   /// <param name="toEmailAddresses">The email addresses to send the email
to.</param>
   /// <param name="subject">The subject of the email.</param>
   /// <param name="htmlContent">The HTML content of the email.</param>
   /// <param name="textContent">The text content of the email.</param>
   /// <param name="templateName">The name of the email template to use
(optional).</param>
   /// <param name="templateData">The data to replace placeholders in the email
template (optional).</param>
   /// <param name="contactListName">The name of the contact list for
unsubscribe functionality (optional).</param>
   /// <returns>The MessageId response from the SendEmail operation.</returns>
   public async Task<string> SendEmailAsync(string fromEmailAddress,
List<string> toEmailAddresses, string? subject,
       string? htmlContent, string? textContent, string? templateName = null,
string? templateData = null, string? contactListName = null)
   {
       var request = new SendEmailRequest
       {
```

```
            FromEmailAddress = fromEmailAddress
        };

        if (toEmailAddresses.Any())
        {
            request.Destination = new Destination { ToAddresses =
toEmailAddresses };
        }

        if (!string.IsNullOrEmpty(templateName))
        {
            request.Content = new EmailContent()
            {
                Template = new Template
                {
                    TemplateName = templateName,
                    TemplateData = templateData
                }
            };
        }
        else
        {
            request.Content = new EmailContent
            {
                Simple = new Message
                {
                    Subject = new Content { Data = subject },
                    Body = new Body
                    {
                        Html = new Content { Data = htmlContent },
                        Text = new Content { Data = textContent }
                    }
                }
            };
        }

        if (!string.IsNullOrEmpty(contactListName))
        {
            request.ListManagementOptions = new ListManagementOptions
            {
                ContactListName = contactListName
            };
        }
```

```
        try
        {
            var response = await _sesClient.SendEmailAsync(request);
            return response.MessageId;
        }
        catch (AccountSuspendedException ex)
        {
            Console.WriteLine("The account's ability to send email has been
permanently restricted.");
            Console.WriteLine(ex.Message);
        }
        catch (MailFromDomainNotVerifiedException ex)
        {
            Console.WriteLine("The sending domain is not verified.");
            Console.WriteLine(ex.Message);
        }
        catch (MessageRejectedException ex)
        {
            Console.WriteLine("The message content is invalid.");
            Console.WriteLine(ex.Message);
        }
        catch (SendingPausedException ex)
        {
            Console.WriteLine("The account's ability to send email is currently
paused.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while sending the email:
{ex.Message}");
        }

        return string.Empty;
    }
```

- Untuk API detailnya, lihat SendEmaildi AWS SDK for .NET APIReferensi.

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Mengirim pesan.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.sesv2.model.Body;
import software.amazon.awssdk.services.sesv2.model.Content;
import software.amazon.awssdk.services.sesv2.model.Destination;
import software.amazon.awssdk.services.sesv2.model.EmailContent;
import software.amazon.awssdk.services.sesv2.model.Message;
import software.amazon.awssdk.services.sesv2.model.SendEmailRequest;
import software.amazon.awssdk.services.sesv2.model.SesV2Exception;
import software.amazon.awssdk.services.sesv2.SesV2Client;

/**
 * Before running this AWS SDK for Java (v2) example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class SendEmail {
        public static void main(String[] args) {
                final String usage = """

                                Usage:
                                    <sender> <recipient> <subject>\s

                                Where:
                                    sender - An email address that represents the
 sender.\s
```

```
                                    recipient - An email address that represents
 the recipient.\s
                                    subject - The subject line.\s
                                """;

                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;
                SesV2Client sesv2Client = SesV2Client.builder()
                                .region(region)
                                .build();

                // The HTML body of the email.
                String bodyHTML = "<html>" + "<head></head>" + "<body>" +
 "<h1>Hello!</h1>"
                                + "<p> See the list of customers.</p>" + "</
body>" + "</html>";

                send(sesv2Client, sender, recipient, subject, bodyHTML);
        }

        public static void send(SesV2Client client,
                        String sender,
                        String recipient,
                        String subject,
                        String bodyHTML) {

                Destination destination = Destination.builder()
                                .toAddresses(recipient)
                                .build();

                Content content = Content.builder()
                                .data(bodyHTML)
                                .build();

                Content sub = Content.builder()
                                .data(subject)
```

```
                                .build();

                Body body = Body.builder()
                                .html(content)
                                .build();

                Message msg = Message.builder()
                                .subject(sub)
                                .body(body)
                                .build();

                EmailContent emailContent = EmailContent.builder()
                                .simple(msg)
                                .build();

                SendEmailRequest emailRequest = SendEmailRequest.builder()
                                .destination(destination)
                                .content(emailContent)
                                .fromEmailAddress(sender)
                                .build();

                try {
                        System.out.println("Attempting to send an email through
 Amazon SES "
                                        + "using the AWS SDK for Java...");
                        client.sendEmail(emailRequest);
                        System.out.println("email was sent");

                } catch (SesV2Exception e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

Mengirim pesan menggunakan template.

```
      String coupons = Files.readString(Paths.get("resources/coupon_newsletter/
sample_coupons.json"));
      for (String emailAddress : contactEmails) {
        SendEmailRequest newsletterRequest = SendEmailRequest.builder()
            .destination(Destination.builder().toAddresses(emailAddress).build())
```

```
            .content(EmailContent.builder()
                .template(Template.builder()
                    .templateName(TEMPLATE_NAME)
                    .templateData(coupons)
                    .build())
                .build())
            .fromEmailAddress(this.verifiedEmail)
            .listManagementOptions(ListManagementOptions.builder()
                .contactListName(CONTACT_LIST_NAME)
                .build())
            .build();
        SendEmailResponse newsletterResponse =
 sesClient.sendEmail(newsletterRequest);
        System.out.println("Newsletter sent to " + emailAddress + ": " +
 newsletterResponse.messageId());
     }
```

- Untuk API detailnya, lihat SendEmaildi AWS SDK for Java 2.x APIReferensi.

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Mengirim pesan ke semua anggota daftar kontak.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
```

```
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
        self.ses_client = ses_client
        self.sleep = sleep


                self.ses_client.send_email(
                    FromEmailAddress=self.verified_email,
                    Destination={"ToAddresses": [email]},
                    Content={
                        "Simple": {
                            "Subject": {
                                "Data": "Welcome to the Weekly Coupons
 Newsletter"
                            },
                            "Body": {
                                "Text": {"Data": welcome_text},
                                "Html": {"Data": welcome_html},
                            },
                        }
                    },
                )
                print(f"Welcome email sent to '{email}'.")
```

Mengirim pesan ke semua anggota daftar kontak menggunakan template.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
```

```
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """

    def __init__(self, ses_client, sleep=True):
        self.ses_client = ses_client
        self.sleep = sleep


                self.ses_client.send_email(
                    FromEmailAddress=self.verified_email,
                    Destination={"ToAddresses": [email_address]},
                    Content={
                        "Template": {
                            "TemplateName": TEMPLATE_NAME,
                            "TemplateData": coupon_items,
                        }
                    },
                    ListManagementOptions={"ContactListName": CONTACT_LIST_NAME},
                )
```

- Untuk API detailnya, lihat SendEmail AWSSDKReferensi Python (Boto3). API

Ruby

SDKuntuk Ruby

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
require 'aws-sdk-sesv2'
require_relative 'config' # Recipient and sender email addresses.

# Set up the SESv2 client.
client = Aws::SESV2::Client.new(region: AWS_REGION)

def send_email(client, sender_email, recipient_email)
  response = client.send_email(
    {
      from_email_address: sender_email,
      destination: {
        to_addresses: [recipient_email]
      },
      content: {
        simple: {
          subject: {
            data: 'Test email subject'
          },
          body: {
            text: {
              data: 'Test email body'
            }
          }
        }
      }
    }
  )
  puts "Email sent from #{SENDER_EMAIL} to #{RECIPIENT_EMAIL} with message ID:
 #{response.message_id}"
end

send_email(client, SENDER_EMAIL, RECIPIENT_EMAIL)
```

- Untuk API detailnya, lihat SendEmaildi AWS SDK for Ruby APIReferensi.

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Mengirim pesan ke semua anggota daftar kontak.

```
async fn send_message(
    client: &Client,
    list: &str,
    from: &str,
    subject: &str,
    message: &str,
) -> Result<(), Error> {
    // Get list of email addresses from contact list.
    let resp = client
        .list_contacts()
        .contact_list_name(list)
        .send()
        .await?;

    let contacts = resp.contacts();

    let cs: Vec<String> = contacts
        .iter()
        .map(|i| i.email_address().unwrap_or_default().to_string())
        .collect();

    let mut dest: Destination = Destination::builder().build();
    dest.to_addresses = Some(cs);
    let subject_content = Content::builder()
        .data(subject)
        .charset("UTF-8")
```

```
        .build()
        .expect("building Content");
    let body_content = Content::builder()
        .data(message)
        .charset("UTF-8")
        .build()
        .expect("building Content");
    let body = Body::builder().text(body_content).build();

    let msg = Message::builder()
        .subject(subject_content)
        .body(body)
        .build();

    let email_content = EmailContent::builder().simple(msg).build();

    client
        .send_email()
        .from_email_address(from)
        .destination(dest)
        .content(email_content)
        .send()
        .await?;

    println!("Email sent to list");

    Ok(())
}
```

Mengirim pesan ke semua anggota daftar kontak menggunakan template.

```
            let coupons = std::fs::read_to_string("../resources/newsletter/
sample_coupons.json")
                .unwrap_or_else(|_| r#"{"coupons":[]}"#.to_string());
            let email_content = EmailContent::builder()
                .template(
                    Template::builder()
                        .template_name(TEMPLATE_NAME)
                        .template_data(coupons)
                        .build(),
                )
                .build();
```

```
            match self
                .client
                .send_email()
                .from_email_address(self.verified_email.clone())

.destination(Destination::builder().to_addresses(email.clone()).build())
                .content(email_content)
                .list_management_options(
                    ListManagementOptions::builder()
                        .contact_list_name(CONTACT_LIST_NAME)
                        .build()?,
                )
                .send()
                .await
            {
                Ok(output) => {
                    if let Some(message_id) = output.message_id {
                        writeln!(
                            self.stdout,
                            "Newsletter sent to {} with message ID {}",
                            email, message_id
                        )?;
                    } else {
                        writeln!(self.stdout, "Newsletter sent to {}", email)?;
                    }
                }
                Err(e) => return Err(anyhow!("Error sending newsletter to {}:
{}", email, e)),
            }
```

- Untuk API detailnya, lihat SendEmail AWSSDKuntuk API referensi Rust.

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Skenario untuk Amazon SES API v2 menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menerapkan skenario umum di Amazon SES API v2 dengan AWS SDKs. Skenario ini menunjukkan kepada Anda cara menyelesaikan tugas tertentu dengan

memanggil beberapa fungsi dalam Amazon SES API v2 atau digabungkan dengan yang lain Layanan AWS. Setiap skenario menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode.

Skenario menargetkan tingkat pengalaman menengah untuk membantu Anda memahami tindakan layanan dalam konteks.

Contoh

- Alur kerja Newsletter Amazon SES API v2 lengkap menggunakan AWS SDK

Alur kerja Newsletter Amazon SES API v2 lengkap menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menjalankan alur kerja buletin Amazon SES API v2.

.NET

AWS SDK for .NET

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

Jalankan alur kerja.

```
using System.Diagnostics;
using System.Text.RegularExpressions;
using Amazon.SimpleEmailV2;
using Amazon.SimpleEmailV2.Model;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Logging.Console;
using Microsoft.Extensions.Logging.Debug;

namespace Sesv2Scenario;

public static class NewsletterWorkflow
{
    /*
```

```
     This workflow demonstrates how to use the Amazon Simple Email Service (SES)
v2 to send a coupon newsletter to a list of subscribers.
     The workflow performs the following tasks:

     1. Prepare the application:
        - Create a verified email identity for sending and replying to emails.
        - Create a contact list to store the subscribers' email addresses.
        - Create an email template for the coupon newsletter.

     2. Gather subscriber email addresses:
        - Prompt the user for a base email address.
        - Create 3 variants of the email address using subaddress extensions
(e.g., user+ses-weekly-newsletter-1@example.com).
        - Add each variant as a contact to the contact list.
        - Send a welcome email to each new contact.

     3. Send the coupon newsletter:
        - Retrieve the list of contacts from the contact list.
        - Send the coupon newsletter using the email template to each contact.

     4. Monitor and review:
        - Provide instructions for the user to review the sending activity and
metrics in the AWS console.

     5. Clean up resources:
        - Delete the contact list (which also deletes all contacts within it).
        - Delete the email template.
        - Optionally delete the verified email identity.

   */

   public static SESv2Wrapper _sesv2Wrapper;
   public static string? _baseEmailAddress = null;
   public static string? _verifiedEmail = null;
   private static string _contactListName = "weekly-coupons-newsletter";
   private static string _templateName = "weekly-coupons";
   private static string _subject = "Weekly Coupons Newsletter";
   private static string _htmlContentFile = "coupon-newsletter.html";
   private static string _textContentFile = "coupon-newsletter.txt";
   private static string _htmlWelcomeFile = "welcome.html";
   private static string _textWelcomeFile = "welcome.txt";
   private static string _couponsDataFile = "sample_coupons.json";

   // Relative location of the shared workflow resources folder.
```

```
    private static string _resourcesFilePathLocation = "../../../../../../../
workflows/sesv2_weekly_mailer/resources/";

    public static async Task Main(string[] args)
    {
        // Set up dependency injection for the Amazon service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonSimpleEmailServiceV2>()
                    .AddTransient<SESv2Wrapper>()
            )
            .Build();

        ServicesSetup(host);

        try
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Welcome to the Amazon SES v2 Coupon Newsletter
 Workflow.");
            Console.WriteLine("This workflow demonstrates how to use the Amazon
 Simple Email Service (SES) v2 " +
                              "\r\nto send a coupon newsletter to a list of
 subscribers.");

            // Prepare the application.
            var emailIdentity = await PrepareApplication();

            // Gather subscriber email addresses.
            await GatherSubscriberEmailAddresses(emailIdentity);

            // Send the coupon newsletter.
            await SendCouponNewsletter(emailIdentity);

            // Monitor and review.
            MonitorAndReview(true);
```

```
            // Clean up resources.
            await Cleanup(emailIdentity, true);

            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Amazon SES v2 Coupon Newsletter Workflow is
complete.");
            Console.WriteLine(new string('-', 80));
            Console.WriteLine(new string('-', 80));
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred: {ex.Message}");
        }
    }

    /// <summary>
    /// Populate the services for use within the console application.
    /// </summary>
    /// <param name="host">The services host.</param>
    private static void ServicesSetup(IHost host)
    {
        _sesv2Wrapper = host.Services.GetRequiredService<SESv2Wrapper>();
    }

    /// <summary>
    /// Set up the resources for the workflow.
    /// </summary>
    /// <returns>The email address of the verified identity.</returns>
    public static async Task<string?> PrepareApplication()
    {
        var htmlContent = await File.ReadAllTextAsync(_resourcesFilePathLocation
+ _htmlContentFile);
        var textContent = await File.ReadAllTextAsync(_resourcesFilePathLocation
+ _textContentFile);

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("1. In this step, we will prepare the application:" +
                          "\r\n  - Create a verified email identity for sending
and replying to emails." +
                          "\r\n  - Create a contact list to store the
subscribers' email addresses." +
                          "\r\n  - Create an email template for the coupon
newsletter.\r\n");
```

```
        // Prompt the user for a verified email address.
        while (!IsEmail(_verifiedEmail))
        {
            Console.Write("Enter a verified email address or an email to verify:
");
            _verifiedEmail = Console.ReadLine();
        }

        try
        {
            // Create an email identity and start the verification process.
            await _sesv2Wrapper.CreateEmailIdentityAsync(_verifiedEmail);
            Console.WriteLine($"Identity {_verifiedEmail} created.");
        }
        catch (AlreadyExistsException)
        {
            Console.WriteLine($"Identity {_verifiedEmail} already exists.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error creating email identity: {ex.Message}");
        }

        // Create a contact list.
        try
        {
            await _sesv2Wrapper.CreateContactListAsync(_contactListName);
            Console.WriteLine($"Contact list {_contactListName} created.");
        }
        catch (AlreadyExistsException)
        {
            Console.WriteLine($"Contact list {_contactListName} already
exists.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error creating contact list: {ex.Message}");
        }

        // Create an email template.
        try
        {
            await _sesv2Wrapper.CreateEmailTemplateAsync(_templateName, _subject,
htmlContent, textContent);
```

```
            Console.WriteLine($"Email template {_templateName} created.");
        }
        catch (AlreadyExistsException)
        {
            Console.WriteLine($"Email template {_templateName} already exists.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error creating email template: {ex.Message}");
        }

        return _verifiedEmail;
    }

    /// <summary>
    /// Generate subscriber addresses and send welcome emails.
    /// </summary>
    /// <param name="fromEmailAddress">The verified email address from
 PrepareApplication.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> GatherSubscriberEmailAddresses(string
 fromEmailAddress)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("2. In Step 2, we will gather subscriber email
 addresses:" +
                          "\r\n  - Prompt the user for a base email address." +
                          "\r\n  - Create 3 variants of the email address using
 subaddress extensions (e.g., user+ses-weekly-newsletter-1@example.com)." +
                          "\r\n  - Add each variant as a contact to the contact
 list." +
                          "\r\n  - Send a welcome email to each new contact.\r
\n");

        // Prompt the user for a base email address.
        while (!IsEmail(_baseEmailAddress))
        {
            Console.Write("Enter a base email address (e.g., user@example.com):
 ");
            _baseEmailAddress = Console.ReadLine();
        }

        // Create 3 variants of the email address using +ses-weekly-newsletter-1,
 +ses-weekly-newsletter-2, etc.
```

```
        var baseEmailAddressParts = _baseEmailAddress!.Split("@");
        for (int i = 1; i <= 3; i++)
        {
            string emailAddress = $"{baseEmailAddressParts[0]}+ses-weekly-
newsletter-{i}@{baseEmailAddressParts[1]}";

            try
            {
                // Create a contact with the email address in the contact list.
                await _sesv2Wrapper.CreateContactAsync(emailAddress,
 _contactListName);
                Console.WriteLine($"Contact {emailAddress} added to the
 {_contactListName} contact list.");
            }
            catch (AlreadyExistsException)
            {
                Console.WriteLine($"Contact {emailAddress} already exists in the
 {_contactListName} contact list.");
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Error creating contact {emailAddress}:
 {ex.Message}");
                return false;
            }

            // Send a welcome email to the new contact.
            try
            {
                string subject = "Welcome to the Weekly Coupons Newsletter";
                string htmlContent = await
 File.ReadAllTextAsync(_resourcesFilePathLocation + _htmlWelcomeFile);
                string textContent = await
 File.ReadAllTextAsync(_resourcesFilePathLocation + _textWelcomeFile);

                await _sesv2Wrapper.SendEmailAsync(fromEmailAddress, new
 List<string> { emailAddress }, subject, htmlContent, textContent);
                Console.WriteLine($"Welcome email sent to {emailAddress}.");
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Error sending welcome email to
 {emailAddress}: {ex.Message}");
                return false;
```

```
            }

            // Wait 2 seconds before sending the next email (if the account is in
the SES Sandbox).
            await Task.Delay(2000);
        }

        return true;
    }

    /// <summary>
    ///  Send the coupon newsletter to the subscribers in the contact list.
    /// </summary>
    /// <param name="fromEmailAddress">The verified email address from
PrepareApplication.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> SendCouponNewsletter(string fromEmailAddress)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("3. In this step, we will send the coupon newsletter:"
+
                          "\r\n  - Retrieve the list of contacts from the contact
list." +
                          "\r\n  - Send the coupon newsletter using the email
template to each contact.\r\n");


        // Retrieve the list of contacts from the contact list.
        var contacts = await _sesv2Wrapper.ListContactsAsync(_contactListName);
        if (!contacts.Any())
        {
            Console.WriteLine($"No contacts found in the {_contactListName}
contact list.");
            return false;
        }

        // Load the coupon data from the sample_coupons.json file.
        string couponsData = await
File.ReadAllTextAsync(_resourcesFilePathLocation + _couponsDataFile);

        // Send the coupon newsletter to each contact using the email template.
        try
        {
            foreach (var contact in contacts)
```

```
            {
                // To use the Contact List for list management, send to only one
address at a time.
                await _sesv2Wrapper.SendEmailAsync(fromEmailAddress,
                    new List<string> { contact.EmailAddress },
                    null, null, null, _templateName, couponsData,
_contactListName);
            }

            Console.WriteLine($"Coupon newsletter sent to contact list
{_contactListName}.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error sending coupon newsletter to contact list
{_contactListName}: {ex.Message}");
            return false;
        }

        return true;
    }

    /// <summary>
    /// Provide instructions for monitoring sending activity and metrics.
    /// </summary>
    /// <param name="interactive">True to run in interactive mode.</param>
    /// <returns>True if successful.</returns>
    public static bool MonitorAndReview(bool interactive)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("4. In step 4, we will monitor and review:" +
                          "\r\n  - Provide instructions for the user to review
the sending activity and metrics in the AWS console.\r\n");

        Console.WriteLine("Review your sending activity using the SES Homepage in
the AWS console.");
        Console.WriteLine("Press Enter to open the SES Homepage in your default
browser...");
        if (interactive)
        {
            Console.ReadLine();
            try
            {
                // Open the SES Homepage in the default browser.
```

```
                Process.Start(new ProcessStartInfo
                {
                    FileName = "https://console.aws.amazon.com/ses/home",
                    UseShellExecute = true
                });
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Error opening the SES Homepage:
 {ex.Message}");
                return false;
            }
        }

        Console.WriteLine("Review the sending activity and email metrics, then
 press Enter to continue...");
        if (interactive)
            Console.ReadLine();
        return true;
    }

    /// <summary>
    /// Clean up the resources used in the workflow.
    /// </summary>
    /// <param name="verifiedEmailAddress">The verified email address from
 PrepareApplication.</param>
    /// <param name="interactive">True if interactive.</param>
    /// <returns>Async task.</returns>
    public static async Task<bool> Cleanup(string verifiedEmailAddress, bool
 interactive)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("5. Finally, we clean up resources:" +
                          "\r\n  - Delete the contact list (which also deletes
 all contacts within it)." +
                          "\r\n  - Delete the email template." +
                          "\r\n  - Optionally delete the verified email identity.
\r\n");

        Console.WriteLine("Cleaning up resources...");

        // Delete the contact list (this also deletes all contacts in the list).
        try
        {
```

```
            await _sesv2Wrapper.DeleteContactListAsync(_contactListName);
            Console.WriteLine($"Contact list {_contactListName} deleted.");
        }
        catch (NotFoundException)
        {
            Console.WriteLine($"Contact list {_contactListName} not found.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error deleting contact list {_contactListName}:
{ex.Message}");
            return false;
        }

        // Delete the email template.
        try
        {
            await _sesv2Wrapper.DeleteEmailTemplateAsync(_templateName);
            Console.WriteLine($"Email template {_templateName} deleted.");
        }
        catch (NotFoundException)
        {
            Console.WriteLine($"Email template {_templateName} not found.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error deleting email template {_templateName}:
{ex.Message}");
            return false;
        }

        // Ask the user if they want to delete the email identity.
        var deleteIdentity = !interactive ||
            GetYesNoResponse(
                $"Do you want to delete the email identity
{verifiedEmailAddress}? (y/n) ");
        if (deleteIdentity)
        {
            try
            {
                await
_sesv2Wrapper.DeleteEmailIdentityAsync(verifiedEmailAddress);
                Console.WriteLine($"Email identity {verifiedEmailAddress}
deleted.");
```

```
            }
            catch (NotFoundException)
            {
                Console.WriteLine(
                    $"Email identity {verifiedEmailAddress} not found.");
            }
            catch (Exception ex)
            {
                Console.WriteLine(
                    $"Error deleting email identity {verifiedEmailAddress}:
 {ex.Message}");
                return false;
            }
        }
        else
        {
            Console.WriteLine(
                $"Skipping deletion of email identity {verifiedEmailAddress}.");
        }

        return true;
    }

    /// <summary>
    /// Helper method to get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null && ynResponse.Equals("y",
 StringComparison.InvariantCultureIgnoreCase);
        return response;
    }

    /// <summary>
    /// Simple check to verify a string is an email address.
    /// </summary>
    /// <param name="email">The string to verify.</param>
    /// <returns>True if a valid email.</returns>
    private static bool IsEmail(string? email)
```

```
    {
        if (string.IsNullOrEmpty(email))
            return false;
        return Regex.IsMatch(email, @"^[^@\s]+@[^@\s]+\.[^@\s]+$",
 RegexOptions.IgnoreCase);
    }
}
```

Pembungkus untuk operasi layanan.

```
using System.Net;
using Amazon.SimpleEmailV2;
using Amazon.SimpleEmailV2.Model;

namespace Sesv2Scenario;

/// <summary>
/// Wrapper class for Amazon Simple Email Service (SES) v2 operations.
/// </summary>
public class SESv2Wrapper
{

    private readonly IAmazonSimpleEmailServiceV2 _sesClient;

    /// <summary>
    /// Constructor for the SESv2Wrapper.
    /// </summary>
    /// <param name="sesClient">The injected SES v2 client.</param>
    public SESv2Wrapper(IAmazonSimpleEmailServiceV2 sesClient)
    {
        _sesClient = sesClient;
    }

    /// <summary>
    /// Creates a contact and adds it to the specified contact list.
    /// </summary>
    /// <param name="emailAddress">The email address of the contact.</param>
    /// <param name="contactListName">The name of the contact list.</param>
    /// <returns>The response from the CreateContact operation.</returns>
    public async Task<bool> CreateContactAsync(string emailAddress, string
 contactListName)
    {
```

```
        var request = new CreateContactRequest
        {
            EmailAddress = emailAddress,
            ContactListName = contactListName
        };

        try
        {
            var response = await _sesClient.CreateContactAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AlreadyExistsException ex)
        {
            Console.WriteLine($"Contact with email address {emailAddress} already
exists in the contact list {contactListName}.");
            Console.WriteLine(ex.Message);
            return true;
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The contact list {contactListName} does not
exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while creating the contact:
{ex.Message}");
        }
        return false;
    }

    /// <summary>
    /// Creates a contact list with the specified name.
    /// </summary>
    /// <param name="contactListName">The name of the contact list.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> CreateContactListAsync(string contactListName)
```

```
    {
        var request = new CreateContactListRequest
        {
            ContactListName = contactListName
        };

        try
        {
            var response = await _sesClient.CreateContactListAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AlreadyExistsException ex)
        {
            Console.WriteLine($"Contact list with name {contactListName} already
exists.");
            Console.WriteLine(ex.Message);
            return true;
        }
        catch (LimitExceededException ex)
        {
            Console.WriteLine("The limit for contact lists has been exceeded.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while creating the contact
list: {ex.Message}");
        }
        return false;
    }

    /// <summary>
    /// Creates an email identity (email address or domain) and starts the
verification process.
    /// </summary>
    /// <param name="emailIdentity">The email address or domain to create and
verify.</param>
    /// <returns>The response from the CreateEmailIdentity operation.</returns>
```

```
   public async Task<CreateEmailIdentityResponse>
CreateEmailIdentityAsync(string emailIdentity)
   {
       var request = new CreateEmailIdentityRequest
       {
           EmailIdentity = emailIdentity
       };

       try
       {
           var response = await _sesClient.CreateEmailIdentityAsync(request);
           return response;
       }
       catch (AlreadyExistsException ex)
       {
           Console.WriteLine($"Email identity {emailIdentity} already exists.");
           Console.WriteLine(ex.Message);
           throw;
       }
       catch (ConcurrentModificationException ex)
       {
           Console.WriteLine($"The email identity {emailIdentity} is being
modified by another operation or thread.");
           Console.WriteLine(ex.Message);
           throw;
       }
       catch (LimitExceededException ex)
       {
           Console.WriteLine("The limit for email identities has been
exceeded.");
           Console.WriteLine(ex.Message);
           throw;
       }
       catch (NotFoundException ex)
       {
           Console.WriteLine($"The email identity {emailIdentity} does not
exist.");
           Console.WriteLine(ex.Message);
           throw;
       }
       catch (TooManyRequestsException ex)
       {
           Console.WriteLine("Too many requests were made. Please try again
later.");
```

```
            Console.WriteLine(ex.Message);
            throw;
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while creating the email
identity: {ex.Message}");
            throw;
        }
    }

    /// <summary>
    /// Creates an email template with the specified content.
    /// </summary>
    /// <param name="templateName">The name of the email template.</param>
    /// <param name="subject">The subject of the email template.</param>
    /// <param name="htmlContent">The HTML content of the email template.</param>
    /// <param name="textContent">The text content of the email template.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> CreateEmailTemplateAsync(string templateName, string
subject, string htmlContent, string textContent)
    {
        var request = new CreateEmailTemplateRequest
        {
            TemplateName = templateName,
            TemplateContent = new EmailTemplateContent
            {
                Subject = subject,
                Html = htmlContent,
                Text = textContent
            }
        };

        try
        {
            var response = await _sesClient.CreateEmailTemplateAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AlreadyExistsException ex)
        {
            Console.WriteLine($"Email template with name {templateName} already
exists.");
            Console.WriteLine(ex.Message);
        }
```

```
        catch (LimitExceededException ex)
        {
            Console.WriteLine("The limit for email templates has been
 exceeded.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
 later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while creating the email
 template: {ex.Message}");
        }

        return false;
    }

    /// <summary>
    /// Deletes a contact list and all contacts within it.
    /// </summary>
    /// <param name="contactListName">The name of the contact list to delete.</
param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteContactListAsync(string contactListName)
    {
        var request = new DeleteContactListRequest
        {
            ContactListName = contactListName
        };

        try
        {
            var response = await _sesClient.DeleteContactListAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (ConcurrentModificationException ex)
        {
            Console.WriteLine($"The contact list {contactListName} is being
 modified by another operation or thread.");
            Console.WriteLine(ex.Message);
```

```
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The contact list {contactListName} does not
 exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
 later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while deleting the contact
 list: {ex.Message}");
        }

        return false;
    }

    /// <summary>
    /// Deletes an email identity (email address or domain).
    /// </summary>
    /// <param name="emailIdentity">The email address or domain to delete.</
param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteEmailIdentityAsync(string emailIdentity)
    {
        var request = new DeleteEmailIdentityRequest
        {
            EmailIdentity = emailIdentity
        };

        try
        {
            var response = await _sesClient.DeleteEmailIdentityAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (ConcurrentModificationException ex)
        {
            Console.WriteLine($"The email identity {emailIdentity} is being
 modified by another operation or thread.");
```

```
            Console.WriteLine(ex.Message);
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The email identity {emailIdentity} does not
 exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
 later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while deleting the email
 identity: {ex.Message}");
        }

        return false;
    }

    /// <summary>
    /// Deletes an email template.
    /// </summary>
    /// <param name="templateName">The name of the email template to delete.</
param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteEmailTemplateAsync(string templateName)
    {
        var request = new DeleteEmailTemplateRequest
        {
            TemplateName = templateName
        };

        try
        {
            var response = await _sesClient.DeleteEmailTemplateAsync(request);
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (NotFoundException ex)
        {
```

```
            Console.WriteLine($"The email template {templateName} does not
 exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
 later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while deleting the email
 template: {ex.Message}");
        }

        return false;
    }

    /// <summary>
    /// Lists the contacts in the specified contact list.
    /// </summary>
    /// <param name="contactListName">The name of the contact list.</param>
    /// <returns>The list of contacts response from the ListContacts operation.</
returns>
    public async Task<List<Contact>> ListContactsAsync(string contactListName)
    {
        var request = new ListContactsRequest
        {
            ContactListName = contactListName
        };

        try
        {
            var response = await _sesClient.ListContactsAsync(request);
            return response.Contacts;
        }
        catch (NotFoundException ex)
        {
            Console.WriteLine($"The contact list {contactListName} does not
 exist.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
```

```
        {
            Console.WriteLine("Too many requests were made. Please try again
later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while listing the contacts:
{ex.Message}");
        }

        return new List<Contact>();
    }

    /// <summary>
    /// Sends an email with the specified content and options.
    /// </summary>
    /// <param name="fromEmailAddress">The email address to send the email
from.</param>
    /// <param name="toEmailAddresses">The email addresses to send the email
to.</param>
    /// <param name="subject">The subject of the email.</param>
    /// <param name="htmlContent">The HTML content of the email.</param>
    /// <param name="textContent">The text content of the email.</param>
    /// <param name="templateName">The name of the email template to use
(optional).</param>
    /// <param name="templateData">The data to replace placeholders in the email
template (optional).</param>
    /// <param name="contactListName">The name of the contact list for
unsubscribe functionality (optional).</param>
    /// <returns>The MessageId response from the SendEmail operation.</returns>
    public async Task<string> SendEmailAsync(string fromEmailAddress,
List<string> toEmailAddresses, string? subject,
        string? htmlContent, string? textContent, string? templateName = null,
string? templateData = null, string? contactListName = null)
    {
        var request = new SendEmailRequest
        {
            FromEmailAddress = fromEmailAddress
        };

        if (toEmailAddresses.Any())
        {
```

```
            request.Destination = new Destination { ToAddresses =
toEmailAddresses };
        }

        if (!string.IsNullOrEmpty(templateName))
        {
            request.Content = new EmailContent()
            {
                Template = new Template
                {
                    TemplateName = templateName,
                    TemplateData = templateData
                }
            };
        }
        else
        {
            request.Content = new EmailContent
            {
                Simple = new Message
                {
                    Subject = new Content { Data = subject },
                    Body = new Body
                    {
                        Html = new Content { Data = htmlContent },
                        Text = new Content { Data = textContent }
                    }
                }
            };
        }

        if (!string.IsNullOrEmpty(contactListName))
        {
            request.ListManagementOptions = new ListManagementOptions
            {
                ContactListName = contactListName
            };
        }

        try
        {
            var response = await _sesClient.SendEmailAsync(request);
            return response.MessageId;
        }
```

```
        catch (AccountSuspendedException ex)
        {
            Console.WriteLine("The account's ability to send email has been
 permanently restricted.");
            Console.WriteLine(ex.Message);
        }
        catch (MailFromDomainNotVerifiedException ex)
        {
            Console.WriteLine("The sending domain is not verified.");
            Console.WriteLine(ex.Message);
        }
        catch (MessageRejectedException ex)
        {
            Console.WriteLine("The message content is invalid.");
            Console.WriteLine(ex.Message);
        }
        catch (SendingPausedException ex)
        {
            Console.WriteLine("The account's ability to send email is currently
 paused.");
            Console.WriteLine(ex.Message);
        }
        catch (TooManyRequestsException ex)
        {
            Console.WriteLine("Too many requests were made. Please try again
 later.");
            Console.WriteLine(ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An error occurred while sending the email:
 {ex.Message}");
        }

        return string.Empty;
    }
}
```

- Untuk API detailnya, lihat topik berikut di AWS SDK for .NET APIReferensi.
  - CreateContact
  - CreateContactList

- CreateEmailIdentity
- CreateEmailTemplate
- DeleteContactList
- DeleteEmailIdentity
- DeleteEmailTemplate
- ListContacts
- SendEmail.sederhana
- SendEmail.template

Java

SDKuntuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    try {
      // 2. Create a contact list
      String contactListName = CONTACT_LIST_NAME;
      CreateContactListRequest createContactListRequest =
 CreateContactListRequest.builder()
          .contactListName(contactListName)
          .build();
      sesClient.createContactList(createContactListRequest);
      System.out.println("Contact list created: " + contactListName);
    } catch (AlreadyExistsException e) {
      System.out.println("Contact list already exists, skipping creation: weekly-
coupons-newsletter");
    } catch (LimitExceededException e) {
      System.err.println("Limit for contact lists has been exceeded.");
      throw e;
    } catch (SesV2Exception e) {
      System.err.println("Error creating contact list: " + e.getMessage());
      throw e;
    }
```

```
      try {
        // Create a new contact with the provided email address in the
        CreateContactRequest contactRequest = CreateContactRequest.builder()
            .contactListName(CONTACT_LIST_NAME)
            .emailAddress(emailAddress)
            .build();

        sesClient.createContact(contactRequest);
        contacts.add(emailAddress);

        System.out.println("Contact created: " + emailAddress);

        // Send a welcome email to the new contact
        String welcomeHtml = Files.readString(Paths.get("resources/
coupon_newsletter/welcome.html"));
        String welcomeText = Files.readString(Paths.get("resources/
coupon_newsletter/welcome.txt"));

        SendEmailRequest welcomeEmailRequest = SendEmailRequest.builder()
            .fromEmailAddress(this.verifiedEmail)
            .destination(Destination.builder().toAddresses(emailAddress).build())
            .content(EmailContent.builder()
                .simple(
                    Message.builder()
                        .subject(Content.builder().data("Welcome to the Weekly
 Coupons Newsletter").build())
                        .body(Body.builder()
                            .text(Content.builder().data(welcomeText).build())
                            .html(Content.builder().data(welcomeHtml).build())
                            .build())
                        .build())
                .build())
            .build();
        SendEmailResponse welcomeEmailResponse =
 sesClient.sendEmail(welcomeEmailRequest);
        System.out.println("Welcome email sent: " +
 welcomeEmailResponse.messageId());
      } catch (AlreadyExistsException e) {
        // If the contact already exists, skip this step for that contact and
 proceed
        // with the next contact
        System.out.println("Contact already exists, skipping creation...");
      } catch (Exception e) {
```

```
        System.err.println("Error occurred while processing email address " +
 emailAddress + ": " + e.getMessage());
        throw e;
      }
    }

      ListContactsRequest contactListRequest = ListContactsRequest.builder()
          .contactListName(CONTACT_LIST_NAME)
          .build();

      List<String> contactEmails;
      try {
        ListContactsResponse contactListResponse =
 sesClient.listContacts(contactListRequest);

        contactEmails = contactListResponse.contacts().stream()
            .map(Contact::emailAddress)
            .toList();
      } catch (Exception e) {
        // TODO: Remove when listContacts's GET body issue is resolved.
        contactEmails = this.contacts;
      }


      String coupons = Files.readString(Paths.get("resources/coupon_newsletter/
sample_coupons.json"));
      for (String emailAddress : contactEmails) {
        SendEmailRequest newsletterRequest = SendEmailRequest.builder()
            .destination(Destination.builder().toAddresses(emailAddress).build())
            .content(EmailContent.builder()
                .template(Template.builder()
                    .templateName(TEMPLATE_NAME)
                    .templateData(coupons)
                    .build())
                .build())
            .fromEmailAddress(this.verifiedEmail)
            .listManagementOptions(ListManagementOptions.builder()
                .contactListName(CONTACT_LIST_NAME)
                .build())
            .build();
        SendEmailResponse newsletterResponse =
 sesClient.sendEmail(newsletterRequest);
        System.out.println("Newsletter sent to " + emailAddress + ": " +
 newsletterResponse.messageId());
```

```
      }

    try {
      CreateEmailIdentityRequest createEmailIdentityRequest =
 CreateEmailIdentityRequest.builder()
          .emailIdentity(verifiedEmail)
          .build();
      sesClient.createEmailIdentity(createEmailIdentityRequest);
      System.out.println("Email identity created: " + verifiedEmail);
    } catch (AlreadyExistsException e) {
      System.out.println("Email identity already exists, skipping creation: " +
 verifiedEmail);
    } catch (NotFoundException e) {
      System.err.println("The provided email address is not verified: " +
 verifiedEmail);
      throw e;
    } catch (LimitExceededException e) {
      System.err
          .println("You have reached the limit for email identities. Please
 remove some identities and try again.");
      throw e;
    } catch (SesV2Exception e) {
      System.err.println("Error creating email identity: " + e.getMessage());
      throw e;
    }

    try {
      // Create an email template named "weekly-coupons"
      String newsletterHtml = loadFile("resources/coupon_newsletter/coupon-
newsletter.html");
      String newsletterText = loadFile("resources/coupon_newsletter/coupon-
newsletter.txt");

      CreateEmailTemplateRequest templateRequest =
 CreateEmailTemplateRequest.builder()
          .templateName(TEMPLATE_NAME)
          .templateContent(EmailTemplateContent.builder()
              .subject("Weekly Coupons Newsletter")
              .html(newsletterHtml)
              .text(newsletterText)
              .build())
          .build();

      sesClient.createEmailTemplate(templateRequest);
```

```
      System.out.println("Email template created: " + TEMPLATE_NAME);
    } catch (AlreadyExistsException e) {
      // If the template already exists, skip this step and proceed with the next
      // operation
      System.out.println("Email template already exists, skipping creation...");
    } catch (LimitExceededException e) {
      // If the limit for email templates is exceeded, fail the workflow and
inform
      // the user
      System.err.println("You have reached the limit for email templates. Please
remove some templates and try again.");
      throw e;
    } catch (Exception e) {
      System.err.println("Error occurred while creating email template: " +
e.getMessage());
      throw e;
    }

    try {
      // Delete the contact list
      DeleteContactListRequest deleteContactListRequest =
DeleteContactListRequest.builder()
          .contactListName(CONTACT_LIST_NAME)
          .build();

      sesClient.deleteContactList(deleteContactListRequest);

      System.out.println("Contact list deleted: " + CONTACT_LIST_NAME);
    } catch (NotFoundException e) {
      // If the contact list does not exist, log the error and proceed
      System.out.println("Contact list not found. Skipping deletion...");
    } catch (Exception e) {
      System.err.println("Error occurred while deleting the contact list: " +
e.getMessage());
      e.printStackTrace();
    }

      try {
        // Delete the email identity
        DeleteEmailIdentityRequest deleteIdentityRequest =
DeleteEmailIdentityRequest.builder()
            .emailIdentity(this.verifiedEmail)
            .build();
```

```
        sesClient.deleteEmailIdentity(deleteIdentityRequest);

        System.out.println("Email identity deleted: " + this.verifiedEmail);
      } catch (NotFoundException e) {
        // If the email identity does not exist, log the error and proceed
        System.out.println("Email identity not found. Skipping deletion...");
      } catch (Exception e) {
        System.err.println("Error occurred while deleting the email identity: " +
e.getMessage());
        e.printStackTrace();
      }
    } else {
      System.out.println("Skipping email identity deletion.");
    }

    try {
      // Delete the template
      DeleteEmailTemplateRequest deleteTemplateRequest =
DeleteEmailTemplateRequest.builder()
          .templateName(TEMPLATE_NAME)
          .build();

      sesClient.deleteEmailTemplate(deleteTemplateRequest);

      System.out.println("Email template deleted: " + TEMPLATE_NAME);
    } catch (NotFoundException e) {
      // If the email template does not exist, log the error and proceed
      System.out.println("Email template not found. Skipping deletion...");
    } catch (Exception e) {
      System.err.println("Error occurred while deleting the email template: " +
e.getMessage());
      e.printStackTrace();
    }
```

- Untuk API detailnya, lihat topik berikut di AWS SDK for Java 2.x APIReferensi.
  - CreateContact
  - CreateContactList
  - CreateEmailIdentity
  - CreateEmailTemplate

- DeleteContactList
- DeleteEmailIdentity
- DeleteEmailTemplate
- ListContacts
- SendEmail.sederhana
- SendEmail.template

Python

SDKuntuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
def main():
    """
    The main function that orchestrates the execution of the workflow.
    """
    print(INTRO)
    ses_client = boto3.client("sesv2")
    workflow = SESv2Workflow(ses_client)
    try:
        workflow.prepare_application()
        workflow.gather_subscriber_email_addresses()
        workflow.send_coupon_newsletter()
        workflow.monitor_and_review()
    except ClientError as e:
        print_error(e)
    workflow.clean_up()



class SESv2Workflow:
    """
    A class to manage the SES v2 Coupon Newsletter Workflow.
    """
```

```
   def __init__(self, ses_client, sleep=True):
       self.ses_client = ses_client
       self.sleep = sleep


       try:

self.ses_client.create_contact_list(ContactListName=CONTACT_LIST_NAME)
           print(f"Contact list '{CONTACT_LIST_NAME}' created successfully.")
       except ClientError as e:
           # If the contact list already exists, skip and proceed
           if e.response["Error"]["Code"] == "AlreadyExistsException":
               print(f"Contact list '{CONTACT_LIST_NAME}' already exists.")
           else:
               raise e

           try:
               # Create a new contact
               self.ses_client.create_contact(
                   ContactListName=CONTACT_LIST_NAME, EmailAddress=email
               )
               print(f"Contact with email '{email}' created successfully.")

               # Send the welcome email
               self.ses_client.send_email(
                   FromEmailAddress=self.verified_email,
                   Destination={"ToAddresses": [email]},
                   Content={
                       "Simple": {
                           "Subject": {
                               "Data": "Welcome to the Weekly Coupons
Newsletter"
                           },
                           "Body": {
                               "Text": {"Data": welcome_text},
                               "Html": {"Data": welcome_html},
                           },
                       }
                   },
               )
               print(f"Welcome email sent to '{email}'.")
               if self.sleep:
                   # 1 email per second in sandbox mode, remove in production.
```

```
                    sleep(1.1)
            except ClientError as e:
                # If the contact already exists, skip and proceed
                if e.response["Error"]["Code"] == "AlreadyExistsException":
                    print(f"Contact with email '{email}' already exists.
Skipping...")
                else:
                    raise e

        try:
            contacts_response = self.ses_client.list_contacts(
                ContactListName=CONTACT_LIST_NAME
            )
        except ClientError as e:
            if e.response["Error"]["Code"] == "NotFoundException":
                print(f"Contact list '{CONTACT_LIST_NAME}' does not exist.")
                return
            else:
                raise e

                self.ses_client.send_email(
                    FromEmailAddress=self.verified_email,
                    Destination={"ToAddresses": [email]},
                    Content={
                        "Simple": {
                            "Subject": {
                                "Data": "Welcome to the Weekly Coupons
Newsletter"
                            },
                            "Body": {
                                "Text": {"Data": welcome_text},
                                "Html": {"Data": welcome_html},
                            },
                        }
                    },
                )
                print(f"Welcome email sent to '{email}'.")

                self.ses_client.send_email(
                    FromEmailAddress=self.verified_email,
                    Destination={"ToAddresses": [email_address]},
                    Content={
                        "Template": {
                            "TemplateName": TEMPLATE_NAME,
```

```
                            "TemplateData": coupon_items,
                        }
                    },
                    ListManagementOptions={"ContactListName": CONTACT_LIST_NAME},
                )

        try:

self.ses_client.create_email_identity(EmailIdentity=self.verified_email)
            print(f"Email identity '{self.verified_email}' created
successfully.")
        except ClientError as e:
            # If the email identity already exists, skip and proceed
            if e.response["Error"]["Code"] == "AlreadyExistsException":
                print(f"Email identity '{self.verified_email}' already exists.")
            else:
                raise e

        try:
            template_content = {
                "Subject": "Weekly Coupons Newsletter",
                "Html": load_file_content("coupon-newsletter.html"),
                "Text": load_file_content("coupon-newsletter.txt"),
            }
            self.ses_client.create_email_template(
                TemplateName=TEMPLATE_NAME, TemplateContent=template_content
            )
            print(f"Email template '{TEMPLATE_NAME}' created successfully.")
        except ClientError as e:
            # If the template already exists, skip and proceed
            if e.response["Error"]["Code"] == "AlreadyExistsException":
                print(f"Email template '{TEMPLATE_NAME}' already exists.")
            else:
                raise e

        try:

self.ses_client.delete_contact_list(ContactListName=CONTACT_LIST_NAME)
            print(f"Contact list '{CONTACT_LIST_NAME}' deleted successfully.")
        except ClientError as e:
            # If the contact list doesn't exist, skip and proceed
            if e.response["Error"]["Code"] == "NotFoundException":
                print(f"Contact list '{CONTACT_LIST_NAME}' does not exist.")
            else:
```

```
                print(e)

            try:

self.ses_client.delete_email_identity(EmailIdentity=self.verified_email)
                print(f"Email identity '{self.verified_email}' deleted
successfully.")
            except ClientError as e:
                # If the email identity doesn't exist, skip and proceed
                if e.response["Error"]["Code"] == "NotFoundException":
                    print(f"Email identity '{self.verified_email}' does not
exist.")
                else:
                    print(e)

        try:
            self.ses_client.delete_email_template(TemplateName=TEMPLATE_NAME)
            print(f"Email template '{TEMPLATE_NAME}' deleted successfully.")
        except ClientError as e:
            # If the email template doesn't exist, skip and proceed
            if e.response["Error"]["Code"] == "NotFoundException":
                print(f"Email template '{TEMPLATE_NAME}' does not exist.")
            else:
                print(e)
```

- Untuk API detailnya, lihat topik berikut AWS SDKuntuk Referensi Python (Boto3). API
  - CreateContact
  - CreateContactList
  - CreateEmailIdentity
  - CreateEmailTemplate
  - DeleteContactList
  - DeleteEmailIdentity
  - DeleteEmailTemplate
  - ListContacts
  - SendEmail.sederhana
  - SendEmail.template

Rust

SDKuntuk Rust

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
        match self
            .client
            .create_contact_list()
            .contact_list_name(CONTACT_LIST_NAME)
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Contact list created
successfully.")?,
            Err(e) => match e.into_service_error() {
                CreateContactListError::AlreadyExistsException(_) => {
                    writeln!(
                        self.stdout,
                        "Contact list already exists, skipping creation."
                    )?;
                }
                e => return Err(anyhow!("Error creating contact list: {}", e)),
            },
        }

            match self
                .client
                .create_contact()
                .contact_list_name(CONTACT_LIST_NAME)
                .email_address(email.clone())
                .send()
                .await
            {
                Ok(_) => writeln!(self.stdout, "Contact created for {}", email)?,
                Err(e) => match e.into_service_error() {
                    CreateContactError::AlreadyExistsException(_) => writeln!(
                        self.stdout,
```

```
                        "Contact already exists for {}, skipping creation.",
                        email
                    )?,
                    e => return Err(anyhow!("Error creating contact for {}: {}",
 email, e)),
                },
            }

        let contacts: Vec<Contact> = match self
            .client
            .list_contacts()
            .contact_list_name(CONTACT_LIST_NAME)
            .send()
            .await
        {
            Ok(list_contacts_output) => {
                list_contacts_output.contacts.unwrap().into_iter().collect()
            }
            Err(e) => {
                return Err(anyhow!(
                    "Error retrieving contact list {}: {}",
                    CONTACT_LIST_NAME,
                    e
                ))
            }
        };

            let coupons = std::fs::read_to_string("../resources/newsletter/
sample_coupons.json")
                .unwrap_or_else(|_| r#"{"coupons":[]}"#.to_string());
            let email_content = EmailContent::builder()
                .template(
                    Template::builder()
                        .template_name(TEMPLATE_NAME)
                        .template_data(coupons)
                        .build(),
                )
                .build();

            match self
                .client
                .send_email()
                .from_email_address(self.verified_email.clone())
```

```
.destination(Destination::builder().to_addresses(email.clone()).build())
                .content(email_content)
                .list_management_options(
                    ListManagementOptions::builder()
                        .contact_list_name(CONTACT_LIST_NAME)
                        .build()?,
                )
                .send()
                .await
            {
                Ok(output) => {
                    if let Some(message_id) = output.message_id {
                        writeln!(
                            self.stdout,
                            "Newsletter sent to {} with message ID {}",
                            email, message_id
                        )?;
                    } else {
                        writeln!(self.stdout, "Newsletter sent to {}", email)?;
                    }
                }
                Err(e) => return Err(anyhow!("Error sending newsletter to {}:
{}", email, e)),
            }

        match self
            .client
            .create_email_identity()
            .email_identity(self.verified_email.clone())
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Email identity created
successfully.")?,
            Err(e) => match e.into_service_error() {
                CreateEmailIdentityError::AlreadyExistsException(_) => {
                    writeln!(
                        self.stdout,
                        "Email identity already exists, skipping creation."
                    )?;
                }
                e => return Err(anyhow!("Error creating email identity: {}", e)),
            },
```

```
        }

        let template_html =
            std::fs::read_to_string("../resources/newsletter/coupon-
newsletter.html")
                .unwrap_or_else(|_| "Missing coupon-
newsletter.html".to_string());
        let template_text =
            std::fs::read_to_string("../resources/newsletter/coupon-
newsletter.txt")
                .unwrap_or_else(|_| "Missing coupon-newsletter.txt".to_string());

        // Create the email template
        let template_content = EmailTemplateContent::builder()
            .subject("Weekly Coupons Newsletter")
            .html(template_html)
            .text(template_text)
            .build();

        match self
            .client
            .create_email_template()
            .template_name(TEMPLATE_NAME)
            .template_content(template_content)
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Email template created
 successfully.")?,
            Err(e) => match e.into_service_error() {
                CreateEmailTemplateError::AlreadyExistsException(_) => {
                    writeln!(
                        self.stdout,
                        "Email template already exists, skipping creation."
                    )?;
                }
                e => return Err(anyhow!("Error creating email template: {}", e)),
            },
        }

        match self
            .client
            .delete_contact_list()
            .contact_list_name(CONTACT_LIST_NAME)
```

```
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Contact list deleted
successfully.")?,
            Err(e) => return Err(anyhow!("Error deleting contact list: {e}")),
        }

            match self
                .client
                .delete_email_identity()
                .email_identity(self.verified_email.clone())
                .send()
                .await
            {
                Ok(_) => writeln!(self.stdout, "Email identity deleted
successfully.")?,
                Err(e) => {
                    return Err(anyhow!("Error deleting email identity: {}", e));
                }
            }

        match self
            .client
            .delete_email_template()
            .template_name(TEMPLATE_NAME)
            .send()
            .await
        {
            Ok(_) => writeln!(self.stdout, "Email template deleted
successfully.")?,
            Err(e) => {
                return Err(anyhow!("Error deleting email template: {e}"));
            }
        }
```

- Untuk API detailnya, lihat topik berikut AWS SDKuntuk API referensi Rust.
  - CreateContact
  - CreateContactList
  - CreateEmailIdentity
  - CreateEmailTemplate

- DeleteContactList
- DeleteEmailIdentity
- DeleteEmailTemplate
- ListContacts
- SendEmail.sederhana
- SendEmail.template

Untuk daftar lengkap panduan AWS SDK pengembang dan contoh kode, lihatMenggunakan Amazon SES dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang SDK versi sebelumnya.

Keamanan di Amazon Simple Email Service

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon Simple Email Service, lihat AWS Layanan dalam Lingkup berdasarkan Program Kepatuhan AWS .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, termasuk kepekaan data Anda, persyaratan perusahaan Anda, serta peraturan perundangan yang berlaku

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon Simple Email Service. Topik berikut menunjukkan kepada Anda cara mengonfigurasi Amazon Simple Email Service untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amazon Simple Email Service Anda.

> **Note**
>
> Jika Anda perlu melaporkan penyalahgunaan AWS sumber daya, termasuk spam email dan distribusi malware, jangan gunakan tautan umpan balik di salah satu halaman panduan pengembang ini, karena formulir diterima oleh tim AWS Dokumentasi, bukan AWS Trust & Safety. Sebaliknya, di Bagaimana cara melaporkan penyalahgunaan sumber AWS daya? halaman, ikuti petunjuk untuk menghubungi tim AWS Trust & Safety untuk melaporkan semua jenis AWS penyalahgunaan Amazon.

Konten

- Perlindungan data di Amazon Simple Email Service
- Identity and access management di Amazon SES
- Pencatatan dan Pemantauan di Amazon SES
- Validasi kepatuhan untuk Amazon Simple Email Service
- Ketahanan di Amazon Simple Email Service
- Keamanan infrastruktur di Amazon Simple Email Service
- Menyiapkan VPC endpoint dengan Amazon SES

Perlindungan data di Amazon Simple Email Service

Bagian AWS model tanggung jawab bersama model berlaku untuk perlindungan data di Amazon Simple Email Service. Seperti yang dijelaskan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk menjaga kontrol atas konten Anda yang di-host di infrastruktur ini. Anda juga bertanggung jawab atas konfigurasi keamanan dan tugas manajemen untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat Privasi Data FAQ. Untuk informasi tentang perlindungan data di Eropa, lihat AWS Model Tanggung Jawab Bersama dan posting GDPR blog di AWS Blog Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda untuk melindungi Akun AWS kredensi dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan otentikasi multi-faktor (MFA) dengan setiap akun.
- GunakanSSL/TLSuntuk berkomunikasi dengan AWS sumber daya. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang menggunakan CloudTrail jalur untuk menangkap AWS kegiatan, lihat Bekerja dengan CloudTrail jalan setapak di AWS CloudTrail Panduan Pengguna.
- Gunakan AWS solusi enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.

- Jika Anda memerlukan FIPS 140-3 modul kriptografi yang divalidasi saat mengakses AWS melalui antarmuka baris perintah atauAPI, gunakan FIPS titik akhir. Untuk informasi selengkapnya tentang FIPS titik akhir yang tersedia, lihat Federal Information Processing Standard (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk ketika Anda bekerja dengan Amazon Simple Email Service atau lainnya Layanan AWS menggunakan konsol,API, AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Jika Anda memberikan URL ke server eksternal, kami sangat menyarankan agar Anda tidak menyertakan informasi kredensil dalam URL untuk memvalidasi permintaan Anda ke server tersebut.

Daftar Isi

- Enkripsi data saat istirahat untuk Amazon SES
- Enkripsi bergerak
- Menghapus data pribadi dari Amazon SES

Enkripsi data saat istirahat untuk Amazon SES

Secara default, Amazon SES mengenkripsi semua data saat istirahat. Enkripsi secara default membantu mengurangi overhead operasional dan kompleksitas yang terlibat dalam melindungi data. Enkripsi juga memungkinkan Anda untuk membuat arsip Mail Manager yang memenuhi kepatuhan enkripsi yang ketat dan persyaratan peraturan.

SESmenyediakan opsi enkripsi berikut:

- AWS kunci yang dimiliki - SES menggunakan ini secara default. Anda tidak dapat melihat, mengelola, atau menggunakan AWS memiliki kunci, atau mengaudit penggunaannya. Namun, Anda tidak perlu mengambil tindakan apa pun atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda. Untuk informasi selengkapnya, silakan lihat AWS kunci yang dimiliki di AWS Key Management Service Panduan Pengembang.
- Kunci terkelola pelanggan — SES mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola. Karena Anda memiliki kontrol penuh atas enkripsi, Anda dapat melakukan tugas-tugas seperti:
  - Menetapkan dan memelihara kebijakan utama

- Menetapkan dan memelihara IAM kebijakan dan hibah
- Mengaktifkan dan menonaktifkan kebijakan utama
- Memutar bahan kriptografi kunci
- Menambahkan tanda
- Membuat alias kunci
- Kunci penjadwalan untuk penghapusan

Untuk menggunakan kunci Anda sendiri, pilih kunci yang dikelola pelanggan saat Anda membuat SES sumber daya.

Untuk informasi selengkapnya, lihat Kunci terkelola pelanggan di AWS Key Management Service Panduan Pengembang.

> **Note**
>
> SESsecara otomatis mengaktifkan enkripsi saat istirahat menggunakan AWS kunci yang dimiliki tanpa biaya.
> Namun, AWS KMS dikenakan biaya untuk menggunakan kunci yang dikelola pelanggan.
> Untuk informasi selengkapnya tentang harga, lihat AWS Key Management Service harga.

Buat kunci terkelola pelanggan

Anda dapat membuat kunci terkelola pelanggan simetris dengan menggunakan AWS Management Console, atau AWS KMS APIs.

Untuk membuat kunci terkelola pelanggan simetris

Ikuti langkah-langkah untuk Membuat KMS kunci enkripsi simetris di AWS Key Management Service Panduan Pengembang.

> **Note**
>
> Untuk pengarsipan, kunci Anda harus memenuhi persyaratan berikut:
>
> - Kuncinya harus simetris.
> - Asal bahan utama harus`AWS_KMS`.

- Penggunaan kunci harusENCRYPT_DECRYPT.

Kebijakan utama

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi selengkapnya, lihat Mengelola akses ke kunci terkelola pelanggan di AWS Key Management Service Panduan Pengembang.

Untuk menggunakan kunci yang dikelola pelanggan dengan pengarsipan Mail Manager, kebijakan kunci Anda harus mengizinkan API operasi berikut:

- kms: DescribeKey — Memberikan detail kunci yang dikelola pelanggan yang memungkinkan SES untuk memvalidasi kunci.
- kms: GenerateDataKey — Memungkinkan SES untuk menghasilkan kunci data untuk mengenkripsi data saat istirahat.
- KMS: Decrypt — Memungkinkan SES untuk mendekripsi data yang disimpan sebelum mengembalikannya ke klien. API

Contoh berikut menunjukkan kebijakan kunci tipikal:

```
{
            "Sid": "Allow SES to encrypt/decrypt",
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
```

Untuk informasi selengkapnya, lihat menentukan izin dalam kebijakan, di AWS Key Management Service Panduan Pengembang.

Untuk informasi selengkapnya tentang pemecahan masalah, lihat pemecahan masalah akses kunci, di AWS Key Management Service Panduan Pengembang.

Menentukan kunci terkelola pelanggan untuk pengarsipan Mail Manager

Anda dapat menentukan kunci yang dikelola pelanggan sebagai alternatif untuk menggunakan AWS kunci yang dimiliki. Saat Anda membuat arsip, Anda dapat menentukan kunci data dengan memasukkan KMSkunci ARN, yang digunakan pengarsipan Mail Manager untuk mengenkripsi semua data pelanggan dalam arsip.

- KMSkey ARN — Sebuah pengidentifikasi kunci untuk AWS KMS kunci yang dikelola pelanggan. Masukkan ID kunci, kunciARN, nama alias, atau aliasARN.

Konteks SES enkripsi Amazon

Konteks enkripsi adalah kumpulan opsional pasangan kunci-nilai yang berisi informasi kontekstual tambahan tentang data.

AWS KMS menggunakan konteks enkripsi sebagai data otentikasi tambahan untuk mendukung enkripsi yang diautentikasi. Bila Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda menyertakan konteks enkripsi yang sama dalam permintaan.

> **Note**
>
> Amazon SES tidak mendukung konteks enkripsi untuk pembuatan arsip. Sebaliknya, Anda menggunakan KMS kebijakan IAM atau kebijakan. Misalnya kebijakan, lihatKebijakan pembuatan arsip, nanti di bagian ini.

Konteks SES enkripsi Amazon

SESmenggunakan konteks enkripsi yang sama di semua AWS KMS operasi kriptografi, di mana kuncinya `aws:ses:arn` dan nilainya adalah sumber daya Amazon Resource Name (ARN).

Example

```
"encryptionContext": {
    "aws:ses:arn": "arn:aws:ses:us-west-2:111122223333:ExampleResourceName/
ExampleResourceID"
}
```

Menggunakan konteks enkripsi untuk pemantauan

Saat Anda menggunakan kunci terkelola pelanggan simetris untuk mengenkripsi SES sumber daya Anda, Anda juga dapat menggunakan konteks enkripsi dalam catatan audit dan log untuk mengidentifikasi bagaimana kunci yang dikelola pelanggan digunakan. Konteks enkripsi juga muncul di log yang dihasilkan oleh AWS CloudTrail atau CloudWatch Log Amazon.

Menggunakan konteks enkripsi untuk mengontrol akses ke kunci terkelola pelanggan Anda

Anda dapat menggunakan konteks enkripsi dalam kebijakan dan IAM kebijakan utama `conditions` untuk mengontrol akses ke kunci terkelola pelanggan simetris Anda. Anda juga dapat menggunakan kendala konteks enkripsi dalam hibah.

SESmenggunakan batasan konteks enkripsi dalam hibah untuk mengontrol akses ke kunci yang dikelola pelanggan di akun atau wilayah Anda. Batasan hibah mengharuskan operasi yang diizinkan oleh hibah menggunakan konteks enkripsi yang ditentukan.

Example

Berikut ini adalah contoh pernyataan kebijakan kunci untuk memberikan akses ke kunci yang dikelola pelanggan untuk konteks enkripsi tertentu. Kondisi dalam pernyataan kebijakan ini mengharuskan hibah memiliki batasan konteks enkripsi yang menentukan konteks enkripsi.

```
{
    "Sid": "Enable DescribeKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
     },
     "Action": "kms:DescribeKey",
     "Resource": "*"
},
{
     "Sid": "Enable CreateGrant",
     "Effect": "Allow",
```

```
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
    },
    "Action": "kms:CreateGrant",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:ses:arn": "arn:aws:ses:us-
west-2:111122223333:ExampleResourceName/ExampleResourceID"
         }
    }
}
```

Kebijakan pembuatan arsip

Contoh kebijakan berikut menunjukkan cara mengaktifkan pembuatan arsip. Kebijakan bekerja pada semua aset.

IAMkebijakan

```
{
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ses:CreateArchive",
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "ses.us-east-1.amazonaws.com",
                    "kms:CallerAccount": "012345678910"
                }
            }
        }
```

AWS KMS kebijakan

```
{
            "Sid": "Allow SES to encrypt/decrypt",
            "Effect": "Allow",
            "Principal": {
                "Service": "ses.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
```

Memantau kunci enkripsi Anda untuk Amazon SES

Saat Anda menggunakan AWS KMS kunci terkelola pelanggan dengan SES sumber daya Amazon Anda, Anda dapat menggunakan AWS CloudTrailatau Amazon CloudWatch Logs untuk melacak permintaan yang SES dikirim ke AWS KMS.

Contoh berikut adalah AWS CloudTrail peristiwa untuk`GenerateDataKey`,`Decrypt`, dan `DescribeKey` untuk memantau KMS operasi yang dipanggil oleh SES untuk mengakses data yang dienkripsi oleh kunci yang dikelola pelanggan Anda:

GenerateDataKey

Saat Anda mengaktifkan AWS KMS kunci terkelola pelanggan untuk sumber daya Anda, SES membuat kunci tabel unik. Ini mengirimkan `GenerateDataKey` permintaan ke AWS KMS yang menentukan AWS KMS kunci yang dikelola pelanggan untuk sumber daya.

Saat Anda mengaktifkan AWS KMS kunci terkelola pelanggan untuk sumber daya arsip Manajer Mail Anda, itu akan digunakan `GenerateDataKey` saat mengenkripsi data arsip saat istirahat.

Contoh peristiwa berikut mencatat `GenerateDataKey` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "ses.amazonaws.com"
```

```
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:ses:arn": "arn:aws:ses:us-west-2:111122223333:ExampleResourceName/
ExampleResourceID"
        },
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "57f5dbee-16da-413e-979f-2c4c6663475e"
}
```


Decrypt

Saat Anda mengakses sumber daya terenkripsi, SES panggil `Decrypt` operasi untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.

Contoh peristiwa berikut mencatat `Decrypt` operasi:


```
{
    "eventVersion": "1.08",
```

```
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "ses.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:10:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:ses:arn": "arn:aws:ses:us-west-2:111122223333:ExampleResourceName/
ExampleResourceID"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}
```


DescribeKey

SESmenggunakan `DescribeKey` operasi untuk memverifikasi apakah AWS KMS kunci terkelola pelanggan yang terkait dengan sumber daya Anda ada di akun dan wilayah.

Contoh peristiwa berikut mencatat `DescribeKey` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-04-22T17:02:00Z"
            }
        },
        "invokedBy": "ses.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
```

```
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

Pelajari selengkapnya

Sumber daya berikut memberikan informasi lebih lanjut tentang enkripsi data saat istirahat.

- Untuk informasi lebih lanjut tentang AWS Key Management Service konsep dasar, lihat AWS Key Management Service Panduan Pengembang.
- Untuk informasi lebih lanjut tentang praktik terbaik Keamanan untuk AWS Key Management Service, lihat AWS Key Management Service Panduan Pengembang.

Enkripsi bergerak

Secara default, Amazon SES menggunakan oportunistikTLS. Ini berarti bahwa Amazon SES selalu mencoba untuk membuat koneksi aman ke server email penerima. Jika tidak dapat membuat koneksi yang aman, ia akan mengirimkan pesan yang tidak dienkripsi. Anda dapat mengubah perilaku ini sehingga Amazon SES mengirim pesan ke server email penerima hanya jika dapat membuat koneksi aman. Untuk informasi selengkapnya, lihat Amazon SES dan protokol keamanan.

Menghapus data pribadi dari Amazon SES

Tergantung cara Anda menggunakannya, Amazon SES mungkin menyimpan data tertentu yang dapat dianggap pribadi. Misalnya, untuk mengirim email menggunakan Amazon SES, Anda harus memberikan setidaknya satu identitas terverifikasi (alamat email atau domain). Anda dapat menggunakan konsol Amazon SES atau API Amazon SES untuk menghapus data pribadi ini secara permanen.

Bab ini menyediakan prosedur untuk menghapus berbagai tipe data yang mungkin dianggap pribadi.

Daftar Isi

- Hapus Alamat Email dari Daftar Penekanan Tingkat-Akun
- Hapus Data Tentang Email yang Dikirim Menggunakan Amazon SES

- Hapus Data Tentang Identitas
- Hapus Data Autentikasi Pengirim
- Hapus Data Terkait dengan Aturan Penerimaan
- Hapus Data Terkait dengan Filter Alamat IP
- Hapus Data dalam Templat Email
- Hapus Data dalam Templat Email Verifikasi Kustom
- Hapus Semua Data Pribadi dengan Menutup AWS Akun Anda

Hapus Alamat Email dari Daftar Penekanan Tingkat-Akun

Amazon SES menyertakan daftar penekanan tingkat-akun opsional. Saat Anda mengaktifkan fitur ini, alamat email secara otomatis ditambahkan ke daftar penekanan ketika email tersebut mengakibatkan pentalan atau aduan. Alamat email tetap ada di daftar ini sampai Anda menghapusnya. Untuk informasi selengkapnya tentang daftar penekanan tingkat-akun, lihat Menggunakan daftar SES penindasan tingkat akun Amazon.

Anda dapat menghapus alamat email dari daftar penekanan tingkat akun dengan menggunakan operasi `DeleteSuppressedDestination` dalam API Amazon SES v2. Bagian ini mencakup prosedur untuk menghapus alamat email dengan menggunakan AWS CLI. Untuk informasi selengkapnya tentang menginstal dan mengonfigurasi AWS CLI, lihat Panduan Pengguna AWS Command Line Interface.

Untuk menghapus alamat dari daftar penekanan tingkat-akun dengan menggunakan AWS CLI

- Di baris perintah, masukkan perintah berikut:

  ```
  aws sesv2 delete-suppressed-destination --email-address recipient@example.com
  ```

  Pada perintah sebelumnya, ganti *recipient@example.com* dengan alamat email yang ingin Anda hapus dari daftar penekanan tingkat-akun.

Hapus Data Tentang Email yang Dikirim Menggunakan Amazon SES

Saat Anda menggunakan Amazon SES untuk mengirim email, Anda dapat mengirim informasi tentang email tersebut ke AWS layanan lain. Misalnya, Anda dapat mengirim informasi tentang peristiwa email (seperti pengiriman, pembukaan, dan klik) ke Firehose. Data peristiwa ini biasanya

berisi alamat email Anda dan alamat IP asal email tersebut dikirim. Data tersebut juga berisi alamat email semua penerima yang dikirimi email.

Anda dapat menggunakan Firehose untuk mengalirkan data peristiwa email ke beberapa tujuan—termasuk Amazon Simple Storage Service, Amazon Service, dan Amazon Redshift OpenSearch . Untuk menghapus data ini, Anda harus terlebih dahulu menghentikan streaming data ke Firehose, dan kemudian menghapus data yang telah dialirkan. Untuk menghentikan streaming data peristiwa Amazon SES ke Firehose, Anda harus menghapus tujuan acara Firehose.

Untuk menghapus tujuan peristiwa Firehose dengan menggunakan konsol Amazon SES

1. Buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di bawah Pengiriman Email, pilih Set Konfigurasi.
3. Dalam daftar set konfigurasi, pilih set konfigurasi yang berisi tujuan peristiwa Firehose.
4. Di samping tujuan acara Firehose yang ingin Anda hapus, pilih tombol delete ().
5. Jika perlu, hapus data yang ditulis Firehose ke layanan lain. Untuk informasi selengkapnya, lihat the section called “Hapus Data Peristiwa yang Tersimpan”.

Anda juga dapat menggunakan API Amazon SES untuk menghapus tujuan peristiwa. Prosedur berikut menggunakan AWS Command Line Interface (AWS CLI) untuk berinteraksi dengan Amazon SES API. Anda juga dapat berinteraksi dengan API dengan menggunakan AWS SDK, atau dengan membuat permintaan HTTP secara langsung.

Untuk menghapus tujuan acara Firehose dengan menggunakan AWS CLI

1. Di baris perintah, ketik perintah berikut:

   ```
   aws sesv2 delete-configuration-set-event-destination --configuration-set-
   name configSet \
   --event-destination-name eventDestination
   ```

   Dalam perintah ini, ganti *ConfigSet* dengan nama set konfigurasi yang berisi tujuan peristiwa Firehose. Ganti *EventDestination* dengan nama tujuan acara Firehose.
2. Jika perlu, hapus data yang ditulis Firehose ke layanan lain. Untuk informasi selengkapnya, lihat the section called “Hapus Data Peristiwa yang Tersimpan”.

Hapus Data Peristiwa yang Tersimpan

Untuk informasi selengkapnya tentang menghapus informasi dari AWS layanan lain, lihat dokumen berikut:

- Menghapus Objek dan Bucket di Panduan Pengguna Amazon Simple Storage Service
- Menghapus Domain OpenSearch Layanan di Panduan Pengembang OpenSearch Layanan Amazon
- Menghapus Klaster dalam Panduan Pengelolaan Klaster Amazon Redshift

Anda juga dapat menggunakan Firehose untuk mengalirkan data email ke Splunk, layanan pihak ketiga yang tidak didukung AWS atau dikelola di. AWS Management Console Untuk informasi selengkapnya tentang menghapus data dari Splunk, konsultasikan dengan administrator sistem Anda atau dokumentasi di situs web Splunk.

Hapus Data Tentang Identitas

Identitas mencakup alamat email dan domain yang Anda gunakan untuk mengirim email menggunakan Amazon SES. Di beberapa yurisdiksi, alamat email atau domain mungkin dianggap sebagai data yang dapat diidentifikasi secara pribadi.

Untuk menghapus identitas dengan menggunakan konsol Amazon SES

1. Buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di bawah Pengelolaan Identitas, lakukan salah satu langkah berikut:
   - Pilih Domain jika Anda ingin menghapus domain.
   - Pilih Alamat Email jika Anda ingin menghapus alamat email.
3. Pilih identitas yang ingin Anda hapus, lalu pilih Hapus.
4. Di kotak dialog konfirmasi, pilih Ya, Hapus Identitas.

Anda juga dapat menggunakan API Amazon SES untuk menghapus identitas. Prosedur berikut menggunakan AWS Command Line Interface (AWS CLI) untuk berinteraksi dengan API Amazon SES. Anda juga dapat berinteraksi dengan API dengan menggunakan AWS SDK, atau dengan membuat permintaan HTTP secara langsung.

Untuk menghapus identitas dengan menggunakan AWS CLI

- Di baris perintah, ketik perintah berikut:

```
aws ses delete-identity --identity sender@example.com
```

  Dalam perintah ini, ganti *sender@example.com* dengan identitas yang ingin Anda hapus.

Hapus Data Autentikasi Pengirim

Autentikasi pengirim mengacu pada proses pengonfigurasian Amazon SES sehingga pengguna lain dapat mengirim email atas nama Anda. Untuk mengaktifkan otorisasi pengirim, Anda harus membuat kebijakan, seperti yang dijelaskan di Menggunakan otorisasi pengiriman dengan Amazon SES. Kebijakan ini berisi identitas (milik Anda), selain AWS ID (yang terkait dengan orang atau grup yang mengirim email atas nama Anda). Anda dapat menghapus data pribadi ini dengan mengubah atau menghapus kebijakan autentikasi pengirim. Prosedur berikut menunjukkan cara menghapus kebijakan ini.

Untuk menghapus kebijakan autentikasi pengirim dengan menggunakan konsol Amazon SES

1. Buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di bawah Pengelolaan Identitas, lakukan salah satu langkah berikut:
   - Pilih Domain jika kebijakan autentikasi pengirim yang ingin Anda hapus terkait dengan domain.
   - Pilih Alamat Email jika kebijakan autentikasi pengirim yang ingin Anda hapus terkait dengan alamat email.
3. Di bawah Kebijakan Identitas, pilih kebijakan yang ingin Anda hapus, lalu pilih Hapus Kebijakan.

Anda juga dapat menggunakan API Amazon SES untuk menghapus kebijakan autentikasi pengirim. Prosedur berikut menggunakan AWS Command Line Interface (AWS CLI) untuk berinteraksi dengan Amazon SES API. Anda juga dapat berinteraksi dengan API dengan menggunakan AWS SDK, atau dengan membuat permintaan HTTP secara langsung.

Untuk menghapus kebijakan otentikasi pengirim dengan menggunakan AWS CLI

- Di baris perintah, ketik perintah berikut:

```
aws ses delete-identity-policy --identity example.com --policy-name samplePolicy
```

Dalam perintah ini, ganti *example.com* dengan identitas yang berisi kebijakan autentikasi pengirim. Ganti *samplePolicy* dengan nama kebijakan autentikasi pengirim.

Hapus Data Terkait dengan Aturan Penerimaan

Jika Anda menggunakan Amazon SES untuk menerima email masuk, Anda dapat membuat aturan penerimaan yang diterapkan ke satu atau beberapa identitas (alamat email atau domain). Aturan ini menentukan apa yang dilakukan Amazon SES dengan email masuk yang dikirim ke identitas tertentu.

Untuk menghapus aturan penerimaan dengan menggunakan konsol Amazon SES

1. Buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di bawah Penerimaan Email, pilih Set Aturan.
3. Jika aturan penerimaan adalah bagian dari set aturan aktif, pilih Tampilkan Set Aturan Aktif. Jika tidak, pilih set aturan yang berisi aturan penerimaan yang ingin Anda hapus.
4. Dalam daftar aturan penerimaan, pilih aturan yang ingin Anda hapus.
5. Di menu Tindakan, pilih Edit.
6. Pada kotak dialog konfirmasi, pilih Hapus.

Anda juga dapat menggunakan API Amazon SES untuk menghapus aturan penerimaan. Prosedur berikut menggunakan AWS Command Line Interface (AWS CLI) untuk berinteraksi dengan Amazon SES API. Anda juga dapat berinteraksi dengan API dengan menggunakan AWS SDK, atau dengan membuat permintaan HTTP secara langsung.

Untuk menghapus aturan tanda terima dengan menggunakan AWS CLI

- Di baris perintah, ketik perintah berikut:

```
aws ses delete-receipt-rule --rule-set myRuleSet --rule-name myReceiptRule
```

Dalam perintah ini, ganti *myRuleSet*dengan nama set aturan tanda terima yang berisi aturan tanda terima. Ganti *myReceiptRule*dengan nama aturan tanda terima yang ingin Anda hapus.

Hapus Data Terkait dengan Filter Alamat IP

Jika Anda menggunakan Amazon SES untuk menerima email masuk, Anda dapat membuat filter untuk secara eksplisit menerima atau memblokir pesan yang dikirim dari alamat IP tertentu.

Untuk menghapus filter alamat IP dengan menggunakan konsol Amazon SES

1. Buka konsol Amazon SES di https://console.aws.amazon.com/ses/.
2. Di bawah Penerimaan Email, pilih Filter Alamat IP.
3. Dalam daftar filter alamat IP, pilih filter yang ingin Anda hapus, lalu pilih Hapus.

Anda juga dapat menggunakan API Amazon SES untuk menghapus filter alamat IP. Prosedur berikut menggunakan AWS Command Line Interface (AWS CLI) untuk berinteraksi dengan Amazon SES API. Anda juga dapat berinteraksi dengan API dengan menggunakan AWS SDK, atau dengan membuat permintaan HTTP secara langsung.

Untuk menghapus filter alamat IP dengan menggunakan AWS CLI

- Di baris perintah, ketik perintah berikut:

  ```
  aws ses delete-receipt-filter --filter-name IPfilter
  ```

  Pada perintah ini, ganti *IPfilter* dengan nama filter alamat IP yang ingin Anda hapus.

Hapus Data dalam Templat Email

Jika Anda menggunakan templat email untuk mengirim email, mungkin templat tersebut berisi data pribadi, tergantung pada cara Anda mengonfigurasinya. Misalnya, Anda mungkin telah menambahkan alamat email ke templat yang dapat dihubungi penerima untuk informasi selengkapnya.

Anda hanya dapat menghapus templat email menggunakan API Amazon SES.

Untuk menghapus template email dengan menggunakan AWS CLI

- Di baris perintah, ketik perintah berikut:

  ```
  aws ses delete-template --template-name sampleTemplate
  ```

Pada perintah ini, ganti *sampleTemplate* dengan nama templat email yang ingin Anda hapus.

Hapus Data dalam Templat Email Verifikasi Kustom

Jika Anda menggunakan templat yang dikustomisasi untuk memverifikasi alamat pengiriman email baru, mungkin templat tersebut berisi data pribadi, tergantung pada cara Anda mengonfigurasinya. Misalnya, Anda mungkin telah menambahkan alamat email ke templat email verifikasi yang dapat dihubungi penerima untuk informasi selengkapnya.

Anda hanya dapat menghapus templat email verifikasi kustom dengan menggunakan API Amazon SES.

Untuk menghapus template email verifikasi kustom dengan menggunakan AWS CLI

- Di baris perintah, ketik perintah berikut:

```
aws ses delete-custom-verification-email-template --template-
name verificationEmailTemplate
```

Dalam perintah ini, ganti *verificationEmailTemplate*dengan nama templat email verifikasi khusus yang ingin Anda hapus.

Hapus Semua Data Pribadi dengan Menutup AWS Akun Anda

Anda juga dapat menghapus semua data pribadi yang disimpan di Amazon SES dengan menutup akun AWS Anda. Namun, tindakan ini juga menghapus semua data lain—pribadi atau non-pribadi—yang telah Anda simpan di setiap layanan lainnya. AWS

Ketika Anda menutup AWS akun Anda, data di AWS akun Anda disimpan selama 90 hari. Setelah periode retensi tersebut, maka data tersebut akan dihapus secara permanen dan tidak dapat diubah.

Untuk menutup AWS akun Anda

Petunjuk lengkap tentang cara menutup AWS akun Anda tercakup dalam Tutup AWS akun.

Identity and access management di Amazon SES

Anda dapat menggunakan AWS Identity and Access Management (IAM) dengan Amazon Simple Email Service (Amazon SES) untuk menentukan tindakan SES API yang dapat dilakukan pengguna,

grup, atau peran. (Dalam topik ini kita mengacu pada entitas ini secara kolektif sebagai pengguna.) Anda juga dapat mengontrol alamat email pengguna agar dapat digunakan untuk alamat email "Dari", penerima, dan "Jalur-Kembali".

Misalnya, Anda dapat membuat kebijakan IAM yang memungkinkan pengguna di organisasi Anda untuk mengirim email, namun tidak melakukan tindakan administratif seperti memeriksa statistik pengiriman. Sebagai contoh lain, Anda dapat menulis kebijakan yang memungkinkan pengguna mengirim email melalui SES dari akun Anda, tetapi hanya jika mereka menggunakan alamat “Dari” tertentu.

Untuk menggunakan IAM, Anda menentukan kebijakan IAM, yang merupakan dokumen yang secara eksplisit menentukan izin, dan melampirkan kebijakan untuk pengguna. Untuk mempelajari cara membuat kebijakan IAM, lihat Panduan Pengguna IAM. Selain menerapkan batasan yang Anda tetapkan dalam kebijakan Anda, tidak ada perubahan pada cara pengguna berinteraksi dengan SES atau cara SES melakukan permintaan.

> **Note**
>
> - Jika akun Anda berada di kotak pasir SES, batasannya mencegah penerapan beberapa kebijakan ini - lihat. Minta akses produksi
> - Anda juga dapat mengontrol akses ke SES dengan menggunakan kebijakan otorisasi pengiriman. Sementara kebijakan IAM membatasi apa yang dapat dilakukan pengguna individu, mengirimkan kebijakan otorisasi membatasi bagaimana identitas terverifikasi individu dapat digunakan. Selanjutnya, hanya kebijakan otorisasi pengiriman yang dapat memberikan akses lintas akun. Untuk informasi selengkapnya tentang otorisasi pengiriman, lihat Menggunakan otorisasi pengiriman dengan Amazon SES.

Jika Anda mencari informasi tentang cara menghasilkan kredenal SES SMTP untuk pengguna yang sudah ada, lihat. Memperoleh SES SMTP kredensi Amazon

Membuat Kebijakan IAM untuk Akses ke SES

Bagian ini menjelaskan bagaimana Anda dapat menggunakan kebijakan IAM secara khusus dengan SES. Untuk mempelajari cara membuat kebijakan IAM secara umum, lihat Panduan Pengguna IAM.

Ada tiga alasan Anda mungkin menggunakan IAM dengan SES:

- Untuk membatasi tindakan pengiriman-email.

- Untuk membatasi alamat email "Dari", penerima, dan "Jalur-Kembali" yang dikirim oleh pengguna.
- Untuk mengontrol aspek-aspek umum penggunaan API seperti periode waktu yang pengguna izinkan untuk memanggil API agar dapat digunakan.

Membatasi Tindakan

Untuk mengontrol tindakan SES mana yang dapat dilakukan pengguna, Anda menggunakan `Action` elemen kebijakan IAM. Anda dapat mengatur `Action` elemen ke tindakan SES API apa pun dengan mengawali nama API dengan string huruf kecil. `ses:` Misalnya, Anda dapat mengatur `Action` ke `ses:SendEmail`, `ses:GetSendStatistics`, atau `ses:*` (untuk semua tindakan).

Kemudian, tergantung pada `Action`, tentukan elemen `Resource` sebagai berikut:

Jika **`Action`** elemen hanya mengizinkan akses ke API pengiriman email (yaitu, dan/atau): **`ses:SendEmail`** **`ses:SendRawEmail`**

- Untuk memungkinkan pengguna mengirim dari identitas apa pun di Anda Akun AWS, atur `Resource` ke*
- Untuk membatasi identitas yang diizinkan untuk dikirim oleh pengguna, atur `Resource` ke ARN identitas yang Anda izinkan untuk digunakan pengguna.

Jika **`Action`** elemen mengizinkan akses ke semua API:

- Jika Anda tidak ingin membatasi identitas yang dapat dikirim oleh pengguna, atur `Resource` menjadi *
- Jika Anda ingin membatasi identitas yang diizinkan untuk dikirim oleh pengguna, Anda perlu membuat dua kebijakan (atau dua pernyataan dalam satu kebijakan):
  - Satu dengan `Action` disetel ke daftar eksplisit non-email-sending API yang diizinkan dan `Resource` disetel ke *
  - Satu dengan `Action` diatur ke salah satu API pengiriman-email (`ses:SendEmail` dan/atau `ses:SendRawEmail`), dan `Resource` diatur ke ARN identitas yang Anda izinkan untuk digunakan oleh pengguna.

Untuk daftar tindakan SES yang tersedia, lihat Referensi API Amazon Simple Email Service. Jika pengguna akan menggunakan antarmuka SMTP, Anda harus mengizinkan akses `ses:SendRawEmail` minimum.

Membatasi Alamat Email

Jika Anda ingin membatasi pengguna ke alamat email tertentu, maka Anda dapat menggunakan blok `Condition`. Di blok `Condition`, Anda menentukan syarat dengan menggunakan kunci syarat seperti yang dijelaskan dalam Panduan Pengguna IAM. Dengan menggunakan kunci syarat, Anda dapat mengontrol alamat email berikut:

> **Note**
>
> Kunci syarat alamat email ini hanya berlaku untuk API yang tercantum dalam tabel berikut.

Kunci Syarat	Deskripsi	API
`ses:Recipients`	Membatasi alamat penerima, yang meliputi alamat Kepada:, "CC", dan "BCC".	`SendEmail` , `SendRawEmail`
`ses:FromAddress`	Membatasi alamat "Dari".	`SendEmail` , `SendRawEmail` , `SendBounce`
`ses:FromDisplayName`	Membatasi alamat "Dari" yang digunakan sebagai nama tampilan.	`SendEmail` , `SendRawEmail`
`ses:FeedbackAddress`	Membatasi alamat "Jalur-Kembali", yang merupakan alamat tempat pentalan dan aduan dapat dikirim kepada Anda melalui penerusan umpan balik email. Untuk informasi tentang penerusan umpan balik email, lihat Menerima notifikasi Amazon SES melalui email.	`SendEmail` , `SendRawEmail`

Membatasi berdasarkan versi SES API

Dengan menggunakan `ses:ApiVersion` kunci dalam kondisi, Anda dapat membatasi akses ke SES berdasarkan versi API SES.

> **Note**
>
> Antarmuka SES SMTP menggunakan SES API versi 2 dari. `ses:SendRawEmail`

Membatasi Penggunaan API Umum

Dengan menggunakan kunci AWS-wide dalam kondisi, Anda dapat membatasi akses ke SES berdasarkan aspek-aspek seperti tanggal dan waktu pengguna diizinkan mengakses API. SES hanya mengimplementasikan kunci kebijakan AWS-wide berikut:

- `aws:CurrentTime`
- `aws:EpochTime`
- `aws:SecureTransport`
- `aws:SourceIp`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:UserAgent`
- `aws:VpcSourceIp`

Untuk informasi selengkapnya tentang kunci ini, lihat Panduan Pengguna IAM.

Contoh Kebijakan IAM untuk SES

Topik ini memberikan contoh kebijakan yang mengizinkan pengguna mengakses SES, tetapi hanya dalam kondisi tertentu.

Contoh kebijakan di bagian ini:

- Mengizinkan Akses Penuh ke Semua Tindakan SES
- Mengizinkan Akses ke hanya SES API versi 2
- Mengizinkan Akses Hanya ke Tindakan Pengiriman-Email

- Membatasi Periode Waktu Pengiriman
- Membatasi Alamat Penerima
- Membatasi Alamat "Dari"
- Membatasi Nama Tampilan Pengirim Email
- Membatasi Tujuan dari Umpan Balik Pentalan dan Aduan

Mengizinkan Akses Penuh ke Semua Tindakan SES

Kebijakan berikut memungkinkan pengguna untuk memanggil tindakan SES apa pun.

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

Mengizinkan Akses ke hanya SES API versi 2

Kebijakan berikut memungkinkan pengguna untuk memanggil hanya tindakan SES dari API versi 2.

```
{
                    "Version":"2012-10-17",
                    "Statement":[
                        {
                            "Effect":"Allow",
                            "Action":[
                            "ses:*"
                            ],
                            "Resource":"*",
                            "Condition": {
                                "StringEquals" : {
                                "ses:ApiVersion" : "2"
                                }
```

```
                        }
                    }
                ]
            }
```

Mengizinkan Akses Hanya ke Tindakan Pengiriman-Email

Kebijakan berikut mengizinkan pengguna untuk mengirim email menggunakan SES, tetapi tidak mengizinkan pengguna untuk melakukan tindakan administratif seperti mengakses statistik pengiriman SES.

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

Membatasi Periode Waktu Pengiriman

Kebijakan berikut mengizinkan pengguna untuk memanggil API pengiriman email SES hanya selama bulan September 2018.

```
{
  "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"
        }
      }
    }
  ]
}
```

Membatasi Alamat Penerima

Kebijakan berikut mengizinkan pengguna untuk memanggil API pengiriman email SES, tetapi hanya ke alamat penerima di domain example.com (peka huruf besar/kecil). *`StringLike`*

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource":"*",
      "Condition":{
        "ForAllValues:StringLike":{
          "ses:Recipients":[
            "*@example.com"
          ]
        }
      }
    }
  ]
}
```

Membatasi Alamat "Dari"

Kebijakan berikut mengizinkan pengguna untuk memanggil API pengiriman email SES, tetapi hanya jika alamat “Dari” adalah marketing@example.com.

```
{
```

```
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource":"*",
      "Condition":{
        "StringEquals":{
          "ses:FromAddress":"marketing@example.com"
        }
      }
    }
  ]
}
```

Kebijakan berikut mengizinkan pengguna untuk memanggil SendBounceAPI, tetapi hanya jika alamat "Dari" adalah bounce@example.com.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "ses:SendBounce"
      ],
      "Resource":"*",
      "Condition":{
        "StringEquals":{
          "ses:FromAddress":"bounce@example.com"
        }
      }
    }
  ]
}
```

Membatasi Nama Tampilan Pengirim Email

Kebijakan berikut mengizinkan pengguna untuk memanggil API pengiriman email SES, tetapi hanya jika nama tampilan alamat "Dari" termasuk Pemasaran (*`StringLike`*peka huruf besar/kecil).

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

Membatasi Tujuan dari Umpan Balik Pentalan dan Aduan

Kebijakan berikut mengizinkan pengguna untuk memanggil API pengiriman email SES, tetapi hanya jika “Jalur Kembali” email disetel ke feedback@example.com.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "ses:SendEmail",
        "ses:SendRawEmail"
      ],
      "Resource":"*",
      "Condition":{
        "StringEquals":{
          "ses:FeedbackAddress":"feedback@example.com"
        }
      }
    }
  ]
}
```

AWS kebijakan terkelola untuk Amazon Simple Email Service

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk membuat kebijakan yang dikelola pelanggan IAM yang hanya memberi tim Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di AWS akun Anda. Untuk informasi selengkapnya tentang kebijakan AWS AWS terkelola, lihat kebijakan terkelola di Panduan Pengguna IAM.

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan AWS terkelola untuk mendukung fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan AWS terkelola saat fitur baru diluncurkan atau saat operasi baru tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan ReadOnlyAccess AWS terkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS tambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk melihat daftar dan deskripsi dari kebijakan fungsi tugas, lihat kebijakan yang dikelola AWS untuk fungsi tugas di Panduan Pengguna IAM.

AWS kebijakan terkelola: AmazonSes FullAccess

Anda dapat melampirkan kebijakan `AmazonSESFullAccess` ke identitas IAM Anda. Menyediakan akses penuh ke Amazon SES.

Untuk melihat izin kebijakan ini, lihat AmazonSes FullAccess di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonSes ReadOnlyAccess

Anda dapat melampirkan kebijakan `AmazonSESReadOnlyAccess` ke identitas IAM Anda. Menyediakan akses baca saja ke Amazon SES.

Untuk melihat izin kebijakan ini, lihat AmazonSes ReadOnlyAccess di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonSes ServiceRolePolicy

Anda tidak dapat melampirkan kebijakan `AmazonSESServiceRolePolicy` ke entitas IAM Anda. Kebijakan ini dilampirkan pada peran terkait layanan yang memungkinkan Amazon SES melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk Amazon SES.

Untuk melihat izin kebijakan ini, lihat AmazonSes ServiceRolePolicy di Referensi Kebijakan AWS Terkelola.

Amazon Simple Email Service memperbarui kebijakan AWS terkelola

Lihat detail dan tentang pembaruan kebijakan AWS terkelola untuk Amazon Simple Email Service sejak layanan ini mulai melacak perubahan ini.

Perubahan	Deskripsi	Tanggal
Amazon Simple Email Service menambahkan kebijakan terkelola baru	Amazon Simple Email Service ditambahkan `AmazonSES ServiceRolePolicy` ke peran terkait layanan `AWSServiceRoleForA mazonSES` yang memungkin kan SES untuk melakukan tindakan atas nama Anda	13 Mei 2024
Amazon Simple Email Service memperbarui definisi kebijakan	Amazon Simple Email Service mengklarifikasi entri sebelumnya dalam tabel ini (baris di bawah) menjadi: Amazon Simple Email Service ditambahkan `ses:Batch GetMetricData` ke kebijakan ReadOnlyAccess terkelola AmazonSes—ini akan memberikan akses ke SES API BatchGetMetricData	Apr 30, 2024

Perubahan	Deskripsi	Tanggal
Amazon Simple Email Service memperbarui definisi kebijakan	Amazon Simple Email Service ditambahkan `ses:BatchGet*` ke kebijakan ReadOnlyAccess terkelola AmazonSes—ini akan memberikan akses ke SES API BatchGetMetricData	Feb 16, 2024
Amazon Simple Email Service mengubah dua definisi kebijakan	Amazon Simple Email Service menghapus “via AWS Management Console” dari akhir definisi AmazonSes FullAccess dan AmazonSes ReadOnlyAccess	3 Mei 2023
Amazon Simple Email Service mulai melacak perubahan	Amazon Simple Email Service mulai melacak perubahan pada kebijakan yang AWS dikelola	5 April 2023

Menggunakan peran terkait layanan untuk Amazon SES

Amazon Simple Email Service (SES) AWS Identity and Access Management menggunakan peran terkait layanan (IAM). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Amazon SES. Peran terkait layanan telah ditentukan sebelumnya oleh SES dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan SES lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. SES mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya SES yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran tertaut layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini melindungi sumber daya SES Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, silakan lihat layanan AWS yang bisa digunakan dengan IAM dan carilah layanan yang memiliki opsi Ya di kolom Peran terkait layanan. Pilih Ya bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran terkait layanan untuk Amazon SES

SES menggunakan peran terkait layanan bernama AWSServiceRoleForAmazonSES—Memungkinkan SES mempublikasikan metrik pemantauan CloudWatch dasar Amazon atas nama sumber daya SES Anda.

Peran AWSServiceRoleForAmazonSES terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `ses.amazonaws.com`

Kebijakan izin peran bernama AmazonSes ServiceRolePolicy adalah kebijakan AWS terkelola yang memungkinkan SES menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `cloudwatch:PutMetricData` di `AWS/SES` CloudWatch namespace. Tindakan ini memberikan izin kepada SES untuk memasukkan data metrik ke dalam CloudWatch `AWS/SES` namespace. Untuk informasi selengkapnya tentang metrik SES yang tersedia di CloudWatch, lihatPencatatan dan Pemantauan di Amazon SES.
- Tindakan: `cloudwatch:PutMetricData` di `AWS/SES/MailManager` CloudWatch namespace. Tindakan ini memberikan izin kepada SES untuk memasukkan data metrik ke dalam CloudWatch `AWS/SES/MailManager` namespace. Untuk informasi selengkapnya tentang metrik SES yang tersedia di CloudWatch, lihatPencatatan dan Pemantauan di Amazon SES.
- Tindakan: `cloudwatch:PutMetricData` di `AWS/SES/Addons` CloudWatch namespace. Tindakan ini memberikan izin kepada SES untuk memasukkan data metrik ke dalam CloudWatch `AWS/SES/Addons` namespace. Untuk informasi selengkapnya tentang metrik SES yang tersedia di CloudWatch, lihatPencatatan dan Pemantauan di Amazon SES.

Anda harus mengonfigurasi izin agar pengguna, grup, atau peran Anda membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Amazon SES

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat sumber daya SES di AWS Management Console, API AWS CLI, atau AWS API, SES membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda membuat sumber daya SES, SES membuat peran terkait layanan untuk Anda lagi.

Mengedit peran terkait layanan untuk Amazon SES

SES tidak mengizinkan Anda mengedit peran AWSServiceRoleForAmazonSES terkait layanan. Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM.

Menghapus peran terkait layanan untuk SES

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan dan tidak dipantau atau dipelihara secara aktif. Namun, Anda harus membersihkan peran tertaut layanan terlebih dahulu sebelum dapat menghapusnya secara manual.

Membersihkan peran terkait layanan

Sebelum Anda dapat menggunakan IAM untuk menghapus peran terkait layanan, Anda harus terlebih dahulu menghapus semua sumber daya SES.

> **Note**
>
> Jika layanan SES menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus peran terkait layanan secara manual

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran AWSServiceRoleForAmazonSES terkait layanan. Untuk informasi selengkapnya, lihat Menghapus peran tertaut layanan dalam Panduan Pengguna IAM.

Wilayah yang Didukung untuk peran terkait layanan Amazon SES

SES tidak mendukung penggunaan peran terkait layanan di setiap Wilayah di mana layanan tersedia. Anda dapat menggunakan AWSServiceRoleForAmazonSES peran di Wilayah berikut.

Nama Wilayah	Identitas wilayah	Support di SES
US East (Northern Virginia)	us-east-1	Ya
US East (Ohio)	us-east-2	Ya
Asia Pacific (Sydney)	ap-southeast-2	Ya
Asia Pacific (Tokyo)	ap-northeast-1	Ya
Eropa (Frankfurt)	eu-central-1	Ya
Eropa (Irlandia)	eu-west-1	Ya

Pencatatan dan Pemantauan di Amazon SES

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa Amazon SES dan solusi AWS Anda. AWS menyediakan beberapa alat untuk membantu Anda memantau Amazon SES dan merespons potensi insiden.

- Amazon CloudWatch memantau sumber daya AWS Anda dan aplikasi yang Anda jalankan di AWS secara langsung. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Untuk informasi selengkapnya, lihat Mengambil data acara Amazon SES dari CloudWatch dan Membuat alarm pemantauan reputasi menggunakan CloudWatch.
- AWS CloudTrail merekam panggilan API dan kejadian terkait yang dilakukan oleh atau atas Akun AWS Anda dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat

mengidentifikasi pengguna dan akun yang memanggil AWS, alamat IP asal panggilan dilakukan, dan waktu panggilan terjadi. Untuk informasi selengkapnya, lihat Mencatat panggilan API Amazon SES dengan AWS CloudTrail.

- Peristiwa pengiriman email Amazon SES dapat membantu Anda menyempurnakan strategi pengiriman email Anda. Amazon SES menangkap informasi detail, termasuk jumlah kiriman, penyampaian, membuka, klik, pentalan, aduan, dan penolakan. Untuk informasi selengkapnya, lihat Memantau aktivitas pengiriman.
- Metrik reputasi Amazon SES melacak tingkat pentalan dan aduan untuk akun Anda. Untuk informasi selengkapnya, lihat Pemantauan reputasi pengirim.

Mencatat panggilan API Amazon SES dengan AWS CloudTrail

Amazon SES terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atauAWSlayanan di Amazon SES. CloudTrail menangkap panggilan API untuk Amazon SES sebagai acara. Panggilan yang direkam mencakup panggilan dari konsol Amazon SES dan panggilan kode ke operasi API Amazon SES. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman yang berkelanjutan CloudTrail acara ke bucket Amazon S3, termasuk acara untuk Amazon SES. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol diSejarah acara. Menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon SES, alamat IP tempat permintaan dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail tambahan.

Untuk mempelajari lebih lanjut tentang CloudTrail, termasuk cara mengkonfigurasi dan mengaktifkannya, lihatAWS CloudTrailPanduan Pengguna.

Amazon SES Informasi di CloudTrail

CloudTrail diaktifkan pada AndaAkun AWSsaat Anda membuat akun. Bila aktivitas acara yang didukung terjadi di Amazon SES, aktivitas tersebut dibuat di Amazon SES, aktivitas tersebut dibuat ke CloudTrail Event bersama dengan lainnyaAWSacara layanan diSejarah acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di Akun AWS Anda. Untuk informasi selengkapnya, lihatMelihat Acara dengan CloudTrail Sejarah acara.

Untuk catatan kejadian yang sedang berlangsung di Akun AWS Anda, termasuk kejadian untuk Amazon SES, buatlah jejak. SEBUAHjejakmenyalakan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan

mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengkonfigurasi lainnya.AWSlayanan untuk menganalisis dan menindaklanjuti data peristiwa yang dikumpulkan di CloudTrail log. Untuk informasi selengkapnya, lihat yang berikut:

- Ikhtisar untuk Membuat Jejak
- CloudTrail Layanan dan Integrasi yang Didukung
- Mengonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima CloudTrail File Log dari Beberapa WilayahdanMenerima CloudTrailFile Log dari Beberapa Account

Amazon SES mendukung pencatatan semua tindakan yang tercantum dalamSES API ReferensidanSES API v2 Referensisebagai peristiwa di CloudTrail file log, kecuali yang tercantum dalam kotak catatan di bawah ini:

> **Note**
>
> Amazon SES memberikanacara manajemenkepada CloudTrail. Peristiwa pengelolaan mencakup tindakan terkait membuat dan mengelola sumber daya dalam perangkat Akun AWS. Di Amazon SES, peristiwa pengelolaan mencakup tindakan seperti membuat dan menghapus identitas atau aturan penerimaan.
> Peristiwa manajemen berbeda dari peristiwa data. Peristiwa data adalah peristiwa yang terkait dengan mengakses dan berinteraksi dengan data dalam Akun AWS Anda. Di Amazon SES, peristiwa data mencakup tindakan seperti mengirim email.
> Karena Amazon SES hanya mengirimkan acara manajemen ke CloudTrail, peristiwa-peristiwa berikuttidaktercatat di CloudTrail:
>
> - SendEmail
> - SendRawEmail
> - SendTemplatedEmail
> - SendBulkTemplatedEmail
>
> Anda dapat menggunakan penerbitan peristiwa untuk mencatat peristiwa yang berkaitan dengan pengiriman email. Untuk informasi lebih lanjut, lihat Pantau pengiriman email menggunakan penerbitan SES acara Amazon.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Bahwa permintaan dibuat dengan kredensial pengguna root atau pengguna AWS Identity and Access Management (IAM).
- Bahwa permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Bahwa permintaan dibuat oleh layanan AWS lain.

Untuk informasi lain, lihat Elemen userIdentity CloudTrail .

Contoh: Entri Berkas Log Amazon SES

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Sebuah peristiwa mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukan jejak tumpukan yang diurutkan dari panggilan API publik, sehingga mereka tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan CloudTrail entri log yang menunjukkan`DeleteIdentity`dan`VerifyEmailIdentity`tindakan.

```
{
  "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"
      }
    }
  ]
}
```

Validasi kepatuhan untuk Amazon Simple Email Service

Auditor pihak ke tiga menilai keamanan dan kepatuhan Amazon Simple Email Service sebagai bagian dari beberapa program kepatuhan AWS. Ini mencakup SOC, PCI, FedRAMP, HIPAA, dan sebagainya.

Untuk daftar layanan AWS dalam cakupan program kepatuhan tertentu, lihat Layanan AWS dalam Cakupan berdasarkan Program Kepatuhan. Untuk informasi umum, lihat Program Kepatuhan AWS.

Anda bisa mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi lebih lanjut, lihat Pengunduhan Laporan di dalam AWS Artifact.

Tanggung jawab kepatuhan Anda ketika menggunakan Amazon Simple Email Service ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta peraturan perundang-undangan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu dalam hal kepatuhan:

- Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektural dan menyediakan langkah-langkah untuk men-deploy lingkungan dasar yang berfokus pada keamanan dan kepatuhan pada AWS.
- Merancang Laporan Resmi Keamanan dan Kepatuhan HIPAA – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang patuh-HIPAA.
- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan Developer AWS Config – AWS Config; menilai seberapa patuh konfigurasi sumber daya Anda terhadap praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub – Layanan AWS ini akan menyediakan tampilan komprehensif status keamanan dalam AWS yang akan membantu Anda dalam memeriksa kepatuhan terhadap standar industri keamanan dan praktik terbaik.

Ketahanan di Amazon Simple Email Service

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah menyediakan beberapa Availability Zone yang terpisah dan terisolasi secara fisik, yang terhubung melalui jaringan latensi rendah, throughput tinggi, dan sangat redundan. Dengan Availability Zone, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis mengalami failover antar zona tanpa gangguan. Availability Zone lebih tersedia, memiliki toleransi kesalahan, dan dapat diskalakan dibandingkan dengan satu atau beberapa infrastruktur pusat data tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur Global AWS.

Keamanan infrastruktur di Amazon Simple Email Service

Sebagai layanan terkelola, Amazon Simple Email Service dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan API panggilan yang AWS dipublikasikan untuk mengakses Amazon Simple Email Service melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Transportasi (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju yang sempurna (PFS) seperti (Ephemeral Diffie-Hellman) atau DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan IAM prinsipal. Atau Anda bisa menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Menyiapkan VPC endpoint dengan Amazon SES

Banyak pelanggan Amazon SES memiliki kebijakan perusahaan yang membatasi kemampuan sistem internal mereka untuk terhubung ke internet publik. Kebijakan ini mencegah penggunaan titik akhir Amazon SES publik.

Jika Anda memiliki kebijakan serupa, Anda dapat bekerja dalam batasan ini dengan menggunakan Amazon Virtual Private Cloud. Dengan Amazon VPC, Anda dapat menyebarkan AWS sumber daya ke jaringan virtual yang ada di area terisolasi. AWS Cloud Untuk informasi tentang Amazon VPC selengkapnya, lihat Panduan Pengguna Amazon VPC.

Anda dapat terhubung langsung dari Amazon VPC ke SES melalui VPC Endpoint dengan cara yang aman dan terukur. Saat Anda menggunakan titik akhir VPC antarmuka, ini memberikan postur keamanan yang lebih baik karena Anda tidak perlu membuka firewall lalu lintas keluar serta memberikan manfaat lain menggunakan titik akhir Amazon VPC.

Saat menggunakan VPC Endpoint, lalu lintas ke SES tidak mengirimkan melalui internet dan tidak pernah meninggalkan jaringan Amazon untuk menghubungkan VPC Anda ke SES dengan aman tanpa risiko ketersediaan atau kendala bandwidth pada lalu lintas jaringan Anda. Anda dapat memusatkan SES di seluruh infrastruktur multi-akun Anda dan menyediakannya sebagai layanan ke akun Anda tanpa perlu menggunakan gateway internet.

Batasan

- SES tidak mendukung titik akhir VPC di Availability Zone berikut:`use1-az2`,,,`use1-az3`,`use1-az5`,, `usw1-az2` `usw2-az4`apne2-az4, `cac1-az3` dan. `cac1-az4`
- Titik akhir SMTP yang digunakan dalam VPC dibatasi untuk yang Wilayah AWS saat ini digunakan untuk akun Anda.

Contoh panduan pengaturan SES di Amazon VPC

Prasyarat

Sebelum Anda menyelesaikan prosedur di bagian ini, Anda harus menyelesaikan langkah-langkah berikut:

- Memiliki virtual private cloud (VPC) yang sudah ada atau buat VPC baru. Untuk prosedur, lihat Memulai Amazon VPC.
- Luncurkan instans Amazon EC2 di VPC Anda untuk menguji konektivitas ke titik akhir VPC yang dibuat di langkah selanjutnya. Untuk informasi selengkapnya, lihat VPC default.

Note

Sementara titik akhir VPC untuk SES dapat digunakan dengan sumber daya apa pun, untuk kemudahan metode pengujian, contoh ini akan membuat Anda menggunakan instance EC2 sebagai sumber daya. Karena Amazon EC2 membatasi lalu lintas email melalui port 25 secara default, Anda harus menggunakan port yang berbeda selain TCP 25, seperti TCP 465, 587, 2465, atau 2587.

Menyiapkan SES di Amazon VPC

Proses pengaturan titik akhir VPC untuk digunakan dengan SES terdiri dari beberapa langkah terpisah. Pertama, Anda harus membuat grup keamanan yang memungkinkan instance berkomunikasi dengan port SMTP, lalu membuat titik akhir VPC untuk Amazon SES, dan terakhir, uji koneksi ke titik akhir VPC untuk memastikan bahwa itu dikonfigurasi dengan benar.

Langkah 1: Buat grup keamanan

Pada langkah ini, Anda membuat grup keamanan yang memungkinkan instans Amazon EC2 berkomunikasi dengan titik akhir antarmuka VPC yang akan Anda buat.

Untuk membuat grup keamanan

1. Di panel navigasi konsol Amazon EC2, di Jaringan & Keamanan, pilih Grup Keamanan.
2. Pilih Buat grup keamanan.
3. Di Detail dasar, lakukan hal berikut:
   - Untuk Nama grup keamanan, masukkan nama unik yang mengidentifikasi grup keamanan.
   - Untuk Deskripsi, masukkan beberapa teks yang menjelaskan tujuan grup keamanan.
   - Untuk VPC, pilih VPC yang ingin Anda menggunakan Amazon SES.
4. Di Aturan masuk, pilih Tambahkan aturan.
5. Untuk aturan Inbound baru, lakukan hal berikut:
   - Untuk Tipe, pilih TCP Khusus.
   - Untuk Jangkauan port, masukkan nomor port yang ingin Anda gunakan untuk mengirim email. Anda dapat menggunakan salah satu nomor port berikut:**465**,**587**,**2465**, atau**2587**.
   - Untuk Tipe sumber, pilih Kustom.
   - Untuk Sumber, masukkan rentang CIDR IP pribadi atau ID Grup Keamanan lainnya yang berisi sumber daya yang akan menggunakan titik akhir VPC untuk berkomunikasi dengan layanan SES.
   - (Ulangi langkah 4 - 5 untuk setiap rentang CIDR atau Grup Keamanan yang ingin Anda izinkan aksesnya.)
6. Setelah selesai, pilih Buat grup keamanan.

Langkah 2: Buat titik akhir VPC

Di Amazon VPC, titik akhir VPC memungkinkan Anda menghubungkan VPC ke layanan yang didukung. AWS Dalam contoh ini, Anda mengonfigurasi Amazon VPC sehingga grup keamanan Amazon EC2 Anda dapat terhubung ke Amazon SES.

Untuk membuat VPC endpoint

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di Virtual Private Cloud, pilih Titik akhir.
3. Pilih Create Endpoint untuk membuka halaman Create Endpoint.
4. (Opsional) Di panel pengaturan Endpoint, buat tag di kolom Tag nama.
5. Untuk kategori Layanan, pilih AWS layanan.
6. Di panel Layanan, filter pada smtp di bilah pencarian, lalu pilih tombol radionya.
7. Di panel VPC, klik di dalam bilah pencarian dan pilih VPC dari kotak daftar (lihat). the section called “Prasyarat”
8. Di panel Subnet, pilih Availability Zones dan Subnet ID.

> **Note**
>
> Amazon SES tidak mendukung titik akhir VPC di Availability Zone berikut:`use1-az2`,,,`use1-az3`,,`use1-az5`, `usw1-az2` `usw2-az4`apne2-az4, `cac1-az3` dan. `cac1-az4`

9. Di panel Grup keamanan, pilih grup keamanan yang Anda buat sebelumnya.
10. (Opsional) Di panel Tag, Anda dapat membuat satu atau beberapa tag.
11. Pilih Buat Titik Akhir. Tunggu sekitar 5 menit sementara Amazon VPC membuat titik akhir. Saat titik akhir siap digunakan, nilai di kolom Status berubah menjadi Tersedia.

(Opsional) Langkah 3: Uji koneksi ke titik akhir VPC

Saat Anda menyelesaikan proses konfigurasi titik akhir VPC, Anda dapat menguji koneksi untuk memastikan bahwa titik akhir VPC dikonfigurasi dengan benar. Anda dapat menguji koneksi dengan menggunakan alat baris perintah yang disertakan dengan sebagian besar sistem operasi.

Untuk menguji koneksi ke VPC endpoint

1. Luncurkan instans Amazon EC2 di VPC yang sama tempat Anda baru saja membuat titik akhir VPC email-smtp.

   Untuk informasi tentang menghubungkan ke instans Linux, lihat Connect ke instans Linux Anda di Panduan Pengguna Amazon EC2.

   Untuk informasi tentang menghubungkan ke instans Windows, lihat tutorial Memulai di Panduan Pengguna Amazon EC2.
2. Kirim email pengujian, misalnya, dengan menggunakan antarmuka SES SMTP.

> **Note**
>
> Anda harus memverifikasi alamat email atau domain sebelum Anda dapat mengirim email melalui Amazon SES. Untuk informasi selengkapnya tentang memverifikasi identitas, lihat Membuat dan memverifikasi identitas di Amazon SES.

Pemecahan masalah Amazon SES

Bagian ini berisi topik berikut yang dapat membantu ketika Anda mengalami masalah:

- Untuk informasi tentang masalah verifikasi domain yang mungkin Anda alami, lihat Verifikasi alamat email.
- Solusi untuk masalah terkait DKIM, lihat Mengatasi masalah DKIM di Amazon SES.
- Untuk daftar masalah pengiriman umum yang mungkin Anda alami ketika mengirim email, bersama dengan tindakan perbaikan yang dapat Anda ambil, lihat Masalah pengiriman Amazon SES.
- Untuk deskripsi masalah yang mungkin dilihat penerima saat mereka menerima email yang dikirim melalui Amazon SES, lihat Masalah dengan email yang diterima dari Amazon SES.
- Untuk solusi masalah dengan notifikasi pentalan, aduan, dan pengiriman, lihat Masalah notifikasi Amazon SES.
- Untuk daftar kesalahan yang dapat terjadi saat Anda mengirim email dengan Amazon SES, lihat Kesalahan pengiriman email Amazon SES.
- Untuk tips tentang cara meningkatkan kecepatan pengiriman email ketika Anda membuat beberapa panggilan ke Amazon SES menggunakan API atau antarmuka SMTP, lihat Meningkatkan throughput dengan Amazon SES.
- Untuk solusi untuk masalah umum yang mungkin Anda alami ketika menggunakan Amazon SES melalui antarmuka Protokol Transfer Surat Sederhana (SMTP), serta daftar kode respons SMTP yang dikembalikan Amazon SES, lihat Masalah SMTP Amazon SES.
- Untuk daftar kode kesalahan umum yang dikembalikan oleh Amazon SES API v2, lihat Kesalahan Umum.
- Untuk deskripsi masalah umum yang terkait dengan proses peninjauan pengiriman kami, dan cara menanganinya, lihat FAQ proses peninjauan Pengiriman Amazon SES.
- Untuk diskusi tentang bagaimana daftar Blackhole berbasis DNS (DNSBLs) mempengaruhi pengiriman Anda dengan Amazon SES, lihat FAQ DNS Blackhole List (DNSBL).

Jika Anda memanggil API Amazon SES secara langsung, lihat Referensi API Amazon Simple Email Service untuk kesalahan HTTP yang mungkin Anda terima.

> **Note**
>
> Jika Anda perlu meminta dukungan teknis, jangan gunakan tautan umpan balik di salah satu halaman panduan pengembang ini, karena formulir diterima olehAWSTim dokumentasi, bukanAWSSupport. Sebagai gantinya, diHubungi Kamihalaman, jelajahi berbagai opsi dukungan yang tersedia.

Konten

- Masalah Amazon SES umum
- Verifikasi alamat email
- Mengatasi masalah DKIM di Amazon SES
- Masalah pengiriman Amazon SES
- Masalah dengan email yang diterima dari Amazon SES
- Masalah notifikasi Amazon SES
- Kesalahan pengiriman email Amazon SES
- Meningkatkan throughput dengan Amazon SES
- Masalah SMTP Amazon SES

Masalah Amazon SES umum

Informasi di halaman ini akan menjelaskan dan membantu mendiagnosis masalah yang mungkin Anda alami saat menggunakan Amazon SES.

Perubahan yang saya buat tidak selalu langsung terlihat

Sebagai layanan yang diakses melalui komputer di pusat data di seluruh dunia, Amazon SES menggunakan model komputasi terdistribusi yang disebut eventual consistency. Setiap perubahan yang Anda lakukan di Amazon SES (atau layanan AWS lainnya) membutuhkan waktu agar terlihat dari semua titik akhir yang memungkinkan. Beberapa penundaan dihasilkan dari waktu yang diperlukan untuk mengirim data dari server ke server dan dari wilayah ke wilayah di seluruh dunia. Pada sebagian besar kasus, penundaan ini tidak lebih dari beberapa menit.

Beberapa area di tempat Anda mungkin melihat penundaan yang meliputi:

- Membuat dan mengubah set konfigurasi – Ketika Anda membuat atau mengubah set konfigurasi (misalnya, jika Anda mengaitkan kolam IP khusus dengan set konfigurasi yang ada), mungkin ada penundaan singkat dari waktu yang Anda buat atau ubah waktu perubahan tersebut menjadi aktif.
- Membuat dan mengubah tujuan kejadian – Saat Anda membuat atau mengubah tujuan kejadian (misalnya, untuk memberitahu Amazon SES mengirim data pengiriman email Anda ke layanan AWS lain), mungkin ada penundaan antara waktu yang Anda buat atau ubah tujuan kejadiannya dan kejadian pengiriman email benar-benar masuk di tujuan yang ditentukan.

Verifikasi alamat email

Untuk memverifikasi domain atau alamat email dengan Amazon SES, Anda memulai proses menggunakan konsol Amazon SES atau API Amazon SES. Bagian ini berisi informasi yang dapat membantu menyelesaikan masalah dengan proses verifikasi.

> **Note**
>
> Dalam prosedur berikut, referensi ke catatan DNS dapat merujuk pada catatan CNAME atau TXT tergantung pada bentuk DKIM yang Anda gunakan. Mudah DKIM menggunakan catatan CNAME dan Bring Your Own DKIM (BYODKIM) menggunakan catatan TXT. Prosedur verifikasi terperinci disediakan untuk masing-masingEasy DKIMatauBYODKIM.

Masalah umum verifikasi domain

Jika Anda mencoba memverifikasi domain menggunakan prosedur di the section called “Memverifikasi identitas domain” dan mengalami masalah, tinjau kemungkinan penyebab dan solusi di bawah ini.

- Mencoba memverifikasi domain yang bukan milik Anda - Anda tidak dapat memverifikasi domain yang bukan milik Anda. Misalnya, jika Anda ingin mengirim email melalui Amazon SES dari alamat di domain gmail.com, Anda perlu memverifikasi alamat email tersebut secara khusus. Anda tidak dapat memverifikasi seluruh domain gmail.com.
- Anda mencoba memverifikasi domain pribadi— Anda tidak dapat memverifikasi domain jika data DNS tidak dapat diselesaikan melalui DNS publik.
- Penyedia DNS Anda tidak mengizinkan garis bawah dalam nama rekaman DNS- Sejumlah kecil penyedia DNS tidak mengizinkan Anda untuk menyertakan garis bawah (_) dalam nama

catatan. Namun, garis bawah di nama catatan DKIM diperlukan. Jika penyedia DNS Anda tidak mengizinkan Anda untuk memasukkan garis bawah di nama catatan, kontak tim dukungan pelanggan penyedia untuk mendapatkan bantuan.

- Penyedia DNS Anda menambahkan nama domain ke akhir catatan DNS- Beberapa penyedia DNS secara otomatis menambahkan nama domain Anda ke nama atribut catatan DNS. Misalnya, jika Anda membuat catatan di mana nama atribut_domainkey.example.com, penyedia mungkin menambahkan nama domain, sehingga_domainkey.example.com.example.com). Untuk menghindari duplikasi nama domain, tambahkan titik ke akhir nama domain saat Anda memasukkan catatan DNS. Langkah ini memberi tahu penyedia DNS Anda bahwa tidak perlu menambahkan nama domain ke catatan.
- Penyedia DNS Anda mengubah nilai catatan DNS - Beberapa penyedia secara otomatis mengubah nilai catatan DNS hanya untuk menggunakan huruf kecil. Amazon SES hanya memverifikasi domain Anda ketika mendeteksi catatan verifikasi yang nilai atributnya sama persis dengan nilai yang disediakan Amazon SES ketika Anda memulai proses verifikasi domain. Jika penyedia DNS untuk domain Anda mengubah nilai catatan DNS Anda menjadi hanya menggunakan huruf kecil, hubungi penyedia DNS untuk bantuan tambahan.
- Anda ingin memverifikasi domain yang sama beberapa kali - Anda mungkin perlu memverifikasi domain Anda lebih dari sekali karena Anda mengirim di wilayah yang berbeda, atau karena Anda menggunakan domain yang sama untuk mengirim dari beberapa akun AWS. Jika penyedia DNS tidak mengizinkan Anda memiliki lebih dari satu catatan DNS dengan nama atribut yang sama, Anda mungkin masih dapat memverifikasi dua domain. Jika penyedia DNS mengizinkannya, Anda dapat menetapkan beberapa nilai atribut ke catatan DNS yang sama. Misalnya, jika DNS dikelola oleh Amazon Route 53, Anda dapat mengatur beberapa nilai untuk catatan CNAME yang sama dengan menyelesaikan langkah-langkah berikut:

  1. Di konsol Route 53, pilih catatan CNAME yang Anda buat saat memverifikasi domain di wilayah pertama.
  2. Di kotak Nilai, pergi ke akhir nilai atribut yang ada, dan kemudian tekan Enter.
  3. Tambahkan nilai atribut untuk wilayah tambahan, dan kemudian simpan kumpulan catatan.

  Jika penyedia DNS tidak mengizinkan Anda untuk menetapkan beberapa nilai ke data DNS yang sama, Anda dapat memverifikasi domain sekali dengan_domainkeydalam nama atribut dari catatan DNS, dan lain waktu dengan_domainkeydihapus dari nama atribut. Kelemahan dari solusi ini adalah Anda hanya dapat memverifikasi domain yang sama dua kali.

Memeriksa pengaturan verifikasi domain

Anda dapat memeriksa bahwa catatan DNS verifikasi domain Amazon SES diterbitkan dengan benar ke server DNS Anda dengan menggunakan prosedur berikut. Prosedur ini menggunakan alat nslookup, yang tersedia untuk Windows dan Linux. Di Linux, Anda juga dapat menggunakan dig.

Perintah dalam instruksi ini dijalankan pada Windows 7, dan contoh domain yang kami gunakan adalahses-contoh.comdikonfigurasi dengan Mudah DKIM yang menggunakan catatan CNAME.

Dalam prosedur ini, pertama-tama Anda menemukan server DNS yang melayani domain Anda, lalu kueri server tersebut untuk melihat catatan CNAME. Anda kueri server DNS yang melayani domain karena server tersebut berisi palingup-to-dateinformasi untuk domain Anda, yang dapat membutuhkan waktu lama untuk disebarkan ke server DNS lainnya.

Untuk memverifikasi bahwa catatan CNAME verifikasi domain diterbitkan ke server DNS

1. Temukan server nama untuk domain Anda dengan mengambil langkah-langkah berikut.

   a. Pergi ke baris perintah. Untuk menuju ke baris perintah pada Windows 7, pilih Mulai kemudian ketik cmd. Pada sistem operasi berbasis Linux, buka jendela terminal.

   b. Di perintah prompt, ketik berikut, tempat <domain> adalah domain Anda. Hal ini akan mencantumkan semua server nama yang melayani domain Anda.

      ```
      nslookup -type=NS <domain>
      ```

      Jika domain Anda adalah ses-example.com, perintah ini akan terlihat seperti:

      ```
      nslookup -type=NS ses-example.com
      ```

      Output perintah akan mencantumkan server nama yang melayani domain Anda. Anda akan meng-kueri salah satu server ini di langkah berikutnya.

2. Verifikasi bahwa catatan CNAME diterbitkan dengan benar dengan mengambil langkah-langkah berikut. Perlu diingat bahwa Amazon SES menghasilkan tiga catatan CNAME untuk otentikasi Easy DKIM, jadi ulangi prosedur berikut untuk masing-masing dari ketiganya.

   a. Di perintah prompt, ketik berikut, tempat<random string>adalah SES dihasilkan nama CNAME,<domain>adalah domain Anda, dan<name server>adalah salah satu server nama yang Anda temukan di langkah 1.

```
nslookup -type=CNAME <random string>_domainkey.<domain> <name server>
```

Di dalam kamises-contoh.comcontoh, jika server nama yang kami temukan di langkah 1 disebutns1.name-server.net, dan<random string>dihasilkan oleh SES adalah`4hzwn5lmznmmjyl2pqf2agr3uzzzzxyz`, kita akan mengetik berikut ini:

```
nslookup -type=CNAME 4hzwn5lmznmmjyl2pqf2agr3uzzzzxyz_domainkey.ses-example.com
 ns1.name-server.net
```

b. Dalam output perintah, verifikasi bahwa string yang mengikuti`canonical name =`cocok dengan nilai CNAME yang Anda lihat saat memilih domain di daftar Identitas konsol Amazon SES.

   Dalam contoh kita, kita mencari catatan CNAME di bawah`4hzwn5lmznmmjyl2pqf2agr3uzzzzxyz_domainkey.ses-example.com`dengan nilai`4hzwn5lmznmmjyl2pqf2agr3uzzzzxyz.dkim.amazonses.com`. Jika catatan diterbitkan dengan benar, kami mengharapkan perintah untuk memiliki output sebagai berikut:

```
4hzwn5lmznmmjyl2pqf2agr3uzzzzxyz_domainkey.ses-example.com canonical name =
 "4hzwn5lmznmmjyl2pqf2agr3uzzzzxyz.dkim.amazonses.com"
```

Masalah umum verifikasi email

- Email verifikasi tidak diterima - Jika Anda menyelesaikan prosedur di Memverifikasi identitas alamat email namun Anda tidak menerima email verifikasi dalam beberapa menit, selesaikan langkah-langkah berikut:
  - Periksa folder spam atau email sampah untuk alamat email yang sedang Anda coba verifikasi.
  - Konfirmasikan bahwa alamat yang Anda coba verifikasi dapat menerima email. Menggunakan alamat email terpisah (seperti alamat email pribadi Anda), kirim email percobaan ke alamat yang ingin Anda verifikasi.
  - Periksa daftar alamat terverifikasi di konsol Amazon SES. Pastikan tidak ada kesalahan dalam alamat email yang sedang Anda coba verifikasi.

Mengatasi masalah DKIM di Amazon SES

Bagian ini berisi daftar beberapa masalah yang mungkin Anda alami ketika Anda mengonfigurasi autentikasi DKIM di Amazon SES. Jika Anda mencoba mengatur DKIM dan Anda menghadapi masalah, tinjau kemungkinan penyebab dan solusi di bawah ini.

Anda berhasil mengatur DKIM, tetapi pesan Anda tidak ditandatangani DKIM

Jika Anda menggunakan Easy DKIM atau BYODKIM untuk mengonfigurasi DKIM untuk domain, namun pesan yang Anda kirim tidak ditandatangani oleh DKIM, lakukan hal berikut:

- Pastikan bahwa DKIM diaktifkan untuk identitas yang sesuai. Untuk mengaktifkan DKIM pada identitas di konsol Amazon SES, pilih domain email di daftar Identitas. Pada halaman detail untuk domain, memperluas DKIM, lalu pilih Aktifkan untuk mengaktifkan DKIM.
- Pastikan Anda tidak mengirim dari alamat email terverifikasi pada domain yang sama. Jika Anda mengatur DKIM untuk domain, maka semua pesan yang Anda kirim dari domain tersebut ditandatangani DKIM, kecuali untuk alamat email yang Anda verifikasi satu per satu. Alamat email yang diverifikasi secara satu per satu menggunakan pengaturan terpisah. Misalnya, jika Anda mengonfigurasi DKIM untuk domain contoh.com, dan Anda secara terpisah memverifikasi alamat email mary@example.com (tetapi tidak mengonfigurasi DKIM untuk alamat), lalu email yang Anda kirim dari mary@example.com dikirim tanpa autentikasi DKIM. Anda dapat mengatasi masalah ini dengan menghapus identitas alamat email dari daftar identitas untuk akun Anda.
- Jika Anda menggunakan identitas yang sama di lebih dari satu wilayah AWS, Anda harus mengonfigurasi DKIM untuk setiap wilayah secara terpisah. Demikian pula, jika Anda menggunakan domain yang sama dengan lebih dari satu AWS, Anda harus mengonfigurasi DKIM untuk setiap akun. Jika Anda menghapus data DNS yang diperlukan untuk wilayah atau akun tertentu, Amazon SES menonaktifkan DKIM masuk wilayah atau akun tersebut. Jika penandatanganan DKIM dinonaktifkan, Amazon SES mengirimkan notifikasi melalui email.

Detail DKIM domain Anda di tampilan konsol Amazon SES DKIM: menunggu verifikasi pengirim... Status Verifikasi DKIM: verifikasi tertunda.

Jika Anda menyelesaikan prosedur di Easy DKIM atau BYODKIM - Bawa DKIM Anda Sendiri untuk mengonfigurasi DKIM pada domain, tetapi konsol Amazon SES masih menunjukkan bahwa verifikasi DKIM tertunda, lakukan hal berikut:

- Tunggu hingga 72 jam. Dalam kasus yang jarang terjadi, dapat mengambil waktu untuk catatan DNS untuk menjadi terlihat oleh Amazon SES.

- Konfirmasi jika catatan CNAME (untuk Easy DKIM) atau data TXT (untuk BYODKIM) menggunakan nama yang benar. Beberapa penyedia DNS secara otomatis menambahkan nama domain ke catatan yang Anda buat. Misalnya, jika Anda membuat catatan dengan nama `example._domainkey.example.com`, penyedia DNS Anda dapat menambahkan nama domain Anda ke bagian akhir string ini, sehingga menyebabkan `example._domainkey.example.com.example.com`. Untuk informasi lebih lanjut, lihat dokumentasi untuk penyedia DNS Anda.

Anda menerima email dari Amazon SES yang mengatakan bahwa pengaturan DKIM Anda telah (atau akan) dicabut.

Ini berarti bahwa Amazon SES tidak dapat lagi menemukan catatan CNAME yang diperlukan (jika Anda menggunakan Easy DKIM) atau catatan TXT yang diperlukan (jika Anda menggunakan BYODKIM) pada server DNS Anda. Notifikasi email akan memberitahu Anda tentang durasi waktu ketika Anda harus menerbitkan ulang data DNS sebelum status pengaturan DKIM dicabut dan penandatanganan DKIM dinonaktifkan. Jika pengaturan DKIM Anda dicabut, Anda harus memulai prosedur pengaturan DKIM dari awal.

Ketika mencoba untuk mengatur BYODKIM, proses verifikasi DKIM gagal.

Pastikan kunci privat Anda menggunakan format yang tepat. Kunci privat harus dalam format PKCS #1 atau PKCS #8 dan menggunakan enkripsi RSA 1024 atau 2048 bit. Selain itu, kunci privat harus dikodekan dalam bentuk base64.

Saat menyiapkan pengaturan BYODKIM, Anda menerima **`BadRequestException`** kesalahan ketika Anda mencoba untuk menentukan kunci publik untuk domain.

Jika Anda menerima kesalahan `BadRequestException`, lakukan hal berikut ini:

- Pastikan bahwa pemilih yang Anda tentukan untuk kunci publik berisi setidaknya 1 dan kurang dari atau sama dengan 63 karakter alfanumerik. Pemilih tidak dapat menyertakan titik atau simbol atau tanda baca lainnya.
- Pastikan bahwa Anda telah menghapus garis header dan footer dari kunci publik, dan bahwa Anda telah menghapus semua jeda baris dari kunci publik.

Bila menggunakan Easy DKIM, server DNS Anda berhasil mengembalikan catatan Amazon SES DKIM CNAME, tapi kembali **`SERVFAIL`** untuk rekaman TXT verifikasi domain.

Penyedia DNS Anda mungkin tidak dapat mengalihkan catatan CNAME. Amazon SES dan kueri ISP untuk catatan TXT. Untuk memenuhi spesifikasi DKIM, server DNS Anda harus dapat menanggapi kueri catatan TXT serta kueri catatan CNAME. Jika penyedia DNS Anda tidak

dapat menanggapi kueri catatan TXT, cara alternatifnya adalah menggunakan Route 53 sebagai penyedia hosting DNS Anda.

Email Anda berisi dua tanda tangan DKIM

Tanda tangan tambahan DKIM, yang berisi `d=amazonses.com`, secara otomatis ditambahkan oleh Amazon SES. Anda dapat mengabaikannya.

Masalah pengiriman Amazon SES

Setelah Anda berhasil membuat permintaan ke Amazon SES, pesan Anda akan sering langsung dikirim. Di lain waktu, mungkin ada penundaan singkat. Bagaimanapun, Anda boleh yakin bahwa email Anda akan dikirim.

Namun, ketika Amazon SES mengirim pesan Anda, beberapa faktor dapat mencegahnya berhasil terkirim, dan dalam beberapa kasus Anda akan menyadari bahwa pengiriman gagal hanya ketika pesan yang Anda kirim tidak sampai. Gunakan proses berikut untuk mengatasi situasi ini.

Jika email tidak sampai, coba hal berikut:

- Verifikasi bahwa Anda membuat `SendEmail` atau meminta `SendRawEmail` untuk email yang dipermasalahkan dan bahwa Anda menerima respons yang berhasil. Jika Anda membuat permintaan ini secara terprogram, periksa log perangkat lunak Anda guna memastikan bahwa program tersebut membuat permintaan dan menerima respons yang berhasil.
- Baca artikel blog Tiga tempat biasanya email Anda tertunda saat mengirim melalui SES karena masalah sebenarnya mungkin adalah penundaan, alih-alih tidak terkirim.
- Periksa alamat email pengirim (alamat "Dari") untuk memverifikasi bahwa alamat tersebut valid. Periksa juga alamat Jalur Kembali, yang menjadi tujuan pesan pentalan dikirim. Jika email Anda terpental, akan ada pesan kesalahan di sana yang menjelaskan pentalan tersebut.
- Periksa AWS Service Health Dashboard untuk mengonfirmasi bahwa tidak ada masalah yang diketahui dengan Amazon SES.
- Hubungi penerima email atau ISP penerima. Verifikasi bahwa penerima menggunakan alamat email yang benar, dan tanyakan jika ada masalah pengiriman yang diketahui dengan ISP penerima. Selain itu, tentukan jika email memang masuk tetapi difilter sebagai spam.
- Jika Anda telah mendaftar untuk Rencana AWS Support berbayar, maka Anda dapat membuka kasus dukungan teknis yang baru. Dalam korespondensi Anda dengan kami, harap berikan alamat penerima yang relevan, bersama dengan ID permintaan atau ID pesan yang dikembalikan dari respons `SendEmail` atau `SendRawEmail`.

- Tunggu untuk melihat jika masalah sebenarnya adalah penundaan, bukan kegagalan pengiriman permanen. Untuk melawan spammer, beberapa ISP untuk sementara menolak pesan masuk dari server email pengiriman yang tidak dikenal. Proses ini, disebut greylisting, dapat menyebabkan keterlambatan pengiriman. Amazon SES akan mencoba lagi pesan-pesan ini. Jika greylisting merupakan masalah, ISP dapat menerima email pada salah satu upaya coba lagi ini.
- Bahkan ketika Anda memiliki minat terbaik pelanggan, Anda mungkin masih menghadapi situasi yang berdampak pada kemampuan pengiriman pesan Anda. Lihatthe section called "Mempertahankan reputasi pengirim yang positif"untuk membantu memastikan komunikasi email Anda menjangkau audiens yang diinginkan.

Masalah dengan email yang diterima dari Amazon SES

Bagian ini membahas beberapa masalah umum yang mungkin Anda temui saat menerima email yang dikirim dari Amazon SES.

Klien email menampilkan "dikirim melalui amazonses.com" sebagai sumber email

Beberapa klien email menampilkan "melalui" domain saat domain pengirim tidak cocok dengan domain asal email tersebut dikirim (dalam hal ini, amazonses.com). Untuk informasi lebih lanjut, lihat, Info tambahan di sebelah nama pengirim di situs web Support Gmail. Atau, Anda dapat mengaturDomainKeysSurat yang Diidentifikasi(DKIM). Bila Anda mengautentikasi email menggunakan DKIM, klien email biasanya tidak menampilkan "melalui" domain karena tanda tangan DKIM menunjukkan bahwa email tersebut berasal dari domain yang diklaimnya. Untuk informasi tentang pengaturan DKIM, lihat Mengautentikasi Email dengan DKIM di Amazon SES.

> **Note**
>
> Jika Anda telah menerima spam atau pesan email lain yang tidak diminta dari pengguna SES, gunakan alat pelaporan spam di klien email Anda, dan ikuti langkah-langkah untuk melaporkan penyalahgunaan email SES yang tercantum di bawahMenghubungi Kami.

Pesan berisi karakter kacau atau omong kosong

Jika pesan Anda menyertakan karakter yang tidak ada dalam set karakter ASCII (seperti karakter Latin beraksen, karakter Cina, atau karakter Arab), Anda harus mengode karakter tersebut menggunakan pengodean karakter HTML. Anda dapat menggunakan alat berbasis web untuk

mengodekan karakter dalam email Anda, seperti Pengonversi Karakter HTML di situs web Email On Acid.

Atau, Anda dapat menyusun pesan MIME sendiri. Dalam pesan MIME, Anda dapat menentukan pesan harus menggunakan pengodean UTF-8. Ketika Anda menggunakan pengodean UTF-8, Anda dapat menggunakan karakter non-ASCII langsung dalam pesan Anda. Setelah selesai membuat pesan MIME, Anda dapat mengirimkannya menggunakanSendRawEmailAPI atauSendMailAPI v2.

Salah satu penyebab umum masalah ini adalah fitur Smart Quotes dari Microsoft Word. Jika Anda sering menyalin konten dari Word dan menempelkannya ke email, Anda mungkin mengalami masalah ini. Fitur Smart Quotes menggantikan karakter kutipan langsung ("...") dengan karakter kutip keriting (“...”). karakter kutip keriting bukan karakter ASCII standar. Akibatnya, karakter tersebut mungkin ditampilkan di beberapa klien email sebagai "??" atau sebagai grup karakter seperti "â€œ". Untuk memperbaiki masalah ini, Anda dapat menonaktifkan fitur Smart Quotes di Word. Atau, Anda dapat menggunakanSendRawEmailsolusi dari paragraf sebelumnya. Untuk mempelajari cara menonaktifkan fitur ini, lihat Kutipan quotes di Word di situs web Microsoft Office Support.

Masalah notifikasi Amazon SES

Jika Anda menghadapi masalah dengan notifikasi pentalan, aduan, atau pengiriman, tinjau kemungkinan penyebab dan solusi di bawah ini.

- Anda menerima notifikasi pentalan melalui Amazon SNS, tetapi Anda tidak tahu penerima mana yang sesuai dengan notifikasi—Pada masa yang akan datang, untuk mengaitkan notifikasi pentalan dengan penerima tertentu, Anda mempunyai pilihan berikut:
  - Sejak Amazon SES tidak mempertahankan ID pesan kustom yang telah Anda tambahkan, simpan pemetaan antara pengenal dan ID pesan Amazon SES bahwa Amazon SES kembali kepada Anda ketika menerima email.
  - Dalam setiap panggilan ke Amazon SES, kirim ke penerima tunggal, daripada mengirim pesan tunggal ke beberapa penerima.
  - Anda dapat mengaktifkan penerusan umpan balik melalui email, yang akan meneruskan pesan pentalan penuh kepada Anda.
- Anda menerima notifikasi aduan atau pengiriman melalui Amazon SNS atau penerusan umpan balik email, namun Anda tidak tahu penerima mana yang sesuai dengan notifikasi—Beberapa ISP menyunting alamat email penerima yang mengeluh sebelum meneruskan notifikasi aduan

ke Amazon SES. Untuk memungkinkan Anda untuk menemukan alamat email penerima, pilihan terbaik Anda adalah untuk menyimpan pemetaan Anda sendiri antara pengenal dan ID pesan Amazon SES bahwa Amazon SES kembali kepada Anda ketika menerima email. Perhatikan bahwa Amazon SES tidak mempertahankan ID pesan kustom yang Anda tambahkan.

- Anda ingin mengatur notifikasi untuk pergi ke topik Amazon SNS yang Anda tidak memiliki—Pemilik topik tersebut harus mengonfigurasi kebijakan akses Amazon SNS yang memungkinkan akun Anda untuk menghubungi tindakan `SNS:Publish` pada topik mereka. Untuk informasi tentang bagaimana mengontrol akses ke topik Amazon SNS Anda melalui penggunaan kebijakan IAM, lihat Mengelola Akses ke Topik Amazon SNS di Panduan Developer Amazon Simple Notification Service.

Kesalahan pengiriman email Amazon SES

Topik ini meninjau tipe kesalahan pengiriman email tertentu yang mungkin Anda alami ketika Anda mengirim email melalui Amazon SES. Jika Anda mencoba untuk mengirim email melalui Amazon SES dan panggilan ke Amazon SES gagal, Amazon SES mengembalikan pesan kesalahan untuk aplikasi Anda dan tidak mengirim email. Cara yang Anda amati pada pesan kesalahan ini tergantung pada cara yang Anda sebut Amazon SES.

- Jika Anda memanggil API Amazon SES secara langsung, tindakan Kueri akan mengembalikan kesalahan. Kesalahan mungkin `MessageRejected` atau salah satu kesalahan yang ditentukan dalam topik Kesalahan Umum dari Referensi API Amazon Simple Email Service.
- Jika Anda memanggil Amazon SES menggunakan AWS SDK yang menggunakan bahasa pemrograman yang mendukung pengecualian, Amazon SES mungkin melemparkan pengecualian. Tipe pengecualian tergantung pada SDK dan kesalahan. Misalnya, pengecualian bisa menjadi Amazon SES `MessageRejectedException` (nama sebenarnya dapat beragam tergantung pada SDK) atau pengecualian AWS umum. Terlepas dari tipe pengecualian, tipe kesalahan dan pesan kesalahan dalam pengecualian akan memberi Anda lebih banyak informasi.
- Jika Anda memanggil Amazon SES melalui antarmuka SMTP, cara Anda mengalami kesalahan tergantung pada aplikasi. Beberapa aplikasi mungkin menampilkan pesan kesalahan tertentu, dan aplikasi lainnya mungkin tidak. Untuk daftar kode respons SMTP yang mengembalikan Amazon SES, lihat Kode respons SMTP dikembalikan oleh Amazon SES.

> **Note**
>
> Ketika panggilan Anda ke Amazon SES untuk mengirim email gagal, Anda tidak ditagih untuk email itu.

Berikut ini adalah tipe masalah spesifik Amazon SES yang dapat menyebabkan Amazon SES untuk mengembalikan kesalahan ketika Anda mencoba untuk mengirim email. Kesalahan ini adalah tambahan ke kesalahan AWS umum seperti `MalformedQueryString` seperti yang ditentukan dalam topik Kesalahan Umum dari Referensi API Amazon Simple Email Service.

- Alamat email tidak diverifikasi. Identitas berikut gagal dalam wilayah check in wilayah: identity1, identity2, identity3—Anda mencoba mengirim email dari alamat email atau domain yang belum diverifikasi dengan Amazon SES. Kesalahan ini dapat diterapkan untuk alamat "Dari", "Sumber", "Pengirim", atau "Jalur-Kembali". Jika akun Anda masih dalam sandbox Amazon SES, Anda juga harus memverifikasi setiap alamat email penerima kecuali penerima yang disediakan oleh simulator kotak surat Amazon SES. Jika Amazon SES tidak mampu menunjukkan semua identitas yang gagal, pesan kesalahan berakhir dengan elipsis.

  > **Note**
  >
  > Amazon SES memiliki titik akhir di beberapa Wilayah AWS, dan status verifikasi alamat email terpisah untuk setiap Wilayah AWS. Anda harus menyelesaikan proses verifikasi untuk setiap pengirim di Wilayah AWS yang ingin Anda gunakan.

- Akun dijeda—Kemampuan akun Anda untuk mengirim email dijeda. Anda masih dapat mengakses konsol Amazon SES dan melakukan sebagian besar operasi. Namun, jika Anda mencoba untuk mengirim email, Anda menerima pesan ini.

  Jika kami menjeda kemampuan akun Anda untuk mengirim email, kami secara otomatis mengirimkan notifikasi ke alamat email yang terkait dengan Akun AWS Anda. Untuk informasi lebih lanjut, lihat the section called “FAQ proses peninjauan Pengiriman”.

- Throttling—Aplikasi Anda mungkin mencoba mengirim terlalu banyak pesan per detik, atau Anda mungkin telah mengirim terlalu banyak email selama 24 jam terakhir. Dalam kasus ini, pesan kesalahan mungkin mirip dengan contoh berikut:

- Kuota pesan harian terlampaui—Anda telah mengirim jumlah pesan maksimum yang diizinkan dalam waktu 24 jam. Jika Anda telah melampaui kuota harian, Anda harus menunggu hingga waktu 24 jam berikutnya sebelum dapat mengirim email lainnya.
- Laju pengiriman maksimum terlampaui—Anda mencoba untuk mengirim lebih banyak email per detik daripada yang diizinkan oleh laju pengiriman maksimum Anda. Jika Anda telah melampaui laju pengiriman Anda, Anda dapat terus mengirim email, tetapi akan perlu untuk mengurangi laju pengiriman Anda. Untuk informasi lebih lanjut, lihat Cara menangani kesalahan "Throttling - laju pengiriman maksimum terlampaui" di Blog Pesan dan Target AWS.
- Laju pengiriman SMTP SigV2 maksimum terlampaui—Anda mencoba mengirim pesan menggunakan kredensial SMTP yang dibuat sebelum tanggal 10 Januari 2019; kredensial SMTP Anda dibuat menggunakan versi Tanda Tangan AWS. Demi keamanan, Anda harus menghapus kredensial yang Anda buat sebelum tanggal ini, dan menggantinya dengan kredensial yang lebih baru. Anda dapat menghapus kredensial menggunakan konsol IAM. Untuk informasi lebih lanjut, lihat the section called “Memperoleh SMTP kredensi” untuk membuat kredensial.

Anda harus memantau aktivitas pengiriman Anda secara teratur untuk melihat seberapa dekat Anda dengan kuota pengiriman Anda. Untuk informasi lebih lanjut, lihat Memantau kuota SES pengiriman Amazon Anda. Untuk informasi umum tentang kuota pengiriman, lihat Mengelola batas pengiriman Amazon SES Anda. Untuk informasi tentang cara meningkatkan kuota pengiriman Anda, lihat Meningkatkan kuota pengiriman Amazon SES Anda.

> **Important**
>
> Jika teks kesalahan yang menjelaskan kesalahan throttling tidak terkait dengan Anda yang melampaui kuota harian Anda atau laju pengiriman maksimum, maka mungkin ada masalah di seluruh sistem yang menyebabkan kemampuan pengiriman berkurang. Untuk informasi tentang status layanan, buka AWS Service Health Dashboard.

- Tidak ada penerima yang ditentukan—Tidak ada penerima yang diberikan.
- Ada karakter non-ASCII di alamat email—String alamat email harus berupa ASCII 7-bit. Jika Anda ingin mengirim ke atau dari alamat email yang berisi karakter Unicode di bagian domain alamat, Anda harus mengodekan domain menggunakan Punycode. Punycode tidak diizinkan di bagian lokal dari alamat email (bagian sebelum tanda @) atau dalam nama "friendly from". Jika Anda ingin menggunakan karakter Unicode dalam nama "friendly from", Anda harus mengodekan nama "friendly from" menggunakan sintaksis kata yang dikodekan MIME, seperti yang dijelaskan dalam

Mengirim email mentah menggunakan Amazon SES API v2. Untuk informasi selengkapnya tentang Punycode, lihat RFC 3492.

- Domain surat FROM tidak diverifikasi—Amazon SES tidak dapat membaca catatan MX yang diperlukan untuk menggunakan domain MAIL FROM yang ditentukan. Untuk informasi pengaturan domain MAIL FROM kustom, lihat Menggunakan domain MAIL FROM kustom.
- Set konfigurasi tidak ada—Set konfigurasi yang Anda tentukan tidak ada. Set konfigurasi adalah parameter opsional yang Anda gunakan untuk memublikasikan kejadian pengiriman email. Untuk informasi lebih lanjut, lihat Pantau pengiriman email menggunakan penerbitan SES acara Amazon.

Meningkatkan throughput dengan Amazon SES

Ketika Anda mengirim email, Anda dapat menghubungi Amazon SES sesering yang diizinkan oleh laju pengiriman maksimum Anda. (Untuk informasi selengkapnya tentang laju pengiriman maksimum Anda, lihat Mengelola batas pengiriman Amazon SES Anda.) Namun, setiap panggilan ke Amazon SES membutuhkan waktu untuk menyelesaikannya.

Jika Anda membuat beberapa panggilan ke Amazon SES menggunakan API Amazon SES atau antarmuka SMTP, Anda mungkin ingin mempertimbangkan tips berikut untuk membantu Anda meningkatkan throughput Anda:

- Ukur performa Anda saat ini untuk mengidentifikasi kemacetan—Tes performa yang mungkin melibatkan pengiriman beberapa uji email secepat mungkin dalam kode loop dalam aplikasi Anda. Ukur latensi bolak-balik setiap permintaan `SendEmail`. Kemudian, secara bertahap meluncurkan instans tambahan dari aplikasi pada mesin yang sama, dan melihat setiap dampak pada latensi jaringan. Anda mungkin juga ingin menjalankan tes ini pada beberapa mesin dan pada jaringan yang berbeda untuk membantu menentukan kemungkinan kemacetan pada sumber daya mesin atau kemacetan pada jaringan yang ada.
- (Hanya API) Pertimbangkan untuk menggunakan koneksi HTTP yang persisten—Daripada menimbulkan overhead dalam membangun koneksi HTTP baru yang terpisah untuk setiap permintaan API, gunakan koneksi HTTP yang persisten. Artinya, gunakan kembali koneksi HTTP yang sama untuk beberapa permintaan API.
- Pertimbangkan untuk menggunakan beberapa utas—Ketika aplikasi menggunakan utas tunggal, kode aplikasi memanggil API Amazon SES kemudian bersamaan menunggu respons API. Mengirim email biasanya merupakan operasi I/O yang terikat, dan melakukan pekerjaan dari beberapa utas untuk memberikan throughput yang lebih baik. Anda dapat mengirim secara bersamaan menggunakan banyak utas eksekusi yang Anda inginkan.

- Pertimbangkan untuk menggunakan beberapa proses—Menggunakan beberapa proses dapat membantu meningkatkan throughput Anda karena Anda akan memiliki lebih banyak koneksi aktif bersamaan ke Amazon SES. Misalnya, Anda dapat memisah email yang Anda maksudkan ke beberapa bucket, kemudian menjalankan beberapa instans dari skrip pengiriman email secara bersamaan.
- Pertimbangkan untuk menggunakan relay surat lokal—Aplikasi Anda dapat dengan cepat mengirimkan pesan ke server email lokal Anda, yang kemudian dapat membantu untuk mem-buffer pesan dan secara asinkron mengirimkan mereka ke Amazon SES. Beberapa server mail mendukung pengiriman konkurensi, yang berarti bahwa bahkan jika aplikasi Anda menghasilkan email ke server mail secara utas tunggal, server mail akan menggunakan beberapa utas saat mengirim ke Amazon SES. Untuk informasi selengkapnya, lihat Mengintegrasikan Amazon SES dengan server email yang ada.
- Pertimbangkan hosting aplikasi Anda lebih dekat ke titik akhir API Amazon SES—Anda mungkin ingin mempertimbangkan hosting aplikasi Anda di pusat data dekat dengan titik akhir API Amazon SES, atau pada instans Amazon EC2 di Wilayah AWS sebagai titik akhir API Amazon SES. Hal ini dapat membantu untuk mengurangi latensi jaringan antara aplikasi Anda dan Amazon SES, dan meningkatkan throughput. Untuk daftar wilayah tempat Amazon SES tersedia, lihat Amazon Simple Email Service (Amazon SES) di Amazon SES di Referensi Umum AWS.
- Pertimbangkan untuk menggunakan beberapa mesin—Tergantung pada konfigurasi sistem pada mesin host Anda, mungkin ada batasan pada jumlah koneksi HTTP serentak ke satu alamat IP, yang dapat membatasi manfaat paralelisme setelah Anda melebihi sejumlah koneksi bersamaan pada satu mesin. Jika ini adalah kemacetan, Anda mungkin ingin mempertimbangkan membuat permintaan Amazon SES secara bersamaan menggunakan beberapa mesin.
- Pertimbangkan untuk menggunakan API kueri Amazon SES bukan titik akhir SMTP—Menggunakan API kueri Amazon SES memungkinkan Anda untuk mengirim email permintaan pengiriman menggunakan panggilan jaringan tunggal, sedangkan interfacing dengan titik akhir SMTP melibatkan percakapan SMTP yang terdiri dari beberapa permintaan jaringan (misalnya, EHLO, MAIL FROM, RCPT TO, DATA, QUIT). Untuk informasi selengkapnya tentang API kueri Amazon SES, lihat Menggunakan Amazon SES API untuk mengirim email.
- Menggunakan simulator kotak pesan Amazon SES untuk menguji throughput maksimum—Untuk menguji setiap perubahan yang mungkin Anda terapkan, Anda dapat menggunakan simulator kotak surat. Simulator kotak surat dapat membantu Anda menentukan throughput maksimum sistem tanpa menggunakan kuota pengiriman harian Anda. Untuk informasi selengkapnya tentang simulator kotak surat, lihat Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual.

Jika Anda mengakses Amazon SES melalui antarmuka SMTP, lihat Masalah SMTP Amazon SES untuk masalah terkait SMTP tertentu yang dapat mempengaruhi throughput.

Masalah SMTP Amazon SES

Bagian ini berisi solusi untuk beberapa masalah umum yang terkait dengan pengiriman email melalui antarmuka Amazon SES Protokol Transfer Surat Sederhana (SMTP). Hal ini juga berisi daftar kode respons SMTP yang dikembalikan Amazon SES.

Untuk mempelajari selengkapnya tentang mengirim email melalui antarmuka Amazon SES SMTP, lihat Menggunakan SES SMTP antarmuka Amazon untuk mengirim email.

- Anda tidak dapat terhubung ke titik akhir SMTP Amazon SES.

  Masalah menghubungkan ke titik akhir Amazon SES SMTP yang paling sering terkait dengan masalah berikut:

  - Kredensial yang salah — Kredensi yang Anda gunakan untuk terhubung ke titik akhir SMTP berbeda dari kredensil Anda. AWS Untuk mendapatkan kredensial SMTP Anda, lihat Memperoleh SES SMTP kredensi Amazon. Untuk informasi selengkapnya tentang jenis kredensial, lihat Tipe kredensial Amazon SES.
  - Masalah jaringan atau firewall - Jaringan Anda mungkin memblokir hubungan keluar melalui port tempat Anda mencoba mengirim email. Untuk menentukan apakah masalah di jaringan lokal Anda menyebabkan masalah hubungan, ketik perintah berikut di baris perintah, ganti *`port`* dengan port yang Anda coba gunakan (biasanya 465, 587, 2465, atau 2587): `telnet email-smtp.us-west-2.amazonaws.com` *`port`*

    Jika Anda dapat terhubung ke server SMTP menggunakan perintah ini, dan mencoba untuk terhubung ke Amazon SES menggunakan TLS Wrapper atau STARTTLS, menyelesaikan prosedur yang ditampilkan dalam Menguji koneksi Anda ke SES SMTP antarmuka Amazon menggunakan baris perintah.

    Jika Anda tidak dapat terhubung ke titik akhir Amazon SES SMTP menggunakan `telnet` atau `openssl`, hal ini menunjukkan bahwa sesuatu di jaringan Anda (seperti firewall) memblokir koneksi keluar melalui port yang sedang Anda coba gunakan. Bekerja dengan administrator jaringan Anda untuk mendiagnosis dan memperbaiki masalah.
- Anda mengirim ke Amazon SES dari instans Amazon EC2 menggunakan port 25, dan Anda menerima kesalahan batas waktu.

Amazon EC2 membatasi port 25 secara default. Untuk menghapus batasan ini, kirimkan Permintaan Amazon EC2 untuk Menghapus Batasan Pengiriman Email. Anda juga dapat terhubung ke Amazon SES menggunakan port 465 atau 587, yang keduanya tidak dibatasi.

- Kesalahan jaringan menyebabkan email jatuh.

  Pastikan aplikasi Anda menggunakan logika coba lagi saat terhubung ke titik akhir SMTP Amazon SES, dan aplikasi Anda dapat mendeteksi dan mencoba lagi pengiriman pesan jika terjadi kesalahan jaringan. SMTP adalah protokol terperinci, dan mengirim email menggunakan protokol ini memerlukan beberapa jaringan berulang. Karena sifat SMTP, potensi kesalahan jaringan meningkat.

- Anda kehilangan koneksi dengan titik akhir SMTP.

  Koneksi yang hilang paling sering disebabkan oleh masalah berikut:

  - Ukuran MTU - Jika Anda menerima pesan kesalahan waktu habis, Unit Transmisi Maksimum (MTU) antarmuka jaringan untuk komputer yang Anda gunakan untuk menghubungkan ke antarmuka SMTP Amazon SES mungkin terlalu besar. Untuk mengatasi masalah ini, atur ukuran MTU di komputer tersebut ke 1500 byte.

    Untuk informasi selengkapnya tentang menyetel ukuran MTU di sistem operasi Windows, Linux, dan macOS, lihat Pertanyaan yang Muncul Menggantung di Klien dan Jangan Mencapai Cluster di Panduan Manajemen Amazon Redshift.

    Untuk informasi selengkapnya tentang menyetel ukuran MTU untuk instans Amazon EC2, lihat Unit Transmisi Maksimum Jaringan (MTU) untuk Instans EC2 Anda di Panduan Pengguna Amazon EC2.

  - Koneksi berumur panjang - Titik akhir Amazon SES SMTP berjalan pada armada instans Amazon EC2 di belakang Elastic Load Balancer (ELB). Untuk memastikan bahwa sistem ini up-to-date toleran terhadap kesalahan, instans Amazon EC2 aktif dihentikan secara berkala dan diganti dengan instans baru. Karena aplikasi Anda terhubung ke instans Amazon EC2 melalui ELB, koneksi menjadi tidak valid ketika instans Amazon EC2 dihentikan. Anda harus membuat hubungan SMTP baru setelah Anda mengirimkan sejumlah pesan tetap melalui satu koneksi SMTP, atau jika koneksi SMTP telah aktif selama beberapa waktu. Anda perlu bereksperimen untuk menemukan ambang batas yang sesuai tergantung tempat aplikasi Anda di-hosting dan bagaimana aplikasi mengirimkan email ke Amazon SES.

- Anda ingin mengetahui alamat IP server email Amazon SES SMTP sehingga Anda dapat mengizinkan daftar alamat IP dengan jaringan Anda.

Alamat IP untuk titik akhir Amazon SES SMTP berada di belakang penyeimbang beban. Akibatnya, alamat IP ini sering berubah. IP tidak mungkin memberikan daftar definitif semua alamat IP untuk titik akhir Amazon SES. Kami menyarankan Anda mengizinkan daftar `amazonses.com` domain, daripada mengizinkan daftar alamat IP individual.

Kode respons SMTP dikembalikan oleh Amazon SES

Bagian ini berisi daftar kode respons yang dikembalikan oleh antarmuka SMTP Amazon SES.

Anda harus mencoba lagi permintaan SMTP yang menerima kesalahan 400. Kami menyarankan Anda untuk menerapkan sistem yang mencoba kembali permintaan dengan waktu tunggu yang semakin lama (misalnya, tunggu 5 detik sebelum mencoba lagi, lalu tunggu 10 detik, lalu tunggu 30 detik). Jika percobaan ketiga tidak berhasil, tunggu 20 menit, lalu ulangi prosesnya. Untuk melihat contoh penerapan yang menggunakan kebijakan percobaan ulang eksponensial, lihat Cara menangani kesalahan "Throttling - Laju pengiriman maksimum terlampaui" di Blog Olahpesan dan Penargetan AWS .

> **Note**
>
> AWS SDK menerapkan logika coba lagi secara otomatis, tetapi mereka menggunakan antarmuka HTTPS alih-alih SMTP.

Jika Anda menerima pesan kesalahan 500, Anda harus merevisi permintaan Anda untuk memperbaiki masalah sebelum mengirimkan permintaan lagi. Misalnya, jika kredensi AWS otentikasi Anda tidak valid, Anda harus memperbarui aplikasi Anda untuk menggunakan kredensil yang benar sebelum Anda mengirimkan permintaan Anda lagi.

Deskripsi	Kode respons	Informasi selengkapnya
Autentikasi berhasil	`235 Authentication successful`	Klien SMTP Anda berhasil terhubung dan masuk ke server SMTP.
Pengiriman berhasil	`250 Ok` *`MessageID`*	*`MessageID`* adalah string karakter unik yang digunakan

Deskripsi	Kode respons	Informasi selengkapnya
		Amazon SES untuk mengident ifikasi pesan.
Layanan tidak tersedia	`421 Too many concurrent SMTP connections`	Amazon SES tidak dapat memproses permintaan karena saat ini terlalu banyak koneksi ke server SMTP.
Kesalahan pemrosesan lokal	`451 Temporary service failure`	Amazon SES tidak bisa memproses permintaan. Mungkin ada masalah dengan permintaan yang mencegah Amazon SES diproses.
Waktu habis	`451 Timeout waiting for data from client`	Terlalu banyak waktu berlalu di antara permintaan, sehingga server SMTP menutup sambungan.
Kuota pengiriman harian terlampaui	`454 Throttling failure: Daily message quota exceeded`	Anda telah melampaui jumlah email maksimum yang diizinkan Amazon SES untuk dikirim dalam periode 24 jam. Untuk informasi lebih lanjut, lihat Mengelola batas pengiriman Amazon SES Anda.
Laju pengiriman maksimum terlampaui	`454 Throttling failure: Maximum sending rate exceeded`	Anda telah melampaui jumlah email maksimum yang diizinkan Amazon SES untuk Anda kirim per detik. Untuk informasi lebih lanjut, lihat Mengelola batas pengiriman Amazon SES Anda.

Deskripsi	Kode respons	Informasi selengkapnya
Masalah Amazon SES saat memvalidasi kredensial SMTP	`454 Temporary authentication failure`	Masalah yang dapat menyebabkan kasus ini termasuk (namun tidak terbatas pada): • Ada masalah dengan enkripsi antara aplikasi pengirima n email Anda dan Amazon SES. Perhatikan bahwa Anda harus menggunakan koneksi terenkripsi ketika Anda terhubung ke Amazon SES. Untuk informasi lebih lanjut, lihat Menghubungkan ke titik SES SMTP akhir Amazon. • Amazon SES bisa mengalami masalah. Periksa Service Health Dashboard AWS untuk pembaruan.
Masalah saat menerima permintaan	`454 Temporary service failure`	Amazon SES tidak berhasil menerima permintaan. Akibatnya, pesan tidak terkirim.
Kredensial tidak benar	`530 Authentication required`	Aplikasi yang Anda gunakan untuk mengirim email tidak mencoba mengautentikasi saat terhubung ke antarmuka SMTP Amazon SES.

Deskripsi	Kode respons	Informasi selengkapnya
Kredensial Autentikasi tidak valid	`535 Authentication Credentials Invalid`	Aplikasi yang Anda gunakan untuk mengirim email tidak memberikan kredensial SMTP yang benar untuk Amazon SES. Perhatikan bahwa kredenal SMTP Anda tidak sama dengan kredensil Anda. AWS Untuk informasi selengkapnya, lihat Memperoleh SES SMTP kredensi Amazon.
Akun tidak berlangganan Amazon SES	`535 Account not subscribed to SES`	Akun AWS Yang memiliki kredensi SMTP tidak mendaftar untuk Amazon SES.
Pesan terlalu panjang	`552 Message is too long.`	Pesan yang Anda coba kirim lebih besar dari ukuran pesan maksimum.
Akun tidak berlangganan Amazon SES	`535 Account not subscribed to SES`	Akun AWS Yang memiliki kredensi SMTP tidak mendaftar untuk Amazon SES.
MAIL DARI kesalahan sintaks	`553 <email-address > Invalid email address`	Ada kesalahan sintaks di bagian MAIL FROM dari pesan SMTP. Harap periksa apakah Anda mengikuti format yang benar dan jangan lupa untuk melampirkan alamat email di '<>'.

Deskripsi	Kode respons	Informasi selengkapnya
Kesalahan sintaks RCPT TO	`553 <email-address > address unknown`	Ada kesalahan sintaks di bagian RCPT TO dari pesan SMTP. Harap periksa apakah Anda mengikuti format yang benar dan jangan lupa untuk melampirkan alamat email di '<>'.
Pengguna tidak berwenang untuk memanggil titik akhir Amazon SES SMTP	`554 Access denied: User UserARN is not authorized to perform ses:SendRawEmail on resource IdentityARN`	Kebijakan AWS Identity and Access Management (IAM) atau kebijakan otorisasi pengiriman Amazon SES dari pengguna yang memiliki kredenal SMTP tidak diizinkan untuk memanggil titik akhir SMTP Amazon SES.

Deskripsi	Kode respons	Informasi selengkapnya
Alamat email belum diverifikasi	`554 Message rejected: Email address is not verified. The following identities failed the check in region` *`region`*`: `*`identity0`*` , `*`identity1`*` , `*`identity2`*	Anda mencoba mengirim email dari alamat email atau domain yang tidak diverifikasi untuk mengirim email dari akun Amazon SES Anda. Kesalahan ini dapat berlaku untuk alamat “Dari”, “Sumber”, “Pengirim”, atau “Jalur Kembali”. Jika akun Anda masih berada di sandbox, Anda juga harus memverifikasi setiap alamat email penerima (kecuali penerima yang disediakan oleh Simulator kotak surat Amazon SES). Jika Amazon SES tidak dapat menampilkan semua identitas yang gagal dalam pemeriksaan verifikasi, pesan kesalahan berakhir dengan tiga titik (...). **Note** Amazon SES memiliki titik akhir di beberapa Wilayah AWS, dan status verifikasi alamat email terpisah untuk masing-masing Wilayah AWS. Anda harus menyelesaikan proses verifikasi untuk setiap pengirim Wilayah AWS yang ingin Anda gunakan.

> **Note**
>
> Untuk masalah SMTP yang tidak ditangani oleh pemecahan masalah di halaman ini, coba opsi dukungan SES yang tercantum di bawah Menghubungi Kami.

Pertanyaan yang sering diajukan Amazon SES (FAQ)

Bagian ini berisi jawaban atas beberapa pertanyaan yang sering diajukan terkait dengan penggunaan Amazon SES.

Bagian ini berisi FAQ untuk topik berikut:

- FAQ proses peninjauan Pengiriman Amazon SES
- FAQ DNS Blackhole List (DNSBL)
- FAQ metrik pengiriman email Amazon SES

FAQ proses peninjauan Pengiriman Amazon SES

Kami memantau email yang dikirim melalui Amazon SES untuk memastikan bahwa layanan tidak digunakan untuk mengirim email berbahaya, tidak diminta, atau berkualitas rendah. Jika kami menentukan bahwa pengguna mengirim konten yang termasuk dalam salah satu kategori ini, kami akan mengambil tindakan pada akun tersebut. Kami menyebut proses ini dengan proses peninjauan pengiriman.

Pada banyak kasus, saat kami mendeteksi masalah pada sebuah akun, kami menempatkan akun tersebut dalam peninjauan. Dalam kasus lain, kami menjeda kemampuan akun untuk mengirim email. Kami mengambil tindakan ini untuk melindungi reputasi pengirim setiap akun, dan untuk mencegah pengguna SES lainnya mengalami gangguan layanan dan masalah pengiriman.

Daftar Isi

- FAQ Akun dalam peninjauan
- FAQ penjedaan pengiriman
- FAQ Pentalan
- FAQ Aduan
- FAQ Jebakan Spam
- FAQ Investigasi manual

FAQ Akun dalam peninjauan

T1. Saya menerima pesan yang menyatakan bahwa akun saya dalam peninjauan. Apa artinya?

Kami telah mendeteksi masalah terkait email yang dikirim dari akun Anda, dan kami memberi Anda waktu untuk memperbaikinya. Anda dapat terus mengirim email seperti biasanya, tetapi Anda juga harus memperbaiki masalah yang menyebabkan akun Anda ditempatkan dalam peninjauan. Jika Anda tidak memperbaiki masalah sebelum periode peninjauan berakhir, kami mungkin menjeda kemampuan akun Anda untuk mengirim email lainnya.

T2. Apakah saya akan selalu diberi tahu jika akun saya ditempatkan dalam peninjauan?

Ya. Anda akan menerima notifikasi di alamat email yang terkait dengan akun AWS Anda.

T3. Mengapa saya tidak menerima notifikasi bahwa akun saya dalam peninjauan?

Saat akun Anda ditinjau, kami secara otomatis mengirimkan pemberitahuan ke alamat email yang terkait dengan AWS akun Anda. Alamat email ini adalah alamat yang Anda tentukan saat Anda membuat AWS akun. Dalam beberapa kasus, alamat email ini mungkin berbeda dari yang Anda gunakan untuk mengirim email menggunakan SES.

Kami menyarankan Anda untuk memantau reputasi pengirim Anda melihat Metrik reputasi secara teratur. Anda juga dapat mengatur alarm otomatis di Amazon CloudWatch. Alarm ini dapat mengirimkan notifikasi bila metrik reputasi Anda melebihi ambang batas tertentu. Anda juga dapat mengonfigurasi Amazon CloudWatch untuk menghubungi Anda dengan cara lain, seperti dengan mengirim pesan teks ke ponsel Anda.

T4. Apakah fakta bahwa akun SES saya sedang ditinjau berdampak pada penggunaan AWS layanan lain oleh saya?

Anda masih dapat menggunakan AWS layanan lain saat akun SES Anda sedang ditinjau. Namun, jika Anda meminta peningkatan kuota layanan untuk AWS layanan lain yang mengirimkan komunikasi keluar (seperti Amazon SNS), permintaan tersebut dapat ditolak hingga periode peninjauan untuk akun SES Anda dicabut.

T5. Apa yang harus saya lakukan jika akun saya sedang dalam peninjauan?

Anda harus melakukan hal berikut:

- Jika situasi Anda memungkinkan, berhenti mengirim email sampai Anda memperbaiki masalah. Anda masih dapat mengirim email saat akun Anda sedang dalam peninjauan. Namun, jika Anda terus mengirim email tanpa membuat perubahan, Anda mungkin secara tidak sengaja memperburuk masalah.
- Lihat email yang Anda terima dari kami untuk ringkasan masalah.
- Selidiki pengiriman Anda untuk menentukan aspek pengiriman Anda yang secara khusus memicu masalah.
- Setelah Anda membuat perubahan yang Anda yakini akan menyelesaikan masalah, masuk ke AWS Konsol dan pergi ke Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah tersebut mencegah masalah terjadi lagi di masa mendatang.
- Pastikan untuk memberikan informasi apa pun yang kami minta secara khusus. Kami membutuhkan informasi ini untuk mengevaluasi kasus Anda.

T6. Bagaimana cara meminta peninjauan?

Anda dapat meminta agar kami meninjau keputusan kami untuk menempatkan akun Anda dalam peninjauan. Untuk meminta peninjauan, masuk ke AWS Konsol dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda.

Dalam permintaan Anda, berikan informasi berikut:

- Informasi tentang akar masalah peristiwa yang menyebabkan akun Anda ditempatkan dalam peninjauan.
- Daftar perubahan yang Anda buat untuk memperbaiki masalah. Cukup sertakan langkah-langkah yang telah Anda terapkan, bukan langkah-langkah yang akan diterapkan di masa mendatang.
- Informasi tentang cara perubahan tersebut mencegah masalah yang sama terjadi lagi di masa mendatang.

Tergantung pada sifat peristiwa yang membuat kami menempatkan akun Anda dalam peninjauan, kami mungkin memerlukan informasi tambahan. Lihat topik FAQ yang terkait dengan masalah yang Anda alami untuk daftar informasi yang harus Anda sertakan dalam permintaan Anda.

T7. Bagaimana jika permintaan peninjauan saya tidak diterima?

Kami akan menanggapi permintaan Anda dengan informasi tentang mengapa kami tidak menerimanya. Dalam beberapa kasus, Anda akan dapat mengirimkan permintaan lain jika Anda dapat menunjukkan bahwa Anda menyelesaikan masalah tersebut, dan bahwa perubahan Anda mencegah masalah terjadi lagi di masa mendatang.

T8. Dapatkah Anda membantu saya mendiagnosis masalah?

Biasanya kami hanya dapat memberikan gambaran umum tingkat tinggi dari masalah Anda (misalnya, Anda memiliki masalah dengan pentalan). Anda harus menyelidiki akar masalah di pihak Anda.

T9. Bagaimana saya tahu jika akun saya sudah tidak dalam peninjauan?

Metrik reputasi mencakup informasi tentang status akun Anda saat ini. Untuk informasi selengkapnya, lihat Menggunakan metrik reputasi untuk melacak tingkat pentalan dan aduan.

T10. Apakah Anda menempatkan akun saya dalam peninjauan setiap kali ada masalah?

Tidak. Dalam beberapa situasi, kami mungkin menjeda kemampuan akun Anda untuk mengirim email tanpa terlebih dahulu menempatkan akun Anda dalam peninjauan. Sebagai contoh:

- Jika masalahnya sangat serius.
- Jika akun Anda telah ditempatkan dalam peninjauan untuk masalah yang sama beberapa kali di masa lalu. Untuk alasan ini, penting untuk mengatasi masalah utamanya, bukan hanya menyelesaikan insiden tertentu yang menyebabkan akun Anda ditempatkan dalam peninjauan. Misalnya, jika kampanye tertentu menyebabkan kami menempatkan akun Anda dalam peninjauan, Anda harus melakukan lebih dari sekadar menghentikan kampanye tersebut. Anda harus menentukan properti kampanye mana yang bermasalah dan pastikan bahwa Anda sudah memprosesnya sehingga kampanye Anda di masa mendatang tidak mendapatkan masalah yang sama.

Dalam salah satu situasi ini, kami secara otomatis mengirim notifikasi saat menjeda kemampuan akun Anda untuk mengirim email.

T11. Bagaimana jika saya melakukan perbaikan sesaat sebelum masa peninjauan berakhir?

Masuk ke AWS Management Console dan pergi ke Support Center. Balas kasus yang kami buka atas nama Anda. Dalam balasan Anda atas kasus ini, beri tahu kami bahwa Anda telah menyelesaikan masalahnya.

T12. Bisakah saya mendapatkan bantuan dari AWS perwakilan saya atau Premium Support?

Jika Anda sudah bekerja dengan perwakilan AWS akun, kami akan secara otomatis menghubunginya saat akun Anda ditinjau. Perwakilan akun Anda mungkin dapat memberikan informasi tambahan untuk membantu Anda memahami masalah ini dengan lebih baik. Jika Anda menggunakan Premium Support, Anda juga harus menghubungi tim tersebut untuk mendapatkan bantuan tambahan.

FAQ penjedaan pengiriman

T1. Saya menerima pesan yang menyatakan bahwa kemampuan akun saya untuk mengirim email dijeda. Apa artinya?

Kami menjeda kemampuan akun Anda untuk mengirim email karena ada masalah serius dengan email yang Anda kirim. Paling sering, kami menjeda akun karena salah satu alasan berikut:

- Kami sebelumnya menempatkan akun Anda dalam peninjauan. Masalah yang menyebabkan kami menempatkan akun Anda dalam peninjauan tidak diperbaiki sebelum akhir periode peninjauan, jadi kami menjeda kemampuan akun Anda untuk mengirim email.
- Kami telah menempatkan akun Anda dalam peninjauan beberapa kali untuk masalah yang sama.
- Akun Anda mengirim email yang melanggar Syarat Layanan AWS. Jika pelanggaran ini serius, kami mungkin menjeda kemampuan akun Anda untuk mengirim email tanpa menempatkan akun dalam peninjauan terlebih dahulu.

T2. Apakah saya akan selalu diberi tahu jika kemampuan akun saya untuk mengirim email dijeda?

Ya. Anda akan menerima notifikasi di alamat email yang terkait dengan akun AWS Anda.

T3. Kemampuan akun saya untuk mengirim email dijeda. Mengapa saya tidak menerima notifikasi?

Ketika kami menjeda kemampuan akun untuk mengirim email, kami secara otomatis mengirim notifikasi ke alamat email yang terkait dengan akun tersebut.

> **Note**
>
> Saat Anda membuat AWS akun, Anda harus memberikan alamat email. Anda dapat mengubah alamat ini kapan saja. Untuk informasi selengkapnya tentang mengubah alamat yang terkait dengan AWS akun Anda, lihat Mengelola AWS Akun di Panduan AWS Billing and Cost Management Pengguna.

Anda dapat menggunakan Amazon CloudWatch untuk membuat alarm yang memberi tahu Anda ketika tingkat bouncing dan keluhan Anda terlalu tinggi. Membuat alarm adalah cara yang baik untuk menerima peringatan dini tentang faktor-faktor yang dapat menyebabkan kami menjeda kemampuan akun Anda untuk mengirim email. Namun, ada beberapa faktor selain pentalan dan aduan yang dapat menyebabkan kami menjeda kemampuan Anda untuk mengirim email. Untuk informasi selengkapnya tentang membuat alarm CloudWatch, lihatMembuat alarm pemantauan reputasi menggunakan CloudWatch.

Anda juga dapat menggunakan dasbor Akun untuk menentukan status akun Anda saat ini. Misalnya, jika kemampuan akun Anda untuk mengirim email saat ini dijeda, bagian Status akun di dasbor Akun menampilkan status Dijeda. Jika akun Anda dapat mengirim email secara normal, itu menampilkan status Sehat.

Terakhir, Anda dapat memeriksa AWS Health Dashboard (PHD) di https://phd.aws.amazon.com/ untuk menentukan apakah kemampuan akun Anda untuk mengirim email saat ini dijeda. Ketika kami menjeda kemampuan akun untuk mengirim email, kami secara otomatis menambahkan peristiwa Pengiriman SES dijeda ke bagian Log Peristiwa dari PHD. Peristiwa Pengiriman SES dijeda selalu memiliki Status Ditutup, terlepas dari apakah kemampuan akun untuk mengirim email saat ini dijeda atau tidak. Log peristiwa juga menyertakan salinan email yang kami kirim ke alamat email yang terkait dengan AWS akun Anda saat peristiwa jeda pengiriman terjadi.

Anda dapat menggunakan CloudWatch untuk membuat alarm yang mengingatkan Anda ketika acara baru muncul di Dashboard Personal Health Anda. Untuk informasi selengkapnya, lihat Memantau AWS Health CloudWatch Acara dengan Peristiwa di Panduan AWS Health Pengguna.

T4. Kemampuan akun saya untuk mengirim email dijeda. Apakah ini memengaruhi kemampuan saya untuk menggunakan AWS layanan lain?

Anda masih dapat menggunakan AWS layanan lain sementara kemampuan akun Anda untuk mengirim email dijeda. Namun, jika Anda meminta peningkatan kuota layanan untuk layanan AWS lain yang mengirimkan komunikasi keluar (seperti Amazon SNS), kami mungkin menolak permintaan Anda hingga kemampuan akun untuk mengirim email dipulihkan.

T5. Apa yang harus saya lakukan jika kemampuan akun saya untuk mengirim email dijeda?

Anda harus melakukan hal berikut:

- Lihat email yang Anda terima dari kami untuk ringkasan masalah.
- Selidiki pengiriman Anda untuk menentukan aspek pengiriman Anda yang secara khusus memicu masalah.
- Setelah Anda membuat perubahan yang Anda yakini akan menyelesaikan masalah, masuk ke AWS Konsol dan pergi ke Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah tersebut mencegah masalah terjadi lagi di masa mendatang.
- Pastikan untuk memberikan informasi apa pun yang kami minta secara khusus. Kami membutuhkan informasi ini untuk mengevaluasi kasus Anda.

T6. Apa itu peninjauan?

Anda dapat meminta agar kami meninjau keputusan kami untuk menempatkan Anda dalam peninjauan. Lihat pertanyaan berikut untuk informasi selengkapnya tentang permintaan peninjauan.

T7. Bagaimana cara meminta peninjauan?

Untuk meminta peninjauan, masuk ke AWS Konsol dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda.

Dalam permintaan Anda, berikan informasi berikut:

- Informasi tentang penyebab masalah.

- Daftar perubahan yang Anda buat untuk memperbaiki masalah. Cukup sertakan langkah-langkah yang telah Anda terapkan, bukan langkah-langkah yang akan diterapkan di masa mendatang.
- Informasi tentang cara perubahan tersebut akan mencegah masalah yang sama terjadi lagi di masa mendatang.

Tergantung pada sifat peristiwa yang membuat kami menjeda kemampuan akun Anda untuk mengirim email, kami mungkin memerlukan informasi tambahan. Lihat topik FAQ yang terkait dengan masalah yang Anda alami untuk daftar informasi yang harus disertakan dalam permintaan Anda.

T8. Bagaimana jika permintaan saya tidak diterima?

Kami akan menanggapi permintaan Anda dengan informasi tentang mengapa kami tidak menerimanya. Dalam beberapa kasus, Anda akan dapat mengirimkan permintaan lain jika Anda dapat menunjukkan bahwa Anda menyelesaikan masalah tersebut, dan bahwa perubahan Anda mencegah masalah terjadi lagi di masa mendatang.

T9. Dapatkah Anda membantu saya mendiagnosis masalah?

Biasanya kami hanya dapat memberikan gambaran umum tingkat tinggi dari masalah Anda (misalnya, Anda memiliki masalah dengan pentalan). Anda bertanggung jawab untuk memperbaiki masalah tersebut.

T10. Bagaimana cara saya mengetahui kemampuan akun saya untuk mengirim email telah dipulihkan?

Metrik reputasi mencakup informasi tentang status akun Anda saat ini. Untuk informasi lebih lanjut, lihat Menggunakan metrik reputasi untuk melacak tingkat pentalan dan aduan.

T11. Bisakah saya mendapatkan bantuan dari AWS perwakilan saya atau Premium Support?

Jika Anda sudah bekerja dengan perwakilan AWS akun, kami akan secara otomatis menghubunginya jika kami menghentikan sementara kemampuan akun Anda untuk mengirim email. Perwakilan akun Anda mungkin dapat memberikan informasi tambahan untuk membantu Anda memahami masalah ini dengan lebih baik. Jika Anda menggunakan Premium Support, Anda juga harus mengontak tim tersebut untuk mendapatkan bantuan tambahan.

FAQ Pentalan

T1. Kenapa Anda peduli dengan pentalan saya?

Tingkat pentalan tinggi sering digunakan oleh entitas seperti penyedia email dan organisasi antispam untuk mendeteksi pengirim yang terlibat dalam praktik pengiriman email yang buruk. Tingkat pentalan tinggi dapat mengakibatkan email dikirim ke folder spam daripada ke kotak masuk.

T2. Apa yang harus saya lakukan jika saya menerima notifikasi yang menyatakan bahwa akun saya dalam peninjauan atau pengiriman saya dijeda karena tingkat pentalan akun saya?

Identifikasi penyebab masalah, lalu perbaiki. Setelah Anda membuat perubahan yang Anda yakini akan menyelesaikan masalah, masuk ke AWS Konsol dan pergi ke Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah tersebut mencegah masalah terjadi lagi di masa mendatang. Juga mencakup informasi berikut:

- Metode yang Anda gunakan untuk melacak pentalan Anda
- Cara Anda memastikan bahwa alamat email penerima baru valid sebelum mengirimkannya kepada mereka. Misalnya, rekomendasi mana yang Anda ikuti dalam T11. Apa yang bisa saya lakukan untuk meminimalkan pentalan?

T3. Tipe pentalan apa yang dihitung dalam tingkat pentalan saya?

Tingkat pentalan Anda hanya mencakup pentalan keras ke domain yang belum Anda verifikasi. Pentalan keras adalah kegagalan pengiriman permanen seperti "alamat tidak ada." Kegagalan sementara dan berselang seperti "kotak surat penuh", atau pentalan karena alamat IP yang diblokir, tidak dihitung ke dalam tingkat pentalan Anda.

T4. Apakah Anda memberitahukan tingkat pentalan yang dapat menyebabkan akun saya ditempatkan dalam peninjauan atau yang dapat menyebabkan pengiriman saya dijeda?

Untuk hasil terbaik, Anda harus mempertahankan tingkat pentalan di bawah 2%. Tingkat pentalan yang lebih tinggi dapat memengaruhi pengiriman email Anda.

Jika tingkat pentalan Anda 5% atau lebih, kami akan menempatkan akun Anda dalam peninjauan. Jika rasio pentalan 10% atau lebih, kami mungkin menjeda kemampuan akun Anda untuk mengirim email lain sampai Anda menyelesaikan masalah yang menghasilkan tingkat pentalan tinggi.

T5. Berapa lama periode waktu tingkat pentalan saya dihitung?

Kami tidak menghitung tingkat pentalan Anda berdasarkan periode waktu yang tetap, karena pengirim yang berbeda mengirim dengan tingkat pengiriman yang berbeda. Sebaliknya, kita melihat volume perwakilan—jumlah email yang mewakili praktik pengiriman Anda biasanya. Agar adil untuk pengirim volume tinggi dan rendah, volume representatif berbeda untuk setiap pengguna dan berubah seiring dengan perubahan pola pengiriman pengguna.

T6. Dapatkah saya menghitung rasio pentalan saya sendiri dengan menggunakan informasi dari konsol SES atau GetSendStatistics API?

Tidak. Tingkat pentalan dihitung menggunakan volume representatif (lihat T5. Berapa lama periode waktu tingkat pentalan saya dihitung?). Bergantung pada tingkat pengiriman Anda, rasio pentalan Anda dapat meregang lebih jauh ke masa lalu daripada konsol SES atau `GetSendStatistics` dapat diambil. Selain itu, hanya email ke domain yang tidak terverifikasi yang dipertimbangkan saat menghitung rasio pentalan Anda. Namun, jika Anda memantau tingkat pentalan secara teratur menggunakan metode tersebut, Anda masih harus memiliki indikator bagus yang dapat digunakan untuk mendeteksi masalah sebelum masalah tersebut mencapai tingkat yang menyebabkan kami menempatkan akun Anda dalam peninjauan atau menjeda kemampuan akun Anda untuk mengirim email.

T7. Bagaimana saya bisa mengetahui alamat email mana yang terpental?

Periksa pemberitahuan bouncing yang dikirimkan SES kepada Anda. Alamat email tempat SES meneruskan notifikasi tergantung pada cara Anda mengirim pesan asli, seperti yang dijelaskan diMenerima notifikasi Amazon SES melalui email. Anda juga dapat menyiapkan notifikasi pentalan melalui Amazon Simple Notification Service (Amazon SNS), seperti yang dijelaskan di Menyiapkan pemberitahuan acara untuk Amazon SES. Perhatikan bahwa hanya menghapus alamat yang terpental dari daftar Anda tanpa investigasi tambahan mungkin tidak menyelesaikan masalah utamanya. Untuk informasi tentang apa yang dapat Anda lakukan untuk mengurangi pentalan, lihat T11. Apa yang bisa saya lakukan untuk meminimalkan pentalan?.

T8. Jika saya belum memantau pentalan saya, dapatkah Anda memberi saya daftar alamat yang telah terpental?

Tidak, kami tidak dapat memberikan daftar lengkap alamat yang telah terpental. Anda bertanggung jawab untuk memantau dan bertindak atas pentalan pada akun Anda.

T9. Bagaimana saya harus menangani pentalan?

Anda perlu menghapus alamat terpental dari milis Anda dan segera hentikan pengiriman surat kepada mereka. Jika Anda adalah pengirim kecil, mungkin cukup untuk hanya memantau pentalan melalui email dan menghapus alamat terpental secara manual dari milis Anda. Jika volume Anda lebih tinggi, Anda mungkin ingin menyiapkan otomatisasi untuk proses ini, baik dengan memproses kotak surat secara terprogram tempat Anda menerima pentalan, atau dengan mengatur notifikasi pentalan melalui Amazon SNS. Untuk informasi lebih lanjut, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.

T10. Mungkinkah email saya terpental karena saya telah mencapai kuota pengiriman saya?

Tidak. Pentalan tidak terkait dengan kuota pengiriman. Jika Anda mencoba melebihi kuota pengiriman, Anda akan menerima kesalahan dari API SES atau antarmuka SMTP ketika Anda mencoba mengirim email.

T11. Apa yang bisa saya lakukan untuk meminimalkan pentalan?

Pertama, pastikan bahwa Anda menyadari pentalan Anda (lihat T7. Bagaimana saya bisa mengetahui alamat email mana yang terpental?). Kemudian ikuti panduan berikut:

- Jangan membeli, menyewa, atau membagikan alamat email. Kirim email hanya ke penerima yang secara eksplisit meminta untuk mendapatkan email dari Anda.
- Hapus alamat email yang terpental dari daftar Anda.
- Pada formulir web, minta pengguna untuk memasukkan alamat email mereka dua kali, dan periksa untuk memastikan kedua alamat tersebut cocok sebelum formulir dapat dikirimkan.
- Gunakan opt-in ganda untuk mendaftarkan pengguna baru. Artinya, saat pengguna baru mendaftar, kirimkan email konfirmasi yang harus mereka klik sebelum menerima surat lainnya. Hal ini mencegah orang mendaftarkan orang lain serta pendaftaran yang tidak disengaja.
- Jika Anda harus mengirim ke alamat yang belum Anda kirimi surat akhir-akhir ini (sehingga Anda tidak yakin bahwa alamat tersebut masih valid), lakukan hanya dengan sebagian kecil dari

keseluruhan pengiriman Anda. Untuk informasi lebih lanjut, lihat kiriman blog kami Jangan pernah mengirim ke alamat lama, tetapi bagaimana jika harus tetap mengirim ke alamat lama?.

- Pastikan Anda tidak menyusun pendaftaran untuk mendorong orang menggunakan alamat fiktif. Misalnya, jangan berikan nilai tambah atau manfaat apa pun sampai penerima memverifikasi alamat mereka.
- Jika Anda memiliki fitur "kirim email ke teman", gunakan CAPTCHA atau mekanisme serupa untuk mencegah penggunaan fitur tersebut secara otomatis, dan jangan izinkan pengguna memasukkan konten sewenang-wenang.
- Jika Anda menggunakan SES untuk pemberitahuan sistem, pastikan Anda mengirim notifikasi ke alamat asli yang dapat menerima email. Pertimbangkan juga untuk mematikan notifikasi yang tidak Anda perlukan.
- Jika Anda menguji sistem baru, pastikan Anda mengirim ke alamat asli yang dapat menerima email, atau Anda menggunakan simulator kotak surat SES. Untuk informasi selengkapnya, lihat Menggunakan simulator kotak surat secara manual menggunakan simulator kotak surat secara manual.

FAQ Aduan

T1. Apa itu aduan?

Aduan terjadi saat penerima melaporkan bahwa mereka tidak ingin menerima email. Mereka mungkin telah mengklik tombol “Laporkan spam” di klien email mereka, mengeluh kepada penyedia email mereka, memberi tahu SES secara langsung, atau melalui beberapa metode lain. Topik ini mencakup informasi umum tentang aduan. Jika pemberitahuan Anda berisi informasi spesifik tentang sumber keluhan, baca juga topik yang relevan:

- Keluhan SES melalui loop umpan balik FAQ
- Keluhan SES langsung dari penerima FAQ
- Keluhan SES melalui penyedia email FAQ

T2. Kenapa Anda peduli dengan aduan saya?

Tingkat aduan yang tinggi sering digunakan oleh entitas seperti penyedia email dan organisasi antispam sebagai indikator bahwa pengirim mengirim ke penerima yang tidak secara khusus mendaftar untuk menerima email, atau pengirim mengirim tipe konten yang berbeda dari yang diinginkan penerima saat mendaftar.

T3. Apa yang harus saya lakukan jika saya menerima pemberitahuan yang mengatakan bahwa akun saya dalam tinjauan atau pengiriman saya dijeda karena ada masalah dengan aduan?

Tinjau proses akuisisi daftar dan konten email Anda untuk mencoba memahami kenapa penerima mungkin tidak menyukai email yang mereka terima dari Anda. Identifikasi penyebab masalah, lalu perbaiki. Setelah Anda membuat perubahan yang Anda yakini akan menyelesaikan masalah, masuk ke AWS Konsol dan pergi ke Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah tersebut mencegah masalah terjadi lagi di masa mendatang.

T4. Apa yang bisa saya lakukan untuk meminimalkan aduan?

Pertama, pastikan Anda memantau keluhan yang dapat diberitahukan SES kepada Anda, yang merupakan keluhan yang diterima SES melalui loop umpan balik (lihatKeluhan SES melalui loop umpan balik FAQ). Kemudian ikuti panduan berikut:

- Jangan membeli, menyewa, atau membagikan alamat email. Gunakan hanya alamat yang secara khusus meminta email Anda.
- Gunakan opt-in ganda untuk mendaftar pengguna baru. Artinya, saat pengguna mendaftar, kirimkan email konfirmasi yang harus mereka klik sebelum menerima surat lainnya. Hal ini mencegah orang mendaftarkan orang lain serta pendaftaran yang tidak disengaja.
- Pantau keterlibatan dengan email yang Anda kirim dan hentikan pengiriman ke penerima yang tidak membuka atau mengeklik pesan Anda.
- Ketika pengguna baru mendaftar, perjelas tentang tipe email yang akan mereka terima dari Anda, dan pastikan bahwa Anda hanya mengirim tipe email yang mereka inginkan saat mendaftar. Contohnya, jika pengguna mendaftar untuk pembaruan berita, jangan mengirimkan iklan kepada mereka.
- Pastikan bahwa surat Anda diformat dengan baik dan terlihat profesional.
- Pastikan bahwa surat Anda jelas berasal dari Anda dan tidak dapat tertukar dengan sesuatu yang lain.
- Berikan pengguna cara yang jelas dan mudah untuk berhenti berlangganan dari email Anda.

Keluhan SES melalui loop umpan balik FAQ

Topik ini memberikan informasi tentang keluhan yang diterima SES dari penyedia email melalui loop umpan balik. Untuk informasi umum yang berlaku untuk semua tipe aduan, lihat FAQ Aduan.

T1. Bagaimana aduan tipe ini dilaporkan?

Sebagian besar program klien email menyediakan tombol berlabel "Tandai sebagai Spam" atau yang serupa, yang memindahkan pesan ke folder spam dan meneruskannya ke penyedia email. Selain itu, sebagian besar penyedia email mempertahankan alamat penyalahgunaan (seperti abuse@example.com), tempat pengguna dapat meneruskan email yang tidak diinginkan dan meminta penyedia mengambil tindakan untuk mencegahnya. Jika SES memiliki loop umpan balik (FBL) yang diatur dengan penyedia email, maka mereka mengirim keluhan kembali ke SES.

> **Note**
>
> SES secara otomatis menetapkan header Feedback-ID saat Anda mengirim pesan, memberikan penyedia kotak pesan cara untuk menggabungkan statistik pengiriman, seperti tingkat keluhan dan spam, dan membuatnya tersedia untuk Anda. Nilai header Feedback-ID yang disediakan SES terdiri dari:
>
> - `FeedBackId:((SESInternalID):(AmazonSES))`, dimana:
> - Sesinternalid adalah pengenal yang digunakan oleh SES untuk mengumpulkan informasi keluhan.
> - AmazonSes adalah tag statis yang mengidentifikasi SES sebagai platform pengiriman.
>
> Secara opsional, selain nilai header Feedback-ID standar yang disediakan SES, Anda juga dapat menentukan ID umpan balik khusus Anda sendiri (hingga dua) menggunakan tag `ses:feedback-id-a` dan `ses:feedback-id-b` pesan, lihat. the section called “Umpan balik halus untuk kampanye email”

T2. Apakah keluhan ini termasuk dalam statistik tingkat keluhan yang ditampilkan di konsol SES dan dikembalikan oleh GetSendStatistics API?

Ya. Namun, statistik tingkat keluhan tidak termasuk keluhan dari penyedia email yang tidak memberikan umpan balik kepada SES. Tingkat aduan dari domain yang memberikan umpan balik kemungkinan juga mewakili pengiriman Anda yang lainnya.

T3. Bagaimana saya bisa diberi tahu tentang aduan ini?

Anda dapat diberi tahu melalui email atau melalui notifikasi Amazon SNS. Lihat petunjuk penyiapan dalam Menyiapkan pemberitahuan acara untuk Amazon SES.

T4. Apa yang harus saya lakukan jika saya menerima notifikasi aduan melalui email atau melalui Amazon SNS?

Pertama, Anda perlu menghapus alamat yang menghasilkan aduan dari milis Anda dan segera berhenti mengirim surat kepada mereka. Jangan pernah mengirim email yang mengatakan bahwa Anda telah menerima permintaan untuk berhenti berlangganan. Pertimbangkan untuk menyiapkan otomatisasi untuk proses ini, baik dengan memproses kotak pesan tempat Anda menerima aduan secara terprogram, atau dengan menyiapkan notifikasi aduan melalui Amazon SNS. Untuk informasi lebih lanjut, lihat Menyiapkan pemberitahuan acara untuk Amazon SES.

Kemudian, perhatikan pengiriman Anda untuk menentukan mengapa penerima Anda tidak menyukai email yang Anda kirim, dan atasi masalah utama itu. Untuk setiap orang yang mengadu, kemungkinan ada lusinan orang yang tidak menyukai email Anda dan tidak (atau tidak mampu) mengadu. Jika Anda hanya menghapus penerima yang benar-benar mengadu, Anda tidak mengatasi masalah utamanya.

T5. Apakah Anda mengungkapkan tingkat keluhan SES yang dapat menyebabkan akun saya ditinjau atau yang dapat mengakibatkan kemampuan akun saya untuk mengirim email dijeda?

Untuk hasil terbaik, Anda harus mempertahankan tingkat aduan di bawah 0,1%. Tingkat aduan yang lebih tinggi dapat memengaruhi pengiriman email Anda.

Jika tingkat aduan Anda 0,1% atau lebih besar, kami akan menempatkan akun Anda dalam peninjauan. Jika tingkat aduan Anda 0,5% atau lebih besar, kami mungkin menjeda kemampuan akun Anda untuk mengirim email lain sampai Anda menyelesaikan masalah yang mengakibatkan tingkat aduan yang tinggi.

T6. Selama periode waktu berapakah tingkat aduan saya dihitung?

Kami tidak menghitung tingkat aduan Anda berdasarkan periode waktu yang tetap, karena pengirim yang berbeda mengirim dengan tingkat pengiriman yang berbeda. Sebagai gantinya, kami melihat volume perwakilan—jumlah surat yang mewakili praktik pengiriman Anda biasanya. Agar adil bagi pengirim volume tinggi dan rendah, volume representatif berbeda untuk setiap pengguna dan berubah seiring dengan perubahan pola pengiriman pengguna. Selain itu, tingkat aduan tidak dihitung berdasarkan setiap email. Sebaliknya, ini dihitung sebagai persentase keluhan pada email yang dikirim ke domain yang mengirim umpan balik keluhan ke SES.

T7. Dapatkah saya menghitung tingkat keluhan saya sendiri dengan menggunakan metrik dari konsol SES atau GetSendStatistics API?

Tidak. Ada dua alasan utama untuk hal ini:

- Tingkat aduan dihitung menggunakan volume perwakilan (lihat T6. Selama periode waktu berapakah tingkat aduan saya dihitung?). Tergantung pada tingkat pengiriman Anda, tingkat keluhan Anda dapat merentang lebih jauh ke masa lalu daripada konsol SES atau `GetSendStatistics` API dapat mengambil. Oleh karenanya, kami menyarankan agar Anda menggunakan metode ini secara teratur untuk memantau tingkat aduan akun Anda. Memantau tingkat aduan Anda dengan cara ini memberikan informasi yang Anda butuhkan untuk mengidentifikasi masalah sebelum mencapai tingkat yang dapat memengaruhi pengiriman email Anda.
- Saat menghitung tingkat aduan, tidak setiap email dihitung. Tingkat keluhan dihitung sebagai persentase keluhan pada surat yang dikirim ke domain yang mengirimkan umpan balik keluhan ke SES.

T8. Bagaimana saya bisa mengetahui alamat email mana yang mengadu?

Periksa pemberitahuan keluhan yang dikirimkan SES kepada Anda melalui email atau melalui Amazon SNS (lihatMenyiapkan pemberitahuan acara untuk Amazon SES). Namun, penyedia email yang berbeda memberikan jumlah informasi yang berbeda, dan beberapa penyedia menyunting alamat email penerima sebelum menyampaikan pemberitahuan keluhan ke SES. Untuk memungkinkan Anda menemukan alamat email penerima di masa mendatang, opsi terbaik Anda adalah menyimpan pemetaan Anda sendiri antara pengenal dan ID pesan SES yang diteruskan SES kepada Anda saat menerima email. Perhatikan bahwa SES tidak menyimpan ID pesan khusus apa pun yang Anda tambahkan.

T9. Jika saya belum memantau aduan saya, bisakah Anda memberi saya daftar alamat yang mengadu?

Sayangnya, kami tidak dapat memberi Anda daftar lengkapnya. Namun, Anda dapat memantau aduan di masa mendatang melalui email atau melalui Amazon SNS.

T10. Bisakah saya mendapatkan email sampel?

Kami tidak dapat mengirimkan email sampel kepada Anda berdasarkan permintaan, tetapi Anda mungkin menemukan informasi ini dalam notifikasi aduan. Untuk informasi selengkapnya, lihat T8. Bagaimana saya bisa mengetahui alamat email mana yang mengadu?.

Keluhan SES langsung dari penerima FAQ

Topik ini memberikan informasi tentang keluhan yang diterima SES langsung dari penerima. Untuk informasi umum yang berlaku untuk semua tipe aduan, lihat FAQ Aduan.

T1. Bagaimana aduan tipe ini dilaporkan?

Beberapa penerima langsung menghubungi SES tentang email Anda melalui email atau cara lain.

T2. Apakah keluhan ini termasuk dalam statistik tingkat keluhan yang ditampilkan di konsol SES dan dikembalikan oleh GetSendStatistics API?

Tidak. Statistik tingkat keluhan yang Anda ambil menggunakan konsol SES atau `GetSendStatistics` API hanya mencakup keluhan yang diterima SES melalui loop umpan balik. Untuk informasi lebih lanjut tentang tipe aduan tersebut, lihat Keluhan SES melalui loop umpan balik FAQ.

T3. Mengapa saya belum mendengar mengenai aduan ini melalui notifikasi umpan balik email atau melalui Amazon SNS?

Penerusan umpan balik email dan notifikasi Amazon SNS hanya mencakup keluhan yang diterima SES melalui loop umpan balik. Anda tidak akan menerima pemberitahuan untuk keluhan yang diajukan penerima langsung ke SES.

T4. Bagaimana saya bisa mengetahui alamat email mana yang mengadu?

Untuk melindungi identitas penerima yang mengadu, kami tidak bisa mencantumkan alamat email yang mengadu perihal email Anda.

Daripada fokus menghapus penerima individu dari daftar Anda, kami menyarankan Anda untuk menentukan masalah yang menyebabkan aduan dikeluarkan. Sebaiknya Anda mulai dengan meninjau proses akuisisi pelanggan Anda, dan menghapus pelanggan dari daftar Anda yang tidak secara eksplisit meminta untuk mendapatkan email dari Anda. Anda juga harus menganalisis isi email Anda untuk mencoba memahami mengapa penerima mengadu.

T5. Bisakah saya mendapatkan email sampel?

Untuk melindungi identitas penerima yang mengadu, kami tidak dapat memberikan salinan email yang menyebabkan penerima Anda mengadu.

T6. Apa yang harus saya lakukan jika saya menerima notifikasi yang menyatakan bahwa akun saya sedang dalam peninjauan atau pengiriman saya dijeda karena aduan langsung?

Segera ubah proses pengiriman Anda sehingga Anda hanya mengirim pesan kepada penerima yang secara khusus mendaftar untuk menerimanya. Selain itu, pastikan Anda mengirim tipe konten sesuai dengan yang diinginkan penerima saat mendaftar. Setelah Anda membuat perubahan yang Anda yakini akan menyelesaikan masalah, masuk ke AWS Konsol dan pergi ke Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah tersebut mencegah masalah terjadi lagi di masa mendatang.

Jika Anda tidak meminta peninjauan dalam waktu tiga minggu, dan kami terus menerima aduan penerima langsung, kami mungkin menjeda kemampuan akun Anda untuk mengirim email.

Keluhan SES melalui penyedia email FAQ

Topik ini memberikan informasi tentang keluhan yang diterima SES melalui penyedia email (juga disebut penyedia kotak surat). Untuk informasi umum yang berlaku untuk semua tipe aduan, lihat FAQ Aduan.

T1. Bagaimana aduan tipe ini dilaporkan?

Penyedia email melaporkan kepada SES bahwa sejumlah besar pelanggannya menandai email Anda sebagai spam. Laporan tersebut diberikan kepada SES melalui cara selain loop umpan balik yang dijelaskan dalamKeluhan SES melalui loop umpan balik FAQ.

T2. Apakah keluhan ini termasuk dalam statistik tingkat keluhan yang ditampilkan di konsol SES dan dikembalikan oleh GetSendStatistics API?

Tidak. Statistik tingkat keluhan yang Anda ambil menggunakan konsol SES atau `GetSendStatistics` API hanya mencakup keluhan yang diterima SES melalui loop umpan balik.

T3. Mengapa saya belum mendengar mengenai aduan ini melalui notifikasi umpan balik email atau melalui Amazon SNS?

Penerusan umpan balik email dan notifikasi Amazon SNS hanya mencakup keluhan yang diterima SES melalui loop umpan balik.

T4. Bagaimana saya bisa mengetahui alamat email mana yang mengadu?

Penyedia email biasanya tidak mengungkapkan informasi ini. Namun, daripada berfokus pada menghapus penerima individu dari daftar Anda, Anda perlu fokus untuk menemukan dan

memperbaiki masalah utamanya. Mulailah dengan meninjau proses akuisisi daftar dan konten email Anda untuk mencoba memahami mengapa penerima mungkin tidak menyukai email Anda.

T5. Bisakah saya mendapatkan email sampel?

Tidak. Penyedia email biasanya tidak memberikan email sampel.

T6. Apa yang harus saya lakukan jika saya menerima notifikasi yang menyatakan bahwa akun saya dalam peninjauan atau pengiriman saya dijeda karena aduan penyedia email?

Identifikasi penyebab masalah, lalu perbaiki. Setelah Anda membuat perubahan yang Anda yakini akan menyelesaikan masalah, masuk ke AWS Konsol dan pergi ke Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah tersebut mencegah masalah terjadi lagi di masa mendatang. Jika Anda tidak meminta peninjauan dalam waktu tiga minggu, dan kami terus menerima aduan dari penyedia, kami mungkin menjeda kemampuan akun Anda untuk mengirim email lain.

FAQ Jebakan Spam

T1. Apa itu jebakan spam?

Jebakan spam adalah alamat email khusus yang dikelola oleh Penyedia Layanan Internet (ISP), penyedia email, atau organisasi antispam. Karena alamat tersebut tidak akan pernah didaftarkan secara sah untuk menerima email, organisasi yang mengelola jebakan spam ini mengetahui bahwa siapa pun yang mengirim email ke salah satu alamat tersebut kemungkinan akan terlibat dalam praktik email yang patut dipertanyakan.

T2. Bagaimana jebakan spam disiapkan?

Alamat jebakan spam dapat disiapkan dengan beberapa cara. Alamat tersebut dapat diubah dari alamat yang dulunya valid, tetapi sudah tidak digunakan (dan terpental) untuk jangka waktu yang lama. Alamat tersebut juga bisa jadi merupakan alamat yang disiapkan hanya untuk menjadi jebakan spam. Alamat tersebut bisa jadi merupakan alamat yang tidak biasa yang sulit ditebak, dan terkadang merupakan alamat yang mendekati alamat sebenarnya (contohnya, memasukkan salah ketik ke nama domain umum). Sering kali, tetapi tidak selalu, jebakan spam "disemai" ke dunia dengan menempatkan mereka di internet dalam berbagai cara.

T3. Bagaimana SES tahu jika saya mengirim ke spamtraps?

Organisasi tertentu yang mengoperasikan spamtraps mengirim pemberitahuan SES ketika spamtraps mereka terkena pengirim SES.

T4. Bagaimana SES menggunakan laporan spamtrap?

Kami meninjau laporan. Jika kami menentukan bahwa akun Anda mengirim email ke jebakan spam, kami menempatkan akun dalam peninjauan dan meminta Anda untuk memperbaiki masalah utamanya. Jika Anda tidak memperbaiki masalah sebelum periode peninjauan berakhir, kami mungkin menjeda kemampuan akun Anda untuk mengirim email lainnya. Jika masalah jebakan spam Anda sangat parah, kami mungkin segera menjeda kemampuan akun Anda untuk mengirim email, tanpa menempatkan akun Anda dalam peninjauan terlebih dahulu.

T5. Apa yang harus saya lakukan jika menerima pemberitahuan yang mengatakan bahwa akun saya dalam peninjauan atau bahwa pengiriman saya dijeda karena masalah dengan jebakan spam?

Pertama, Anda harus mengatasi masalah yang menyebabkan kami menempatkan akun Anda dalam peninjauan atau menjeda kemampuan Anda untuk mengirim email. Selanjutnya, masuk ke AWS Console dan pergi ke Support Center. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah tersebut mencegah masalah terjadi lagi di masa mendatang. Jika kami setuju bahwa perubahan yang telah Anda buat mengatasi masalah tersebut dengan tepat, kami akan membatalkan periode peninjauan atau menghapus jeda pengiriman dari akun Anda.

Karena cara temuan jebakan spam dilaporkan, mungkin diperlukan waktu tiga minggu atau lebih sebelum kami dapat menentukan apakah perubahan yang Anda buat telah menyelesaikan masalah.

T6. Berapa banyak temuan jebakan spam yang dapat saya kenai sebelum Anda menempatkan akun saya dalam peninjauan atau menjeda kemampuan akun saya untuk mengirim email?

Kami tidak mengungkapkan jumlah temuan jebakan spam tertentu yang menyebabkan kami mengambil tindakan di akun Anda. Namun, penting untuk dicatat bahwa bahkan temuan jebakan spam dengan jumlah kecil dapat memiliki efek yang sangat negatif pada reputasi Anda sebagai pengirim, jadi Anda harus menganggap serius laporan jebakan spam.

T7. Apakah Anda mengungkapkan alamat jebakan spam?

Tidak. Agar jebakan spam efektif, penting untuk alamat tersebut tetap menjadi rahasia. Organisasi jebakan spam mengungkapkan hanya terjadinya temuan jebakan spam, bukan alamat jebakan spam yang sebenarnya.

T8. Apa yang dapat saya lakukan untuk menghindari pengiriman ke jebakan spam?

Untuk mengurangi risiko pengiriman ke jebakan spam, ikuti panduan berikut:

- Jangan membeli, menyewa, atau membagikan alamat email. Gunakan hanya alamat yang secara khusus meminta email Anda.
- Pada formulir web, minta pengguna untuk memasukkan alamat email mereka dua kali, dan periksa untuk memastikan kedua alamat tersebut cocok sebelum formulir dapat dikirimkan.
- Gunakan opt-in ganda untuk mendaftarkan pengguna baru. Artinya, saat pengguna mendaftar, kirimkan email konfirmasi yang harus mereka klik sebelum menerima surat lainnya.
- Pastikan bahwa Anda menghapus alamat yang mengalami pentalan keras dari daftar Anda, sehingga alamat tersebut dihapus jauh sebelum diubah menjadi jebakan spam.
- Pastikan Anda memantau keterlibatan penerima Anda, dan berhenti mengirim ke penerima yang belum berinteraksi dengan email atau situs web Anda baru-baru ini. Jangka waktu untuk "pengguna yang terlibat" tergantung pada kasus penggunaan Anda, namun secara umum jika pengguna belum membuka atau mengeklik email Anda dalam beberapa bulan, sebaiknya pertimbangkan untuk menghapusnya kecuali jika Anda memiliki bukti bahwa mereka menginginkan email Anda.
- Berhati-hatilah dengan kampanye keterlibatan kembali yaitu saat Anda dengan sengaja mengontak orang-orang yang belum berinteraksi dengan Anda baru-baru ini. Upaya ini cenderung sangat berisiko, dan sering kali dapat menimbulkan masalah tidak hanya dengan pengiriman jebakan spam, tetapi juga dengan pentalan dan aduan.
- Kirim pesan opt-in ke seluruh milis Anda dan hanya simpan penerima yang mengeklik tautan verifikasi. Selain menghapus penerima yang tidak aktif dari daftar Anda, prosedur ini juga membantu menghapus alamat jebakan spam. Namun, kami tidak menyarankan untuk menggunakan teknik ini jika menurut Anda milis Anda mungkin berisi banyak alamat yang buruk, atau jika akun Anda sudah bermasalah dengan pentalan, karena dapat menyebabkan tingkat pentalan akun Anda meningkat lebih jauh.

FAQ Investigasi manual

T1. Apa yang harus saya lakukan jika saya menerima notifikasi yang menyatakan bahwa akun saya dalam peninjauan atau pengiriman saya dijeda karena investigasi manual?

Penyelidik SES telah mengidentifikasi masalah signifikan dengan pengiriman Anda. Masalah yang biasa terjadi, namun tidak terbatas pada hal berikut:

- Pengiriman Anda melanggar Kebijakan Penggunaan yang Diterima AWS (AUP).
- Email Anda sepertinya tidak diminta.
- Konten Anda terkait phishing (hal ini termasuk simulasi phishing).
- Konten Anda terkait dengan kasus penggunaan yang tidak didukung SES.

Jika kami yakin bahwa masalah tersebut dapat diperbaiki, kami menempatkan akun Anda dalam peninjauan untuk jangka waktu tertentu. Saat akun Anda dalam peninjauan, Anda harus membuat perubahan pada praktik pengiriman email untuk memperbaiki masalah tersebut.

Jika kami tidak yakin bahwa masalahnya dapat diperbaiki, atau jika masalahnya sangat parah, kami mungkin menjeda kemampuan akun Anda untuk mengirim email tanpa terlebih dahulu menempatkan akun Anda dalam peninjauan.

T2. Masalah apa yang dapat menyebabkan Anda melakukan peninjauan manual terhadap pengiriman email saya?

Ada beberapa masalah yang bisa menyebabkan kami memulai peninjauan manual terhadap akun Anda. Alasan ini termasuk, namun tidak terbatas pada hal berikut ini:

- Penerima menghubungi SES untuk mengeluh tentang email yang dikirim dari akun Anda.
- Kami mendeteksi perubahan yang tidak biasa pada pola pengiriman email Anda.
- Filter spam kami menemukan karakteristik email Anda yang biasanya dari konten yang tidak diminta atau berkualitas rendah.

Ketika kami menempatkan akun Anda dalam peninjauan atau menjeda kemampuan akun Anda untuk mengirim email, kami mengirimkan notifikasi kepada Anda. Dalam kebanyakan kasus, notifikasi ini berisi informasi tentang masalah, dan memberikan informasi tentang langkah berikutnya yang dapat Anda ambil.

T3. Apa yang dimaksud dengan email “tidak diminta”?

Email yang tidak diminta adalah email yang penerimanya tidak meminta secara eksplisit untuk menerima email. Ini termasuk kasus ketika penerima mendaftar untuk tipe surat tertentu (misalnya, notifikasi), dan sebagai gantinya dikirim tipe surat yang berbeda (misalnya, iklan).

Ketika kami menempatkan akun Anda dalam peninjauan atau menjeda kemampuan akun Anda untuk mengirim email, kami mengirimkan notifikasi kepada Anda. Jika Anda menerima pemberitahuan yang menyatakan bahwa kami mengambil salah satu tindakan ini karena masalah dengan email yang tidak diminta, masuk ke AWS Konsol dan buka Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, sertakan informasi berikut:

- Apakah semua pesan yang Anda kirim secara khusus diminta oleh penerima, dan apakah pesan tersebut mematuhi Kebijakan Penggunaan yang Diterima AWS?
- Apakah Anda telah memperoleh alamat email dengan cara lain selain pelanggan yang secara khusus berinteraksi dengan Anda atau situs web Anda dan meminta email dari sana? Anda harus menjelaskan bagaimana Anda memperoleh milis Anda.
- Bagaimana cara kerja proses berlangganan dan berhenti berlangganan Anda? Anda harus menyertakan tautan opt-in dan opt-out.

T4. Apa yang harus saya lakukan jika saya menerima notifikasi yang menyatakan bahwa akun saya dalam peninjauan atau pengiriman saya dijeda karena peninjauan manual?

Identifikasi penyebab masalah, lalu perbaiki. Setelah Anda membuat perubahan yang Anda yakini akan menyelesaikan masalah, masuk ke AWS Konsol dan pergi ke Pusat Dukungan. Balas kasus yang kami buka atas nama Anda. Dalam pesan Anda, berikan informasi detail tentang langkah-langkah yang telah Anda ambil untuk menyelesaikan masalah, dan jelaskan cara langkah-langkah tersebut mencegah masalah terjadi lagi di masa mendatang. Jika kami setuju bahwa perubahan yang telah Anda buat mengatasi masalah tersebut dengan tepat, kami akan membatalkan periode peninjauan pada akun Anda.

T5. Tipe masalah apa yang Anda lihat sebagai “dapat diperbaiki?”

Umumnya, kami yakin situasi dapat diperbaiki jika Anda memiliki riwayat praktik pengiriman yang baik, dan jika ada langkah-langkah yang dapat Anda ambil untuk menghilangkan pengiriman yang bermasalah sambil melanjutkan sebagian besar pengiriman Anda. Contohnya, jika Anda

mengirim tiga tipe email yang berbeda dan hanya satu tipe yang bermasalah, Anda mungkin dapat menghentikan pengiriman yang bermasalah dan melanjutkan pengiriman lainnya.

T6. Bagaimana jika saya tidak dapat menemukan sumber masalahnya?

Anda dapat masuk ke AWS Console dan pergi ke Support Center. Balas kasus yang kami buka atas nama Anda dan minta sampel surat yang menyebabkan masalah.

FAQ DNS Blackhole List (DNSBL)

Domain Name System-based Blackhole List (DNSBL)—terkadang disebut sebagai Realtime Blackhole List (RBL), daftar tolak, daftar blokir, atau daftar hitam—yang dimaksudkan untuk memberi tahu penyedia email tentang alamat IP yang diduga mengirimkan email yang tidak diinginkan.

DNSBL yang berbeda memiliki dampak yang berbeda pada kemampuan pengiriman email. Topik ini menjelaskan bagaimana DNSBL memengaruhi pengiriman email yang Anda kirim menggunakan Amazon SES, serta kebijakan kami untuk menghapus alamat IP Amazon SES dari DNSBL.

> **Note**
>
> Topik ini membahas DNSBL yang digunakan penyedia email untuk memblokir pesan masuk. Untuk informasi tentang cara Amazon SES memblokir email keluar yang dikirim ke penerima tempat alamat email telah menghasilkan pentalan sebelumnya, lihat Daftar penindasan SES global Amazon.

Q1. Bagaimana DNSBL memengaruhi penyampaian email?

DNSBL yang berbeda memiliki dampak berbeda pada keberhasilan penyampaian pesan. Penyedia email besar—termasuk Gmail, Hotmail, AOL, dan Yahoo — tampaknya mengenali sejumlah kecil DNSBL yang sangat diperhitungkan, seperti yang ditawarkan oleh Spamhaus. Berdasarkan pengalaman kami, DNSBL lain cenderung memiliki dampak yang rendah, meskipun beberapa sistem email menekankan DNSBL tertentu daripada yang lain.

Akhirnya, banyak penyedia email memiliki daftar tolak internal mereka sendiri. Penyedia email menjaga daftar ini dengan sangat ketat, dan jarang membaginya dengan publik. Jika alamat IP ada di salah satu daftar ini, alamat IP tersebut dapat berdampak besar pada kemampuan Anda mengirim email ke penerima yang menggunakan penyedia tersebut.

Q2. Bagaimana alamat IP berakhir di DNSBL?

Ada beberapa cara alamat IP dapat berakhir pada DNSBL. Alamat IP dapat ditambahkan ke DNSBL ketika mereka mengirim email ke jebakan spam. Jebakan spam adalah alamat email yang bukan milik pengguna manusia. Jebakan spam ada hanya untuk mengumpulkan spam dan mengidentifikasi spammer. Beberapa DNSBL juga mengizinkan pengguna individu untuk mengirimkan alamat IP. Beberapa DNSBL bahkan mengizinkan pengguna untuk mengirimkan seluruh rentang alamat IP. DNSBL lain dikelola melalui kontribusi oleh administrator email, dan dapat menyertakan alamat IP yang dipercaya bahwa administrator menyalahgunakan sistem mereka sendiri.

Q3. Bagaimana Amazon SES mencegah alamat IP-nya muncul di DNSBL?

Sistem kami mencari tanda-tanda penyalahgunaan. Jika kami mendeteksi pola pengiriman atau karakteristik lain yang dapat menyebabkan alamat IP ditambahkan ke DNSBL, kami mengirimkan notifikasi ke pengirim. Jika situasinya sangat serius, atau jika pengirim tidak memperbaiki masalah setelah kami mengirim notifikasi, kami akan menjeda kemampuan pengirim untuk mengirim email hingga mereka menyelesaikan masalah tersebut. Memberlakukan kebijakan pengiriman kami dengan cara ini untuk membantu mengurangi kemungkinan bahwa alamat IP kami berakhir di DNSBL.

T4. Bisakah Amazon SES menghapus alamat IP-nya dari DNSBL?

Kami secara aktif memantau DNSBL yang dapat memengaruhi pengiriman di seluruh layanan Amazon SES, atau yang dapat memengaruhi kemampuan untuk mengirim email ke penerima yang menggunakan penyedia email besar, seperti Gmail, Yahoo, AOL, dan Hotmail. DNSBL yang ditawarkan oleh Spamhaus termasuk dalam kategori ini. Ketika salah satu alamat IP kami muncul pada daftar yang memenuhi salah satu dari kriteria tersebut, kami segera mengambil tindakan agar alamat tersebut dihapus dari DNSBL secepatnya.

Kami tidak memantau DNSBL yang tampaknya tidak mungkin untuk memengaruhi pengiriman di seluruh layanan Amazon SES, atau yang tidak memiliki dampak besar pada pengiriman ke penyedia email besar. DNSBL yang ditawarkan oleh SORBS dan UCEPROTECT termasuk dalam kategori ini. Karena praktik pendaftaran dan penghapusan khusus dari vendor yang mengoperasikan daftar ini, kami tidak dapat menghapus alamat IP kami dari daftar ini.

T5. Penyedia email menolak email saya dikarenakan pengiriman alamat IP sudah dicantumkan oleh DNSBL, selain Spamhaus. Apa yang bisa saya lakukan?

Pertama, konfirmasi bahwa pesan benar-benar diblokir karena DNSBL. Jika email Anda ditolak karena pengiriman alamat IP telah ditambahkan ke DNSBL, maka Anda akan menerima notifikasi pentalan yang menyebutkan nama penyedia DNSBL, seperti dalam contoh berikut:

```
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
```

Jika Anda menerima notifikasi pentalan, tetapi tidak berisi informasi yang mirip dengan pesan yang ditampilkan dalam contoh sebelumnya, maka kemungkinan besar penyedia email menolak pesan Anda dengan alasan yang tidak terkait dengan ditambahkannya alamat IP ke DNSBL.

Jika Anda dapat mengonfirmasi bahwa penyedia email memblokir email Anda karena pengiriman alamat IP berada di DNSBL, ada beberapa hal yang dapat Anda lakukan:

- Hubungi postmaster domain yang menolak pesan Anda untuk meminta pengecualian dari kebijakan pemfilteran spam mereka. Beberapa postmaster memiliki proses dukungan, dan dapat menerbitkan halaman postmaster yang menjelaskan proses ini. Jika domain yang Anda coba hubungi tidak menerbitkan kebijakan yang mendukung postmaster-nya, maka Anda dapat menghubungi postmaster dengan mengirim email ke postmaster@*example.com*, dengan *example.com* adalah domain yang dimaksud. Domain diperlukan oleh RFC 5321 untuk memiliki kotak surat postmaster.

  Ketika Anda menghubungi postmaster, berikan kode pentalan yang Anda terima, header email yang Anda coba kirim, pengukuran dampak DNSBL terhadap pengiriman email Anda, dan informasi tentang alasan Anda meyakini bahwa email Anda diblokir dengan tidak semestinya. Semakin banyak informasi yang dapat Anda berikan kepada postmaster untuk menunjukkan bahwa Anda mengirim email yang sah, akan semakin besar kemungkinan postmaster membuat pengecualian bagi Anda.
- Jika penyedia email tidak merespons, atau tidak mau mengubah kebijakan mereka, pertimbangkan untuk menggunakan alamat IP khusus. Alamat IP khusus adalah alamat yang hanya dapat digunakan oleh Anda. Dengan menerapkan praktik pengiriman yang baik, Anda dapat menjaga tingkat keterlibatan Anda tetap tinggi, dan tingkat pentalan, aduan, serta jebakan spam Anda tetap

rendah. Praktik pengiriman yang baik dapat membantu memastikan alamat Anda tidak berakhir di DNSBL.

T6. Email yang saya kirim ke Gmail, Yahoo, Hotmail, atau penyedia besar lainnya sedang dikirim ke folder spam. Apakah ini terjadi karena alamat IP pengiriman saya ada di DNSBL?

Mungkin tidak. Jika alamat IP sudah terdaftar oleh DNSBL dengan dampak signifikan, seperti salah satu DNSBL dari Spamhaus, penyedia email besar akan menolak email dari alamat IP tersebut sepenuhnya, dibandingkan mengirimkannya ke folder spam.

Ketika penyedia email besar menerima email (bukan menolaknya), mereka biasanya mempertimbangkan keterlibatan pengguna saat mempertimbangkan jika akan menempatkan pesan di kotak masuk atau di folder spam. Keterlibatan pengguna mengacu pada cara pengguna berinteraksi dengan pesan yang Anda kirimkan sebelumnya.

Untuk meningkatkan kemungkinan bahwa pesan Anda akan sampai ke kotak masuk pelanggan, Anda harus menerapkan semua praktik terbaik berikut ini:

- Tidak pernah menyewa atau membeli daftar alamat email. Menyewa atau membeli daftar merupakan pelanggaran Kebijakan Penggunaan yang dapat Diterima AWS (AUP) dan tidak diizinkan di Amazon SES dalam kondisi apa pun.
- Hanya mengirim email ke pelanggan yang secara eksplisit meminta untuk mendapatkan email dari Anda. Di berbagai negara serta yurisdiksi di seluruh dunia, mengirim email ke penerima yang secara eksplisit tidak setuju untuk menerima email dari Anda merupakan hal yang ilegal.
- Berhenti mengirimkan email ke pelanggan yang belum membuka atau mengeklik tautan dalam pesan yang dikirim dalam 30-90 hari terakhir. Langkah ini dapat membantu menjaga tingkat keterlibatan Anda tetap tinggi, yang meningkatkan peluang bahwa pesan yang Anda kirim akan sampai di kotak masuk penerima di kemudian hari.
- Gunakan elemen desain dan gaya penulisan yang konsisten di setiap pesan yang Anda kirim guna memastikan bahwa pelanggan dapat dengan mudah mengidentifikasi pesan dari Anda.
- Gunakan mekanisme autentikasi email, seperti SPF dan DKIM.
- Ketika pelanggan menggunakan formulir web untuk berlangganan konten Anda, kirimkan email kepada mereka untuk mengonfirmasi bahwa mereka ingin menerima email dari Anda. Jangan mengirimkan email tambahan kepada mereka hingga mereka mengonfirmasi bahwa mereka

ingin menerima email dari Anda. Proses ini dikenal sebagai keikutsertaan telah dikonfirmasi atau keikutsertaan ganda.

- Permudah pelanggan Anda untuk berhenti berlangganan, dan segera terima permintaan untuk berhenti berlangganan.
- Jika Anda mengirim email yang berisi tautan, periksa tautan tersebut terhadap Domain Block List (DBL) Spamhaus. Untuk menguji tautan Anda, gunakan Alat Pencarian Domain di situs web Spamhaus.

Dengan menerapkan praktik ini, Anda dapat meningkatkan reputasi pengirim, yang meningkatkan kemungkinan bahwa email yang Anda kirim akan sampai ke kotak masuk penerima. Dengan menerapkan praktik ini juga dapat membantu menjaga tingkat pentalan dan aduan akun Anda tetap rendah, dan mengurangi risiko pengiriman email ke jebakan spam.

FAQ metrik pengiriman email Amazon SES

Amazon SES mengumpulkan beberapa metrik tentang email yang Anda kirim. Metrik ini memungkinkan Anda menganalisis keefektifan program email dan memantau statistik penting, seperti tingkat pentalan dan aduan Anda.

Bagian ini berisi FAQ tentang topik berikut terkait dengan metrik pengiriman email:

- Pertanyaan Umum
- Pelacakan Buka
- Pelacakan Klik

> **Note**
>
> Pelacakan peristiwa tergantung pada penyedia layanan email penerima (ESP) dan bagaimana mereka mengonfigurasi pengaturan privasi mereka yang berada di luar kendali Amazon SES. Hitungan peristiwa pelacakan dapat miring (mengembalikan jumlah yang tidak akurat) dalam kondisi seperti:
>
> - Penerima email menggunakan penyedia layanan email (ESP) yang melindungi privasi mereka.
> - Penerima email secara eksplisit tidak memberikan izin ESP mereka untuk membagikan data mereka.

- ESP penerima email menyimpan gambar atau tautan, SES hanya dapat menghitung pembukaan awal, tetapi tidak akan dapat menghitung bukaan berikutnya.

Pertanyaan Umum

T1. Setelah email terkirim, berapa lama Amazon SES terus mengumpulkan metrik buka dan klik?

Amazon SES mengumpulkan metrik buka dan klik selama 60 hari setelah email dikirim.

T2. Jika pengguna membuka email beberapa kali, atau mengeklik tautan dalam email beberapa kali, apakah setiap peristiwa tersebut dilacak secara terpisah?

Jika penerima membuka email beberapa kali, Amazon SES menghitung setiap pembukaan sebagai peristiwa buka yang unik. Demikian pula, jika penerima mengeklik tautan yang sama beberapa kali, Amazon SES menghitung setiap klik sebagai peristiwa klik yang unik. Namun, jumlah ini dapat dimiringkan oleh skenario yang diuraikan di atas dalam kotak catatan.

T3. Apakah metrik buka dan klik dikumpulkan, atau dapatkah metrik tersebut diukur hingga ke tingkat penerima?

Setiap pembukaan dan pengeklikan dilacak di tingkat penerima. Dengan pelacakan buka dan klik, Anda dapat menentukan penerima mana yang membuka email atau mengeklik tautan di email.

T4. Dapatkah saya mengambil metrik buka dan klik menggunakan API Amazon SES?

API Amazon SES tidak menyediakan metode untuk mengambil metrik buka dan klik. Namun, Anda dapat mengambil metrik terbuka dan klik untuk Amazon SES menggunakan API. CloudWatch Misalnya, Anda dapat menggunakan AWS CLI untuk mengambil metrik klik menggunakan CloudWatch API dengan mengeluarkan perintah berikut:

```
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
```

Perintah di atas menunjukkan pengambilan jumlah total kejadian klik untuk setiap hari di tahun 2017. Untuk mengambil metrik buka, ubah nilai parameter `metric-name` ke `Open`. Anda juga

dapat memodifikasi parameter `start-time` dan `end-time` untuk mengubah periode analisis, atau mengubah parameter `period` untuk analisis yang lebih detail.

Pelacakan Buka

T1. Bagaimana cara kerja pelacakan buka?

Gambar GIF transparan 1 piksel kali 1 piksel disisipkan di setiap email yang dikirim melalui Amazon SES dan menyertakan referensi unik ke file gambar ini; ketika gambar diunduh, SES dapat mengetahui dengan tepat pesan mana yang dibuka dan oleh siapa.

Secara default, piksel ini disisipkan di bagian bawah email; namun, beberapa aplikasi penyedia email memotong pratinjau email ketika melebihi ukuran tertentu dan dapat memberikan tautan untuk melihat sisa pesan. Dalam skenario ini, gambar pelacakan piksel SES tidak dimuat dan akan membuang tarif terbuka yang Anda coba lacak. Untuk menyiasatinya, Anda dapat secara opsional menempatkan piksel di awal email, atau di mana pun, dengan memasukkan `{{ses:openTracker}}` placeholder ke dalam badan email. Setelah SES menerima pesan dengan placeholder, itu akan diganti dengan gambar piksel pelacakan terbuka.

> **Important**
>
> Cukup tambahkan satu `{{ses:openTracker}}` placeholder, karena lebih dari satu akan menghasilkan kode `400 BadRequestException` kesalahan dikembalikan.

Penambahan piksel pelacakan ini tidak mengubah tampilan email Anda.

T2. Apakah pelacakan buka diaktifkan secara default?

Pelacakan buka tersedia untuk semua pengguna Amazon SES secara default. Untuk menggunakan pelacakan buka, Anda harus melakukan hal berikut:

1. Buat set konfigurasi.
2. Dalam set konfigurasi, buatlah tujuan peristiwa.
3. Konfigurasikan tujuan peristiwa untuk menerbitkan notifikasi peristiwa buka ke tujuan.
4. Dalam setiap email yang peristiwa bukanya ingin Anda lacak, tentukan set konfigurasi yang Anda buat di langkah 1.

Untuk detail tentang cara mengaktifkan pelacakan terbuka melalui tujuan acara set konfigurasi, lihatthe section called “Buat tujuan acara”. Anda dapat menggunakan placeholder piksel di email SMTP dengan cara seperti email yang diformat, mentah, dan template.

Pelajari lebih lanjut tentang cara Pantau pengiriman email menggunakan penerbitan acara.

T3. Dapatkah saya menghilangkan piksel pelacakan buka dari email tertentu?

Ada dua cara untuk menghilangkan piksel pelacakan buka dari email Anda. Metode pertama adalah mengirim email tanpa menentukan set konfigurasi. Atau, Anda dapat menentukan set konfigurasi yang tidak dikonfigurasi untuk mempublikasikan data tentang peristiwa buka.

P4. Apakah Anda melacak buka untuk email teks biasa?

Pelacakan buka hanya berhasil dengan email HTML. Karena pelacakan buka tergantung pada penyertaan citra, metrik terbuka tidak mungkin dikumpulkan untuk pengguna yang membuka email menggunakan klien email hanya teks (non-HTML).

Pelacakan Klik

T1. Bagaimana cara kerja pelacakan klik?

Untuk melacak klik, Amazon SES memodifikasi setiap tautan di badan email. Ketika penerima membuka tautan, mereka dikirim ke server Amazon SES, dan segera diteruskan ke alamat tujuan. Seperti halnya pelacakan buka, setiap tautan pengalihan bersifat unik. Hal ini memungkinkan Amazon SES untuk menentukan penerima mana yang mengeklik tautan, waktu mereka mengekliknya, dan dari email yang mana mereka tiba di tautan.

> **Important**
>
> Jika Anda mengirim satu pesan ke beberapa penerima, setiap penerima akan menyimpan tautan pelacakan klik yang sama. Untuk melacak aktivitas klik pada masing-masing penerima, kirim email ke satu penerima per operasi pengiriman.

T2. Dapatkah saya menonaktifkan pelacakan klik?

Anda dapat menonaktifkan pelacakan klik untuk tautan individual dengan menambahkan atribut, `ses:no-track`, ke tanda jangkar di badan email HTML Anda. Contohnya, jika Anda menautkan ke halaman beranda AWS, tautan jangkar yang normal akan menyerupai berikut ini:

```
<a href="https://aws.amazon.com">Amazon Web Services</a>
```

Untuk menonaktifkan pelacakan klik untuk tautan tersebut, modifikasi tautan tersebut menyerupai berikut ini:

```
<a ses:no-track href="aws.amazon.com">Amazon Web Services</a>
```

Karena `ses:no-track` bukan atribut HTML standar, Amazon SES secara otomatis menghapusnya dari versi email yang sampai di kotak masuk penerima Anda.

Anda juga dapat menonaktifkan pelacakan klik untuk semua pesan yang Anda kirim menggunakan set konfigurasi tertentu. Untuk menonaktifkan pelacakan klik, modifikasi tujuan peristiwa set konfigurasi tujuan acara sehingga tidak menangkap peristiwa klik.

Untuk detail tentang cara mengaktifkan dan menonaktifkan pelacakan klik melalui tujuan acara set konfigurasi, lihatthe section called “Buat tujuan acara”.

Pelajari lebih lanjut tentang cara Pantau pengiriman email menggunakan penerbitan acara.

T3. Berapa banyak tautan yang dapat dilacak pada setiap email?

Sistem pelacakan klik dapat melacak maksimum 250 tautan.

T4. Apakah metrik klik dikumpulkan untuk tautan dalam email teks biasa?

Anda hanya dapat melacak klik di email HTML.

T5. Dapatkah saya menandai tautan dengan pengenal unik?

Anda dapat menambahkan tanda dengan jumlah yang tidak terbatas, sebagai pasangan nilai kunci, untuk tautan di email Anda menggunakan atribut `ses:tags`. Ketika Anda menggunakan atribut ini, tentukan kunci dan nilai menggunakan format yang sama dengan yang akan Anda gunakan untuk meneruskan properti inline CSS: ketikkan kuncinya, diikuti oleh titik dua (:), lalu diikuti oleh nilainya. Jika Anda perlu meneruskan beberapa pasangan kunci-nilai, pisahkan masing-masing pasangan dengan titik koma (;).

Misalnya, anggaplah Anda ingin menambahkan tanda `product:book, genre:fiction, subgenre:scifi, type:newrelease` ke sebuah tautan. Tautan yang dihasilkan mirip seperti berikut ini:

```
<a ses:tags="product:book;genre:fiction;subgenre:scifi;type:newrelease;"
    href="http://www.amazon.com/…/">New Releases in Science Fiction</a>
```

Tanda ini diteruskan ke tujuan penerbitan peristiwa Anda sehingga Anda dapat melakukan analisis tambahan pada tautan tertentu yang diklik oleh pengguna Anda.

> **Note**
>
> Tanda tautan dapat meliputi angka 0–9, huruf A–Z (huruf besar dan huruf kecil), tanda hubung (-), dan garis bawah (_).

T6. Apakah tautan yang dilacak menggunakan protokol HTTP atau HTTPS?

Tautan pelacakan menggunakan protokol yang sama dengan tautan asli di email Anda.

Misalnya, jika email Anda menyertakan tautan ke `https://www.amazon.com`, tautan tersebut diganti dengan tautan pelacakan yang menggunakan protokol HTTPS. Jika email Anda menyertakan tautan ke `http://www.example.com`, tautan diganti dengan tautan pelacakan yang menggunakan HTTP. Jika email Anda menyertakan kedua tautan yang disebutkan sebelumnya, tautan HTTPS diganti dengan tautan pelacakan yang menggunakan protokol HTTPS, dan tautan HTTP diganti dengan tautan pelacakan yang menggunakan protokol HTTP.

T7. Tautan di email saya tidak bisa dilacak. Mengapa tidak bisa?

Amazon SES mengharapkan tautan di email Anda berisi URL yang dikodekan dengan benar. Khususnya, URL dalam tautan Anda harus sesuai dengan RFC 3986. Jika tautan di email tidak dikodekan dengan benar, penerima masih akan melihat tautan di email, tetapi Amazon SES tidak akan melacak peristiwa klik untuk tautan tersebut.

Masalah yang terkait dengan pengodean yang tidak tepat biasanya terjadi di URL yang berisi string kueri. Contohnya, jika URL tautan di email berisi karakter spasi yang tidak dikodekan dalam string kueri (seperti spasi di antara "John" dan "Doe" pada contoh berikut: http://www.example.com/path/to/page?name=John Doe), Amazon SES tidak akan melacak tautan tersebut. Namun, jika URL menggunakan karakter spasi yang dikodekan sebagai gantinya (seperti "%20" dalam contoh berikut: http://www.example.com/path/to/page?name=John%20Doe), Amazon SES melacaknya seperti yang diharapkan.

Indeks Cari Cepat

Indeks berikut telah dibuat untuk membantu Anda dengan cepat menemukan hal-hal di Amazon SES dengan menyediakan dua cara pencarian - baik dengan cara atau konsep. Cara menggambarkan “bagaimana” melakukan sesuatu sementara konsep menjelaskan gambaran yang lebih besar.

Beri tahu kami pendapat Anda

Silakan gunakan tombol Umpan Balik di sudut kanan atas untuk memberi tahu kami...

- Apakah indeks ini bermanfaat?
- Apakah ada cara atau konsep yang ingin Anda lihat ditambahkan ke indeks ini?
- Apakah ada sesuatu yang menurut Anda seharusnya dikategorikan secara berbeda?

Tautan Cara & konsep SES

How-tos

Tautan cara SES tercantum menurut abjad dan akan membawa Anda ke bagian yang sesuai untuk menunjukkan kepada Anda “bagaimana” melakukan tindakan yang Anda pilih.

- Pelajari cara...
  - Menambahkan Rekaman SPF sebagai bagian dari pengaturan domain MAIL FROM kustom
  - Tetapkan kolam IP
  - Blokir SPAM untuk menerima email
  - Konfigurasikan domain buka/klik khusus
  - Konfigurasikan notifikasi SNS
  - Connect ke endpoint SMTP
  - Buat set konfigurasi
  - Buat identitas domain
  - Membuat identitas alamat email
  - Buat tujuan acara
  - Buat filter alamat IP
  - Buat kumpulan IP terkelola untuk mengaktifkan IP khusus (dikelola)

- Buat aturan tanda terima
- Buat alarm reputasi menggunakan CloudWatch
- Membuat kebijakan otorisasi pengiriman menggunakan kebijakan khusus
- Membuat kebijakan otorisasi pengiriman menggunakan pembuat kebijakan
- Buat kolam IP khusus standar untuk alamat IP khusus (standar)
- Menghapus identitas
- Hapus data pribadi
- Mengedit identitas
- Aktifkan penerusan umpan balik email
- Metrik reputasi ekspor
- Keluar dari kotak pasir
- Memulai dengan SES
- Memulai dengan Virtual Deliverability Manager
- Berikan izin untuk menerima email
- Meningkatkan throughput
- Tingkatkan kuota pengiriman Anda
- Integrasikan dengan server email yang ada
- Log panggilan API
- Mengelola set konfigurasi
- Mengelola Easy DKIM & BYODKIM
- Pantau kirim dan metrik reputasi
- Pantau statistik pengiriman
- Pantau statistik penggunaan
- Pantau kuota pengiriman Anda
- Mendapatkan catatan DKIM untuk identitas
- Dapatkan kredensi SMTP
- Ganti penekanan tingkat akun dengan penekanan set-level konfigurasi
- Ganti penandatanganan DKIM yang diwarisi pada identitas alamat email
- Jeda pengiriman email
- Publikasikan catatan MX

- Laporkan penyalahgunaan sumber AWS daya
- Minta alamat IP khusus
- Minta dukungan teknis
- Selesaikan masalah pengiriman & reputasi menggunakan penasihat Virtual Deliverability Manager
- Mengambil data peristiwa dari CloudWatch
- Mengambil data peristiwa dari Kinesis Data Firehose
- Mengambil data peristiwa dari SNS
- Mengirim email menggunakan AWS SDK
- Kirim email secara terprogram
- Kirim email menggunakan SES API
- Kirim email menggunakan SMTP
- Kirim email mentah dengan lampiran menggunakan CLI atau SES API
- Kirim email pengujian menggunakan simulator kotak surat
- Mengatur BYODKIM (Bawa DKIM Anda Sendiri)
- Menyiapkan kebijakan DMARC
- Mengatur Easy DKIM
- Mengatur penerimaan email
- Menyiapkan penerbitan acara
- Menyiapkan domain MAIL DARI
- Mengatur otorisasi pengiriman (tugas pemilik identitas)
- Mengatur otorisasi pengiriman (tugas pengirim delegasi)
- Tentukan set konfigurasi saat mengirim email
- Uji koneksi Anda ke antarmuka SMTP
- Lacak tingkat bouncing dan keluhan
- Memahami properti penandatanganan DKIM yang diwarisi
- Gunakan metrik reputasi
- Gunakan paket perangkat lunak untuk mengirim email
- Gunakan manajemen berlangganan
- Gunakan template untuk mengirim email

- Gunakan daftar penindasan tingkat akun Anda
- Verifikasi identitas domain
- Verifikasi identitas alamat email
- Lihat identitas
- Lihat metrik pengiriman akun tingkat tinggi & terperinci menggunakan dasbor Virtual Deliverability Manager
- Lihat metrik SNDS untuk IP khusus
- Lakukan pemanasan alamat IP khusus

Concepts

Tautan konsep SES terdaftar menurut abjad dan akan membawa Anda ke bagian dan bagian yang sesuai untuk menjelaskan konsep yang Anda pilih.

- Temukan informasi tentang...
  - Penyalahgunaan sumber AWS daya, laporkan
  - Dasbor akun
  - Daftar penindasan tingkat akun
  - Opsi tindakan untuk menerima email
  - Tambahkan tindakan header
  - Jenis lampiran, tidak didukung
  - Tindakan respons pantulan, kembali
  - BYODKIM (Bawa DKIM Anda Sendiri)
  - BYOIP (Bawa IP Anda Sendiri)
  - Contoh kode
  - Validasi kepatuhan
  - Penindasan set-level konfigurasi
  - Set konfigurasi
  - Pengkodean konten
  - Dukungan lama pemberitahuan lintas akun
  - KUSTOM MAIL DARI domain
  - Perlindungan data

- Alamat IP khusus
- Alamat IP khusus (dikelola)
- Alamat IP khusus (standar)
- DKIM, mengautentikasi email dengan
- DMARC (Otentikasi, Pelaporan, dan Kesesuaian Pesan Berbasis Domain)
- DMARC melalui DKIM, sesuai dengan
- DMARC melalui SPF, sesuai dengan
- DKIM Mudah
- Tujuan penerusan umpan balik email
- Email menerima otentikasi
- Konsep penerimaan email
- Panduan konsol penerima email
- Email menerima pemindaian malware
- Email menerima izin
- Kasus penggunaan penerimaan email
- Batasan penerimaan email
- Metode otentikasi pengiriman email
- Titik akhir
- Pemberitahuan acara
- Pemberitahuan acara melalui email
- Pemberitahuan acara melalui SNS
- Penerbitan acara
- FAQ (Pertanyaan yang Sering Diajukan)
- Daftar penindasan global
- Bidang header, didukung
- Identitas, mengelola
- Manajemen identitas dan akses
- Keamanan infrastruktur
- Integrasikan dengan WorkMail tindakan Amazon

- Kontrol berbasis IP menggunakan filter alamat IP

- Tindakan fungsi Lambda, panggil
- Manajemen daftar
- Daftar dan langganan
- Penebangan dan pemantauan
- Deteksi malware
- Penandatanganan manual DKIM
- Pantau pengiriman email menggunakan penerbitan acara
- Pantau reputasi pengirim
- Memantau aktivitas pengiriman
- Kuota
- Aturan tanda terima
- Kontrol berbasis penerima menggunakan aturan tanda terima
- Daerah
- Metrik reputasi
- Pesan metrik reputasi
- Ketahanan
- Aksi bucket S3, kirimkan ke
- Sandbox - keluar dari
- Keamanan
- Protokol keamanan, didukung
- Mengirim otorisasi
- Mengirim anatomi kebijakan otorisasi
- Mengirim contoh kebijakan otorisasi
- Mengirim proses otorisasi
- Metrik SNDS untuk IP khusus
- Konten pemberitahuan SNS
- Contoh pemberitahuan SNS
- Tindakan topik SNS, publikasikan ke
- SPF (Kerangka Kebijakan Pengirim)
- Hentikan tindakan pengaturan aturan

- Manajemen berlangganan
- Support, minta teknis
- Template untuk verifikasi email khusus
- Pemecahan masalah
- Identitas terverifikasi
- Manajer Pengiriman Virtual
- Titik akhir VPC

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.