Amazon Simple Email Service

Developer Guide

API Version 2010-12-01

Amazon Simple Email Service: Developer Guide

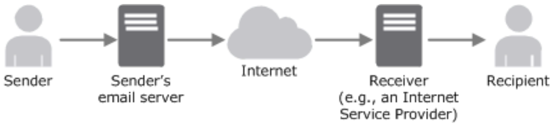
Table of Contents

What Is Amazon SES?

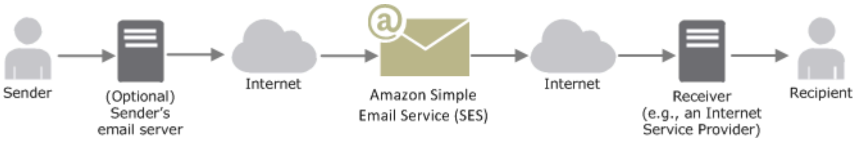
Welcome to the Amazon Simple Email Service (Amazon SES) Developer Guide. Amazon SES is an outbound-only email-sending service that provides an easy, cost-effective way for you to send email. You can use Amazon SES to send marketing emails such as special offers, transactional emails such as order confirmations, and other types of correspondence such as newsletters. You only pay for what you use, so you can send as much or as little email as you like. For more service highlights and pricing information, go to the Amazon Simple Email Service Detail Page.

Where does Amazon SES fit in?

When you send an email, you are sending it through some type of outbound email server. That email server might be provided by your Internet service provider (ISP), your company's IT department, or you might have set it up yourself. The email server accepts your email content, formats it to comply with email standards, and then sends the email out over the Internet. The email may pass through other servers until it eventually reaches a receiver (an entity, such as an ISP, that receives the email on behalf of the recipient). The receiver then delivers the email to the recipient. The following diagram illustrates the basic email-sending process.



When you use Amazon SES, Amazon SES becomes your outbound email server. You can also keep your existing email server and configure it to send your outgoing emails through Amazon SES so that you don't have to change any settings in your email clients. The following diagram shows where Amazon SES fits in to the email-sending process.


Sender
(Optional) Sender's email server
Internet
Amazon Simple Email Service (SES)
Internet
Receiver (e.g., an Internet Service Provider)
Recipient

A sender can generate the email content in different ways. A sender can create the email by using an email client application, or use a program that automatically generates emails, like an application that sends order confirmations in response to purchase transactions.

Why use Amazon SES?

When you use Amazon SES, you can eliminate the complexity and expense of building an in-house email solution or licensing, installing, and operating a third-party email service. Another important factor in any email-sending effort is deliverability—the percentage of your emails that arrive in your recipients' inboxes. ISPs use filters to detect email messages that appear to be spam (unsolicited, undesired emails) and prevent these messages from being delivered. Even if your email is legitimate, an ISP's spam filter could falsely identify your email as spam, and put it in the recipient's junk folder or block it entirely.

If you are not using Amazon SES, you need to take several steps to reduce the likelihood that your emails will be marked as spam. For example, you need to examine the content of your emails to make sure they don't contain material that is typically flagged as questionable. You need to build trust with ISPs so that the ISPs have high confidence that you are sending high-quality emails, and therefore are less likely to block emails coming from you. You need to ensure that you don't send too many emails too soon because sudden spikes in email-sending volume or rate may cause ISPs to block your emails. Amazon SES takes care of all of these tasks for you to maximize the deliverability of your emails.

Another aspect that you would need to manage yourself is to keep track of bounces (email delivery failures that occur, for example, if an email address does not exist) and complaints (emails that recipients mark as spam, for example, by clicking "Mark as spam" in their email client). To receive complaint feedback, you would need to set up feedback loops with individual ISPs. Amazon SES already has feedback loops set up with certain major ISPs, and automatically forwards the information to you. Amazon SES also forwards bounce notifications to you either by email or through Amazon Simple Notification Service (Amazon SNS) and provides you with real-time access to your delivery metrics—the number of emails you have sent and the number that have bounced or generated complaints—to help guide your email-sending strategy.

How do I send emails using Amazon SES?

There are several ways that you can send an email by using Amazon SES. You can use the Amazon SES console, the Simple Mail Transfer Protocol (SMTP) interface, or you can call the Amazon SES API.

- **Amazon SES console—**This method is the quickest way to set up your system and send a couple of test emails, but once you are ready to start your email campaign, you will use the console primarily to monitor your sending activity. For example, you can quickly view the number of emails that you have sent and the number of bounces and complaints that you have received.
- **SMTP Interface—**There are two ways to access Amazon SES through the SMTP interface. The first way, which requires no coding, is to configure any SMTP-enabled software to send email through Amazon SES. For example, you can configure your existing email client or software program to connect to the Amazon SES SMTP endpoint instead of your current outbound email server.

  The second way is to use an SMTP-compatible programming language such as Java and access the Amazon SES SMTP interface by using the language's built-in SMTP functions and data types.
- **Amazon SES API—**You can call the Amazon SES Query (HTTPS) interface directly, or you can use the AWS Command Line Interface, the AWS Tools for Windows PowerShell, or an AWS Software Development Kit (SDK). The AWS SDKs wrap the low-level functionality of the Amazon SES API with higher-level data types and function calls that take care of the details for you. The AWS SDKs provide not only Amazon SES operations, but also basic AWS functionality such as request authentication, request retries, and error handling. AWS SDKs and resources are available for Android, iOS, Java, .NET, Node.js, PHP, Python, and Ruby.

Amazon SES and other AWS services

Amazon SES integrates seamlessly with other AWS products. For example, you can:

- Add email capabilities to any application that runs on an Amazon EC2 instance by using the AWS SDKs or the Amazon SES API. If you want to send email through Amazon SES from an Amazon EC2 instance, you can get started with Amazon SES for free.
- Use Elastic Beanstalk to create an email-enabled application such as a program that uses Amazon SES to send a newsletter to customers.
- Set up Amazon Simple Notification Service (Amazon SNS) to notify you of your emails that bounced, produced a complaint, or were successfully delivered to the recipient's mail server.
- Use the AWS Management Console to set up Easy DKIM, which is a way to authenticate your emails. Although you can use Easy DKIM with any DNS provider, it is especially easy to set up when you manage your domain with Amazon Route 53.
- Control user access to your email sending by using AWS Identity and Access Management (IAM).

How do I start?

If you are a first-time user of Amazon SES, we recommend that you begin by reading the following sections:

- Amazon SES Quick Start (p. 16)—Shows you how to get set up and send a test email as quickly as possible.
- Getting Started with Amazon SES (p. 18)—Shows you how to send an email by using the Amazon SES console, the SMTP interface, and an AWS SDK. Examples are provided in C# and Java.
- Amazon SES Concepts (p. 5)—Explains email deliverability concepts that you should be familiar with when you use Amazon SES.
- Sending Email with Amazon SES (p. 8)—Shows you what happens when you send an email through Amazon SES.
- Email Format and Amazon SES (p. 11)—Reviews the format of emails and identifies the information that you need to provide to Amazon SES.

Then you can learn about Amazon SES in more detail by reading the sections listed in the following table:

Section	Description
Setting up Amazon SES (p. 36)	Shows you how to sign up for AWS, get your AWS access keys, download an AWS SDK, verify email addresses or domains, and move out of the Amazon SES sandbox.
Using the Amazon SES SMTP Interface to Send Email (p. 48)	Shows you how to get your Amazon SES SMTP credentials, connect to the Amazon SES SMTP endpoint, and provides examples of how to configure email clients and software packages to send email through Amazon SES. Also explains how to configure your existing email server to send all outgoing emails through Amazon SES.
Using the Amazon SES API to Send Email (p. 79)	Shows you how to send formatted and raw emails by using the Amazon SES API. Explains how to use non-standard characters and send attachments by using the Multipurpose Internet Mail Extensions (MIME) standard when you send raw emails.

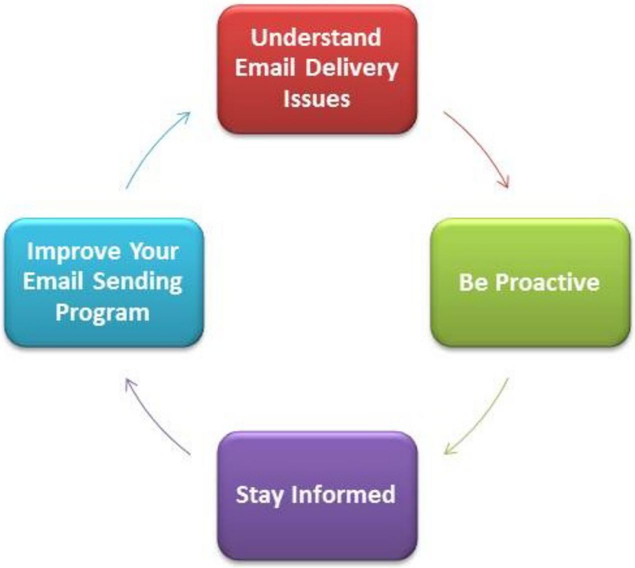
Section	Description
Regions and Amazon SES (p. 173)	Lists the Amazon SES SMTP and API endpoints for the AWS regions in which Amazon SES is available, and contains information you need to know when you use Amazon SES endpoints in multiple regions.
Authenticating Email in Amazon SES (p. 93)	Shows you how to use DKIM with Amazon SES to show ISPs that you own the domain you are sending from.
Monitoring Your Amazon SES Sending Activity (p. 104)	Shows you how to view your usage statistics (such as the number of deliveries, bounces, and complaints) and sending limits by using the Amazon SES console or by calling the Amazon SES API. Also shows you how to receive bounce and complaint notifications by email, and how to receive bounce, complaint, and delivery notifications by setting up Amazon SNS notifications.
Managing Your Amazon SES Sending Limits (p. 124)	Explains the two Amazon SES sending limits (sending quota and maximum send rate), how to increase them, and the errors you receive when you try to exceed them.
Improving Deliverability with Amazon SES (p. 177)	Provides tips about how to improve the percentage of emails that reach your recipients' inboxes. These include monitoring your sending activity and taking preventative measures to keep your bounce and complaint statistics low.
Controlling User Access to Amazon SES (p. 128)	Shows you how to use Amazon SES with AWS Identity and Access Management (IAM) to specify which Amazon SES API actions a user can perform on which Amazon SES resources.
Using Sending Authorization with Amazon SES (p. 134)	Shows you how to authorize other users to send emails from your identities on your behalf.
Testing Amazon SES Email Sending (p. 158)	Explains how to use the Amazon SES mailbox simulator to simulate common email scenarios without affecting your sending statistics such as your bounce and complaint metrics. The scenarios you can test are successful delivery, bounce, complaint, out-of-the-office (OOTO), and address on the suppression list.
Logging Amazon SES API Calls By Using AWS CloudTrail (p. 161)	Provides a list of Amazon SES APIs that can be logged using AWS CloudTrail.
Using Credentials With Amazon SES (p. 170)	Explains the types of credentials that you might use with Amazon SES, and when you might use them.
Limits in Amazon SES (p. 207)	Provides a list of limits within Amazon SES.
Troubleshooting Amazon SES (p. 180)	Explains common causes of delivery problems and provides descriptions of common Amazon SES exceptions and SMTP response codes.

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Amazon SES Concepts

You want your recipients to read your emails, find them valuable, and not label them as spam. In other words, you want to maximize email *deliverability*—the percentage of your emails that arrive in your recipients' inboxes. This topic reviews email deliverability concepts that you should be familiar with when you use Amazon SES.

To maximize email deliverability, you need to understand email delivery issues, proactively take steps to prevent them, stay informed of the status of the emails that you send, and then improve your email-sending program, if necessary, to further increase the likelihood of successful deliveries. The following sections review the concepts behind these steps and how Amazon SES helps you through the process.


Understand Email Delivery Issues
Be Proactive
Stay Informed
Improve Your Email Sending Program

Understand Email Delivery Issues

In most cases, your messages are delivered successfully to recipients who expect them. In some cases, however, a delivery might fail, or a recipient might not want to receive the mail that you are sending. Bounces, complaints, and the suppression list are related to these delivery issues and are described in the following sections.

Bounce

If your recipient's receiver (for example, an ISP) fails to deliver your message to the recipient, the receiver bounces the message back to Amazon SES. Amazon SES then notifies you of the bounced email through email or through Amazon Simple Notification Service (Amazon SNS), depending on how you have your system set up. For more information, see Using Notifications with Amazon SES (p. 105).

There are *hard bounces* and *soft bounces*, as follows:

- **Hard bounce –** A persistent email delivery failure. For example, the mailbox does not exist. Amazon SES does not retry hard bounces, with the exception of DNS lookup failures. We strongly recommend that you do not make repeated delivery attempts to email addresses that hard bounce.
- **Soft bounce –** A temporary email delivery failure. For example, the mailbox is full, there are too many connections (also called *throttling*), or the connection times out. Amazon SES retries soft bounces multiple times. If the email still cannot be delivered, then Amazon SES stops retrying it.

Amazon SES notifies you of hard bounces and soft bounces that will no longer be retried. However, only hard bounces count toward your bounce rate and the bounce metric that you retrieve using the Amazon SES console or the `GetSendStatistics` API.

Bounces can also be *synchronous* or *asynchronous*. A synchronous bounce occurs while the email servers of the sender and receiver are actively communicating. An asynchronous bounce occurs when a receiver initially accepts an email message for delivery and then subsequently fails to deliver it to the recipient.

Complaint

Most email client programs provide a button labeled "Mark as Spam," or similar, which moves the message to a spam folder, and forwards it to the ISP. Additionally, most ISPs maintain an abuse address (e.g., abuse@example.net), where users can forward unwanted email messages and request that the ISP take action to prevent them. In both of these cases, the recipient is making a complaint. If the ISP concludes that you are a spammer, and Amazon SES has a feedback loop set up with the ISP, then the ISP will send the complaint back to Amazon SES. When Amazon SES receives such a complaint, it forwards the complaint to you either by email or by using an Amazon SNS notification, depending on how you have your system set up. For more information, see Using Notifications with Amazon SES (p. 105). We recommend that you do not make repeated delivery attempts to email addresses that generate complaints.

Suppression List

The Amazon SES *suppression list* (formerly called the *blacklist*) is a list of recipient email addresses that Amazon SES considers to be invalid because the addresses have caused a hard bounce for any Amazon SES customer within the past 14 days. If you try to send an email to an address on the suppression list, the call to Amazon SES succeeds, but Amazon SES treats the email as a hard bounce instead of attempting to send it. You are notified by the same means as you are notified of other hard bounces—by email or by using an Amazon SNS notification, depending on how you have your system set up. Like any hard bounce, suppression list bounces count towards your sending quota and your bounce rate. If you are sure that the email address that you're trying to send to is valid, you can submit a suppression list removal request. For more information, see Removing an Email Address from the Amazon SES Suppression List (p. 187).

Be Proactive

One of the biggest issues with email on the Internet is unsolicited bulk email, or spam. ISPs take considerable measures to prevent their customers from receiving spam. Correspondingly, Amazon SES takes proactive steps to decrease the likelihood that ISPs consider your email to be spam. Amazon SES uses verification, authentication, sending limits, and content filtering. Amazon SES also maintains a trusted reputation with ISPs and requires you to send high-quality email. Amazon SES does some of those things for you automatically (like content filtering); in other cases, it provides the tools (like authentication), or guides you in the right direction (sending limits). The following sections provide more information about each concept.

Verification

Unfortunately, it's possible for a spammer to falsify an email header and spoof the originating email address so that it appears as though the email originated from a different source. To maintain trust between ISPs and Amazon SES, Amazon SES needs to ensure that its senders are who they say they are. You are therefore required to verify all email addresses from which you send emails through Amazon SES to protect your sending identity. You can verify email addresses by using the Amazon SES console or by using the Amazon SES API. You can also verify entire domains. For more information, see Verifying Email Addresses in Amazon SES (p. 37) and Verifying Domains in Amazon SES (p. 40).

If your account is still in the Amazon SES sandbox, you also need to verify all recipient addresses except for addresses provided by the Amazon SES mailbox simulator. For information about getting out of the sandbox, see Moving Out of the Amazon SES Sandbox (p. 45). For more information about the mailbox simulator, see Testing Amazon SES Email Sending (p. 158).

Authentication

Authentication is another way that you can indicate to ISPs that you are who you say you are. When you authenticate an email, you provide evidence that you are the owner of the account and that your emails have not been modified in transit. In some cases, ISPs refuse to forward email that is not authenticated. Amazon SES supports two methods of authentication: Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM). For more information, see Authenticating Email in Amazon SES (p. 93).

Sending Limits

If an ISP detects sudden, unexpected spikes in the volume or rate of your emails, the ISP might suspect you are a spammer and block your emails. Therefore, every Amazon SES account has a set of sending limits to regulate the number of email messages that you can send and the rate at which you can send them. These sending limits help you to gradually ramp up your sending activity to protect your trustworthiness with ISPs.

Amazon SES has two sending limits: a sending quota (the maximum number of messages you can send in a 24-hour period) and a maximum send rate (the maximum number of emails that Amazon SES can accept from your account per second, although the actual rate at which Amazon SES accepts your messages might be less than the maximum send rate). If you are a brand-new user, Amazon SES lets you send a small amount of email each day. If the mail that you send is acceptable to ISPs, this limit will gradually increase. Over time, your sending limits will steadily increase so that you can send larger quantities of email at faster rates. You can also file an SES Sending Limits Increase Case to get your quotas increased if you need them to ramp up more quickly.

For more information about sending limits and how to increase them, see Managing Your Amazon SES Sending Limits (p. 124).

Content Filtering

Many ISPs use content filtering to determine if incoming emails are spam. Content filters look for questionable content and block the email if the email fits the profile of spam. Amazon SES uses content filters also. When your application sends a request to Amazon SES, Amazon SES assembles an email message on your behalf and then scans the message header and body to determine if they contain content that ISPs might construe as spam. If your messages look like spam to the content filters that Amazon SES uses, your reputation with Amazon SES will be negatively affected. If a message is infected with a virus, it is rejected by Amazon SES entirely.

Reputation

When it comes to email sending, *reputation*—a measure of confidence that an IP address, email address, or sending domain is not the source of spam—is important. Amazon SES maintains a strong reputation with ISPs so that ISPs deliver your emails to your recipients' inboxes. Similarly, you need to maintain a trusted reputation with Amazon SES. You build your reputation with Amazon SES by sending high-quality content. When you send high-quality content, your reputation becomes more trusted over time and Amazon SES increases your sending limits. Excessive bounces and complaints negatively impact your reputation and can cause Amazon SES to lower your sending limits or terminate your Amazon SES account.

One way to help maintain your reputation is to use the mailbox simulator when you test your system, instead of sending to email addresses that you have created yourself. Emails to the mailbox simulator do not count toward your bounce and complaint metrics. For more information about the mailbox simulator, see Testing Amazon SES Email Sending (p. 158).

High-Quality Email

High-quality email is email that recipients find valuable and want to receive. Value means different things to different recipients and can come in the form of offers, order confirmations, receipts, newsletters, etc. Ultimately, your deliverability rests on the quality of the emails that you send because ISPs block emails that they find to be low quality (spam). For more information about how to send high-quality email, see Improving Deliverability with Amazon SES (p. 177) and the Amazon SES Email Sending Best Practices white paper.

Stay Informed

Whether your deliveries fail, your recipients complain about your emails, or Amazon SES successfully delivers an email to a recipient's mail server, Amazon SES helps you to track down the issue by providing notifications and by enabling you to easily monitor your usage statistics.

Notifications

When an email bounces, the ISP notifies Amazon SES, and Amazon SES notifies you. Amazon SES notifies you of hard bounces and soft bounces that Amazon SES will no longer retry. Many ISPs also forward complaints, and Amazon SES sets up complaint feedback loops with the major ISPs so you don't have to. Amazon SES can notify you of bounces, complaints, and successful deliveries in two ways: you can set your account up to receive notifications through Amazon SNS, or you can receive notifications by email (bounces and complaints only). For more information, see Using Notifications with Amazon SES (p. 105).

Usage Statistics

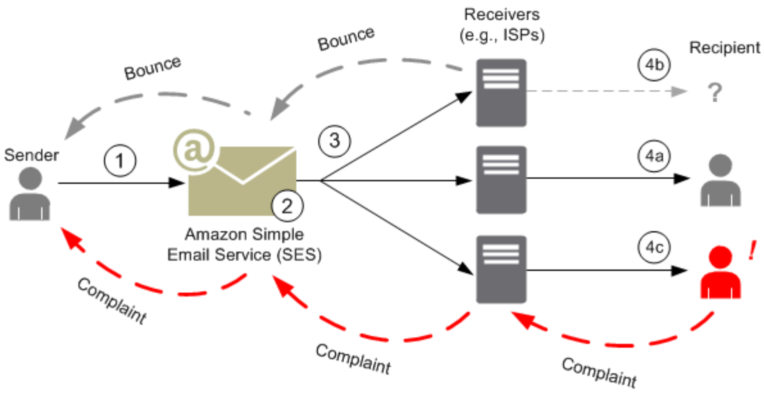
Amazon SES provides usage statistics so that you can view your failed deliveries to determine and resolve the root causes. You can view your usage statistics by using the Amazon SES console or by calling the Amazon SES API. You can view how many deliveries, bounces, complaints, and virus-infected rejected emails you have, and you can also view your sending limits to ensure that you stay within them.

Improve Your Email-Sending Program

If you are getting large numbers of bounces and complaints, it's time to reassess your email-sending strategy. Remember that excessive bounces, complaints, and attempts to send low-quality email constitute abuse and put your AWS account at risk of termination. Ultimately, you need to be sure that you use Amazon SES to send high-quality emails and to only send emails to recipients who want to receive them. For more information, see Improving Deliverability with Amazon SES (p. 177) and the Amazon SES Email Sending Best Practices white paper.

Sending Email with Amazon SES

This topic describes what happens when you send an email with Amazon SES, and the various outcomes that can occur after the email is sent. The following figure is a high-level overview of the sending process:



1. A client application, acting as an email sender, makes a request to Amazon SES to send email to one or more recipients.
2. If the request is valid, Amazon SES accepts the email and sends it over the Internet to the recipient's receiver. Once the message is passed to Amazon SES, it is usually sent immediately, with the first delivery attempt normally occurring within milliseconds.
3. At this point, there are different possibilities. For example:

   a. The ISP successfully delivers the message to the recipient's inbox.
   b. The recipient's email address does not exist, so the ISP sends a bounce notification to Amazon SES. Amazon SES then forwards the notification to the sender.
   c. The recipient receives the message but considers it to be spam and registers a complaint with the ISP. The ISP, which has a feedback loop set up with Amazon SES, sends the complaint to Amazon SES, which then forwards it to the sender.

The following sections review the individual possible outcomes after a sender sends an email request to Amazon SES and after Amazon SES sends an email message to the recipient.

After a Sender Sends an Email Request to Amazon SES

When the sender makes a request to Amazon SES to send an email, the call may succeed or fail. The following sections describe what happens in each case.

Successful Sending Request

If the request to Amazon SES succeeds, Amazon SES returns a success response to the sender. This message includes the *message ID*, a string of characters that uniquely identifies the request. You can use the message ID to identify the sent email or to track problems encountered during sending. Amazon SES then assembles an email message based on the request parameters, scans the message for questionable content and viruses and then sends it out over the Internet using Simple Mail Transfer Protocol (SMTP). Your message is usually sent immediately; the first delivery attempt typically occurs within milliseconds.

> **Note**
> If Amazon SES successfully accepts the sender's request and then an Amazon SES content filter finds that the message contains a virus, Amazon SES drops the message and notifies the sender through email or through Amazon Simple Notification Service (Amazon SNS), depending on the sender's setup.

Failed Sending Request

If the sender's email-sending request to Amazon SES fails, Amazon SES responds to the sender with an error and drops the email. The request could fail for several reasons. For example, the request may not be formatted properly or the email address may not have been verified by the sender.

The method through which you can determine if the request has failed depends on how you call Amazon SES. The following are examples of how errors and exceptions are returned:

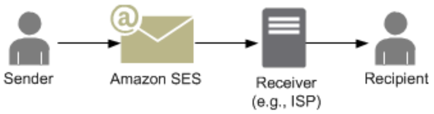
- If you are calling Amazon SES through the Query (HTTPS) API (`SendEmail` or `SendRawEmail`), the actions will return an error. For more information, see the Amazon Simple Email Service API Reference.
- If you are using an AWS SDK for a programming language that uses exceptions, the call to Amazon SES will throw a *MessageRejectedException*. (The name of the exception may vary slightly depending on the SDK.)
- If you are using the SMTP interface, then the sender receives an SMTP response code, but how the error is conveyed depends on the sender's client. Some clients may display an error code; others may not.

For information about errors that can occur when you send an email with Amazon SES, see Amazon SES Email Sending Errors (p. 182).

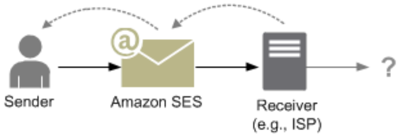
After Amazon SES Sends an Email

If the sender's request to Amazon SES succeeds, then Amazon SES sends the email and one of the following outcomes occurs:

- **Successful delivery and the recipient does not object to the email—**The email is accepted by the ISP, and the ISP delivers the email to the recipient. A successful delivery is shown in the following figure.

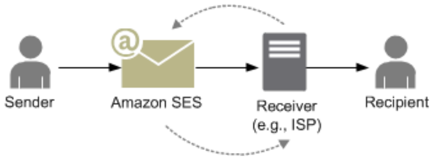


- **Hard bounce—**The email is rejected by the ISP because of a persistent condition or rejected by Amazon SES because the email address is on the Amazon SES suppression list. An email address is on the Amazon SES suppression list if it has caused a hard bounce for any Amazon SES customer within the past 14 days. A hard bounce with an ISP can occur because the recipient's address or domain name is invalid. A hard bounce notification is sent from the ISP back to Amazon SES, which notifies the sender through email or through Amazon Simple Notification Service (Amazon SNS), depending on the sender's setup. Amazon SES notifies the sender of suppression list bounces by the same means. The path of a hard bounce from an ISP is shown in the following figure.

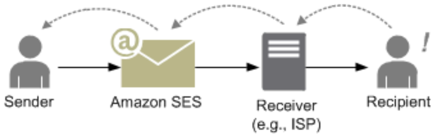


- **Soft bounce—**The ISP cannot deliver the email to the recipient because of a temporary condition. For example, the ISP is too busy to handle the request or the recipient's mailbox is full. The ISP sends a soft bounce notification back to Amazon SES. Amazon SES retries sending the email for a length of time.

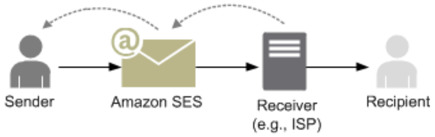
If the ISP can deliver the email to the recipient during a retry, the delivery is successful. If the ISP cannot deliver the email to the recipient by the time Amazon SES stops retrying, then Amazon SES notifies the sender through email or through Amazon SNS. A soft bounce is shown in the following figure. In this case, Amazon SES retries sending the email, and the ISP is eventually able to deliver it to the recipient.



- **Complaint—**The email is accepted by the ISP and delivered to the recipient, but the recipient considers the email to be spam and clicks a button such as "Mark as spam" in his or her email client. If Amazon SES has a feedback loop set up with the ISP, then a complaint notification is sent to Amazon SES, which forwards the complaint notification to the sender. Most ISPs do not provide the email address of the recipient who submitted the complaint, so the complaint notification from Amazon SES provides the sender a list of recipients who might have sent the complaint, based on the recipients of the original message and the ISP from which Amazon SES received the complaint. The path of a complaint is shown in the following figure.



- **Auto response—**The email is accepted by the ISP, and the ISP delivers it to the recipient. The ISP then sends an automatic response such as an out-of-the-office (OOTO) message to Amazon SES. Amazon SES forwards the auto response notification to the sender. An auto response is shown in the following figure.



Make sure that your Amazon SES-enabled program does not retry sending messages that generate an auto response.

> **Tip**
> You can use the Amazon SES mailbox simulator to test a successful delivery, bounce, complaint, OOTO, or what happens when an address is on the suppression list. For more information, see Testing Amazon SES Email Sending (p. 158).

Email Format and Amazon SES

When a client makes a request to Amazon SES, Amazon SES constructs an email message compliant with the Internet Message Format specification (RFC 5322). An email consists of a *header*, a *body*, and an *envelope*, as described below.

- **Header—**Contains routing instructions and information about the message. Examples are the sender's address, the recipient's address, the subject, and the date. The header is analogous to the information at the top of a postal letter, though it can contain many other types of information, such as the format of the message.

- **Body—**Contains the text of the message itself.
- **Envelope—**Contains the actual routing information that is communicated between the email client and the mail server during the SMTP session. This email envelope information is analogous to the information on a postal envelope. The routing information of the email envelope is usually the same as the routing information in the email header, but not always. For example, when you send a blind carbon copy (BCC), the actual recipient address (derived from the envelope) is not the same as the "To" address that is displayed in the recipient's email client, which is derived from the header.

The following is a simple example of an email. The header is followed by a blank line and then the body of the email. The envelope isn't shown because it is communicated between the client and the mail server during the SMTP session, rather than a part of the email itself.

```
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
```

The following sections review email headers and bodies and identify the information that you need to provide when you use Amazon SES.

Email Header

There is one header per email message. Each line of the header contains a field followed by a colon followed by a field body. When you read an email in an email client, the email client typically displays the values of the following header fields:

- **To—**The email addresses of the message's recipients.
- **CC—**The email addresses of the message's carbon copy recipients.
- **From—**The email address from which the email is sent.
- **Subject—**A summary of the message topic.
- **Date—**The time and date the email is sent.

There are many additional header fields that provide routing information and describe the content of the message. Email clients typically do not display these fields to the user. For a full list of the header fields that Amazon SES accepts, see Appendix: Header Fields (p. 212). When you use Amazon SES, you particularly need to understand the difference between "From," "Reply-To," and "Return-Path" header fields. As noted previously, the "From" address is the email address of the message sender, whereas "Reply-To" and "Return-Path" are as follows:

- **Reply-To—**The email address to which replies will be sent. By default, replies are sent to the original sender's email address.
- **Return-Path—**The email address to which message bounces and complaints should be sent. "Return-Path" is sometimes called "envelope from," "envelope sender," or "MAIL FROM."

  Note
  When you use Amazon SES, we recommend that you always set the "Return-Path" parameter so that you can be aware of bounces and take corrective action if they occur.

To easily match a bounced message with its intended recipient, you can use Variable Envelope Return Path (VERP). With VERP, you set a different "Return-Path" for each recipient, so that if the message bounces back, you automatically know which recipient it bounced from, rather than having to open the bounce message and parse it.

Email Body

The email body contains the text of the message. The body can be sent in the following formats:

- **HTML—**If the recipient's email client can interpret HTML, the body can include formatted text and hyperlinks
- **Plain text—**If the recipient's email client is text-based, the body must not contain any nonprintable characters.
- **Both HTML and plain text—**When you use both formats to send the same content in a single message, the recipient's email client decides which to display, based upon its capabilities.

If you are sending an email message to a large number of recipients, then it makes sense to send it in both HTML and text. Some recipients will have HTML-enabled email clients, so that they can click embedded hyperlinks in the message. Recipients using text-based email clients will need you to include URLs that they can copy and open using a web browser.

Email Information You Need to Provide to Amazon SES

When you send an email with Amazon SES, the email information you need to provide depends on how you call Amazon SES. You can provide a minimal amount of information and have Amazon SES take care of all of the formatting for you. Or, if you want to do something more advanced like send an attachment, you can provide the raw message yourself. The following sections review what you need to provide when you send an email by using the Amazon SES API, the Amazon SES SMTP interface, or the Amazon SES console.

Amazon SES API

If you call the Amazon SES API directly, you call either the `SendEmail` or the `SendRawEmail` API. The amount of information you need to provide depends on which API you call.

- The `SendEmail API` requires you to provide only a source address, destination address, message subject, and a message body. You can optionally provide "Reply-To" addresses. When you call this API, Amazon SES automatically assembles a properly formatted multi-part Multipurpose Internet Mail Extensions (MIME) email message optimized for display by email client software. For more information, see Sending Formatted Email Using the Amazon SES API (p. 80).
- The `SendRawEmail` API provides you the flexibility to format and send your own raw email message by specifying headers, MIME parts, and content types. `SendRawEmail` is typically used by advanced users. You need to provide the body of the message and all header fields that are specified as required

in the Internet Message Format specification (RFC 5322). For more information, see Sending Raw Email Using the Amazon SES API (p. 80).

If you use an AWS SDK to call the Amazon SES API, you provide the information listed above to the corresponding functions (for example, `SendEmail` and `SendRawEmail` for Java).

For more information about sending email using the Amazon SES API, see Using the Amazon SES API to Send Email (p. 79).

Amazon SES SMTP Interface

When you access Amazon SES through the SMTP interface, your SMTP client application assembles the message, so the information you need to provide depends on the application you are using. At a minimum, the SMTP exchange between a client and a server requires a source address, a destination address, and message data. If you are using the SMTP interface and have feedback forwarding enabled, then your bounces, complaints, and delivery notifications are sent to the "MAIL FROM" address. Any "Reply-To" address that you specify is not used.

For more information about sending email using the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 48).

Amazon SES Console

When you send an email by using the Amazon SES console, the amount of information you need to provide depends on whether you choose to send a formatted or raw email.

- To send a formatted email, you need to provide a source address, a destination address, a message subject, and a message body. Amazon SES automatically assembles a properly formatted multi-part MIME email message optimized for display by email client software. You can also specify a reply-to and a return path field.
- To send a raw email, you provide the source address, a destination address, and the message content, which must contain the body of the message and all header fields that are specified as required in the Internet Message Format specification (RFC 5322).

Amazon SES and Security Protocols

This topic describes the security protocols that you can use when you connect to Amazon SES, as well as when Amazon SES delivers an email to a receiver.

Email Sender to Amazon SES

The security protocol that you use to connect to Amazon SES depends on whether you are using the Amazon SES API or the Amazon SES SMTP interface, as described next.

HTTP

If you are using the Amazon SES API (either directly or through an AWS SDK), then all communications are encrypted by TLS through Amazon SES's HTTPS endpoint.

SMTP Interface

If you are accessing Amazon SES through the SMTP interface, you are required to encrypt your connection using Transport Layer Security (TLS). Note that TLS is often referred to by the name of its predecessor protocol, Secure Sockets Layer (SSL).

Amazon SES supports two mechanisms for establishing a TLS-encrypted connection: STARTTLS and TLS Wrapper.

- **STARTTLS**—STARTTLS is a means of upgrading an unencrypted connection to an encrypted connection. There are versions of STARTTLS for a variety of protocols; the SMTP version is defined in RFC 3207. For STARTTLS connections, Amazon SES supports SSLv3, TLSv1 and SSLv2Hello.
- **TLS Wrapper**—TLS Wrapper (also known as SMTPS or the Handshake Protocol) is a means of initiating an encrypted connection without first establishing an unencrypted connection. With TLS Wrapper, the Amazon SES SMTP endpoint does not perform TLS negotiation: it is the client's responsibility to connect to the endpoint using TLS, and to continue using TLS for the entire conversation. TLS Wrapper is an older protocol, but many clients still support it. For TLS Wrapper connections, Amazon SES supports SSLv3, TLSv1, TLSv1.1 and TLSv1.2.

For information about connecting to the Amazon SES SMTP interface using these methods, see Connecting to the Amazon SES SMTP Endpoint (p. 53).

If your software does not support STARTTLS or TLS Wrapper, you can set up a secure tunnel to allow your software to communicate with the Amazon SES SMTP endpoint. For information about how to set up a secure tunnel, see Setting Up a Secure Tunnel to Connect to Amazon SES (p. 59).

Amazon SES to Receiver

Amazon SES sends messages over a TLS-protected connection (TLS version 1.0 only) by default. This method, called *opportunistic TLS*, means that when Amazon SES establishes an SMTP connection with a receiving mail server, Amazon SES upgrades the connection using the STARTTLS protocol if the receiving mail server supports TLS. If the receiving server does not advertise STARTTLS or if TLS negotiation fails, the connection proceeds in plaintext.

Amazon SES supports opportunistic TLS in all regions and you don't need to take any action to enable it.

Amazon SES Quick Start

This procedure leads you through the steps to sign up for AWS, verify your email address, send your first email, consider how you will handle bounces and complaints, and move out of the Amazon Simple Email Service (Amazon SES) sandbox.

Use this procedure if you:

- Are just experimenting with Amazon SES.
- Want to send some test emails without doing any programming.
- Want to get set up in as few steps as possible.

Step 1: Sign up for AWS

Before you can use Amazon SES, you need to sign up for AWS. When you sign up for AWS, your account is automatically signed up for all AWS services.

For instructions, see Signing up for AWS (p. 36).

Step 2: Verify your email address

Before you can send email from your email address through Amazon SES, you need to show Amazon SES that you own the email address by verifying it.

For instructions, see Verifying Email Addresses in Amazon SES (p. 37).

Step 3: Send your first email

You can send an email simply by using the Amazon SES console. As a new user, your account is in a test environment called the sandbox, so you can only send email to and from email addresses that you have verified.

For instructions, see Sending an Email Using the Amazon SES Console (p. 19).

Step 4: Consider how you will handle bounces and complaints

Before the next step, you need to think about how you will handle bounces and complaints. If you are sending to a small number of recipients, your process can be as simple as examining the bounce and complaint feedback that you receive by email, and then removing those recipients from your mailing list.

For more information, see Processing Bounces and Complaints (p. 178).

Step 5: Move out of the Amazon SES sandbox

To be able to send emails to unverified email addresses and to raise the number of emails you can send per day and how fast you can send them, your account needs to be moved out of the sandbox. This process involves opening an SES Sending Limits Increase case in Support Center.

For more information about the sandbox restrictions and how to apply to move out of the sandbox, see Moving Out of the Amazon SES Sandbox (p. 45).

Next steps

- After you send a few test emails to yourself, use the Amazon SES mailbox simulator for further testing because emails to the mailbox simulator do not count towards your sending quota or your bounce and complaint rates. For more information on the mailbox simulator, see Testing Amazon SES Email Sending (p. 158).
- Monitor your sending activity, such as the number of emails that you have sent and the number that have bounced or received complaints. For more information, see Monitoring Your Amazon SES Sending Activity (p. 104).
- Verify entire domains so that you can send email from any email address in your domain without verifying addresses individually. For more information, see Verifying Domains in Amazon SES (p. 40).
- Increase the chance that your emails will be delivered to your recipients' inboxes instead of junk boxes by authenticating your emails. For more information, see Authenticating Email in Amazon SES (p. 93).

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Getting Started with Amazon SES

This getting started tutorial provides step-by-step instructions for you to set up Amazon Simple Email Service (Amazon SES) and send an email. First, review the information in Before You Begin with Amazon SES (p. 19). Then, send an email in one of the following ways. You can also watch our Getting Started with Amazon SES video.

For information about Amazon SES email pricing, see Pricing on the Amazon SES detail page.

Using the Amazon SES Console

Use this method if you want to get started sending test emails through Amazon SES with minimal setup. When you are ready to start your production email sending campaign, you will want to move on to one of the other sending methods and use the Amazon SES console primarily to monitor your sending activity.

To start this tutorial, go to Sending an Email Using the Amazon SES Console (p. 19).

Using Simple Mail Transfer Protocol (SMTP)

Use this method if you want to send email through the Amazon SES SMTP interface with or without programming as follows:

- Enable an application to send email through Amazon SES by using a programming language that supports SMTP. Examples are provided in C# and Java.

  To start this tutorial, go to Sending an Email Through the Amazon SES SMTP Interface Programmatically (p. 21).
- Set up your mail server to forward mail to Amazon SES, or configure your email client or SMTP-enabled software package to send email through Amazon SES. Examples are provided for Postfix, Sendmail, and Exim mail servers as well as email client Microsoft Outlook and issue-tracking software Jira.

  To start this tutorial, go to Configuring Your Existing Email Server or SMTP-Enabled Application to Send Email Through Amazon SES (p. 28).

For introductory information on both SMTP sending methods, see Sending an Email Through Amazon SES Using SMTP (p. 20).

Using an AWS SDK

Use this method to call the Amazon SES API using libraries that handle the details of the underlying Amazon SES Query interface.

To start this tutorial, go to Sending an Email Through Amazon SES Using an AWS SDK (p. 28).

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Before You Begin with Amazon SES

Before you get started, you need to set up Amazon SES. Whether you send an email by using the Amazon SES console, the SMTP interface, or the Amazon SES API, you need to:

- **Sign up for AWS—**Before you can use Amazon SES or other AWS services, you need to create an AWS account. For information, see Signing up for AWS (p. 36).
- **Verify your email address or domain—**To send emails using Amazon SES, you always need to verify your "From" address to show that you own it. If you are in the sandbox, you also need to verify your "To" addresses. You can verify email addresses or entire domains. For information, see Verifying Email Addresses and Domains in Amazon SES (p. 37).

This list contains the setup tasks that are mandatory for all email sending methods. Additional setup tasks that are specific to the email sending method are given in the corresponding getting started section. To see a complete list of all setup tasks, see Setting up Amazon SES (p. 36).

Sending an Email Using the Amazon SES Console

Sending an email from the Amazon SES console, as described in the following procedure, is the easiest way to start experimenting with sending emails using Amazon SES. After you get started with Amazon SES, you typically will use the console to monitor your sending activity rather than to send production emails.

> **Important**
> In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator whenever possible. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 158).

Before you follow these steps, make sure you review the setup instructions in Before You Begin with Amazon SES (p. 19).

To send an email message from the Amazon SES console

1. Sign into the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses/home. If you are not currently signed into your AWS account, this link will take you to a sign-in page. After you sign in, you will be directed to the Amazon SES console.

2. In the **Navigation** pane of the Amazon SES console, under **Verified Senders**, click **Email Addresses** to view the email address that you verified in Verifying Email Addresses in Amazon SES (p. 37).
3. In the list of verified senders, select the checkbox of an email address that you have verified.
4. Click **Send a Test Email**.
5. In the **Send Test Email** dialog box, choose the **Email Format**. The two choices are as follows:

   - **Formatted**—This is the simplest option. Choose this if you simply want to type the text of your message into the **Body** text box. When you send the email, Amazon SES will put the text into email format for you.
   - **Raw**—Choose this option if you want to send a more complex message, such as a message that includes HTML or an attachment. Because of this flexibility, you will need to format the message as described in Sending Raw Email Using the Amazon SES API (p. 80) yourself, and then paste the entire formatted message, including the headers, into the **Body** text box. You can use the following example, which contains HTML, to send a test email using the **Raw** email format. Copy and paste this message in its entirety into the **Body** text box. Ensure that there is not a blank line between the `MIME-Version` header and the `Content-Type` header, because that would cause the email to be formatted as plain text instead of 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="ht
   tp://docs.aws.amazon.com/ses/latest/DeveloperGuide/Welcome.html">Amazon 
   SES Developer Guide</a>.</p>
   </body>
   </html>
   ```

6. In the **Send Test Email** dialog box, fill out the rest of the fields. If you are still in the Amazon SES sandbox, make sure that you have verified the address in the **To** field. For more information, see Verifying Email Addresses in Amazon SES (p. 37).
7. Click **Send Test Email**.
8. Sign in to the email client of the address you sent the email to. You should find the email message that you sent.

Sending an Email Through Amazon SES Using SMTP

You can use an SMTP-compatible programming language, application, or software package to send email through the Amazon SES SMTP interface. Before you start, review the instructions in Before You Begin with Amazon SES (p. 19). You also need to obtain the following additional information:

- Your Amazon SES SMTP username and password, which enable you to connect to the Amazon SES SMTP endpoint. To get your Amazon SES SMTP username and password, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

  Important
  Your SMTP credentials are different from your AWS credentials. For more information about credentials, see Using Credentials With Amazon SES (p. 170).
- The Amazon SES SMTP hostname, which is *email-smtp.us-east-1.amazonaws.com* (for region us-east-1), *email-smtp.us-west-2.amazonaws.com* (for region us-west-2), or *email-smtp.eu-west-1.amazonaws.com* (for region eu-west-1).
- The Amazon SES SMTP interface port number, which depends on the connection method. For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 53).

Once you have obtained your SMTP credentials, you can connect to the Amazon SES SMTP endpoint and start sending email. This getting started tutorial shows you how to send email through the Amazon SES SMTP interface by using the following methods:

- Sending an Email Through the Amazon SES SMTP Interface Programmatically (p. 21)
- Configuring Your Existing Email Server or SMTP-Enabled Application to Send Email Through Amazon SES (p. 28)

For more information about the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 48).

Sending an Email Through the Amazon SES SMTP Interface Programmatically

You can send an email through the Amazon SES SMTP interface by using an SMTP-enabled programming language. You provide the Amazon SES SMTP hostname and port number along with your SMTP credentials and then use the programming language's generic SMTP functions to send the email.

Review Sending an Email Through Amazon SES Using SMTP (p. 20) and then select one of the following tutorials:

- Sending an Email Through the Amazon SES SMTP Interface with C# (p. 21)
- Sending an Email Through the Amazon SES SMTP Interface with Java (p. 24)

Sending an Email Through the Amazon SES SMTP Interface with C#

The following procedure shows you how to use Microsoft Visual Studio to create a console application and modify the C# code to send an email through Amazon SES. The process to create a new project based on a project template that is similar across Microsoft Visual Studio editions, but we'll go through the procedure using Microsoft Visual Studio Professional 2012.

Before you perform the following procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 19) and Sending an Email Through Amazon SES Using SMTP (p. 20).

> **Important**
> In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator whenever possible. Emails that you send to the mailbox simulator do not count toward your

sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 158).

To send an email using the Amazon SES SMTP interface with C#

1. Create a console project in Visual Studio by performing the following steps:

   a. Open Microsoft Visual Studio.

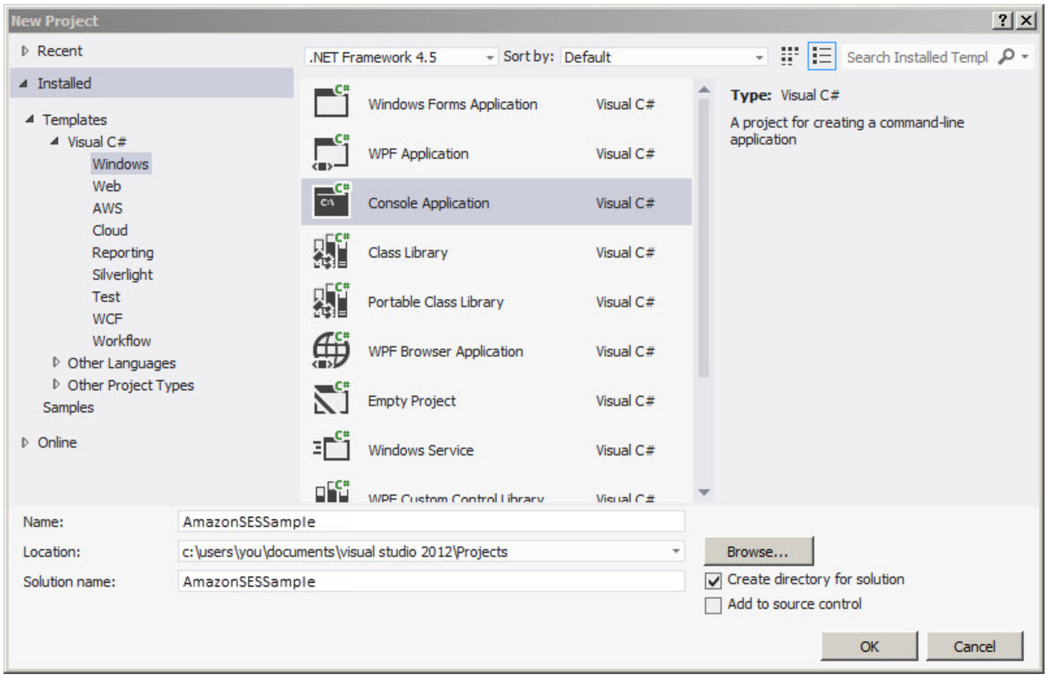
   b. Click **File**, click **New**, and then click **Project**.

   c. In the **New Project** dialog box, in the left pane, expand **Installed**, expand **Templates**, and then expand **Visual C#**.

   d. Under **Visual C#**, click **Windows**.

   e. Click **Console Application**.

   f. In the **Name** field, type `AmazonSESSample`. The dialog box should look similar to the following figure.



.NET Framework 4.5

Sort by: Default

Type: Visual C#

A project for creating a command-line application

Name: AmazonSESSample

Location: c:\users\you\documents\visual studio 2012\Projects

Solution name: AmazonSESSample

☑ Create directory for solution

□ Add to source control

   g. Click **OK**.

2. In your Visual Studio project, replace the entire contents of Program.cs with the following code:

```
using System;

namespace AmazonSESSample
{
    class Program
    {
        static void Main(string[] args)
        {
           const String FROM = "SENDER@EXAMPLE.COM";   // Replace with your
 "From" address. This address must be verified.
            const String TO = "RECIPIENT@EXAMPLE.COM";  // Replace with a
```

```
"To" address. If your account is still in the
                                                        // sandbox, this
address must be verified.

           const String SUBJECT = "Amazon SES test (SMTP interface accessed
 using C#)";
            const String BODY = "This email was sent through the Amazon SES
 SMTP interface by using C#.";

            // Supply your SMTP credentials below. Note that your SMTP cre
dentials are different from your AWS credentials.
            const String SMTP_USERNAME = "YOUR_SMTP_USERNAME";  // Replace
 with your SMTP username.
            const String SMTP_PASSWORD = "YOUR_SMTP_PASSWORD";  // Replace
 with your SMTP password.

            // Amazon SES SMTP host name. This example uses the US West
(Oregon) region.
            const String HOST = "email-smtp.us-west-2.amazonaws.com";

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

            // Create an SMTP client with the specified host name and port.

          using (System.Net.Mail.SmtpClient client = new System.Net.Mail.Smt
pClient(HOST, PORT))
            {
                // Create a network credential with your SMTP user name and
 password.
                client.Credentials = new System.Net.NetworkCreden
tial(SMTP_USERNAME, SMTP_PASSWORD);

                // Use SSL when accessing Amazon SES. The SMTP session will
 begin on an unencrypted connection, and then
                // the client will issue a STARTTLS command to upgrade to
an encrypted connection using SSL.
                client.EnableSsl = true;

                // Send the email.
                try
                {
                    Console.WriteLine("Attempting to send an email through
 the Amazon SES SMTP interface...");
                    client.Send(FROM, TO, SUBJECT, BODY);
                    Console.WriteLine("Email sent!");
                }
                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();
            }
        }
}
```

3. In Program.cs, replace the following email addresses with your own values:

   Important
   The email addresses are case-sensitive. Make sure that the addresses are exactly the same as the ones you verified.

   - SENDER@EXAMPLE.COM—Replace with your "From" email address. You must verify this address before you run this program. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 37).
   - RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 45).

4. In Program.cs, replace the following SMTP credentials with the values that you obtained in Obtaining Your Amazon SES SMTP Credentials (p. 50):

   Important
   Your SMTP credentials are different from your AWS credentials. For more information about credentials, see Using Credentials With Amazon SES (p. 170).

   - YOUR_SMTP_USERNAME—Replace with your SMTP username. Note that your SMTP username credential is a 20-character string of letters and numbers, not an intelligible name.
   - YOUR_SMTP_PASSWORD—Replace with your SMTP password.

5. (Optional) If you want to use an Amazon SES SMTP endpoint in a region other than US West (Oregon), you need to change HOST in Program.cs to the endpoint you want to use. For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 173).
6. Save Program.cs.
7. To build the project, click **Build** and then click **Build Solution**.
8. To run the program, click **Debug** and then click **Start Debugging**.
9. Review the program's console output to verify that the sending was successful. (You should see "Email sent!")
10. Log into the email client of the recipient address. You should find the email message that you sent.

Sending an Email Through the Amazon SES SMTP Interface with Java

This example uses Eclipse IDE for Java EE Developers and the JavaMail API to send email through Amazon SES using the SMTP interface. The JavaMail API is included in the Java EE Platform and is available as an optional package for use with the Java SE Platform. If you do not have the JavaMail API installed, install it from JavaMail.

Before you perform the following procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 19) and Sending an Email Through Amazon SES Using SMTP (p. 20).

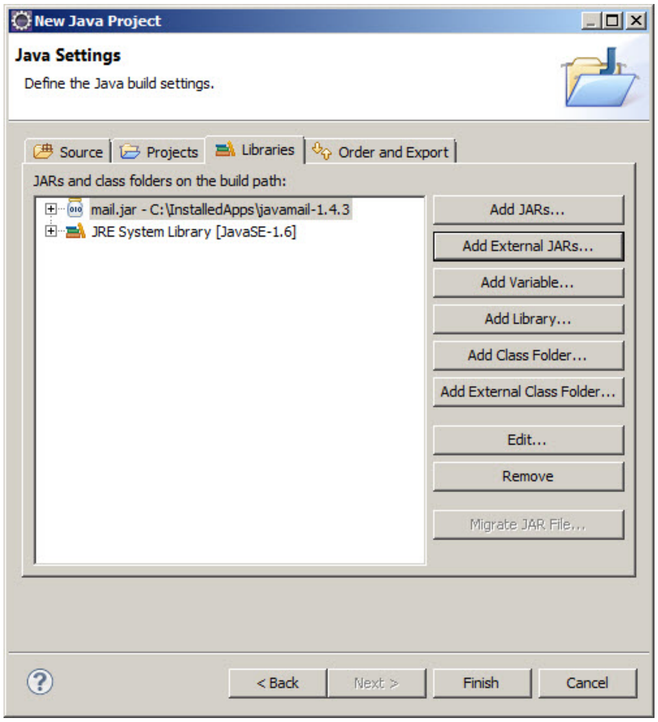
Important
In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator whenever possible. Emails that you send to the mailbox simulator do not count toward your

sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 158).

To send an email using the Amazon SES SMTP interface with Java

1. Create a project in Eclipse by performing the following steps:

   a. Open Eclipse.
   b. In Eclipse, click **File**, click **New**, and then click **Java Project**.
   c. In the **Create a Java Project** dialog box, type a project name and the click **Next.**
   d. In the **Java Settings** dialog box, click the **Libraries** tab.
   e. Click **Add External JARs**.
   f. Browse to your installation of JavaMail, click mail.jar, and then click **Open**. The **Java Settings** dialog box should now look similar to the following figure:



New Java Project

Java Settings

Define the Java build settings.

JARs and class folders on the build path:

mail.jar - C:\InstalledApps\javamail-1.4.3
JRE System Library [JavaSE-1.6]

   g. In the **Java Settings** dialog box, click **Finish**.

2. In Eclipse, in the **Package Explorer** window, expand your project.
3. Under your project, right-click the **src** directory, click **New**, and then click **Class**.
4. In the **New Java Class** dialog box, in the **Name** field, type `AmazonSESSample` and then click **Finish**.
5. Replace the entire contents of AmazonSESSample.java with the following code:

```
import java.util.Properties;
import javax.mail.*;
import javax.mail.internet.*;

public class AmazonSESSample {

    static final String FROM = "SENDER@EXAMPLE.COM";   // Replace with your
```

```
 "From" address. This address must be verified.
    static final String TO = "RECIPIENT@EXAMPLE.COM";  // Replace with a
"To" address. If your account is still in the
                                                      // sandbox, this ad
dress must be verified.

    static final String BODY = "This email was sent through the Amazon SES
 SMTP interface by using Java.";
    static final String SUBJECT = "Amazon SES test (SMTP interface accessed
 using Java)";

    // Supply your SMTP credentials below. Note that your SMTP credentials
 are different from your AWS credentials.
    static final String SMTP_USERNAME = "YOUR_SMTP_USERNAME";  // Replace
with your SMTP username.
    static final String SMTP_PASSWORD = "YOUR_SMTP_PASSWORD";  // Replace
with your SMTP password.

    // Amazon SES SMTP host name. This example uses the US West (Oregon)
region.
    static final String HOST = "email-smtp.us-west-2.amazonaws.com";

    // Port we will connect to on the Amazon SES SMTP endpoint. We are
choosing port 25 because we will use
    // STARTTLS to encrypt the connection.
    static final int PORT = 25;

    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);

     // Set properties indicating that we want to use STARTTLS to encrypt
the connection.
     // The SMTP session will begin on an unencrypted connection, and then
 the client
        // will issue a STARTTLS command to upgrade to an encrypted connec
tion.
     props.put("mail.smtp.auth", "true");
     props.put("mail.smtp.starttls.enable", "true");
     props.put("mail.smtp.starttls.required", "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));
       msg.setRecipient(Message.RecipientType.TO, new InternetAddress(TO));

        msg.setSubject(SUBJECT);
        msg.setContent(BODY,"text/plain");

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

        // Send the message.
        try
        {
            System.out.println("Attempting to send an email through the
Amazon SES SMTP interface...");

            // 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();
        }
    }
}
```

6. In AmazonSESSample.java, replace the following email addresses with your own values:

   Important
   The email addresses are case-sensitive. Make sure that the addresses are exactly the same as the ones you verified.

   - SENDER@EXAMPLE.COM—Replace with your "From" email address. You must verify this address before you run this program. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 37).
   - RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 45).

7. In AmazonSESSample.java, replace the following SMTP credentials with the values that you obtained in Obtaining Your Amazon SES SMTP Credentials (p. 50):

   Important
   Your SMTP credentials are different from your AWS credentials. For more information about credentials, see Using Credentials With Amazon SES (p. 170).

   - YOUR_SMTP_USERNAME—Replace with your SMTP username credential. Note that your SMTP username credential is a 20-character string of letters and numbers, not an intelligible name.
   - YOUR_SMTP_PASSWORD—Replace with your SMTP password.

8. (Optional) If you want to use an Amazon SES SMTP endpoint in a region other than US West (Oregon), you need to change HOST in AmazonSESSample.java to the endpoint you want to use. For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 173).
9. Save AmazonSESSample.java.

10. To build the project, click **Project** and then click **Build Project**. (If this option is disabled, then you may have automatic building enabled.)
11. To start the program and send the email, click **Run** and then click **Run** again.
12. Review the program's console output to verify that the sending was successful. (You should see "Email sent!")
13. Log into the email client of the recipient address. You should find the email message that you sent.

Configuring Your Existing Email Server or SMTP-Enabled Application to Send Email Through Amazon SES

You can configure your mail server, email client, or email sending software package to send messages through Amazon SES without any programming.

First, read Sending an Email Through Amazon SES Using SMTP (p. 20). Then review one of the following topics, which show you how to configure a mail server to forward mail to Amazon SES:

- Integrating Amazon SES with Postfix (p. 60)
- Integrating Amazon SES with Sendmail (p. 63)
- Integrating Amazon SES with Exim (p. 74)

For information about how to configure Microsoft Outlook, an email client, to send email through Amazon SES, see Configuring Email Clients to Send Through Amazon SES (p. 54).

For information about how to configure Jira, an issue-tracking software package, to send email through Amazon SES, see Sending Email Through Amazon SES From Software Packages (p. 57).

Sending an Email Through Amazon SES Using an AWS SDK

You can use an AWS SDK to send email through Amazon SES if you want to call the Amazon SES API, but you do not want to handle low-level details such as assembling and parsing HTTP requests and responses.

Before you send email using an AWS SDK, review the instructions in Before You Begin with Amazon SES (p. 19). For this tutorial, you also need to:

- **Download an AWS SDK—**Download and install an AWS SDK for either .NET or Java. For more information, see Downloading an AWS SDK (p. 45).
- **Get your AWS credentials—**To access Amazon SES programmatically, you need your AWS access keys. For more information, see Getting Your AWS Access Keys (p. 45).

After you have installed the appropriate SDK and retrieved your AWS credentials, you can send an email through Amazon SES using one of the following examples:

- Sending an Email Through Amazon SES Using AWS SDK for .NET (p. 29)
- Sending an Email Through Amazon SES Using AWS SDK for Java (p. 32)

Sending an Email Through Amazon SES Using AWS SDK for .NET

The following procedure shows you how to use Microsoft Visual Studio and AWS Toolkit for Microsoft Visual Studio to create an AWS SDK project and modify the C# code to send an email through Amazon SES. The process to create a new project based on a project template is similar across Microsoft Visual Studio editions, but we'll go through the procedure using Microsoft Visual Studio Professional 2012.

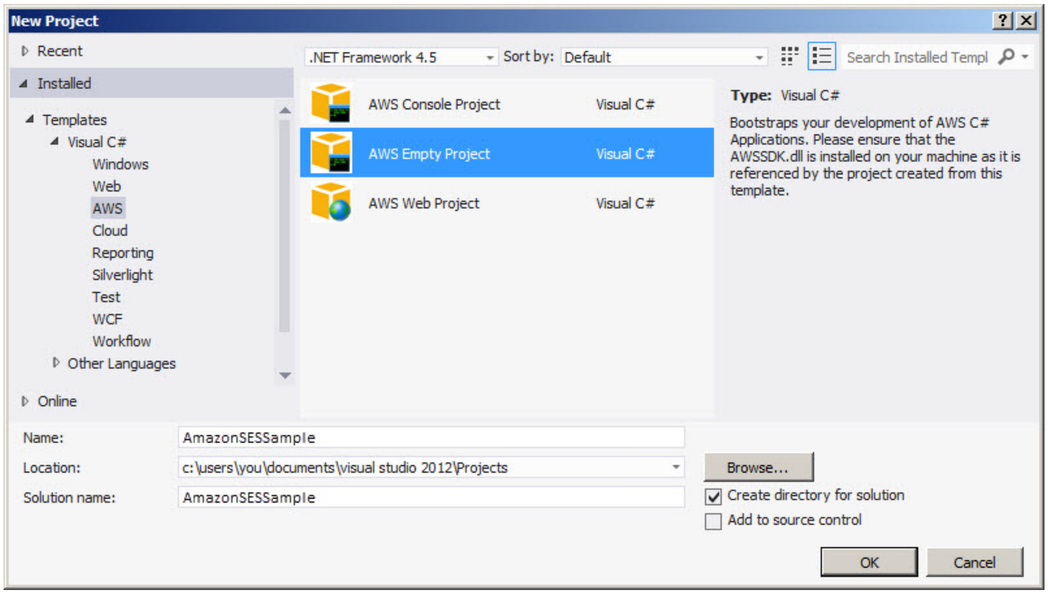
Before you begin this procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 19) and Sending an Email Through Amazon SES Using an AWS SDK (p. 28).

> **Important**
> In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator whenever possible. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 158).

To send an email using the AWS SDK for .NET v.2

1. Create an AWS project in Visual Studio by performing the following steps:

   a. Open Visual Studio.
   b. Click **File**, click **New**, and then click **Project**.
   c. In the **New Project** dialog box, in the left pane, expand **Installed**, expand **Templates**, and then expand **Visual C#**.
   d. Under **Visual C#**, click **AWS**.
   e. Click **AWS Empty Project**.
   f. In the **Name** field, type `AmazonSESSample`. The dialog box should look similar to the following figure.



Type: Visual C#

Bootstraps your development of AWS C# Applications. Please ensure that the AWSSDK.dll is installed on your machine as it is referenced by the project created from this template.

Name: AmazonSESSample

Location: c:\users\you\documents\visual studio 2012\Projects

Solution name: AmazonSESSample

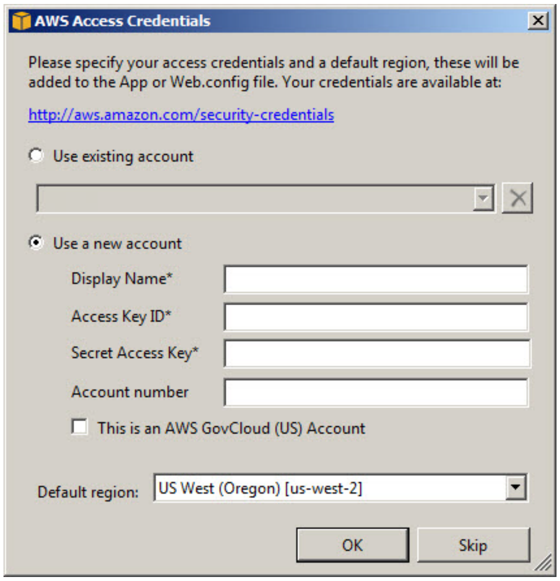
☑ Create directory for solution

□ Add to source control

   g. Click **OK**.

2. In the **AWS Access Credentials** dialog box, select an existing account or enter the following information:

   - **Display Name—**Type a name that identifies your account. Next time you create an AWS project in Visual Studio, you will be able to select this account so you do not have to enter the information again.
   - **Access Key ID—**Enter the AWS access key ID that you obtained in Getting Your AWS Access Keys (p. 45).
   - **Secret Access Key—**Enter the AWS secret access key that you obtained in Getting Your AWS Access Keys (p. 45).
   - **Account Number—**(Optional) Enter your AWS account number. To find your AWS account number, go to the Security Credentials page in the AWS Management Console and click *Account Identifiers*. (If you are not logged into your AWS account, this link will take you to an AWS account sign-in page first.) At the bottom of the page, under **Account Identifiers**, you will see your AWS Account ID.
   - **Default Region—**Select the AWS region of the Amazon SES endpoint you want to connect to. Note that your sandbox status, sending limits, and Amazon SES identity-related settings are specific to a given AWS region, so be sure to select an AWS region in which you set up Amazon SES. For a list of AWS regions that Amazon SES supports, see Regions and Amazon SES (p. 173).



AWS Access Credentials

Please specify your access credentials and a default region, these will be added to the App or Web.config file. Your credentials are available at:

http://aws.amazon.com/security-credentials

○ Use existing account

● Use a new account

Display Name*

Access Key ID*

Secret Access Key*

Account number

□ This is an AWS GovCloud (US) Account

Default region: US West (Oregon) [us-west-2]

3. Click **OK**.
4. In your Visual Studio project, replace the entire contents of Program.cs with the following code:

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

namespace AmazonSESSample
{
    class Program
    {
        public static void Main(string[] args)
        {
            const String FROM = "SENDER@EXAMPLE.COM";  // Replace with your
 "From" address. This address must be verified.
            const String TO = "RECIPIENT@EXAMPLE.COM"; // Replace with a
"To" address. If your account is still in the
            // sandbox, this address must be verified.

            const String SUBJECT = "Amazon SES test (AWS SDK for .NET)";
            const String BODY = "This email was sent through Amazon SES by
 using the AWS SDK for .NET.";

            // Construct an object to contain the recipient address.
            Destination destination = new Destination();
            destination.ToAddresses = (new List<string>() { TO });

            // Create the subject and body of the message.
            Content subject = new Content(SUBJECT);
            Content textBody = new Content(BODY);
            Body body = new Body(textBody);

            // Create a message with the specified subject and body.
            Message message = new Message(subject, body);

            // Assemble the email.
          SendEmailRequest request = new SendEmailRequest(FROM, destination,
 message);

            // Choose the AWS region of the Amazon SES endpoint you want to
 connect to. Note that your sandbox
            // status, sending limits, and Amazon SES identity-related set
tings are specific to a given
            // AWS region, so be sure to select an AWS region in which you
 set up Amazon SES. Here, we are using
            // the US West (Oregon) region. Examples of other regions that
 Amazon SES supports are USEast1
            // and EUWest1. For a complete list, see ht
tp://docs.aws.amazon.com/ses/latest/DeveloperGuide/regions.html
            Amazon.RegionEndpoint REGION = Amazon.RegionEndpoint.USWest2;

           // Instantiate an Amazon SES client, which will make the service
 call.
          AmazonSimpleEmailServiceClient client = new AmazonSimpleEmailSer
viceClient(REGION);

            // Send the email.
            try
            {
              Console.WriteLine("Attempting to send an email through Amazon
 SES by using the AWS SDK for .NET...");
                client.SendEmail(request);
                Console.WriteLine("Email sent!");
            }
            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();
        }
    }
}
```

5. In Program.cs, replace the following with your own values:

   Important
   The email addresses are case-sensitive. Make sure that the addresses are exactly the same as the ones you verified.

   - SENDER@EXAMPLE.COM—Replace with your "From" email address. You must verify this address before you run this program. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 37).
   - RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 45).
   - REGION—Set this to the AWS region of the Amazon SES endpoint you want to connect to. Note that your sandbox status, sending limits, and Amazon SES identity-related settings are specific to a given AWS region, so be sure to select an AWS region in which you set up Amazon SES. In this example, we are using the US West (Oregon) region. Examples of other regions that Amazon SES supports are USEast1 and EUWest1. For a complete list of AWS regions that Amazon SES supports, see Regions and Amazon SES (p. 173).

6. Save Program.cs.
7. To build the project, click **Build** and then click **Build Solution**.
8. To run the program, click **Debug** and then click **Start Debugging**.
9. Review the program's console output to verify that the sending was successful. (You should see "Email sent!")
10. Log into the email client of the recipient address. You should find the email message that you sent.

Sending an Email Through Amazon SES Using AWS SDK for Java

The following procedure shows you how to use Eclipse IDE for Java EE Developers and AWS Toolkit for Eclipse to create an AWS SDK project and modify the Java code to send an email through Amazon SES. It retrieves your AWS credentials from environment variables.

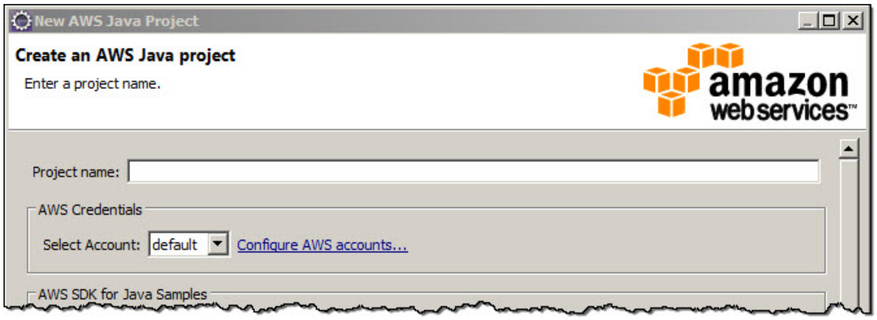
Before you begin this procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 19) and Sending an Email Through Amazon SES Using an AWS SDK (p. 28).

Important
In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator whenever possible. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 158).

To send an email using the AWS SDK for Java

1. Create an environment variable called AWS_ACCESS_KEY_ID and set it to your AWS access key ID. The procedure for setting environment variables depends on your operating system. Your AWS access key ID will look something like: `AKIAIOSFODNN7EXAMPLE`.
2. Create an environment variable called AWS_SECRET_ACCESS_KEY and set it to your AWS secret access key. Your AWS secret access key will look something like: `wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY`.
3. Create an AWS Java Project in Eclipse by performing the following steps:

   a. Open Eclipse.
   b. In Eclipse, click **File**, click **New**, and then click **AWS Java Project**. If you do not see **AWS Java Project** as an option, try selecting **Other**.
   c. In the **Create an AWS Java Project** dialog box, type a project name.



New AWS Java Project

Create an AWS Java project

Enter a project name.

Project name:

AWS Credentials

Select Account: default

   d. Click **Finish**.

4. In Eclipse, in the **Package Explorer** window, expand your project.
5. Under your project, right-click the **src** directory, click **New**, and then click **Class**.
6. In the **Java Class** dialog box, in the **Name** field, type `AmazonSESSample` and then click **Finish**.
7. Replace the entire contents of AmazonSESSample.java with the following code:

```
import java.io.IOException;
import com.amazonaws.services.simpleemail.*;
import com.amazonaws.services.simpleemail.model.*;
import com.amazonaws.regions.*;

public class AmazonSESSample {

    static final String FROM = "SENDER@EXAMPLE.COM";  // Replace with your
 "From" address. This address must be verified.
   static final String TO = "RECIPIENT@EXAMPLE.COM"; // Replace with a "To"
 address. If your account is still in the
                                                  // sandbox, this address
 must be verified.
    static final String BODY = "This email was sent through Amazon SES by
using the AWS SDK for Java.";
    static final String SUBJECT = "Amazon SES test (AWS SDK for Java)";


    public static void main(String[] args) throws IOException {
        // Construct an object to contain the recipient address.
        Destination destination = new Destination().withToAddresses(new 
String[]{TO});

        // Create the subject and body of the message.
        Content subject = new Content().withData(SUBJECT);
        Content textBody = new Content().withData(BODY);
        Body body = new Body().withText(textBody);

        // Create a message with the specified subject and body.
       Message message = new Message().withSubject(subject).withBody(body);


        // Assemble the email.
        SendEmailRequest request = new SendEmailRequest().with
Source(FROM).withDestination(destination).withMessage(message);

        try
        {
            System.out.println("Attempting to send an email through Amazon
 SES by using the AWS SDK for Java...");

           // Instantiate an Amazon SES client, which will make the service
 call. The service call requires your AWS credentials.
            // Because we're not providing an argument when instantiating 
the client, the SDK will attempt to find your AWS credentials
            // using the default credential provider chain. The first place
 the chain looks for the credentials is in environment variables 
            // AWS_ACCESS_KEY_ID and AWS_SECRET_KEY.
            // For more information, see http://docs.aws.amazon.com/AWSSdk
DocsJava/latest/DeveloperGuide/credentials.html
          AmazonSimpleEmailServiceClient client = new AmazonSimpleEmailSer
viceClient();

            // Choose the AWS region of the Amazon SES endpoint you want to
 connect to. Note that your sandbox 
            // status, sending limits, and Amazon SES identity-related set
tings are specific to a given AWS 
            // region, so be sure to select an AWS region in which you set
 up Amazon SES. Here, we are using 
            // the US West (Oregon) region. Examples of other regions that
 Amazon SES supports are US_EAST_1 
            // and EU_WEST_1. For a complete list, see ht
tp://docs.aws.amazon.com/ses/latest/DeveloperGuide/regions.html 
            Region REGION = Region.getRegion(Regions.US_WEST_2);
            client.setRegion(REGION);

            // Send the email.
            client.sendEmail(request);
            System.out.println("Email sent!");
        }
        catch (Exception ex)
        {
            System.out.println("The email was not sent.");
            System.out.println("Error message: " + ex.getMessage());
        }
    }
}
```

8. In AmazonSESSample.java, replace the following with your own values:

   Important
   The email addresses are case-sensitive. Make sure that the addresses are exactly the same as the ones you verified

   - SENDER@EXAMPLE.COM—Replace with your "From" email address. You must verify this address before you run this program. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 37).
   - RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If your account is still in the sandbox, you must verify this address before you use it. For more information, see Moving Out of the Amazon SES Sandbox (p. 45).
   - REGION—Set this to the AWS region of the Amazon SES endpoint you want to connect to. Note that your sandbox status, sending limits, and Amazon SES identity-related settings are specific to a given AWS region, so be sure to select an AWS region in which you set up Amazon SES. In this example, we are using the US West (Oregon) region. Examples of other regions that Amazon SES supports are US_EAST_1 and EU_WEST_1. For a complete list of AWS regions that Amazon SES supports, see Regions and Amazon SES (p. 173).

9. Save AmazonSESSample.java.
10. To build the project, click **Project** and then click **Build Project**. (If this option is disabled, you may have automatic building enabled.)
11. To start the program and send the email, click **Run** and then click **Run** again.
12. Review the program's console output to verify that the sending was successful. (You should see "Email sent!")
13. Log into the email client of the recipient address. You should find the email message that you sent.

Setting up Amazon SES

To set up Amazon Simple Email Service (Amazon SES), you need to perform the following tasks:

- Before you can access Amazon SES or other AWS services, you need to set up an AWS account. For more information, see Signing up for AWS (p. 36).
- Before you send email through Amazon SES, you need to verify that you own the "From" address. If your account is still in the Amazon SES sandbox, you also need to verify your "To" addresses. You can verify email addresses or entire domains. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 37).

The following tasks are optional depending on what you want to do:

- If you want to access Amazon SES through the Amazon SES API, whether by the Query (HTTPS) interface or indirectly through an AWS SDK, the AWS Command Line Interface or the AWS Tools for Windows PowerShell, you need to obtain your AWS access keys. For more information, see Getting Your AWS Access Keys (p. 45).
- If you want to call the Amazon SES API without handling the low-level details of the Query interface, you can use an AWS SDK. For more information, see Downloading an AWS SDK (p. 45).
- If you want to access Amazon SES through its SMTP interface, you need to obtain your SMTP user name and password. Your SMTP credentials are different from your AWS credentials. For more information, see Getting Your SMTP Credentials for Amazon SES (p. 45).
- When you first sign up for Amazon SES, your account is in the Amazon SES sandbox. In the sandbox, you can send emails using the same email-sending methods as any other Amazon SES user, except that you can only send 200 emails per 24-hour period at a maximum rate of one email per second, and you can only send emails to addresses you have verified. To increase your sending limits and to send email to unverified email addresses, see Moving Out of the Amazon SES Sandbox (p. 45).

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Signing up for AWS

You need to create an AWS account before you can use Amazon SES or other AWS services. When you create an AWS account, AWS automatically signs up your account for all services. You are charged only for the services that you use.

Note
If you will be sending your emails from an Amazon EC2 instance either directly or through AWS Elastic Beanstalk, you can get started with Amazon SES for free. For more information, see Amazon SES Pricing.

When you first sign up for Amazon AWS, your account is in the Amazon SES sandbox. In the sandbox, you have full access to the Amazon SES API and SMTP interface. However, the following restrictions are in effect:

- You can only send emails to the Amazon SES mailbox simulator and to email addresses or domains that you have verified. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 37).
- You can send a maximum of 200 messages per 24-hour period.
- You can send a maximum of one message per second.

For information about moving out of the sandbox, see Moving Out of the Amazon SES Sandbox (p. 45).

To create an AWS account

1. Go to http://aws.amazon.com/ses, and click *Sign up now*.
2. Follow the on-screen instructions.

Note
Even if your account is out of the Amazon SES sandbox, you still need to verify your "From" address to confirm that you own it.

Verifying Email Addresses and Domains in Amazon SES

Before you can send an email using Amazon SES, you must verify the address or domain that you are sending the email from to prove that you own it. If your account is still in the Amazon SES sandbox, you also need to verify any email addresses that you send emails to except for email addresses provided by the Amazon SES mailbox simulator. You can verify an email address or domain by using the Amazon SES console or by using the Amazon SES API.

Email address and domain verification status for each AWS region is separate. For more information, see the verification procedures in the following sections.

- For information about verifying email addresses, see Verifying Email Addresses in Amazon SES (p. 37).
- For information about verifying entire domains, see Verifying Domains in Amazon SES (p. 40).

Verifying Email Addresses in Amazon SES

Amazon SES requires that you verify your email address or domain, to confirm that you own it and to prevent others from using it. This section discusses verifying individual email addresses. For information about domain verification, see Verifying Domains in Amazon SES (p. 40).

With the exception of addresses containing labels (see below), you must verify each email address (or the domain of the email address) that you will use as a "From" or "Return-Path" address for your messages. Until your account is out of the Amazon SES sandbox, you must also verify the email address of every recipient except for the recipients provided by the Amazon SES mailbox simulator. For more information about the mailbox simulator, see Testing Amazon SES Email Sending (p. 158). For more information about moving out of the sandbox, see Moving Out of the Amazon SES Sandbox (p. 45).

Important notes about email address verification are as follows:

- The entire email address is case-sensitive. For example, if you verify *sender@example.com*, you cannot send emails from *sender@EXAMPLE.com* unless you verify *sender@EXAMPLE.com* also. (Domain verification, however, is case-insensitive. For more information, see Verifying Domains in Amazon SES (p. 40).)
- If you individually verify an email address and you also verify the domain of that address, the verified identity settings (such as DKIM and feedback notifications) of the email address override the domain-level settings. For example, if you verify *example.com* and *sender@example.com*, and you have DKIM enabled for *example.com* but not enabled for *sender@example.com*, then emails you send from *sender@example.com* will not be DKIM-signed.
- Amazon SES has endpoints in multiple AWS regions, and email address verification status is separate for each AWS region. You must complete the email address verification process for each sender in the AWS region(s) you want to use. For information about using Amazon SES in multiple AWS regions, see Regions and Amazon SES (p. 173).
- You can verify as many as 1000 identities (domains and email addresses, in any combination) per AWS account.
- Verifying an email address also allows you to set the "From" and "Return-Path" address to any address formed by adding a label to the verified address. Addresses that contain labels are of the form *name+label@example.com*, with user-specified text between the plus sign (+) and the at sign (@).

  For example, if you verify *user@example.com*, you can also send email from *user+recipient1@example.com*, *user+recipient2@example.com*, and so on. This makes it possible to support Variable Envelope Return Path (VERP) — the use of a different return path for each recipient. For more information about VERP, see http://en.wikipedia.org/wiki/Variable_envelope_return_path.

  When you verify an unlabeled address, then you are essentially verifying all addresses that are formed by adding a label to the verified address. The opposite, however is not true. Verifying an email address that already contains a label does not allow you to send from other addresses. For example, verifying *andrew+recipient1@example.com* does not allow you to send from *andrew@example.com*, *andrew+recipient2@example.com*, or *andrew+recipient1+recipient2@example.com*.
- If you want to use the `SendRawEmail` API action to send a message that contains a "Sender" header, then you must first verify the email address or domain in that header. For more information, see About Email Header Fields (p. 80).

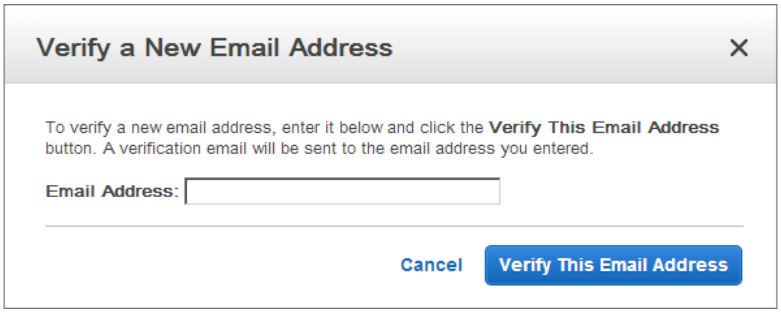
To verify an email address

1. Go to your email address list in the Amazon SES console, or follow these instructions to navigate to it:

   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.

   b. In the navigation pane, under **Identities**, click **Email Addresses**.

2. Click **Verify a New Email Address**.
3. In the **Verify a New Email Address** dialog box, type your email address in the indicated field, and then click **Verify This Email Address**.



Verify a New Email Address

To verify a new email address, enter it below and click the **Verify This Email Address** button. A verification email will be sent to the email address you entered.

Email Address:

4. In your email client, open the message from Amazon SES asking you to confirm that you are the owner of this email address.
5. Click the link in the message.

   Note
   The link in the verification message expires 24 hours after your original verification request.

6. The status of the email address in the Amazon SES console will change from "pending verification" to "verified".
7. You can now use Amazon SES to send email from this address. To send a test email, check the box next to the verified email address, and then click **Send a Test Email**.

To view your verified email addresses

1. Go to your email address list in the Amazon SES console, or follow these instructions to navigate to it:

   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
   b. In the navigation pane, under **Identities**, click **Email Addresses**.

2. In the list of verified email addresses, you can expand one or more email addresses to view the details.

To remove verified email addresses

1. Go to your email address list in the Amazon SES console, or follow these instructions to navigate to it:

   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
   b. In the navigation pane, under **Identities**, click **Email Addresses**.

2. Check the box beside each email address that you want to remove, and then click **Remove**.

Using the Amazon SES API

You can also manage verified email addresses with the Amazon SES API. The following actions are available:

- `VerifyEmailIdentity`
- `ListIdentities`

- `DeleteIdentity`
- `GetIdentityVerificationAttributes`

> **Note**
> The API actions above are preferable to the following older API actions, which are deprecated as of the May 15, 2012 release of Domain Verification.
>
> - `VerifyEmailAddress`
> - `ListVerifiedEmailAddresses`
> - `DeleteVerifiedEmailAddress`

You can use these API actions to write a customized front-end application for email address verification. For a complete description of the API actions related to email verification, go to the Amazon Simple Email Service API Reference.

Verifying Domains in Amazon SES

Amazon SES requires that you verify your email address or domain, to confirm that you own it and to prevent others from using it. When you verify an entire domain, you are verifying all email addresses from that domain, so you don't need to verify email addresses from that domain individually. For example, if you verify the domain *example.com*, you can send email from *user1@example.com*, *user2@example.com*, or any other user at *example.com*.

You can manage your verified domains by using the Amazon SES console or the Amazon SES API. For a complete description of API actions related to domain verification, go to the Amazon Simple Email Service API Reference. This section, which demonstrates the actions using the Amazon SES console, contains the following topics:

Important notes about domain verification are as follows:

- Amazon SES has endpoints in multiple AWS regions, and domain verification applies to each AWS region separately. You must perform the entire domain verification procedure for each region in which you want to send from a given domain. If you want to verify the same domain in multiple regions and your DNS provider does not allow you to have multiple TXT records with the same name, see the workarounds in Common Domain Verification Problems (p. 184).
- If you verify a domain with Amazon SES, you can send from any subdomain of that domain without specifically verifying the subdomain. For example, if you verify *example.com*, you do not need to verify *a.example.com* or *a.b.example.com*. As specified in RFC 1034, each DNS label can have up to 63 characters and the whole domain name must not exceed a total length of 255 characters.
- If you verify a domain, subdomain(s), and/or email address(es) that share a root domain, the verified identity settings (such as feedback notifications and Easy DKIM) apply at the most granular level you verified. That is:
  - Verified email address settings override verified domain settings.
  - Verified subdomain settings override verified domain settings, with lower-level subdomain settings overriding higher-level subdomain settings.

For example, assume you verify *user@a.b.example.com*, *a.b.example.com*, *b.example.com*, and *example.com*. These are the verified identity settings that will be used in the following scenarios:

  - Emails sent from *user@example.com* (an address that is not specifically verified) will use the settings for *example.com*.
  - Emails sent from *user@a.b.example.com* (an address that *is* specifically verified) will use the settings for *user@a.b.example.com*.
  - Emails sent from *user@b.example.com* (an address that is not specifically verified) will use the settings for *b.example.com*.

- Domain names are case-insensitive. If you verify *example.com*, you can send from *EXAMPLE.com* also.
- You can verify as many as 1000 identities (domains and email addresses, in any combination) per AWS account.

Verifying a Domain With Amazon SES

The following procedure shows you how to verify a domain using the Amazon SES console. If you want to use the Amazon SES API instead, see the Amazon Simple Email Service API Reference.

To verify a domain

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:

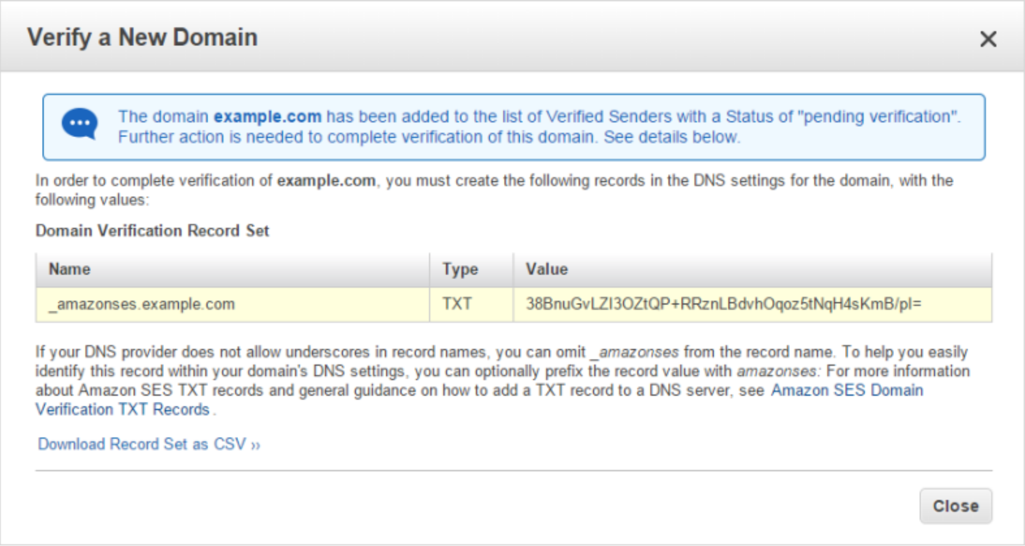
    a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.

    b. In the navigation pane, under **Identities**, click **Domains**.

2. Click **Verify a New Domain**.
3. In the **Verify a New Domain** dialog box, enter the domain name. If you want to set up DKIM signing for this domain, select the **Generate DKIM Settings** option. (For information about DKIM signing, see Authenticating Email with DKIM in Amazon SES (p. 94).) Click **Verify This Domain**.
4. In the **Verify a New Domain** dialog box, you will see a **Domain Verification Record Set** containing a **Name**, a **Type**, and a **Value**. (This information will also be available by clicking the domain name after you close the dialog box.)

   To complete domain verification, add a TXT record with the displayed **Name** and **Value** to your domain's DNS server. For information about Amazon SES TXT records and general guidance about how to add a TXT record to a DNS server, see Amazon SES Domain Verification TXT Records (p. 43). In particular:

   - If your DNS provider does not allow underscores in record names, you can omit *_amazonses* from the **Name**.
   - To help you easily identify this record within your domain's DNS settings, you can optionally prefix the **Value** with *amazonses:*
   - Some DNS providers automatically append the domain name to DNS record names. To avoid duplication of the domain name, you can add a period to the end of the domain name in the DNS record. This indicates that the record name is fully qualified and the DNS provider need not append an additional domain name.



Verify a New Domain

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

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

Domain Verification Record Set

Name	Type	Value
_amazonses.example.com	TXT	38BnuGvLZI3OZtQP+RRznLBdvhOqoz5tNqH4sKmB/pI=

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

Download Record Set as CSV ››

5. If Amazon Route 53 provides the DNS service for the domain that you are verifying, and you are logged in to the AWS Management Console under the same account that you use for Amazon Route 53, then Amazon SES will give you the option of updating your DNS settings immediately from within the Amazon SES console. If you are not using Amazon Route 53, Amazon SES needs to verify that a TXT record with the specified **Name** and **Value** have been added to your domain's DNS server. This may take up to 72 hours.

   When verification is complete, the domain's status in the Amazon SES console will change from "pending verification" to "verified," and you will receive a confirmation success email from Amazon SES to the email address associated with your AWS account.
6. You can now use Amazon SES to send email from any address in the verified domain. To send a test email, check the box next to the verified domain, and then click **Send a Test Email**.

If the DNS settings are not correctly updated, you will receive a domain verification failure email from Amazon SES, and the domain will display a status of "failed" in the **Domains** tab. If this happens, read our troubleshooting page at Amazon SES Domain Verification Problems (p. 183). When you have verified that your TXT record is correctly in place, click the "retry" link next to the "failed" status notification. This will reinitiate the domain verification process.

Viewing Your Domains Verified With Amazon SES

To view your verified domains, follow the procedure below.

To view your verified domains

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:

   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.

   b. In the navigation pane, under **Identities**, click **Domains**.

2. In the list of verified domains, you can expand one or more domains to view the details.

Removing a Domain Verified With Amazon SES

To remove a verified domain, follow the procedure below.

To remove a verified domain

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:

   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.

   b. In the navigation pane, under **Identities**, click **Domains**.

2. Check the box beside each domain that you want to remove, and then click **Remove**.
3. You will no longer be able to send email from the removed domain.

Amazon SES Domain Verification Revocation

Amazon SES periodically reviews domain verification status, and revokes verification in cases where it is no longer valid. If Amazon SES is unable to detect the TXT record information required to confirm ownership of a domain, you will receive an **Amazon SES Domain Verification REVOCATION WARNING** email from Amazon SES.

If you restore the TXT record information to your domain's DNS server within 72 hours, you will receive an **Amazon SES Domain Verification REVOCATION CANCELLATION** email from Amazon SES.

> **Note**
> You can review the required TXT record information in the Amazon SES console by using the following instructions. In the navigation pane, under **Identities**, click **Domains**. In the list of domains, click (not just expand) the domain to display the domain verification settings, which include the TXT record name and value.

If you do not restore the TXT record information to your domain's DNS server within 72 hours, you will receive an **Amazon SES Domain Verification REVOCATION** email from Amazon SES, the domain will be removed from the list of **Verified Senders** on the **Domains** tab, and you will no longer be able to send from the domain.

To reverify a domain for which verification has been revoked, you must restart the verification procedure from the beginning, just as if the revoked domain were an entirely new domain.

Amazon SES Domain Verification TXT Records

Your domain is associated with a set of Domain Name System (DNS) records that you manage through your DNS provider. A TXT record is a type of DNS record that provides additional information about your domain. Each TXT record consists of a name and a value.

When you initiate domain verification using the Amazon SES console or API, Amazon SES gives you the name and value to use for the TXT record. For example, if your domain is *example.com*, the TXT record settings that Amazon SES generates will look similar to the following example:

Name	Type	Value
_amazonses.example.com	TXT	pmBGN/7MjnfhTKUZ06Enqq1PeGUaOkw8lGhcfwefcHU=

Add a TXT record to your domain's DNS server using the specified **Name** and **Value**. Amazon SES domain verification is complete once Amazon SES detects the existence of the TXT record in your domain's DNS settings.

If your DNS provider does not allow DNS record names to contain underscores, you can omit *_amazonses* from the TXT record name. In that case, for the preceding example, the TXT record name would be *example.com* instead of *_amazonses.example.com*. To make the record easier to recognize and maintain, you can also optionally prefix the TXT record's value with *amazonses:*. In the example above, the value of the TXT record would therefore be *amazonses:pmBGN/7MjnfhTKUZ06Enqq1PeGUaOkw8lGhcfwefcHU=*.

> **Note**
> Amazon SES previously allowed TXT record names to contain *amazonses* without an underscore. If you have already verified a domain and your TXT record contains *amazonses* without an underscore, your domain will continue to be verified; there is no action required on your part. However, any new domains you verify will require *amazonses* to be preceded by an underscore.

You can find troubleshooting information and instructions on how to check your domain verification settings in Amazon SES Domain Verification Problems (p. 183).

Adding a TXT Record to Your Domain's DNS Server

The procedure for adding TXT records to your domain's DNS server depends on who provides your DNS service. Your DNS provider might be Amazon Route 53 or another domain name registrar such as GoDaddy. Although we cannot provide specific instructions about how to add TXT records to your domain's DNS server, here is the general procedure DNS providers typically use.

To add a TXT record to your domain's DNS server (general procedure)

1. Go to your DNS provider's website. If you are not sure which DNS provider serves your domain, try looking it up by using a free Whois service.
2. Sign in to your domain's account.
3. Find the page for updating your domain's DNS records. This page often has a name such as DNS Records, DNS Zone File, Advanced DNS, or something similar.
4. Locate the TXT records for your domain.
5. Add a TXT record with the name and value provided by Amazon SES.

   > **Important**
   > Some DNS providers automatically append the domain name to the end of DNS records. Adding a record that already contains the domain name (such as *_amazonses.example.com*) might result in the duplication of the domain name (such as *_amazonses.example.com.example.com*). To avoid duplication of the domain name, add a period to the end of the domain name in the DNS record. This will indicate to your DNS provider that the record name is fully qualified (that is, no longer relative to the domain name), and prevent the DNS provider from appending an additional domain name.

6. Save your changes. DNS record updates can take up to 48 hours to take effect, but they often take effect much sooner. You can verify that the TXT record is correctly published by using the procedure in How to Check Domain Verification Settings (p. 183).

Getting Your AWS Access Keys

After you've signed up for Amazon SES, you'll need to obtain your AWS access keys if you want to access Amazon SES through the Amazon SES API, whether by the Query (HTTPS) interface directly or indirectly through an AWS SDK, the AWS Command Line Interface or the AWS Tools for Windows PowerShell. AWS access keys consist of an access key ID and a secret access key.

For information about getting your AWS access keys, see How Do I Get Security Credentials? in the *AWS General Reference*.

Downloading an AWS SDK

If you want to call the Amazon SES API without having to handle low-level details like assembling raw HTTP requests, you can use an AWS SDK. The AWS SDKs provide functions and data types that encapsulate the functionality of Amazon SES and other AWS services. AWS SDKs and resources are available for Android, iOS, Java, .NET, Node.js, PHP, Python, and Ruby.

To download an AWS SDK, go to the appropriate link listed above. The prerequisites and installation instructions for each SDK are listed on the corresponding page.

> **Note**
> The getting started section of this developer guide provides examples of how to send an email by using the AWS SDKs for .NET and Java. For more information, see Sending an Email Through Amazon SES Using an AWS SDK (p. 28).

To see a list of all the AWS SDKs, go to Sample Code and Libraries.

Setting up SPF Records for Amazon SES

You do not need to make any changes to your DNS records for your emails to pass Sender Policy Framework (SPF) authentication. Amazon SES sends your emails from a "Mail-From" domain that Amazon SES owns.

Getting Your SMTP Credentials for Amazon SES

To use the Amazon SES SMTP interface, you must first create an SMTP user name and password. To get your SMTP Credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

> **Important**
> Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself to the Amazon SES SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 170).

Moving Out of the Amazon SES Sandbox

To help protect our customers from fraud and abuse and to help you establish your trustworthiness to ISPs and email recipients, we do not immediately grant unlimited Amazon SES usage to new users. New users are initially placed in the Amazon SES *sandbox*. In the sandbox, you have full access to all Amazon

SES email-sending methods and features so that you can test and evaluate the service; however, the following restrictions are in effect:

- You can only send mail to the Amazon SES mailbox simulator and to verified email addresses and domains.
- You can only send mail from verified email addresses and domains.
- You can send a maximum of 200 messages per 24-hour period.
- Amazon SES can accept a maximum of one message from your account per second.

To remove the restriction on recipient addresses and increase your sending limits, you need to open a case in Support Center by using the following instructions.

To move out of the Amazon SES sandbox

1. Log into the AWS Management Console.
2. Go to SES Sending Limits Increase. Alternatively, you can go to Support Center, click **Create Case**, click **Service Limit Increase**, and then select **SES Sending Limits** as the limit type.
3. In the form, provide the following information:

   - **Region:** Select the AWS region for which you are requesting a sending limit increase. Note that your Amazon SES sandbox status and sending limits are separate for each AWS region. For more information, see Regions and Amazon SES (p. 173).
   - **Limit:** Select *Desired Daily Sending Quota* or *Desired Maximum Send Rate*. Sending limits are described in Managing Your Amazon SES Sending Limits (p. 124).

     > **Note**
     >
     > The rate at which Amazon SES accepts your messages might be less than the maximum send rate.

   - **New limit value:**Enter the amount you are requesting. **Be sure to only request the amount you think you'll need.** Keep in mind that you are not guaranteed to receive the amount you request, and the higher the limit you request, the more justification you will need to be considered for that amount.
   - **Mail type:** Select *Transactional*, *System Notifications*, *Subscription*, *Marketing*, or *Other*.
   - **Website URL.** Provide a link to your website. Although it isn't required, we highly recommend that you provide one if you have it, because it helps us evaluate your request.
   - **My email-sending complies with the AWS Service Terms and AWS Acceptable Use Policy (AUP).** Select *Yes* or *No*.
   - **I only send to recipients who have specifically requested my mail.** Select *Yes* or *No*. For tips on how to send high-quality mail and keep your recipient list clean, see Obtaining and Maintaining Your Recipient List (p. 178) and the Amazon Simple Email Service Email Sending Best Practices white paper.
   - **I have a process to handle bounces and complaints.** Select *Yes* or *No*. For information on how to monitor and handle bounces and complaints, see Processing Bounces and Complaints (p. 178).
   - **Use Case Description.** Explain your situation in as much detail as possible. For example, describe the type of emails you are sending and how email-sending fits into your business. The more information you can provide that indicates that you are sending high-quality emails to recipients who want and expect it, the more likely we are to approve your request. The higher the limit value you are requesting, the more detail you should provide.

   We will respond to the case after reviewing your request. Please allow one business day for processing. If you are granted a sending limit increase, then you have also been moved out of the sandbox and no longer need to verify your "To" addresses.

The following are three ways to determine whether you have moved out of the sandbox:

- The correspondence in your SES Sending Limits Increase case indicates that your request has been granted.
- You can successfully use Amazon SES to send an email message from your *verified* email address to an unverified address that you own. If you receive a *MessageRejected* error instead, stating that your email address is not verified, then you are still in the sandbox.
- The Amazon SES console shows that your sending quota is higher than 200 messages per 24-hour period. To learn more, see Monitoring Your Amazon SES Sending Limits (p. 122).

Once you are out of the sandbox, you no longer have to verify "To" addresses or domains; however, you must still verify any additional "From" or "Return-Path" addresses or domains. Over time, Amazon SES will gradually increase your sending limits, or you can open another SES Sending Limits Increase case if the gradual increase does not meet your needs. For more information, see Managing Your Amazon SES Sending Limits (p. 124).

Sending Email with Amazon SES

You can send an email with Amazon Simple Email Service (Amazon SES) by using the Amazon SES console, the Amazon SES Simple Mail Transfer Protocol (SMTP) interface, or the Amazon SES API. You typically use the console to send test emails and manage your sending activity. To send bulk emails, you use either the SMTP interface or the API. For information about Amazon SES email pricing, see Pricing on the Amazon SES detail page.

- If you want to use an SMTP-enabled software package, application, or programming language to send email through Amazon SES, or integrate Amazon SES with your existing mail server, use the Amazon SES SMTP interface. For more information, see Using the Amazon SES SMTP Interface to Send Email (p. 48).
- If you want to call Amazon SES by using raw HTTP requests, use the Amazon SES API. For more information, see Using the Amazon SES API to Send Email (p. 79).

Before you send emails, see Setting up Amazon SES (p. 36).

> **Important**
> When you send an email to multiple recipients (recipients are "To", "CC", and "BCC" addresses) and the call to Amazon SES fails, the entire email is rejected and none of the recipients will receive the intended email. We therefore recommend that you send an email to one recipient at a time.

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Using the Amazon SES SMTP Interface to Send Email

To send production email through Amazon SES, you can use the Simple Mail Transfer Protocol (SMTP) interface or the Amazon SES API. For more information about the Amazon SES API, see Using the Amazon SES API to Send Email (p. 79). This section describes the SMTP interface.

Amazon SES sends email using the SMTP, the most common email protocol on the Internet. You can send email through Amazon SES by using a variety of SMTP-enabled programming languages and software to connect to Amazon SES's native SMTP interface. This section explains how to get your

Amazon SES SMTP credentials, how to send email by using the SMTP interface, and how to configure several pieces of software and mail servers to use Amazon SES for email sending.

> **Note**
> For solutions to common problems that you might encounter when you use Amazon SES through its SMTP interface, see Amazon SES SMTP Issues (p. 189).

To send email using the Amazon SES SMTP interface, you will need the following:

- An AWS account. For more information, see Signing up for AWS (p. 36).
- The SMTP interface hostname (i.e., endpoint). For a list of Amazon SES SMTP endpoints, see Connecting to the Amazon SES SMTP Endpoint (p. 53).
- The SMTP interface port number. The port number varies with the connection method. For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 53).
- An SMTP user name and password. You can use the same set of SMTP credentials in all AWS regions.

  > **Important**
  > Your SMTP user name and password are not identical to your AWS access keys or the credentials you use to log into the Amazon SES console. For information about how to generate your SMTP user name and password, see Obtaining Your Amazon SES SMTP Credentials (p. 50).
- Client software that can communicate using Transport Layer Security (TLS). For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 53).
- An email address that you have verified with Amazon SES. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 37).
- Higher sending limits, if you want to send large quantities of email. For more information, see Managing Your Amazon SES Sending Limits (p. 124).

Then, you can send email by doing the following:

- To configure an email client to send email through Amazon SES, including an example for Microsoft Outlook, see Configuring Email Clients to Send Through Amazon SES (p. 54).
- To configure SMTP-enabled software to send email through the Amazon SES SMTP interface, including an example for issue-tracking software Jira, see Sending Email Through Amazon SES From Software Packages (p. 57).
- To program an application to send email through Amazon SES, see Sending Email Through Amazon SES From Your Application (p. 58).
- To configure your existing email server to send all of your outgoing mail through Amazon SES, see Integrating Amazon SES with Your Existing Email Server (p. 58).
- To interact with the Amazon SES SMTP interface using the command line, which can be useful for testing, see Using the Command Line to Send Email Through the Amazon SES SMTP Interface (p. 76).

For a list of SMTP response codes, see SMTP Response Codes Returned by Amazon SES (p. 191).

Email Information to Provide

When you access Amazon SES through the SMTP interface, your SMTP client application assembles the message, so the information you need to provide depends on the application you are using. At a minimum, the SMTP exchange between a client and a server requires a source address, a destination address, and message data.

If you are using the SMTP interface and have feedback forwarding enabled, then your bounces, complaints, and delivery notifications are sent to the "MAIL FROM" address. Any "Reply-To" address that you specify is not used.

Obtaining Your Amazon SES SMTP Credentials

You need an Amazon SES SMTP user name and password to access the Amazon SES SMTP interface. You can use the same set of SMTP credentials in all AWS regions.

> **Important**
> Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 170).

There are two ways to generate your SMTP credentials. You can either use the Amazon SES console or you can generate your SMTP credentials from your AWS credentials.

Use the Amazon SES console to generate your SMTP credentials if:

- You want to get your SMTP credentials using the simplest method.
- You do not need to automate SMTP credential generation using code or a script.

Generate your SMTP credentials from your AWS credentials if:

- You have an existing AWS Identity and Access Management (IAM) user that you created using the IAM interface and you want that user to be able to send emails using the Amazon SES SMTP interface.
- You want to automate SMTP credential generation using code or a script.

For information on each method, see Obtaining Amazon SES SMTP Credentials Using the Amazon SES Console (p. 50) and Obtaining Amazon SES SMTP Credentials by Converting AWS Credentials (p. 51).

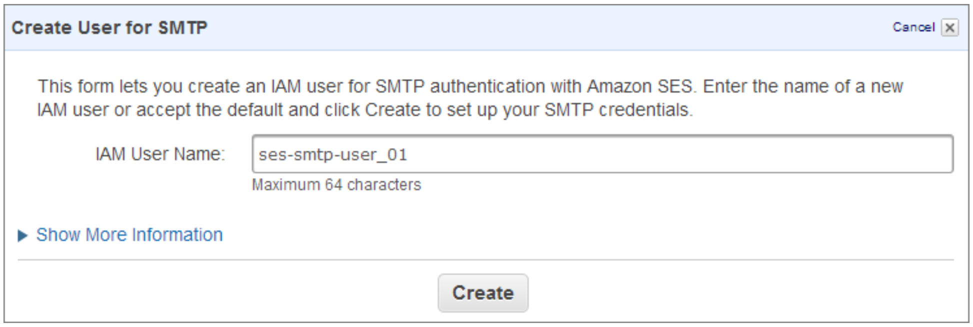
Obtaining Amazon SES SMTP Credentials Using the Amazon SES Console

When you generate SMTP credentials by using the Amazon SES console, the Amazon SES console creates an IAM user with the appropriate policies to call Amazon SES and provides you with the SMTP credentials associated with that user.

> **Note**
> An IAM user can create Amazon SES SMTP credentials, but the IAM user's policy must give them permission to use IAM itself, because Amazon SES SMTP credentials are created through IAM. If the IAM user tries to create Amazon SES SMTP credentials using the console and they don't have IAM permissions, they will get an error that says "… not authorized to perform iam:ListUsers…" In that case, the root account owner needs to modify the IAM user's policy to allow them to access the following IAM actions: "iam:ListUsers", "iam:CreateUser", "iam:CreateAccessKey", and "iam:PutUserPolicy".

To create your SMTP credentials

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the navigation pane, click **SMTP Settings**.
3. In the content pane, click **Create My SMTP Credentials**.
4. In the **Create User for SMTP** dialog box, you will see that an SMTP user name has been filled in for you. You can accept this suggested user name or enter a different one. To proceed, click **Create**.

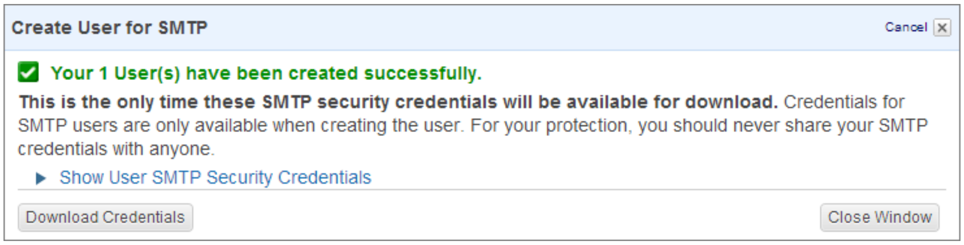


This form lets you create an IAM user for SMTP authentication with Amazon SES. Enter the name of a new IAM user or accept the default and click Create to set up your SMTP credentials.

IAM User Name: ses-smtp-user_01

Maximum 64 characters

5. Click **Show User SMTP Credentials**. Your SMTP credentials will be displayed on the screen; copy them and store them in a safe place. You can also click **Download Credentials** to download a file that contains your credentials.



☑ Your 1 User(s) have been created successfully.

This is the only time these SMTP security credentials will be available for download. Credentials for SMTP users are only available when creating the user. For your protection, you should never share your SMTP credentials with anyone.

> **Important**
> This is the only time that you will be able to view your SMTP credentials! We strongly advise you to download these credentials and refrain from sharing them with others.

6. Click **Close Window**.

If you want to delete your SMTP credentials, go to the IAM console at https://console.aws.amazon.com/iam/home and delete the IAM user name that corresponds with your SMTP credentials. To learn more, go to the Using IAM guide.

If you want to change your SMTP password, go to the IAM console and delete your existing IAM user, and then go to the Amazon SES console to re-generate your SMTP credentials.

Obtaining Amazon SES SMTP Credentials by Converting AWS Credentials

If you have an IAM user that you set up using the IAM interface, you can derive the user's Amazon SES SMTP credentials from their AWS credentials.

> **Important**
> Do not use temporary AWS credentials to derive SMTP credentials. The Amazon SES SMTP interface does not support SMTP credentials that have been generated from temporary security credentials.

To enable the IAM user to send email using the Amazon SES SMTP interface, you need to do the following two steps:

- Derive the user's SMTP credentials from their AWS credentials using the algorithm provided in this section. A user's SMTP username is the same as their AWS Access Key ID, so you just need to generate the SMTP password.
- Apply the following policy to the IAM user:

```
{ "Statement": [{
        "Effect":"Allow",
        "Action":"ses:SendRawEmail",
        "Resource":"*"
}]}
```

For more information about using Amazon SES with IAM, see Controlling User Access to Amazon SES (p. 128).

Note
Although you can generate Amazon SES SMTP credentials for any existing IAM user, we recommend for security reasons that you create a separate IAM user for the AWS credentials that you will use to generate the SMTP password. For information about why it is good practice to create users for specific purposes, go to IAM Best Practices.

The following pseudocode shows the algorithm that converts an AWS Secret Access Key to an Amazon SES SMTP password.

```
key = AWS Secret Access Key;
message = "SendRawEmail";
versionInBytes = 0x02;
signatureInBytes = HmacSha256(message, key);
signatureAndVer = Concatenate(versionInBytes, signatureInBytes);
smtpPassword = Base64(signatureAndVer);
```

The following is an example Java implementation that converts an AWS Secret Access Key to an Amazon SES SMTP password. Before you run the program, put the AWS Secret Access Key of the IAM user into an environment variable called AWS_SECRET_ACCESS_KEY. The output of the program is the SMTP password. That password, along with the SMTP username (which is the same as the AWS Access Key ID) are the user's Amazon SES SMTP credentials.

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class SesSmtpCredentialGenerator {
       private static final String KEY_ENV_VARIABLE = "AWS_SECRET_ACCESS_KEY";
 // Put your AWS secret access key in this environment variable.
      private static final String MESSAGE = "SendRawEmail"; // Used to generate
 the HMAC signature. Do not modify.
       private static final byte VERSION =  0x02; // Version number. Do not
modify.

       public static void main(String[] args) {

              // Get the AWS secret access key from environment variable
AWS_SECRET_ACCESS_KEY.
              String key = System.getenv(KEY_ENV_VARIABLE);
              if (key == null)
              {
                 System.out.println("Error: Cannot find environment variable
AWS_SECRET_ACCESS_KEY.");
                 System.exit(0);
              }
```

```
             // Create an HMAC-SHA256 key from the raw bytes of the AWS secret
 access key.
              SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(),
"HmacSHA256");

              try {
                     // Get an HMAC-SHA256 Mac instance and initialize it with
 the AWS secret access key.
                     Mac mac = Mac.getInstance("HmacSHA256");
                     mac.init(secretKey);

                     // Compute the HMAC signature on the input data bytes.
                     byte[] rawSignature = mac.doFinal(MESSAGE.getBytes());

                     // Prepend the version number to the signature.
                     byte[] rawSignatureWithVersion = new byte[rawSigna
ture.length + 1];
                     byte[] versionArray = {VERSION};
                    System.arraycopy(versionArray, 0, rawSignatureWithVersion,
 0, 1);
                    System.arraycopy(rawSignature, 0, rawSignatureWithVersion,
 1, rawSignature.length);

                    // To get the final SMTP password, convert the HMAC signa
ture to base 64.
                     String smtpPassword = DatatypeConverter.printBase64Bin
ary(rawSignatureWithVersion);
                     System.out.println(smtpPassword);
              }
              catch (Exception ex) {
                     System.out.println("Error generating SMTP password: " +
ex.getMessage());
              }
       }
}
```

Connecting to the Amazon SES SMTP Endpoint

The following table shows the Amazon SES SMTP endpoints for the regions in which Amazon SES is available.

Region name	SMTP endpoint
US East (N. Virginia)	email-smtp.us-east-1.amazonaws.com
US West (Oregon)	email-smtp.us-west-2.amazonaws.com
EU (Ireland)	email-smtp.eu-west-1.amazonaws.com

The Amazon SES SMTP endpoint requires that all connections be encrypted using Transport Layer Security (TLS). (Note that TLS is often referred to by the name of its predecessor protocol, SSL.) Amazon SES supports two mechanisms for establishing a TLS-encrypted connection: STARTTLS and TLS Wrapper. Check the documentation for your software to determine whether it supports STARTTLS, TLS Wrapper, or both.

If your software does not support STARTTLS or TLS Wrapper, you can use the open source *stunnel* program to set up an encrypted connection (called a "secure tunnel"), then use the secure tunnel to connect to the Amazon SES SMTP endpoint.

> **Important**
> Amazon Elastic Compute Cloud (Amazon EC2) throttles email traffic over port 25 by default. To avoid timeouts when sending email through the SMTP endpoint from EC2, use a different port (587 or 2587) or fill out a Request to Remove Email Sending Limitations to remove the throttle.

STARTTLS

STARTTLS is a means of upgrading an unencrypted connection to an encrypted connection. There are versions of STARTTLS for a variety of protocols; the SMTP version is defined in RFC 3207.

To set up a STARTTLS connection, the SMTP client connects to the Amazon SES SMTP endpoint on port 25, 587, or 2587, issues an EHLO command, and waits for the server to announce that it supports the STARTTLS SMTP extension. The client then issues the STARTTLS command, initiating TLS negotiation. When negotiation is complete, the client issues an EHLO command over the new encrypted connection, and the SMTP session proceeds normally.

TLS Wrapper

TLS Wrapper (also known as SMTPS or the Handshake Protocol) is a means of initiating an encrypted connection without first establishing an unencrypted connection. With TLS Wrapper, the Amazon SES SMTP endpoint does not perform TLS negotiation: it is the client's responsibility to connect to the endpoint using TLS, and to continue using TLS for the entire conversation. TLS Wrapper is an older protocol, but many clients still support it.

To set up a TLS Wrapper connection, the SMTP client connects to the Amazon SES SMTP endpoint on port 465 or 2465. The server presents its certificate, the client issues an EHLO command, and the SMTP session proceeds normally.

Secure Tunnel

If your software does not support STARTTLS or TLS Wrapper, you can set up a secure tunnel to allow your software to communicate with the Amazon SES SMTP endpoint. As this option is most commonly used by mail server administrators, details are given under Integrating Amazon SES with Your Existing Email Server (p. 58).

Configuring Email Clients to Send Through Amazon SES

After you have obtained your SMTP credentials, you can begin sending email through Amazon SES. You can use any email client application, provided that it can communicate via SMTP and connect to an SMTP endpoint using Transport Layer Security (TLS). Most email clients can send email using an SMTP server.

To configure your email client, you must provide the Amazon SES SMTP interface hostname and port number (see Connecting to the Amazon SES SMTP Endpoint (p. 53)), along with your SMTP user name and password (see Obtaining Your Amazon SES SMTP Credentials (p. 50)).

The following procedure shows how to configure one such client, Microsoft Outlook 2010, so that it can send email through the Amazon SES SMTP interface. If you are using a different email client, follow the instructions provided by the client vendor, and use the settings described in the following procedure.

To configure Microsoft Outlook 2010 for sending via Amazon SES

1. On the **File** menu, click the **Info** link on the side of the page to reveal the **Account Information** page.
2. Click the **Add Account** button.
3. On the **Auto Account Setup** page, select the **Manually configure server settings or additional server types** option, and then click **Next**.
4. On the **Choose Service** window, choose **Internet E-Mail** and then click **Next**.
5. On the **Internet E-mail settings** form, fill in the following fields:

   a. **Your Name**—Your real name.

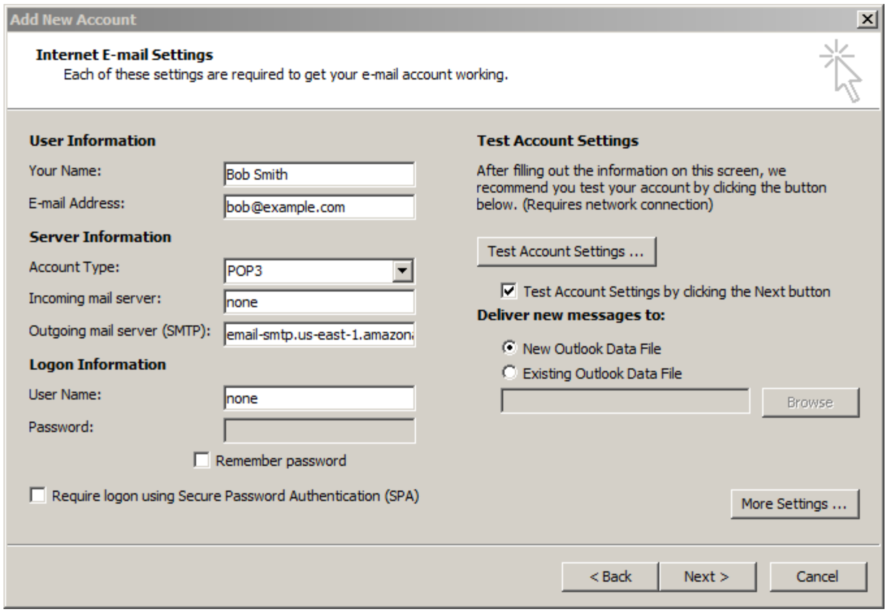
   b. **E-Mail Address**—The address from which emails will be sent. You will need to verify this email address or its domain before you can send from it. For more information about verifying email addresses and domains, see Verifying Email Addresses and Domains in Amazon SES (p. 37).

   c. **Account Type**—POP3

   d. **Incoming mail server**—type the word *none*. (Even though Amazon SES does not receive email, Outlook still requires that you fill in this field.)

   e. **Outgoing mail server (SMTP)**—See Connecting to the Amazon SES SMTP Endpoint (p. 53) for a list of Amazon SES SMTP endpoints. For example, if you want to use the Amazon SES endpoint in the US West (Oregon) region, the outgoing mail server would be *email-smtp.us-west-2.amazonaws.com*.

   f. **User Name**—type the word *none*. (You will configure your credentials later in this procedure.)



Add New Account

Internet E-mail Settings
Each of these settings are required to get your e-mail account working.

User Information
Your Name: Bob Smith
E-mail Address: bob@example.com

Server Information
Account Type: POP3
Incoming mail server: none
Outgoing mail server (SMTP): email-smtp.us-east-1.amazon

Logon Information
User Name: none
Password:
□ Remember password
□ Require logon using Secure Password Authentication (SPA)

Test Account Settings
After filling out the information on this screen, we recommend you test your account by clicking the button below. (Requires network connection)
☑ Test Account Settings by clicking the Next button
Deliver new messages to:
● New Outlook Data File
○ Existing Outlook Data File

6. Click **More Settings**.
7. In the **Internet E-mail Settings** dialog box, click the **Outgoing Server** tab and fill in the following fields:

   a. **My outgoing server (SMTP) requires authentication**—check this box.

   b. **Log on using**—choose this option.

   c. **User Name**—enter your SMTP user name.

Important

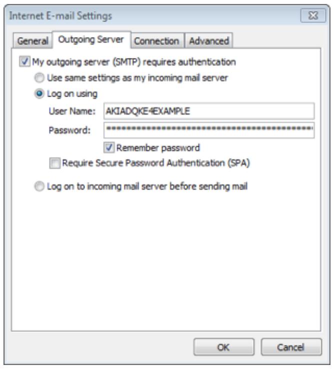
Use your SMTP user name, not your AWS access key ID. Your SMTP credentials and your AWS credentials are not the same. For information about how to obtain your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

d. **Password**—enter your SMTP password.

Important

Use your SMTP password, not your AWS secret access key. Your SMTP credentials and your AWS credentials are not the same. For information about how to obtain your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

e. **Remember Password**—check this box.



Internet E-mail Settings

General | Outgoing Server | Connection | Advanced

☑ My outgoing server (SMTP) requires authentication

○ Use same settings as my incoming mail server

● Log on using

User Name: AKIADQKE4EXAMPLE

Password: **

☑ Remember password

☐ Require Secure Password Authentication (SPA)

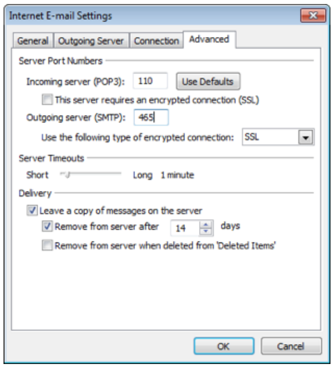
○ Log on to incoming mail server before sending mail

8. In the **Internet E-mail Settings** dialog box, click the **Advanced** tab and fill in the following fields:

   a. **Outgoing server (SMTP)**—25, 587, or 2587 (to connect using STARTTLS), or 465 or 2465 (to connect using TLS Wrapper).

   b. **Use the following type of encrypted connection**—TLS (to connect using STARTTLS) or SSL (to connect using TLS Wrapper).

   Settings for TLS Wrapper are shown below.



Internet E-mail Settings

General | Outgoing Server | Connection | Advanced

Server Port Numbers

Incoming server (POP3): 110 [Use Defaults]

☐ This server requires an encrypted connection (SSL)

Outgoing server (SMTP): 465

Use the following type of encrypted connection: SSL

Server Timeouts

Short — Long 1 minute

Delivery

☑ Leave a copy of messages on the server

☑ Remove from server after 14 days

☐ Remove from server when deleted from 'Deleted Items'

9. Click **OK**.

10. On the **Internet E-mail Settings** page of the wizard, click **Test Account Settings**. This lets you test your setup by having Outlook send an email through Amazon SES.

    Note

    Because you did not specify an incoming mail server (Amazon SES is only used for outgoing email), the *Log onto incoming mail server* test is expected to fail. The *Send test e-mail* test should pass.

11. If the test message that Outlook sends through Amazon SES arrives successfully, clear the **Test Account Settings by clicking the Next button** checkbox (because the test will fail without setting up incoming email) and then click **Next**.
12. Click **Finish**.
13. Amazon SES is for mail sending only. To prevent Microsoft Outlook from attempting to receive messages through the account you just set up, you need to disable mail retrieval for the account by using the following steps.

    a. In Microsoft Outlook, click the **Send/Receive** tab.
    b. On the **Send/Receive** tab, click **Send/Receive Groups**, and then click **Define Send/Receive Groups**.
    c. In the **Send/Receive Groups** dialog box, click **Edit**.
    d. In the left panel of the **Send/Receive Settings - All Accounts** dialog box, click the account you just created for sending mail through Amazon SES.
    e. Under **Account Options**, clear **Receive mail items**.
    f. Click **OK**, and then click **Close**.

Sending Email Through Amazon SES From Software Packages

There are a number of commercial and open source software packages that support sending email via SMTP. Here are some examples:

- Blogging platforms
- RSS aggregators
- List management software
- Workflow systems

You can configure any such SMTP-enabled software to send email through the Amazon SES SMTP interface. For instructions on how to configure SMTP for a particular software package, see the documentation for that software.

The following procedure shows how to set up Amazon SES sending in JIRA, a popular issue-tracking solution. With this configuration, JIRA can notify users via email whenever there is a change in the status of a software issue.

To Configure JIRA to Send Email Using Amazon SES

1. Using your web browser, log in to JIRA with administrator credentials.
2. In the browser window, click **Administration**.
3. On the **System** menu, click **Mail**.
4. On the **Mail administration** page, click **Mail Servers**.
5. Click **Configure new SMTP mail server**.
6. On the **Add SMTP Mail Server** form, fill in the following fields:

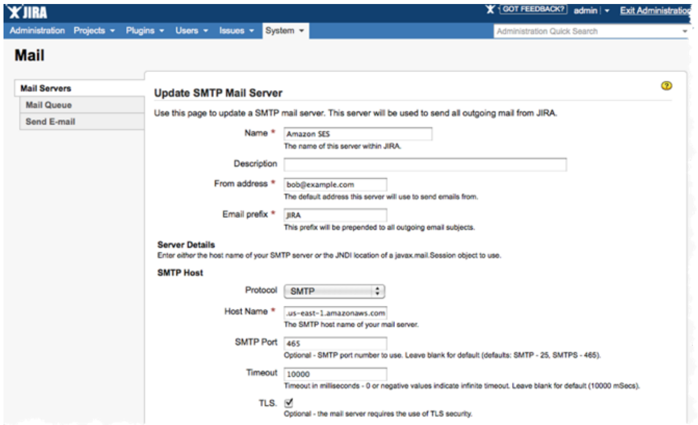
    a. **Name**—A descriptive name for this server.
    b. **From address**—The address from which email will be sent. You will need to verify this email address with Amazon SES before you can send from it. For more information about verification, see Verifying Email Addresses and Domains in Amazon SES (p. 37).

c. **Email prefix**—A string that JIRA prepends to each subject line prior to sending.
d. **Protocol**—Choose **SMTP**.

> **Note**
> If you cannot connect to Amazon SES using this setting, try **SECURE_SMTP**.

e. **Host Name**—See Connecting to the Amazon SES SMTP Endpoint (p. 53) for a list of Amazon SES SMTP endpoints. For example, if you want to use the Amazon SES endpoint in the US West (Oregon) region, the host name would be *email-smtp.us-west-2.amazonaws.com*.
f. **SMTP Port**—25, 587, or 2587 (to connect using STARTTLS), or 465 or 2465 (to connect using TLS Wrapper).
g. **TLS**—Select this check box.
h. **Username**—Your SMTP username.
i. **Password**—Your SMTP password.

Settings for TLS Wrapper are shown below.



Mail

Update SMTP Mail Server

Use this page to update a SMTP mail server. This server will be used to send all outgoing mail from JIRA.

Name * Amazon SES
The name of this server within JIRA.

Description

From address * bob@example.com
The default address this server will use to send emails from.

Email prefix * JIRA
This prefix will be prepended to all outgoing email subjects.

Server Details
Enter either the host name of your SMTP server or the JNDI location of a javax.mail.Session object to use.

SMTP Host

Protocol SMTP

Host Name * .us-east-1.amazonaws.com
The SMTP host name of your mail server.

SMTP Port 465
Optional - SMTP port number to use. Leave blank for default (defaults: SMTP - 25, SMTPS - 465).

Timeout 10000
Timeout in milliseconds - 0 or negative values indicate infinite timeout. Leave blank for default (10000 mSecs).

TLS. ☑
Optional - the mail server requires the use of TLS security.

7. Click **Test Connection**. If the test email that JIRA sends through Amazon SES arrives successfully, then your configuration is complete.

Sending Email Through Amazon SES From Your Application

Many programming languages support sending email using SMTP. This capability might be built into the programming language itself, or it might be available as an add-on, plug-in, or library. You can take advantage of this capability by sending email through Amazon SES from within application programs that you write.

For examples in C# and Java, see Sending an Email Through the Amazon SES SMTP Interface Programmatically (p. 21) in the Getting Started section.

Integrating Amazon SES with Your Existing Email Server

If you currently administer your own email server, you can use the Amazon SES SMTP endpoint to send all of your outgoing email to Amazon SES. There is no need to modify your existing email clients and applications; the changeover to Amazon SES will be transparent to them.

Several mail transfer agents (MTAs) support sending email through SMTP relays. This section provides general guidance on how to configure some popular MTAs to send email using Amazon SES SMTP interface.

- To configure Postfix to send email through Amazon SES, see Integrating Amazon SES with Postfix (p. 60).
- To configure Sendmail to send email through Amazon SES, see Integrating Amazon SES with Sendmail (p. 63).
- To configure Microsoft Exchange to send email through Amazon SES, see Integrating Amazon SES with Microsoft Exchange (p. 68).
- To configure Microsoft Windows Server's IIS SMTP server to send email through Amazon SES, see Integrating Amazon SES with Microsoft Windows Server IIS SMTP (p. 72).
- To configure Exim to send email through Amazon SES, see Integrating Amazon SES with Exim (p. 74).

The Amazon SES SMTP endpoint requires that all connections be encrypted using Transport Layer Security (TLS). If you want to use TLS Wrapper but your MTA does not support TLS Wrapper, you can set up a "secure tunnel" to provide TLS Wrapper support. For more information, see Setting Up a Secure Tunnel to Connect to Amazon SES (p. 59).

Setting Up a Secure Tunnel to Connect to Amazon SES

The Amazon SES SMTP endpoint requires that all connections be encrypted using Transport Layer Security (TLS). If you want to use TLS Wrapper to connect to the Amazon SES SMTP endpoint, but your MTA does not support TLS Wrapper, you can set up a "secure tunnel" to provide TLS Wrapper support. One way to do this is by using the open source *stunnel* program. Note that *stunnel* is intended to be used for port 465, the SSL port, only.

> **Important**
> Some MTAs have native support for TLS Wrapper, while others do not. Check the documentation for your mail server to determine whether it supports TLS Wrapper. If it supports TLS Wrapper, then you do not need to set up a secure tunnel.

These instructions were tested on a 64-bit Amazon EC2 instance using the following Amazon Machine Image (AMI), which is based on Red Hat:

- Amazon Linux AMI 2014.09.2 (HVM) (ami-146e2a7c).

To launch an Amazon EC2 instance, which includes selecting an AMI, see Amazon Machine Images (AMIs).

To set up a secure tunnel to the Amazon SES US West (Oregon) endpoint using *stunnel*

1. Download and install the *stunnel* software. For information, go to http://www.stunnel.org.
2. If you are using Ubuntu Linux, stunnel may require a certificate. To generate the certificate, go to the */etc/stunnel* directory and at a command prompt, type the following:

```
sudo openssl req -new -out mail.pem -keyout mail.pem -nodes -x509 -days 365
```

3. Open or create a file called */etc/stunnel/stunnel.conf*.
4. To configure the secure tunnel, add the following lines to *stunnel.conf*. For the *accept* line, specify a port number that is outside the range of reserved ports and is not currently being used. For this example, we will use port 2525 for this purpose.

> **Important**
>
> These instructions assume that you want to use Amazon SES in the US West (Oregon) AWS region. If you want to use a different region, replace the instance of *email-smtp.us-west-2.amazonaws.com* in these instructions with the SMTP endpoint of the desired region. For a list of SMTP endpoints, see Regions and Amazon SES (p. 173).

```
[smtp-tls-wrapper]
accept = 2525
client = yes
connect = email-smtp.us-west-2.amazonaws.com:465
delay = yes
```

5. If you are using Ubuntu Linux, add this additional line to *stunnel.conf*:

   ```
   cert = /etc/stunnel/mail.pem
   ```

6. Save *stunnel.conf*.
7. At a command prompt, issue the following command to start stunnel:
   `sudo stunnel /etc/stunnel/stunnel.conf`
8. At a command prompt, type the following command to verify that the tunnel has been created. We are using port 2525 for this example; if you have specified a different port number, modify the command accordingly.
   `telnet localhost 2525`

Integrating Amazon SES with Postfix

Postfix was created as an alternative to the widely used Sendmail MTA. For information about Postfix, go to http://www.postfix.org.

These instructions were tested on a 64-bit Amazon EC2 instance using the following Amazon Machine Image (AMI), which is based on Red Hat:

- Amazon Linux AMI 2014.09.2 (HVM) (ami-146e2a7c).

To launch an Amazon EC2 instance, which includes selecting an AMI, see Amazon Machine Images (AMIs).

Prerequisites

Before you perform one of the following procedures, verify the following:

- You have uninstalled Sendmail (if you are not sure how to switch between Sendmail and Postfix).
- You have installed Postfix.
- You are able to successfully send an email using Postfix without Amazon SES.
- You have verified your "From" address and, if your account is still in the sandbox, you have also verified your "To" addresses. For more information, see Verifying Email Addresses in Amazon SES (p. 37).
- (Optional) If you are sending email through Amazon SES from an Amazon EC2 instance, you may need to assign an Elastic IP Address to your Amazon EC2 instance for the receiving ISP to accept your email. For more information, see Amazon EC2 Elastic IP Addresses.
- (Optional) If you are sending email through Amazon SES from an Amazon EC2 instance, you can fill out a Request to Remove Email Sending Limitations to remove the additional sending limit restrictions that are applied to port 25 by default.

To configure integration with the Amazon SES US West (Oregon) endpoint using STARTTLS

1. On your mail server, open the *main.cf* file. On many systems, this file resides in the */etc/postfix* folder.

   Important
   These instructions assume that you want to use Amazon SES in the US West (Oregon) AWS region. If you want to use a different region, replace all instances of *email-smtp.us-west-2.amazonaws.com* in these instructions with the SMTP endpoint of the desired region. For a list of SMTP endpoints, see Regions and Amazon SES (p. 173).

2. Add the following lines to the *main.cf* file.

```
relayhost = [email-smtp.us-west-2.amazonaws.com]:25
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_use_tls = yes
smtp_tls_security_level = encrypt
smtp_tls_note_starttls_offer = yes
```

   Save and close the *main.cf* file.

3. On your mail server, open the *master.cf* file. On many systems, this file resides in the */etc/postfix* folder.

4. Comment out the following line of the *master.cf* file by putting a # in front of it: `-o smtp_fallback_relay=`

   Save and close the *master.cf* file.

5. Edit the */etc/postfix/sasl_passwd* file. If the file does not exist, create it. Add the following lines to the file, replacing *USERNAME* and *PASSWORD* with your SMTP user name and password. If Postfix cannot authenticate with the Amazon SES SMTP endpoint because the hostname does not match, try adding the additional line specified in Amazon SES SMTP Issues (p. 189).

   Important
   Use your SMTP user name and password, not your AWS access key ID and secret access key. Your SMTP credentials and your AWS credentials are not the same. For information about how to obtain your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

```
[email-smtp.us-west-2.amazonaws.com]:25 USERNAME:PASSWORD
```

   Save and close the *sasl_passwd* file.

6. At a command prompt, issue the following command to create a hashmap database file containing your SMTP credentials.

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

7. (Optional but recommended) Remove the */etc/postfix/sasl_passwd* file.

8. (Optional but recommended) The */etc/postfix/sasl_passwd* and */etc/postfix/sasl_passwd.db* files you created in the previous steps are not encrypted. Because these files contain your SMTP credentials, it is a good idea to use the following commands to change the owner to root and set permissions to restrict access to the files as much as possible. (Note that if you deleted */etc/postfix/sasl_passwd* in the previous step, you should omit it from the commands below.)

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

sudo chmod 0600 /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
```

9. Tell Postfix where to find the CA certificate (needed to verify the Amazon SES server certificate). You could use a self-signed certificate or you could use default certificates as follows:

   If running on the Amazon Linux AMI:

   ```
   sudo postconf -e 'smtp_tls_CAfile = /etc/ssl/certs/ca-bundle.crt'
   ```

   If running on Ubuntu Linux:

   ```
   sudo postconf -e 'smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt'
   ```
10. When you have finished updating the configuration, stop and start Postfix by typing the following at the command line:

    ```
    sudo postfix stop
    ```

    ```
    sudo postfix start
    ```
11. Send a test email by typing the following at a command line, pressing Enter after each line. Note that you must replace *from@example.com* with your "From" email address, which you must have previously verified with Amazon SES. Replace *to@example.com* with your "To" address. If your account is still in the sandbox, the "To" address must also be verified. Also note that the final line is a single period.

    ```
    sendmail -f from@example.com to@example.com

    From: from@example.com

    Subject: Test

    This email was sent through Amazon SES!

    .
    ```
12. Check your inbox for the email. If the message was not delivered, check your Junk box, and then check your system's mail log (typically */var/log/maillog*) for errors. For example, you will get an "Email address not verified" error if you have not verified the "From" address that follows "-f" on the command line.

To configure integration using a secure tunnel

1. To begin, you will need to set up a secure tunnel as described in Setting Up a Secure Tunnel to Connect to Amazon SES (p. 59). In the following procedure, we use port 2525 as your *stunnel* port. If you are using a different port, modify the settings that you actually use accordingly.
2. On your mail server, open the *main.cf* file. On many systems, this file resides in the */etc/postfix* folder.
3. Add the following lines to the *main.cf* file.

   ```
   relayhost = 127.0.0.1:2525
   smtp_sasl_auth_enable = yes
   smtp_sasl_security_options = noanonymous
   smtp_tls_security_level = may
   smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
   ```

   Save and close the *main.cf* file.
4. On your mail server, open the *master.cf* file. On many systems, this file resides in the */etc/postfix* folder.
5. Comment out the following line of the *master.cf* file by putting a # in front of it: `-o smtp_fallback_relay=`

   Save and close the *master.cf* file.

6. Edit the */etc/postfix/sasl_passwd* file. If the file does not exist, create it. Add the following line to the file, replacing *USERNAME* and *PASSWORD* with your SMTP user name and password.

   Important
   Use your SMTP user name and password, not your AWS access key ID and secret access key. Your SMTP credentials and your AWS credentials are not the same. For information about how to obtain your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

   ```
   127.0.0.1:2525 USERNAME:PASSWORD
   ```

   Save the *sasl_passwd* file.
7. At a command prompt, issue the following command to create a hashmap database file containing your SMTP credentials.

   ```
   sudo postmap hash:/etc/postfix/sasl_passwd
   ```
8. (Optional but recommended) Remove the */etc/postfix/sasl_passwd* file.
9. (Optional but recommended) The */etc/postfix/sasl_passwd* and */etc/postfix/sasl_passwd.db* files you created in the previous steps are not encrypted. Because these files contain your SMTP credentials, it is a good idea to use the following commands to change the owner to root and set permissions to restrict access to the files as much as possible. (Note that if you deleted */etc/postfix/sasl_passwd* in the previous step, you should omit it from the commands below.)

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

   sudo chmod 0600 /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
   ```
10. When you have finished updating the configuration, stop and start Postfix by typing the following at the command line:

    ```
    sudo postfix stop

    sudo postfix start
    ```
11. Send a test email by typing the following at a command line, pressing Enter after each line. Note that you must replace *from@example.com* with your "From" email address, which you must have previously verified with Amazon SES. Replace *to@example.com* with your "To" address. If your account is still in the sandbox, the "To" address must also be verified. Also note that the final line is a single period.

    ```
    sendmail -f from@example.com to@example.com

    From: from@example.com

    Subject: Test

    This email was sent through Amazon SES!

    .
    ```
12. Check your inbox for the email. If the message was not delivered, check your Junk box, and then check your system's mail log (typically */var/log/maillog*) for errors. For example, you will get an "Email address not verified" error if you have not verified the "From" address that follows "-f" on the command line.

Integrating Amazon SES with Sendmail

Sendmail was released in the early 1980s, and has been continuously improved ever since. It is a very flexible and configurable MTA, and it has a large installed base. For information about Sendmail, go to http://www.sendmail.com/sm/open_source/.

The following instructions show you how to configure Sendmail to send email through Amazon SES using two ways to encrypt the connection: STARTTLS and a secure tunnel.

These instructions were tested on a 64-bit Amazon EC2 instance using the following Amazon Machine Image (AMI):

- Amazon Linux AMI 2012.09 (ami-1624987f), which runs Linux 3.2

To launch an Amazon EC2 instance, which includes selecting an AMI, see Amazon Machine Images (AMIs).

Prerequisites

Before you perform one of the following procedures, verify the following:

- The Sendmail package is installed on your computer, and you are able to successfully send an email using Sendmail without Amazon SES.

  Tip
  To see if a package is installed on a computer running Red Hat Linux, type `rpm -qa | grep <package>, where <package> is the package name`. To see if a package is installed on a computer running Ubuntu Linux, type `dpkg -s <package>`.
- In addition to the Sendmail package, the following packages are installed on your computer: sendmail-cf, m4, and cyrus-sasl-plain.
- You have verified your "From" address and, if your account is still in the sandbox, you have also verified your "To" addresses. For more information, see Verifying Email Addresses in Amazon SES (p. 37).
- (Optional) If you are sending email through Amazon SES from an Amazon EC2 instance, you may need to assign an Elastic IP Address to your Amazon EC2 instance for the receiving ISP to accept your email. For more information, see Amazon EC2 Elastic IP Addresses.
- (Optional) If you are sending email through Amazon SES from an Amazon EC2 instance, you can fill out a Request to Remove Email Sending Limitations to remove the additional sending limit restrictions that are applied to port 25 by default.

To configure Sendmail to send email through the Amazon SES endpoint in US West (Oregon) using STARTTLS

1. Open the */etc/mail/authinfo* file for editing. If the file does not exist, create it.

   Important
   These instructions assume that you want to use Amazon SES in the US West (Oregon) AWS region. If you want to use a different region, replace all instances of *email-smtp.us-west-2.amazonaws.com* in these instructions with the SMTP endpoint of the desired region. For a list of SMTP endpoints, see Regions and Amazon SES (p. 173).
2. Add the following line to */etc/mail/authinfo*, where:

   - U:root—Do not modify.
   - I:USERNAME—Replace with the Amazon SES username you obtained using the instructions in Obtaining Your Amazon SES SMTP Credentials (p. 50). This is NOT the same as your AWS Access Key ID.
   - P:PASSWORD—Replace with the Amazon SES password you obtained using the instructions in Obtaining Your Amazon SES SMTP Credentials (p. 50). This is NOT the same as your AWS Secret Key.
   - M:LOGIN—Replace with the method of authentication to use. For example, PLAIN, DIGEST-MD5, etc.

```
AuthInfo:email-smtp.us-west-2.amazonaws.com "U:root" "I:USERNAME" "P:PASSWORD"
 "M:LOGIN"
```

If Sendmail cannot authenticate with the Amazon SES SMTP endpoint because the hostname does not match, try adding the additional line specified in Amazon SES SMTP Issues (p. 189).

3. Save the *authinfo* file.
4. At a command prompt, type the following command to generate */etc/mail/authinfo.db*:

   `sudo makemap hash /etc/mail/authinfo.db < /etc/mail/authinfo`

5. Open the */etc/mail/access* file and include support for relaying to the Amazon SES SMTP endpoint by adding the following line. If Sendmail cannot authenticate with the Amazon SES SMTP endpoint because the hostname does not match, try adding the additional line specified in Amazon SES SMTP Issues (p. 189).

   ```
   Connect:email-smtp.us-west-2.amazonaws.com RELAY
   ```

   Save the file.

6. At a command prompt, type the following command to regenerate */etc/mail/access.db*:

   `sudo makemap hash /etc/mail/access.db < /etc/mail/access`

7. Save a back-up copy of */etc/mail/sendmail.mc* and */etc/mail/sendmail.cf*.
8. Add the following group of lines to the */etc/mail/sendmail.mc* file before any MAILER() definitions. If you add a FEATURE() line after a MAILER() definition, when you run `m4` in a subsequent step, you will get the following error: "`ERROR: FEATURE() should be before MAILER().`":

   > **Important**
   > If you are using an AWS region other than US West (Oregon), replace the `SMART_HOST` value with the Amazon SES SMTP endpoint of the region you're using, and be sure to use the ` character and the apostrophe exactly as shown.

   ```

   define(`SMART_HOST', `email-smtp.us-west-2.amazonaws.com')dnl
   define(`RELAY_MAILER_ARGS', `TCP $h 25')dnl
   define(`confAUTH_MECHANISMS', `LOGIN PLAIN')dnl
   FEATURE(`authinfo', `hash -o /etc/mail/authinfo.db')dnl
   MASQUERADE_AS(`YOUR_DOMAIN')dnl
   FEATURE(masquerade_envelope)dnl
   FEATURE(masquerade_entire_domain)dnl
   ```

9. In the text you just added to *sendmail.mc*, in the line that starts with `MASQUERADE_AS`, replace *YOUR_DOMAIN* with the domain name from which you are sending your email. By adding this masquerade, you are making email from this host appear to be sent from your domain. Otherwise, the email will appear as if the email is being sent from the host name of the mail server, and you may get an "Email address not verified" error when you try to send an email.
10. Save the *sendmail.mc* file.
11. At a command prompt, type the following command to make *sendmail.cf* writeable:

    `sudo chmod 666 /etc/mail/sendmail.cf`

12. At a command prompt, type the following command to regenerate *sendmail.cf*:

    `sudo m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf`

> **Note**
>
> If you encounter errors such as "Command not found" and "No such file or directory," make sure you have installed the m4 and sendmail-cf packages as specified in the prerequisites section above.

13. At a command prompt, type the following command to reset the permissions of *sendmail.cf* to read only:

    ```
    sudo chmod 644 /etc/mail/sendmail.cf
    ```

14. At a command prompt, type the following command to restart Sendmail:

    ```
    sudo /etc/init.d/sendmail restart
    ```

15. Send a test email by doing the following:

    1. At a command prompt, type the following. Note that you should replace *from@example.com* with your "From" email address, which you must have verified with Amazon SES. Replace *to@example.com* with your "To" address. If your account is still in the sandbox, the "To" address must also be verified.

       ```
       sudo /usr/sbin/sendmail -f from@example.com to@example.com
       ```

    2. Press <Enter>. Type the body of the message, pressing <Enter> after each line.
    3. When you are finished typing the email, press CTRL+D to send the email.

16. Check the recipient email's client for the email. If you cannot find the email, check the Junk box in the recipient's email client. If you still cannot find the email, look at the Sendmail log on the mail server. The log is typically in */var/spool/mail/<user>*.

To configure Sendmail to send email through Amazon SES using a secure tunnel

1. To begin, you will need to set up a secure tunnel as described in Setting Up a Secure Tunnel to Connect to Amazon SES (p. 59). In the following procedure, we use port 2525 as your *stunnel* port. If you are using a different port, modify the settings that you actually use accordingly.
2. Open the */etc/mail/authinfo* file for editing. If the file does not exist, create it.
3. Add the following lines to */etc/mail/authinfo*, where:

   - U:root—Do not modify.
   - I:USERNAME—Replace with the Amazon SES username you obtained using the instructions in Obtaining Your Amazon SES SMTP Credentials (p. 50). This is NOT the same as your AWS Access Key ID.
   - P:PASSWORD—Replace with the Amazon SES password you obtained using the instructions in Obtaining Your Amazon SES SMTP Credentials (p. 50). This is NOT the same as your AWS Secret Key.
   - M:LOGIN—Replace with the method of authentication to use. For example, PLAIN, DIGEST-MD5, etc.

   ```
   AuthInfo:127.0.0.1 "U:root" "I:USERNAME" "P:PASSWORD" "M:LOGIN"
   ```

4. Save the *authinfo* file.
5. At a command prompt, type the following command:

   ```
   sudo makemap hash /etc/mail/authinfo.db < /etc/mail/authinfo
   ```

6. Open the */etc/mail/access* file to ensure that relaying is allowed for 127.0.0.1. This is the default behavior. If relaying is not allowed for localhost, open your */etc/hosts* file and add another hostname pointing to 127.0.0.1.
7. If you modified */etc/mail/access* in the last step, at a command prompt, type the following command to regenerate */etc/mail/access.db*:
   ```
   sudo makemap hash /etc/mail/access.db < /etc/mail/access
   ```
8. Open the */etc/mail/sendmail.mc* file and add the following group of lines before any MAILER() definitions. If you add a FEATURE() line after a MAILER() definition, when you run `m4` in a subsequent step, you will get the following error: `"ERROR: FEATURE() should be before MAILER()."`:

   Important

   Be sure to use the ` character and the apostrophe exactly as shown.

   ```
   FEATURE(`authinfo', `hash -o /etc/mail/authinfo.db')dnl
   define(`SMART_HOST', `[127.0.0.1]')dnl
   define(`RELAY_MAILER_ARGS', `TCP $h 2525')dnl
   define(`ESMTP_MAILER_ARGS', `TCP $h 2525')dnl
   MASQUERADE_AS(`YOUR_DOMAIN')dnl
   FEATURE(masquerade_envelope)dnl
   FEATURE(masquerade_entire_domain)dnl
   ```
9. In the text you just added to *sendmail.mc*, in the line that starts with `MASQUERADE_AS`, replace *YOUR_DOMAIN* with the domain name from which you are sending your email. By adding this masquerade, you are making email from this host appear to be sent from your domain. Otherwise, the email will appear as if the email is being sent from the host name of the mail server, and you may get an "Email address not verified" error when you try to send an email.

   Also, if you found in Step 5 that relaying was not allowed for 127.0.0.1, change the `SMART_HOST' line you added to *sendmail.mc* to use the hostname that you entered in the */etc/hosts* file. That is:

   ```
   define(`SMART_HOST', `hostname')dnl
   ```
10. Save and close the *sendmail.mc* file.
11. At a command prompt, type the following command to make *sendmail.cf* writeable:

    ```
    sudo chmod 666 /etc/mail/sendmail.cf
    ```
12. At a command prompt, type the following command to regenerate *sendmail.cf*:

    ```
    sudo m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
    ```

    Note

    If you encounter errors such as "Command not found" and "No such file or directory," make sure you have installed the m4 and sendmail-cf packages as specified in the prerequisites section above.
13. At a command prompt, type the following command to reset the permissions of *sendmail.cf* to read only:

    ```
    sudo chmod 644 /etc/mail/sendmail.cf
    ```
14. At a command prompt, type the following command to restart Sendmail:

    ```
    sudo /etc/init.d/sendmail restart
    ```
15. Send a test email by doing the following:

    1. At a command prompt, type the following. Note that you should replace *from@example.com* with your "From" email address, which you must have verified with Amazon SES. Replace

to@example.com with your "To" address. If your account is still in the sandbox, the "To" address must also be verified.

```
sudo /usr/sbin/sendmail -f from@example.com to@example.com
```

2. Press <Enter>. Type the body of the message, pressing <Enter> after each line.
3. When you are finished typing the email, press CTRL+D to send the email.

16. Check the recipient email's client for the email. If you cannot find the email, check the Junk box in the recipient's email client. If you still cannot find the email, look at the Sendmail log on the email sending computer. The log is typically in */var/spool/mail/<user>*.

Integrating Amazon SES with Microsoft Exchange

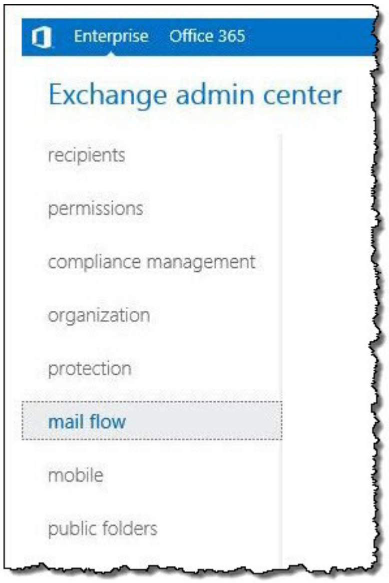
You can configure Microsoft Exchange to send email through Amazon SES. The following procedures show you how to integrate Microsoft Exchange with Amazon SES using the Microsoft Exchange GUI or Windows PowerShell.

Important
Follow only one of the following procedures (Microsoft Exchange GUI or Windows PowerShell). If you follow both procedures, you will get an error stating that you have two send connectors with the same name.

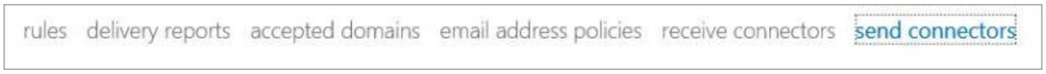
These instructions were written using Microsoft Exchange 2013.

To integrate Microsoft Exchange with Amazon SES using the Microsoft Exchange GUI

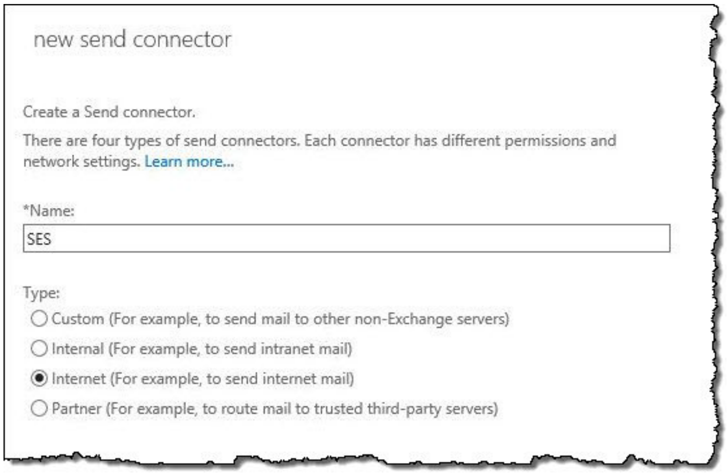
1. Go to the Microsoft Exchange admin center (typically *https://<CASServerName>/ecp*) and sign in as a user who is part of the Exchange administrators group.
2. From the left menu, click **mail flow**.



3. Click **send connectors**.



4. Click the plus sign.
5. Enter a name for the send connector (for example, SES).
6. Under **Type**, select **Internet**.



new send connector

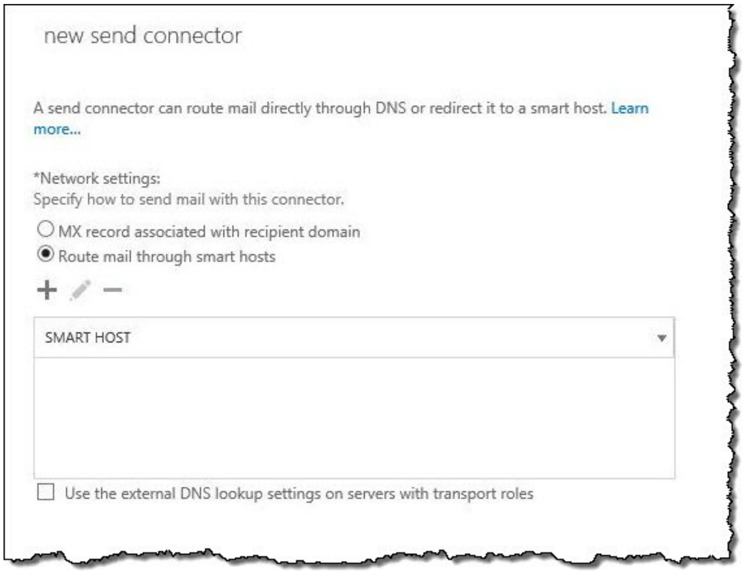
Create a Send connector.

There are four types of send connectors. Each connector has different permissions and network settings. Learn more...

*Name:
SES

Type:
○ Custom (For example, to send mail to other non-Exchange servers)
○ Internal (For example, to send intranet mail)
● Internet (For example, to send internet mail)
○ Partner (For example, to route mail to trusted third-party servers)

7. Click **Next**.
8. Select **Route mail through smart hosts**.



new send connector

A send connector can route mail directly through DNS or redirect it to a smart host. Learn more...

*Network settings:
Specify how to send mail with this connector.
○ MX record associated with recipient domain
● Route mail through smart hosts

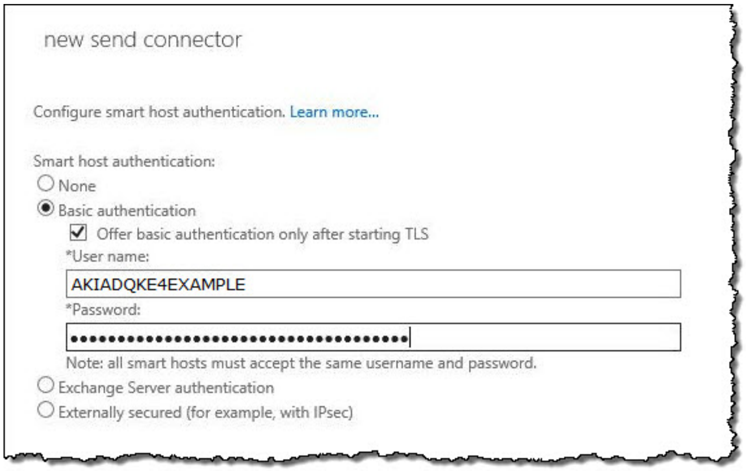
SMART HOST

□ Use the external DNS lookup settings on servers with transport roles

9. Click the plus sign and then enter the Amazon SES endpoint that you will use (for example, *email-smtp.us-west-2.amazonaws.com*). For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 173).
10. Click **Save**. The endpoint you entered will appear in the **SMART HOST** box.
11. Click **Next**.
12. Select **Basic authentication**, then select **Offer basic authentication only after starting TLS**, and then enter your Amazon SES SMTP user name and password.

Important

Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 170).



new send connector

Configure smart host authentication. Learn more...

Smart host authentication:

○ None

● Basic authentication

☑ Offer basic authentication only after starting TLS

*User name:

AKIADQKE4EXAMPLE

*Password:

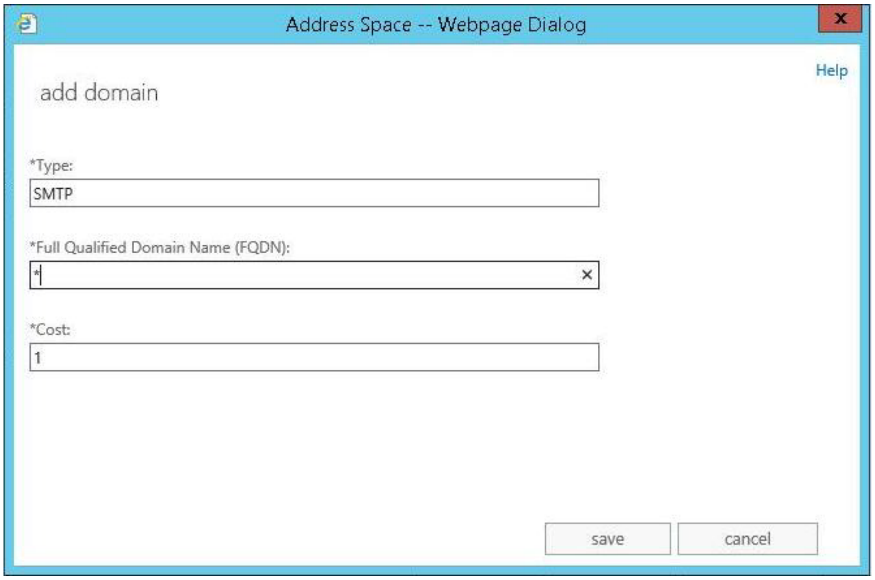
••••••••••••••••••••••••••••••••••••

Note: all smart hosts must accept the same username and password.

○ Exchange Server authentication

○ Externally secured (for example, with IPsec)

13. Click **Next**.
14. Click the plus sign.
15. Verify that **Type** is `SMTP`, **FQDN** is `*`, and **Cost** is `1`.



add domain

*Type:

SMTP

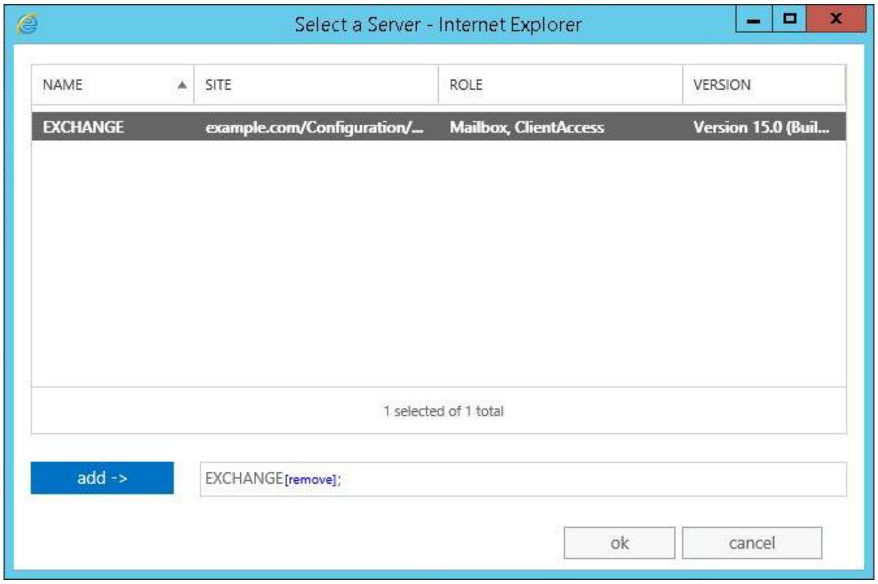
*Full Qualified Domain Name (FQDN):

*

*Cost:

1

16. Click **Save** and then click **Next**.
17. Click the plus sign.
18. Select all transport servers you would like to apply this rule to and click **Add**. When you have added all the servers you want to send email through Amazon SES, click **ok**.



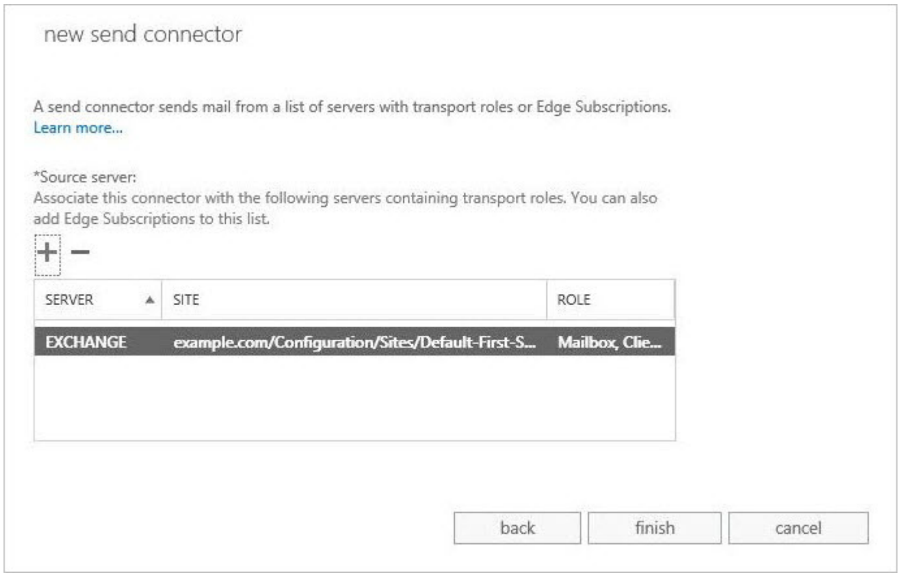
Select a Server - Internet Explorer

NAME	SITE	ROLE	VERSION
EXCHANGE	example.com/Configuration/...	Mailbox, ClientAccess	Version 15.0 (Buil...

1 selected of 1 total

EXCHANGE[remove];

19. Verify that the servers are added and then click **finish**.



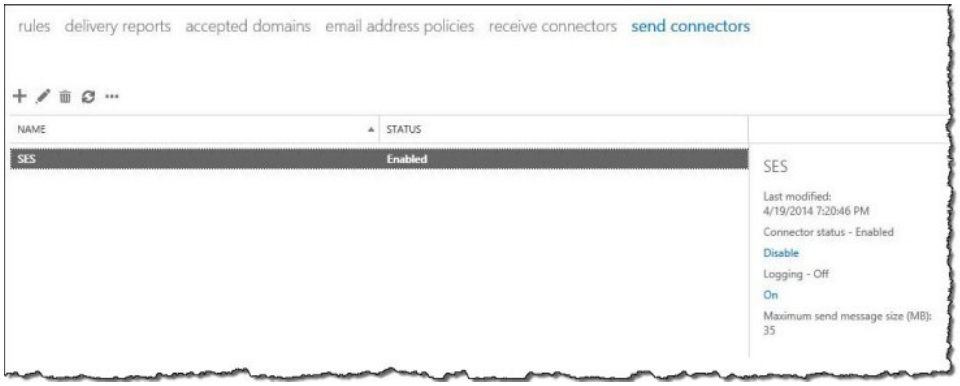
new send connector

A send connector sends mail from a list of servers with transport roles or Edge Subscriptions.
Learn more...

*Source server:
Associate this connector with the following servers containing transport roles. You can also add Edge Subscriptions to this list.

SERVER	SITE	ROLE
EXCHANGE	example.com/Configuration/Sites/Default-First-S...	Mailbox, Clie...

You should now see a send connector for Amazon SES with an enabled status. All outbound mail will now flow through Amazon SES.



NAME	STATUS
SES	Enabled

SES

Last modified:
4/19/2014 7:20:46 PM

Connector status - Enabled

Disable

Logging - Off

On

Maximum send message size (MB):
35

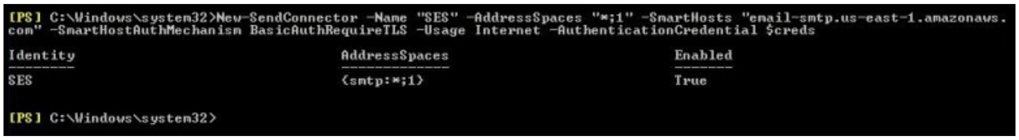
To integrate Microsoft Exchange with Amazon SES using Windows PowerShell

1. Open the Exchange Management Shell and type `$creds = Get-Credential`. A Windows PowerShell Credential Request dialog box will appear.
2. In the dialog box, enter your Amazon SES SMTP user name and password and then click **OK**.

   Important
   Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 170).
3. At the command prompt, type the following line, replacing ENDPOINT with an Amazon SES SMTP endpoint (for example, *email-smtp.us-west-2.amazonaws.com*). For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 173).

   ```
   New-SendConnector -Name "SES" -AddressSpaces "*;1" -SmartHosts "ENDPOINT"
   -SmartHostAuthMechanism BasicAuthRequireTLS -Usage Internet
   -AuthenticationCredential $creds
   ```

   The command line should now display a send connector for Amazon SES with an enabled status. All outbound mail will now flow through Amazon SES.



```
[PS] C:\Windows\system32>New-SendConnector -Name "SES" -AddressSpaces "*;1" -SmartHosts "email-smtp.us-east-1.amazonaws.
com" -SmartHostAuthMechanism BasicAuthRequireTLS -Usage Internet -AuthenticationCredential $creds

Identity                          AddressSpaces                     Enabled
--------                          -------------                     -------
SES                               {smtp:*;1}                        True

[PS] C:\Windows\system32>
```

Integrating Amazon SES with Microsoft Windows Server IIS SMTP

You can configure Microsoft Windows Server's IIS SMTP server to send email through Amazon SES. These instructions were written using Microsoft Windows Server 2012 on an Amazon EC2 instance. You can use the same configuration on Microsoft Windows Server 2008 and Microsoft Windows Server 2008 R2.

To integrate the Microsoft Windows Server IIS SMTP server with Amazon SES

1. First, set up Microsoft Windows Server 2012 using the following instructions.

a. From the Amazon EC2 management console, launch a new Microsoft Windows Server 2012 Base Amazon EC2 instance.

b. Connect to the instance and log into it using Remote Desktop by following the instructions in Getting Started with Amazon EC2 Windows Instances.

c. Launch the Server Manager Dashboard.

d. Install the **Web Server** role. Be sure to include the **IIS 6 Management Compatibility tools** (an option under the **Web Server** checkbox).

e. Install the **SMTP Server** feature.

2. Next, configure the IIS SMTP service using the following instructions.

a. Return to the Server Manager Dashboard.

b. From the **Tools** menu, click **Internet Information Services (IIS) 6.0 Manager**.

c. Right-click **SMTP Virtual Server #1** and then select **Properties**.

d. On the **Access** tab, under **Relay Restrictions**, click **Relay**.

e. In the **Relay Restrictions** dialog box, click **Add**.

f. Under **Single Computer**, enter **127.0.0.1** for the IP address. You have now granted access for this server to relay email to Amazon SES through the IIS SMTP service.

> **Note**
> In this procedure, we assume that your emails are generated on this server. If the application that generates the email runs on a separate server, you need to grant relaying access for that server in IIS SMTP.

3. Finally, configure the server to send email through Amazon SES using the following instructions.

a. Return to the **SMTP Virtual Server #1 Properties** dialog box and then click the **Delivery** tab.

b. On the **Delivery** tab, click **Outbound Security**.

c. Select **Basic Authentication** and then enter your Amazon SES SMTP username and password. You can obtain these credentials from the Amazon SES console using the procedure in Obtaining Your Amazon SES SMTP Credentials (p. 50).

> **Important**
> Your SMTP user name and password are not the same as your AWS access key ID and secret access key. Do not attempt to use your AWS credentials to authenticate yourself against the SMTP endpoint. For more information about credentials, see Using Credentials With Amazon SES (p. 170).

d. Ensure that **TLS encryption** is selected.

e. Return to the **Delivery** tab.

f. Click **Outbound Connections**.

g. In the **Outbound Connections** dialog box, ensure that the port is 25 or 587.

h. Click **Advanced**.

i. For the **Smart host** name, enter the Amazon SES endpoint that you will use (for example, *email-smtp.us-west-2.amazonaws.com*). For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 173).

j. Return to the Server Manager Dashboard.

k. On the Server Manager Dashboard, right-click **SMTP Virtual Server #1** and then restart the service to pick up the new configuration.

l. Send an email through this server. You can examine the message headers to confirm that it was delivered through Amazon SES.

Integrating Amazon SES with Exim

Exim is an MTA that was originally developed for Unix-like systems. It is a general purpose mail program that is very flexible and configurable.

To learn more about Exim, go to http://www.exim.org.

To configure integration with the Amazon SES US West (Oregon) endpoint using STARTTLS

1. Open the */etc/exim/exim.conf* file for editing. If the file does not exist, create it.

   Important
   These instructions assume that you want to use Amazon SES in the US West (Oregon) AWS region. If you want to use a different region, replace all instances of *email-smtp.us-west-2.amazonaws.com* in these instructions with the SMTP endpoint of the desired region. For a list of SMTP endpoints, see Regions and Amazon SES (p. 173).

2. In */etc/exim/exim.conf*, make the following changes:

   a. In the *routers* section, after the *begin routers* line, add the following:

   ```
   send_via_ses:
   driver = manualroute
   domains = ! +local_domains
   transport = ses_smtp
   route_list = * email-smtp.us-west-2.amazonaws.com;
   ```

   b. In the *transports* section, after the *begin transports* line, add the following:

   ```
   ses_smtp:
   driver = smtp
   port = 25
   hosts_require_auth = $host_address
   hosts_require_tls = $host_address
   ```

   c. In the *authenticators* section, after the *begin authenticators* line, add the following, replacing USERNAME and PASSWORD with your SMTP user name and password:

      Important
      Use your SMTP user name and password, not your AWS access key ID and secret access key. Your SMTP credentials and your AWS credentials are not the same. For information about how to obtain your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

   ```
   ses_login:
   driver = plaintext
   public_name = LOGIN
   client_send = : USERNAME : PASSWORD
   ```

3. Save the */etc/exim/exim.conf* file.

To configure integration using a secure tunnel

1. To begin, you will need to set up a secure tunnel as described in Secure Tunnel (p. 54). In the following procedure, we use port 2525 as your *stunnel* port. If you are using a different port, modify the settings that you actually use accordingly.
2. Open the */etc/exim/exim.conf* file for editing. If the file does not exist, create it.

   Important
   These instructions assume that you want to use Amazon SES in the US West (Oregon) AWS region. If you want to use a different region, replace all instances of *email-smtp.us-west-2.amazonaws.com* in these instructions with the SMTP endpoint of the desired region. For a list of SMTP endpoints, see Regions and Amazon SES (p. 173).
3. In */etc/exim/exim.conf*, make the following changes:

   a. In the *routers* section, after the *begin routers* line, add the following:

   ```
   send_via_ses:
   driver = manualroute
   domains = ! +local_domains
   transport = ses_smtp
   self = send
   route_list = * localhost
   ```

   b. In the *transports* section, after the *begin transports* line, add the following:

   ```
   ses_smtp:
   driver = smtp
   port = 2525
   hosts_require_auth = localhost
   hosts_avoid_tls = localhost
   ```

   c. In the *authenticators* section, after the *begin authenticators* line, add the following, replacing USERNAME and PASSWORD with your SMTP user name and password:

   ```
   ses_login:
   driver = plaintext
   public_name = LOGIN
   client_send = : USERNAME : PASSWORD
   ```

4. Save the */etc/exim/exim.conf* file.

When you have finished updating the configuration, restart Exim. At the command line, type the following command and press ENTER.

```
sudo /etc/init.d/exim restart
```

Note
This command may not be exactly the same on your particular server.

When you have completed this procedure, your outgoing email will be sent via Amazon SES. To test your configuration, send an email message through your Exim server, and then verify that arrives at its destination. If the message is not delivered, then check your system's mail log for errors. On many systems, this is the /var/log/exim/main.log file.

Using the Command Line to Send Email Through the Amazon SES SMTP Interface

You can use a command line utility to interact directly with the Amazon SES SMTP interface. The command line interface can be helpful for testing purposes or for writing software that must communicate directly using the SMTP protocol.

To protect our customers, all communication with the SMTP interface must take place using TLS (Transport Layer Security). For SMTP command line usage, we recommend that you use OpenSSL. OpenSSL, which is available at http://www.openssl.org, includes a command line utility for communicating over a TLS-secured connection.

Example : Using OpenSSL to Send Email Using Amazon SES

This example shows how to connect to the Amazon SES SMTP endpoint in the US West (Oregon) region and use standard SMTP commands to send an email message. Some of the output in the example is omitted for brevity.

Using TLS Wrapper:

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

- `s_client`—Specifies that this connection will use TLS (SSL).
- `-crlf`—Translates line feed characters (LF) to CR+LF (carriage return and line feed).
- `-quiet`—Inhibits printing of session and certificate information. This implicitly turns on -ign_eof as well.
- `-connect`—Specifies the SMTP host and port.

Using STARTTLS:

```
openssl s_client -crlf -quiet -starttls smtp -connect email-smtp.us-west-
2.amazonaws.com:25
```

- `s_client`—Specifies that this connection will use TLS (SSL).
- `-crlf`—Translates line feed characters (LF) to CR+LF (carriage return and line feed).
- `-quiet`—Inhibits printing of session and certificate information. This implicitly turns on -ign_eof as well.
- `-starttls smtp`—Specifies STARTTLS negotiation.
- `-connect`—specifies the SMTP host and port.

After you make the connection using one of the preceding commands, the Amazon SES SMTP interface identifies itself by presenting its server certificate.

```


CONNECTED(00000003) ... <output omitted> Server certificate -----BEGIN
CERTIFICATE-----
MIID2jCCAue4gAwIBAgIAMEkqjWRxm3cqMA0tGC2GxSI37DQEBQ6UAMIGHjswCQD
VQQEwVUzErTMBEGaxA51UECBfMKV2Fza7GluZ3RvbjEMxA4GAUEByEXAMPLECERT
... <output omitted>
--- 220 localhost ESMTP SES 2010-12-03
```

At this point, use the `EHLO` command to identify your client. Specify the hostname of the system from which you are logging in.

```
EHLO bob-desktop.example.com

250-localhost
... <output omitted>
250-AUTH LOGIN
250 Ok
```

You can now use the `AUTH LOGIN` command to supply your SMTP credentials. You must base64-encode your SMTP username and password. The server prompts ("Username:" and "Password:") are similarly encoded, and appear with the SMTP response code 334.

Note

To base64-encode a string in Linux, you can use the following command (replace "SMTP-USERNAME" with your SMTP username): `echo -n "SMTP-USERNAME" | base64`

For example, if your SMTP username credential is *c2VzLXNtdHAtdXNlcEXAMPLE* and your password is *SkFYTVpaM3k0U2paVEYwOFpLEXAMPLE*, you would supply your base64-encoded credentials as follows:

```
AUTH LOGIN

334 VXNlcm5hbWU6
YzJWekxYTnRkSEF0ZFhObGNFWEFNUExF
334 UGFzc3dvcmQ6
U2tGWVRWcGFNM2swVTJwYVZFWXdPRnBMRVhBTVBMRQ==
235 Authentication successful.
```

Specify the sender and recipient by using the `MAIL FROM` and `RCPT TO` commands. For `MAIL FROM`, you must use an email address that you have already verified with Amazon SES. For more information about verification, see Verifying Email Addresses and Domains in Amazon SES (p. 37).

```
MAIL FROM:bob@example.com

250 Ok

RCPT TO:alice@example.com
250 Ok
```

Issue the `DATA` command to specify the email headers and the body of the message. The headers and the body must be separated by at least one blank line. In this example, only the *Subject:* header is being used. A dot (".") on a line by itself signifies the end of the message body.

```
DATA

354 End data with <CR><LF>.<CR><LF>
Subject:Hello from Amazon SES!

This email was sent using the Amazon SES SMTP interface.
.
250 Ok
```

Now that the message has been sent, use the QUIT command to close the SMTP connection.

```
QUIT

221 Bye
closed
```

Note

For more information about SMTP, go to http://tools.ietf.org/html/rfc5321.

Using the Amazon SES API to Send Email

To send production email through Amazon SES, you can use the Simple Mail Transfer Protocol (SMTP) interface or the Amazon SES API. For more information about the SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 48). This section describes how to send email by using the API.

The Amazon SES API has a Query (HTTPS) interface. See Regions and Amazon SES (p. 173) for a list of Amazon SES API endpoints. When you send an email by using the API, you can provide limited information and have Amazon SES assemble the email for you, or you can assemble the email yourself so that you have complete control over the content and formatting. For more information about the API, see the Amazon Simple Email Service API Reference. You can call the API in the following four ways:

- **Make raw Query requests and responses—**This is the most advanced method, because you are calling the API directly. For information about how to make Query requests and responses, see Using the Amazon SES Query Interface (p. 84).
- **Use an AWS SDK—**AWS SDKs wrap the low-level functionality of the Amazon SES API with higher-level data types and function calls that take care of the details for you, and provide basic functions (not included in the Amazon SES API), such as request authentication, request retries, and error handling. AWS SDKs and resources are available for Android, iOS, Java, .NET, Node.js, PHP, Python, and Ruby.
- **Use a command line interface—**The AWS Command Line Interface is the command line tool for Amazon SES. For documentation on the prior Amazon SES Perl scripts, see Appendix: Amazon SES Scripts (p. 214). We also offer the AWS Tools for Windows PowerShell for those who script in the PowerShell environment.

Regardless of whether you access the Amazon SES API directly or indirectly through an AWS SDK, the AWS Command Line Interface or the AWS Tools for Windows PowerShell, the Amazon SES API provides two different ways for you to send an email, depending on how much control you want over the composition of the email message:

- **Formatted—**Amazon SES composes and sends a properly formatted email message. You need only supply "From:" and "To:" addresses, a subject, and a message body. Amazon SES takes care of all the rest. For more information, see Sending Formatted Email Using the Amazon SES API (p. 80).
- **Raw—**You manually compose and send an email message, specifying your own email headers and MIME types. If you are experienced in formatting your own email, the raw interface gives you more control over the composition of your message. For more information, see Sending Raw Email Using the Amazon SES API (p. 80).

Sending Formatted Email Using the Amazon SES API

You can send a formatted email by using the AWS Management Console or by calling the Amazon SES API through an application directly, or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell.

The Amazon SES API provides the `SendEmail` action, which lets you compose and send a formatted email. `SendEmail` requires a From: address, To: address, message subject, and message body—text, HTML, or both. For a complete description of `SendEmail`, go to the Amazon Simple Email Service API Reference.

> **Note**
> When you use `SendEmail`, the email address string must be 7-bit ASCII. If you want it to contain any other characters, then you must use MIME encoded-word syntax (RFC 2047) instead of a literal string. To use MIME-encoded word syntax, you need to use `SendRawEmail` as described in Using MIME (p. 80). MIME encoded-word syntax uses the following form: *=?charset?encoding?encoded-text?=*. For more information, see RFC 2047.

For an example of how to compose a formatted message using the AWS SDK for Java or the AWS SDK for .NET, see Sending an Email Through Amazon SES Using AWS SDK for Java (p. 32) or Sending an Email Through Amazon SES Using AWS SDK for .NET (p. 29), respectively.

For tips on how to increase your email sending speed when you make multiple calls to `SendEmail`, see Increasing Throughput with Amazon SES (p. 188).

Sending Raw Email Using the Amazon SES API

Sometimes you might want more control over how Amazon SES composes and sends email than automatic formatting provides. If so, you can use the Amazon SES raw email interface to specify email headers and MIME types, to send highly customized messages to your recipients.

This section introduces you to some common email standards and how Amazon SES uses them. It also shows how to construct and send raw email from the command line and from the Amazon SES API.

About Email Header Fields

Simple Mail Transfer Protocol (SMTP) specifies how email messages are to be sent by defining the mail envelope and some of its parameters, but it does not concern itself with the content of the message. Instead, the Internet Message Format (RFC 5322) defines how the message is to be constructed.

With the Internet Message Format specification, every email message consists of a header and a body. The header consists of message metadata, and the body contains the message itself. For more information about email headers and bodies, see Email Format and Amazon SES (p. 11).

Using MIME

The SMTP protocol is designed for sending email messages composed of 7-bit ASCII characters. While this works well for many use cases, it is insufficient for non-ASCII text encodings (such as Unicode), binary content, or attachments. The Multipurpose Internet Mail Extensions standard (MIME) was developed to overcome these limitations, making it possible to send many other kinds of content using SMTP.

The MIME standard works by breaking the message body into multiple parts and then specifying what is to be done with each part. For example, one part of an email message body might be plain text, while another might be an image. In addition, MIME allows email messages to contain one or more attachments.

Message recipients can view the attachments from within their email clients, or they can save the attachments.

The message header and content are separated by a blank line. Each part of the email is separated by a boundary, a string of characters that denotes the beginning and ending of each part.

Here is an example of the raw text of a multipart MIME email message:

```
            Received: from smtp-out.example.com (123.45.67.89) by
            in.example.com (87.65.43.210); Wed, 2 Mar 2011 11:39:39 -0800
            From: "Bob" <bob@example.com>
            To: "Andrew" <andrew@example.com>
            Date: Wed, 2 Mar 2011 11:39:34 -0800
            Subject: Customer service contact info
            Message-ID: <97DCB304-C529-4779-BEBC-FC8357FCC4D2@example.com>
            Accept-Language: en-US
            Content-Language: en-US
            Content-Type: multipart/mixed;
             boundary="_003_97DCB304C5294779BEBCFC8357FCC4D2"
            MIME-Version: 1.0

            --_003_97DCB304C5294779BEBCFC8357FCC4D2
            Content-Type: text/plain; charset="us-ascii"
            Content-Transfer-Encoding: quoted-printable

          Hi Andrew.  Here are the customer service names and telephone numbers
 I promised you.

            See attached.

            -Bob

            --_003_97DCB304C5294779BEBCFC8357FCC4D2
            Content-Type: text/plain; name="cust-serv.txt"
            Content-Description: cust-serv.txt
          Content-Disposition: attachment; filename="cust-serv.txt"; size=1180;

             creation-date="Wed, 02 Mar 2011 11:39:39 GMT";
             modification-date="Wed, 02 Mar 2011 11:39:39 GMT"
            Content-Transfer-Encoding: base64

            TWFyeSBEYXZpcyAtICgzMjEpIDU1NS03NDY1DQpDYXJsIFRob21hcyAtICg
zMjEpIDU1NS01MjM1
            DQpTYW0gRmFycmlzIC0gKDMyMSkgNTU1LTIxMzQ=

            --_003_97DCB304C5294779BEBCFC8357FCC4D2
```

Note the following aspects of this example:

- A blank line separates the header from the body.
- The content type is "multipart/mixed," which indicates that the message has many parts and the receiver must handle each part separately.
- The "boundary" parameter specifies where each part begins and ends. In this case, the boundary is a unique string of characters that the sender's email client generates.

- There are two parts to the body, a plain text message and an attachment. The email client will display the plain text part, and it will handle the attachment separately.
- The "Content-Disposition" field specifies how the client should handle the attachment: When the reader clicks the attachment, the email client will attempt to save it to a text file named "cust-serv.txt".

MIME Encoding

Because of the 7-bit ASCII restriction of SMTP, any content containing 8-bit characters must first be converted to 7-bit ASCII before sending. MIME defines a *Content-Transfer-Encoding* header field for this purpose.

By convention, the most common encoding scheme is base64, where 8-bit binary content is encoded using a well-defined set of 7-bit ASCII characters. Upon receipt, the email client inspects the Content-Transfer-Encoding header field, and can immediately perform a base64 decode operation on the content, thus returning it to its original form. With most email clients, the encoding and decoding occur automatically, and the user need not be aware of it.

In the example above, the "cust-serv.txt" attachment must be decoded from base64 format in order to be read. Some email clients will encode all MIME parts in base64 format, even if they were originally in plain text. This is not normally an issue, since email clients perform the encoding and decoding automatically.

> **Note**
> For a list of MIME types that Amazon SES accepts, see Appendix: Unsupported Attachment Types (p. 214).

If you want the email address string to contain characters other than 7-bit ASCII, then you must use MIME encoded-word syntax (RFC 2047) instead of a literal string. MIME encoded-word syntax uses the following form: *=?charset?encoding?encoded-text?=*. For more information, see RFC 2047.

API

The Amazon SES API provides the `SendRawEmail` action, which lets you compose and send an email message in the format that you specify. For a complete description of `SendRawEmail`, go to the Amazon Simple Email Service API Reference.

> **Note**
> For tips on how to increase your email sending speed when you make multiple calls to `SendRawEmail`, see Increasing Throughput with Amazon SES (p. 188).

The message body must contain a properly formatted, raw email message, with appropriate header fields and message body encoding. Although it is possible to construct the raw message manually within an application, it is much easier to do so using existing mail libraries.

Example

The following code sample shows how to use the JavaMail library to compose an email message:

```
import java.io.IOException;
import java.io.PrintStream;
import java.util.Properties;

import javax.mail.*;
import javax.mail.internet.*;


public class ComposeEmail {

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

        // JavaMail representation of the message
        Session s = Session.getInstance(new Properties(), null);
        MimeMessage msg = new MimeMessage(s);

        // Sender and recipient
        msg.setFrom(new InternetAddress("bob@example.com"));
        msg.setRecipient( Message.RecipientType.TO, new InternetAddress("an
drew@example.com"));

        // Subject
        msg.setSubject("MIME email test");

        // Add a MIME part to the message
        MimeMultipart mp = new MimeMultipart();

        BodyPart part = new MimeBodyPart();
        String myText = "<h1>Hello, this is a test</h1>";
        part.setContent(myText, "text/html");
        mp.addBodyPart(part);

         msg.setContent(mp);

         // Print the raw email content on the console
         PrintStream out = System.out;
         msg.writeTo(out);
     }
}
```

The program produces the following output:

```


            From: bob@example.com
            To: andrew@example.com
            Message-ID: <697745840.1.1292371199964.JavaMail.user@0023df9374dc>

            Subject: MIME email test
            MIME-Version: 1.0
            Content-Type: multipart/mixed;
             boundary="----=_Part_0_405642246.1292371199511"
            ------=_Part_0_405642246.1292371199511
            Content-Type: text/html; charset=us-ascii
            Content-Transfer-Encoding: 7bit

            <h1>Hello, this is a test</h1>
            ------=_Part_0_405642246.1292371199511--
```

You would pass that raw email content to the `SendRawEmail` action.

Using the Amazon SES Query Interface

This section describes how to make Query requests to Amazon SES. The various topics acquaint you with the Amazon SES Query interface, the components of a request, how to authenticate a request, and the content of responses.

- For information about Query requests, see Query Requests and Amazon SES (p. 84).
- For information about request authentication, see Request Authentication and Amazon SES (p. 87).
- For examples of GET and POST requests, see GET and POST Examples for Amazon SES (p. 89).
- For information about Query responses, see Query Responses and Amazon SES (p. 91).

Query Requests and Amazon SES

Amazon Simple Email Service supports Query requests for service actions. Query requests are simple HTTPS requests that use the GET or POST method. Query requests must contain an *`Action`* parameter to indicate the action to be performed.

> **Important**
> For security reasons, Amazon SES does not support HTTP requests. You must use HTTPS instead.

Structure of a GET Request

This guide presents the Amazon SES GET requests as URLs. Each URL consists of the following:

- **Endpoint—**The resource the request is acting on. See Regions and Amazon SES (p. 173) for a list of Amazon SES endpoints.
- **Action—**The action you want to perform on the endpoint, such as sending a message.
- **Parameters—**Any request parameters.

The following is an example GET request to send a message using the Amazon SES endpoint in the US West (Oregon) region.

```
https://email.us-west-2.amazonaws.com?Action=SendEmail&Source=user%40ex
ample.com&Destination.ToAddresses.member.1=allan%40example.com&Message.Sub
ject.Data=This%20is%20the%20subject%20line.&Mes
sage.Body.Text.Data=Hello.%20I%20hope%20you%20are%20having%20a%20good%20day.
```

> **Important**
> Because the GET requests are URLs, you must URL-encode the parameter values. For example, in the preceding example request, the value for the *`Source`* parameter is actually

`user@example.com`. However, the "@" character is not allowed in URLs, so each "@" is URL-encoded as "`%40`".

To make the GET examples easier to read, this guide presents them in the following parsed format.

```
https://email.us-west-2.amazonaws.com
?Action=SendEmail
&Source=user%40example.com
&Destination.ToAddresses.member.1=allan%40example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20having%20a%20good%20day.
```

The first line represents the *endpoint* of the request. After the endpoint is a question mark (?), which separates the endpoint from the parameters. Each parameter is separated by an ampersand (&).

The *`Action`* parameter indicates the action to perform. See the Amazon Simple Email Service API Reference for a complete list of actions, and the parameters used with each action.

Some operations take lists of parameters. For example, when you send an email to multiple recipients, you can provide a list of email addresses. You specify this type of list with *param.n* notation, where values of *n* are integers starting from 1. For example, you would specify the first "To:" address using *Destination.ToAddresses.1*, the second with *Destination.ToAddresses.2*, etc.

In Amazon SES, no spaces are allowed in any of the parameter values. In this guide, any example Query request parameter value that includes spaces is displayed in one of two different ways:

- URL-encoded (as `%20`).
- Represented by a plus sign ("+"). Within a Query request, a plus sign is reserved as a shorthand notation for a space. (If you want to include a literal, uninterpreted plus sign in any parameter, you must URL-encode it as `%2B`.)

Note
Every request must be accompanied by an `X-Amzn-Authorization` HTTP header. For instructions on how to create this header, see Authentication Process (p. 87).

Structure of a POST Request

Amazon SES also accepts POST requests. With a POST request, you send the query parameters as a form in the HTTP request body as described in the following procedure.

To create a POST request

1. Assemble the query parameter names and values into a form.

   Put the parameters and values together as you would for a GET request (with an ampersand separating each name-value pair). The following example shows a `SendEmail` request with the line breaks we use in this guide to make the information easier to read.

   ```
   Action=SendEmail
   &Source=user@example.com
   &Destination.ToAddresses.member.1=allan@example.com
   &Message.Subject.Data=This is the subject line.
   &Message.Body.Text.Data=Hello. I hope you are having a good day.
   ```

2. Form-URL-encode the form according to the Form Submission section of the HTML specification.

For more information, go to http://www.w3.org/MarkUp/html-spec/html-spec_toc.html#SEC8.2.1.

```
Action=SendEmail
&Source=user%40example.com
&Destination.ToAddresses.member.1=allan%40example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20hav
ing%20a%20good%20day.
```

3. Provide the resulting form as the body of the POST request.
4. Include the following HTTP headers in the request:

   - `Content-Type`, with the value set to `application/x-www-form-urlencoded`
   - `Content-Length`
   - `Date`
   - `X-Amzn-Authorization` (for more information, see Authentication Process (p. 87))

5. Send the completed request.

```
POST / HTTP/1.1
Date: Thu, 26 May 2011 06:49:50 GMT
Host: email.us-west-2.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amzn-Authorization: AWS3 AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Signa
ture=lBP67vCvGlDMBQ=dofZxg8E8SUEXAMPLE,Algorithm=HmacSHA256,SignedHead
ers=Date;Host
Content-Length: 230

Action=SendEmail
&Source=user%40example.com
&Destination.ToAddresses.member.1=allan%40example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20hav
ing%20a%20good%20day.
```

The `X-Amzn-Authorization` header you provide is the same header you would provide if you sent a GET request (for information about this header, see Authentication Process (p. 87)).

Note

Your HTTP client typically adds other items to the HTTP request as required by the version of HTTP that the client uses. We don't include those additional items in the examples in this guide.

Request Authentication and Amazon SES

When you make a request to the Amazon SES API, you must provide proof that you are truly the account holder so that Amazon SES can verify your identity and whether you are registered to use services offered by AWS. If either test fails, Amazon SES returns an error and does not process the request.

Authentication Process

To provide proof of your identity, you must provide the following items as part of the `X-Amzn-Authorization` HTTPS header in your request to the Amazon SES API:

- **AWSAccessKeyId**—Your AWS account is identified by your access key ID, which AWS uses to look up your secret access key. For information about how to get your access key ID, see Getting Your AWS Access Keys (p. 45).
- **Signature**—Each request must contain a valid request signature, or the request will be rejected. A request signature is calculated using your secret access key, which is a shared secret known only to you and AWS.

  Note
  Amazon SES supports signature version 3 and version 4. Version 4 is preferred. For information about using signature version 4, see Signature Version 4 Signing Process in the AWS general reference documentation.
- **Algorithm**—Identify which HMAC hash algorithm you used to calculate your signature, either SHA256 or SHA1. For information about HMAC, go to http://www.faqs.org/rfcs/rfc2104.html

When Amazon SES receives your request, it does the following:

1. Uses the access key ID to look up your secret access key.
2. Generates a signature from the request data and the secret access key using the same algorithm you used to calculate the signature you sent in the request.
3. If the signature generated by Amazon SES matches the one you sent in the request, Amazon SES handles the request. If the comparison fails, the request is discarded, and Amazon SES returns an error response.

To create the *X-Amzn-Authorization* header

1. Create a `Date` header to be used in the request. For more information, go to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.18.

   Here is an example of what a `Date` header might look like:

   ```
   Date: Tue, 25 May 2010 21:20:27 +0000
   ```

2. To create the *string to sign*, calculate an RFC 2104-compliant HMAC hash with the `Date` header value, your secret access key as the key, and SHA256 or SHA1 as the hash algorithm. For more information, go to http://www.ietf.org/rfc/rfc2104.txt.

   Note
   Use only the *value* of the header when calculating the hash; do not include the word "Date", nor the trailing colon and space.
3. To create the *request signature*, convert the HMAC hash to base64. The resulting value is the *signature* for this request.
4. Create an `X-Amzn-Authorization` header, consisting of the following elements:

   a. `AWS3-HTTPS`.

b. `AWSAccessKeyId`=your AWS Access Key ID.

c. `Algorithm`=the algorithm you used when creating the string to sign—either HmacSHA1 or HmacSHA256.

d. `Signature`=the signature for this request.

All of the elements, except for `AWS3-HTTPS`, must be separated by commas.

Here is an example of what an `X-Amzn-Authorization` header might look like, using placeholders for the AWS Access Key ID and the signature:

```
X-Amzn-Authorization: AWS3-HTTPS AWSAccessKeyId=<Your AWS Access Key ID>,
Algorithm=HmacSHA256, Signature=<Signature>
```

GET and POST Examples for Amazon SES

The following are examples of GET and POST requests, using the Query API.

Example GET Request

Here is an example of what a GET request might look like, including the calculated signature. Notice that all of the parameters have been URL-encoded.

```
https://email.us-west-2.amazonaws.com/
?Action=SendEmail
&Source=user%40example.com
&Destination.ToAddresses.member.1=allan%40example.com
&Message.Subject.Data=This%20is%20the%20subject%20line.
&Message.Body.Text.Data=Hello.%20I%20hope%20you%20are%20having%20a%20good%20day.
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Signature=RhU864jFu893mg7g9N9j9nr6h7EXAMPLE
&Algorithm=HMACSHA256
```

Example POST Request

Here is an example of what a POST request might look like, before calculating the signature. Notice that all of the parameters have been URL-encoded.

```
POST / HTTP/1.1
Host: email.us-west-2.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Date: Tue, 25 May 2010 21:20:27 +0000
Content-Length: 174

Action=SendRawEmail
&Destinations.member.1=allan%40example.com
&RawMessage.Data=RnJvbTp1c2VyQGV4YW1wbGUuY29tDQpTdWJqZWN0OiBUZXN0DQoNCk1lc3 ...
```

The value for *`RawMessage.Data`* is a base64-encoded representation of the following text.

```
From:user@example.com
Subject: Test

Message sent using SendRawEmail.
```

Following is the complete POST request to *SendRawEmail*, with the `X-Amzn-Authorization` header. None of the headers should be URL-encoded.

```
POST / HTTP/1.1
Host: email.us-west-2.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Date: Tue, 25 May 2010 21:20:27 +0000
Content-Length: 174
X-Amzn-Authorization: AWS3-HTTPS AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE,Al
gorithm=HMACSHA256,Signature=lBP67vCvGl ...

Action=SendRawEmail
```

```
&Destinations.member.1=allan%40example.com
&RawMessage.Data=RnJvbTp1c2VyQGV4YW1wbGUuY29tDQpTdWJqZWN0OiBUZXN0DQoNCk1lc3 ...
```

Query Responses and Amazon SES

In response to a Query request, Amazon Simple Email Service returns an XML data structure that contains the results of the request.

Every Amazon SES response includes a request ID in a *`RequestId`* element. The value is a unique string that AWS assigns. If you ever have issues with a particular request, AWS will ask for the request ID to help troubleshoot the issue.

Successful Amazon SES responses also include one or more message IDs. You can think of a message ID as a receipt for an email message that Amazon SES sends. If a message is rejected or bounced, the message ID will appear in any complaint or bounce notifications that you receive; you can then use the message ID to identify any problematic email messages that you have sent, and take corrective action.

Structure of a Successful Response

If the request succeeded, the main response element is named after the action, but with "Response" appended. For example, `SendEmailResponse` is the response element returned for a successful `SendEmail` request. This element contains the following child elements:

- `ResponseMetadata`, which contains the `RequestId` child element.
- An optional element containing action-specific results. For example, the `SendEmailResponse` element includes an element called `SendEmailResult`.

The XML schema describes the XML response message for each Amazon SES action.

The following is an example of a successful response.

```
<SendEmailResponse xmlns="https://email.amazonaws.com/doc/2010-03-31/">
  <SendEmailResult>
    <MessageId>000001271b15238a-fd3ae762-2563-11df-8cd4-6d4e828a9ae8-
000000</MessageId>
  </SendEmailResult>
  <ResponseMetadata>
    <RequestId>fd3ae762-2563-11df-8cd4-6d4e828a9ae8</RequestId>
  </ResponseMetadata>
</SendEmailResponse>
```

Structure of an Error Response

If a request is unsuccessful, the main response element is called `ErrorResponse` regardless of the action that was called. This element contains an `Error` element and a `RequestId` element. Each `Error` includes:

- A `Type` element that identifies whether the error was a receiver or sender error
- A `Code` element that identifies the type of error that occurred
- A `Message` element that describes the error condition in a human-readable form
- A `Detail` element that might give additional details about the error or might be empty

The following is an example of an error response.

```
<ErrorResponse>
   <Error>
      <Type>
         Sender
      </Type>
      <Code>
         ValidationError
      </Code>
      <Message>
         Value null at 'message.subject' failed to satisfy constraint: Member
must not be null
      </Message>
   </Error>
   <RequestId>
      42d59b56-7407-4c4a-be0f-4c88daeea257
   </RequestId>
</ErrorResponse>
```

Authenticating Email in Amazon SES

Amazon Simple Email Service (Amazon SES) uses the Simple Mail Transfer Protocol (SMTP) to send email. Because SMTP does not provide any authentication by itself, spammers can send email messages that claim to originate from someone else, while hiding their true origin. By falsifying email headers and spoofing source IP addresses, spammers can mislead recipients into believing that the email messages that they are receiving are authentic.

Most ISPs that forward email traffic take measures to evaluate whether email is legitimate. One such measure that ISPs take is to determine whether an email is *authenticated*. Authentication requires senders to verify that they are the owner of the account that they are sending from. In some cases, ISPs refuse to forward email that is not authenticated. To ensure optimal deliverability, we recommend that you authenticate your emails.

The following sections describe two authentication mechanisms ISPs use—Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM)—and provide instructions for how to use these standards with Amazon SES.

- To learn about SPF, which provides a way to trace an email message back to the system from which it was sent, see Authenticating Email with SPF in Amazon SES (p. 93).
- To learn about DKIM, a standard that allows you to sign your email messages to show ISPs that your messages are legitimate and have not been modified in transit, see Authenticating Email with DKIM in Amazon SES (p. 94).

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Authenticating Email with SPF in Amazon SES

Sender Policy Framework (SPF) provides a means for tracing an email message back to the system from which it was sent.

To be SPF-compliant, an email sender publishes one or more DNS records that establish the sending domain's identity. These DNS records are usually specified as TXT (text); they identify a set of hosts that are authorized to send email. After these DNS records are created and published, ISPs can authenticate a host by comparing its IP address with the set of IP addresses specified in the SPF record.

Amazon SES sends your emails from a "Mail-From" domain that Amazon SES owns. You therefore do not need to make any changes to your DNS records for your emails to pass SPF authentication. For more information, see SPF and Amazon SES on the Amazon SES blog.

> **Important**
> If you want to implement Domain-based Message Authentication, Reporting and Conformance (DMARC), you must enable DKIM. DMARC requires authentication via SPF and/or DKIM to verify your domain. SPF alone will not comply with DMARC because the "Mail From" domain of email sent through Amazon SES is *amazonses.com* (or a subdomain of that), which is different from your sending domain. Using DKIM enables DMARC to verify your sending domain. For information about how to set up DKIM with Amazon SES, see Authenticating Email with DKIM in Amazon SES (p. 94).

For more information about SPF, go to http://www.openspf.net and RFC 7208.

Authenticating Email with DKIM in Amazon SES

DomainKeys Identified Mail (*DKIM*) is a standard that allows senders to sign their email messages and ISPs to use those signatures to verify that those messages are legitimate and have not been modified by a third party in transit.

An email message that is sent using DKIM includes a *DKIM-Signature* header field that contains a cryptographically signed representation of all, or part, of the message. An ISP receiving the message can decode the cryptographic signature using a public key, published in the sender's DNS record, to ensure that the message is authentic. For more information about DKIM, see http://www.dkim.org.

DKIM signatures are optional. You might decide to sign your email using a DKIM signature to enhance deliverability with DKIM-compliant ISPs. Amazon SES provides two options to sign your messages using a DKIM signature:

- To set up your domain so that Amazon SES automatically adds a DKIM signature to every message sent from that domain, see Easy DKIM in Amazon SES (p. 94).
- To add your own DKIM signature to any email that you send using the `SendRawEmail` API, see Manual DKIM Signing in Amazon SES (p. 103).

Easy DKIM in Amazon SES

Easy DKIM is a feature of Amazon SES that signs every message that you send from a verified email address or domain with a DKIM signature that uses a 1024-bit DKIM key. You can use the Amazon SES console to configure Easy DKIM settings, and to enable or disable automatic DKIM signing for your email messages. You must be able to edit your domain’s DNS records to set up Easy DKIM. With the appropriate DNS records in place, you can enable Easy DKIM signing for any verified email address or domain.

> **Important**
> If you set up Easy DKIM for a domain, it will apply to all email addresses in that domain except for email addresses that you individually verified. Individually verified email addresses use separate settings.

Once you set up Easy DKIM, your messages will automatically be DKIM-signed regardless of whether you call Amazon SES through the SMTP interface or the API (`SendEmail` or `SendRawEmail`). Note that you only need to set up Easy DKIM for the domain you use in your "From" address, not for the domain in a "Return-Path" or "Reply-To" address.

If you are verifying a new domain, you can set up Easy DKIM at that time. If you already have a verified domain or email address, you can add Easy DKIM capability to it whenever you want.

> **Note**
> Amazon SES has endpoints in multiple AWS regions, and Easy DKIM setup applies to each AWS region separately. You must perform the Easy DKIM setup procedure for each region in which you want to use Easy DKIM. For information about using Amazon SES in multiple AWS regions, see Regions and Amazon SES (p. 173).

This topic contains the following sections:

- To set up Easy DKIM while you verify a new domain, see Setting Up Easy DKIM for a New Domain (p. 95).
- To set up Easy DKIM for an email address or domain that you have already verified, see Setting Up Easy DKIM for an Existing Verified Identity (p. 98).

Setting Up Easy DKIM for a New Domain

When you use the AWS Management Console to verify a new domain, you can also set up Easy DKIM at the same time.

These instructions are for new domains only. If you want to set up Easy DKIM for an email address or domain that you have already verified, see Setting Up Easy DKIM for an Existing Verified Identity (p. 98).

> **Important**
> Easy DKIM only works with fully qualified domain names (FQDNs). If you wanted to set up Easy DKIM for both *example.com* and *newyork.example.com*, you would need to set up Easy DKIM for each of these FQDNs separately.

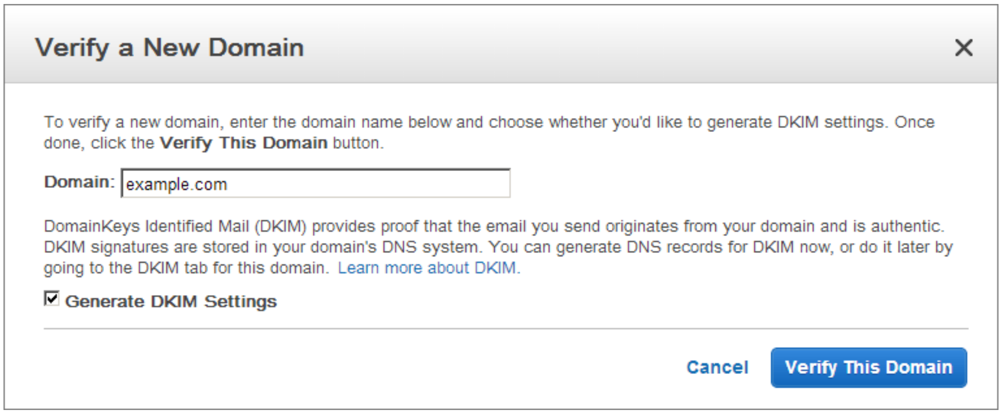
To set up Easy DKIM for a new domain

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:

   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.

   b. In the navigation pane, under **Identities**, click **Domains**.

2. Click **Verify a New Domain**.
3. In the **Verify a New Domain** dialog box, enter your domain name, select the **Generate DKIM settings** check box, and then click **Verify This Domain**.



Verify a New Domain

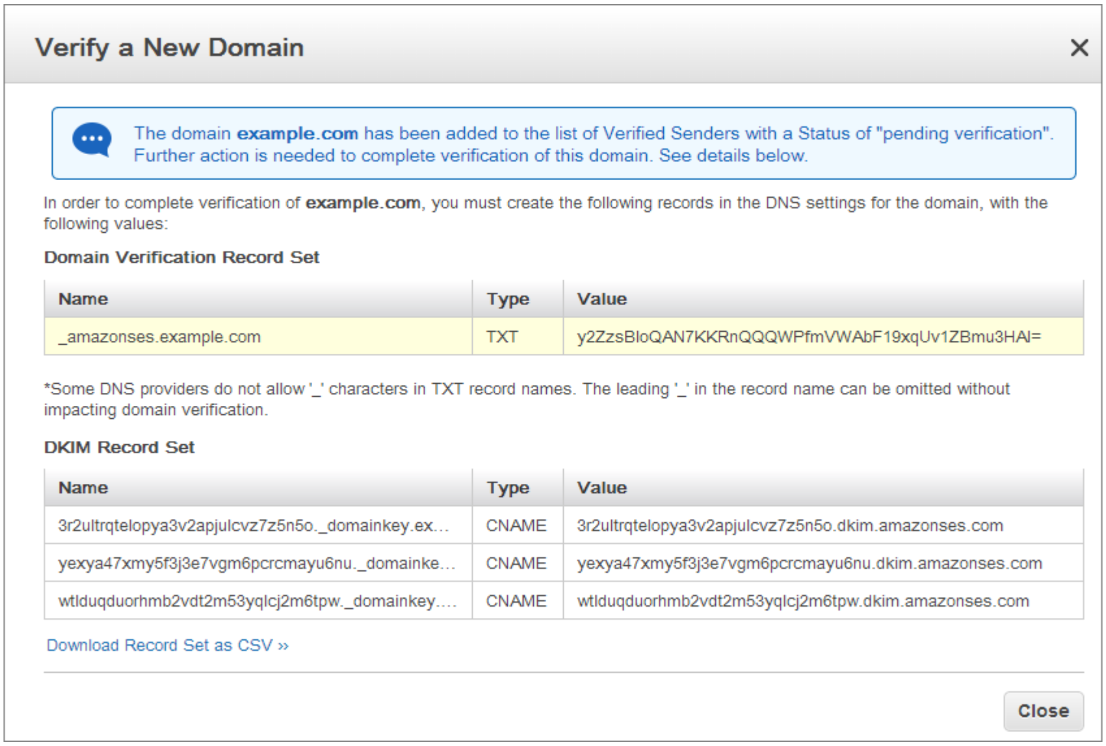
To verify a new domain, enter the domain name below and choose whether you'd like to generate DKIM settings. Once done, click the **Verify This Domain** button.

Domain: example.com

DomainKeys Identified Mail (DKIM) provides proof that the email you send originates from your domain and is authentic. DKIM signatures are stored in your domain's DNS system. You can generate DNS records for DKIM now, or do it later by going to the DKIM tab for this domain. Learn more about DKIM.

☑ Generate DKIM Settings

In the resulting dialog box, you will see all of the DNS records that you need for setting up domain verification and Easy DKIM. This information will also be available by clicking the domain name after you close the dialog box.



Verify a New Domain

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

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

Domain Verification Record Set

Name	Type	Value
_amazonses.example.com	TXT	y2ZzsBloQAN7KKRnQQQWPfmVWAbF19xqUv1ZBmu3HAI=

*Some DNS providers do not allow '_' characters in TXT record names. The leading '_' in the record name can be omitted without impacting domain verification.

DKIM Record Set

Name	Type	Value
3r2ultrqtelopya3v2apjulcvz7z5n5o._domainkey.ex...	CNAME	3r2ultrqtelopya3v2apjulcvz7z5n5o.dkim.amazonses.com
yexya47xmy5f3j3e7vgm6pcrcmayu6nu._domainke...	CNAME	yexya47xmy5f3j3e7vgm6pcrcmayu6nu.dkim.amazonses.com
wtlduqduorhmb2vdt2m53yqlcj2m6tpw._domainkey....	CNAME	wtlduqduorhmb2vdt2m53yqlcj2m6tpw.dkim.amazonses.com

4. To complete domain verification, you must update your domain's DNS settings with the TXT record information from the **Domain Verification Record Set** in the **Verify a New Domain** dialog box. Note that some domain name providers use the term **Host** instead of **Name**. If your DNS provider does not allow underscores in TXT record names, you can omit the underscore before *amazonses* in the TXT record name for domain verification. (You cannot, however, omit the underscore in the DKIM records, as described in the next step.)

   Highlight and copy individual records, or select **Download Record Set as CSV** to download all of the records.

   > **Important**
   > DNS providers may append the domain name to the end of DNS records. Adding a record that already contains the domain name (such as *_amazonses.example.com*) may result in the duplication of the domain name (such as *_amazonses.example.com.example.com*). To avoid duplication of the domain name, add a period to the end of the domain name in the DNS record. This will indicate to your DNS provider that the record name is fully qualified (that is, no longer relative to the domain name), and prevent the DNS provider from appending an additional domain name.

5. To set up DKIM, you must update your domain's DNS settings with the CNAME record information from the dialog box. Unlike for domain verification, you cannot omit the underscore from *_domainkey* in this case because the underscore is required by RFC 4871.

   Highlight and copy individual CNAME records, or select **Download Record Set as CSV** to download all of the records.

   a. If Amazon Route 53 provides the DNS service for the domain you are verifying, and you are logged in to Amazon SES console with the same email address and password you use for

Amazon Route 53, then you will have the option of immediately updating your DNS settings for both domain verification and DKIM from within the Amazon SES console.

b. If you are not using Amazon Route 53, you will need to update your DNS settings according to the procedure established by your DNS service provider. (Ask your system administrator if you are not sure who provides your DNS service.) Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours.

When verification is complete, the domain's **Status** in the Amazon SES console will change from *pending verification* to *verified*, and you will receive an *Amazon SES Domain Verification SUCCESS* confirmation email from Amazon Web Services. (AWS emails are sent to the email address you used when you signed up for Amazon SES.)

When Amazon SES has successfully detected the changes to your DNS records, the **DKIM Verification Status** for that domain in the Amazon SES console will change from *in progress* to *success*, and you will receive an *Amazon SES DKIM Setup Successful* confirmation email from Amazon Web Services.

Important
TO COMPLETE EASY DKIM SETUP, YOU MUST ENABLE DKIM SIGNING FOR THE VERIFIED IDENTITY IN THE NEXT STEP.

6. To sign your messages using a DKIM signature, you must enable Easy DKIM for the appropriate verified sending identity as follows:

   a. In the navigation pane, under **Identities**, click either **Email Addresses** or **Domains**, depending whether you want to enable Easy DKIM signing for an email address or a domain.
   b. Click the email address or domain for which you wish to enable Easy DKIM signing.
   c. On the Details page of the email address or domain, expand **DKIM**.
   d. In the **DKIM:** field, click **enable**.

7. You can now use Amazon SES to send email that is signed using a DKIM signature from any valid address in the verified domain. To send a test email using the Amazon SES console, check the box next to the verified domain, and then click **Send a Test Email**. View the email headers in the email you receive. Email providers typically provide this capability through an option such as **Show original** or **View message source**. Look for a header named *DKIM-Signature* with the *"d"* tag set to your domain. Note that when DKIM is enabled, there will be two *DKIM-Signature* headers added to the message: one header for your domain, and one header with *d=amazonses.com*. (Amazon SES adds a signature for *amazonses.com* automatically whether you have DKIM enabled or not. You can ignore it.) For example, for a domain called *ses-example.com*, the DKIM signature header you are looking for might look like:

```
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple;
s=xtk53kxcy4p3t6ztbrffs6d54rsrrhh6; d=ses-example.com;
t=1366720445;
h=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Date:Mes
sage-ID;
bh=lcj/Sl5qKl6K6zwFUwb7Flgnngl892pW574kmS1hrS0=;
b=nhVMQLmSh7/DM5PW7xPV4K/PN4iVY0a5OF4YYk2L7jgUq9hHQlckopxe82TaAr64
eVTcBhHHj9Bwtzkmuk88g4G5UUN8J+AAsd/JUNGoZOBS1OofSkuAQ6cGfRGanF68Ag7
nmmEjEi+JL5JQh//u+EKTH4TVb4zdEWlBuMlrdTg=
```

Important

How you update the DNS settings depends on who provides your DNS service. DNS service may be provided by a domain name registrar such as GoDaddy or Network Solutions, or by a separate service such as Amazon Route 53.

What if Easy DKIM fails?

If your DNS settings are not correctly updated, you will first receive an *Amazon SES DKIM FAILURE* email from Amazon Web Services, and you will see a status of *failed* in the Domains area when you click on the DKIM tab.

Note

If this happens, Amazon SES will still send your email, but *it will not be signed using a DKIM signature*.

Setting Up Easy DKIM for an Existing Verified Identity

If you have already verified a domain or email address, you can use the AWS Management Console to set up Easy DKIM for that identity at any time.

These instructions are for adding DKIM signing to a domain that has already been verified. If you are verifying a new domain and want to set up Easy DKIM at the same time, see Setting Up Easy DKIM for a New Domain (p. 95).

Important

Easy DKIM only works with fully qualified domain names (FQDNs). If you wanted to set up Easy DKIM for both *example.com* and *newyork.example.com*, you would need to set up Easy DKIM for each of these FQDNs separately.

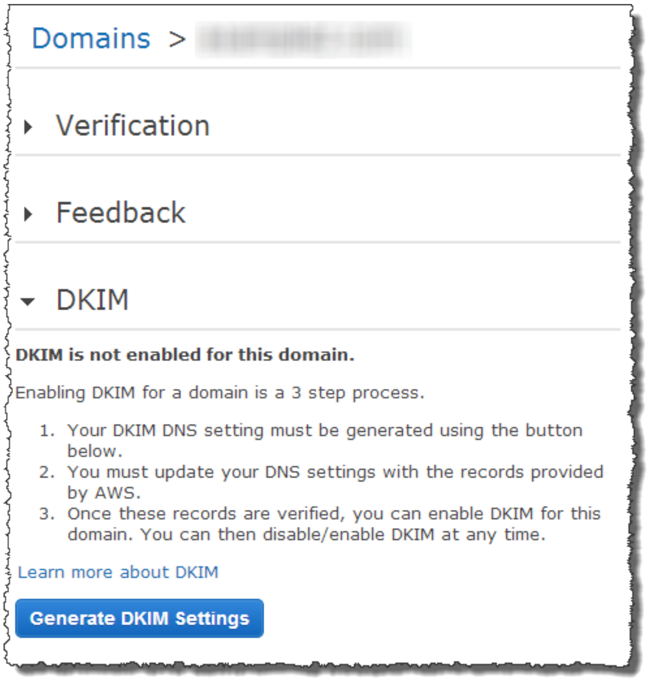
To set up Easy DKIM for an existing verified domain

1. Go to your verified domain list in the Amazon SES console, or follow these instructions to navigate to it:

   a. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.

   b. In the navigation pane, under **Identities**, click **Domains**.

2. In the content pane, click the verified domain for which you would like to set up Easy DKIM.
3. On the domain's Details page, expand **DKIM**.



Domains > [redacted]

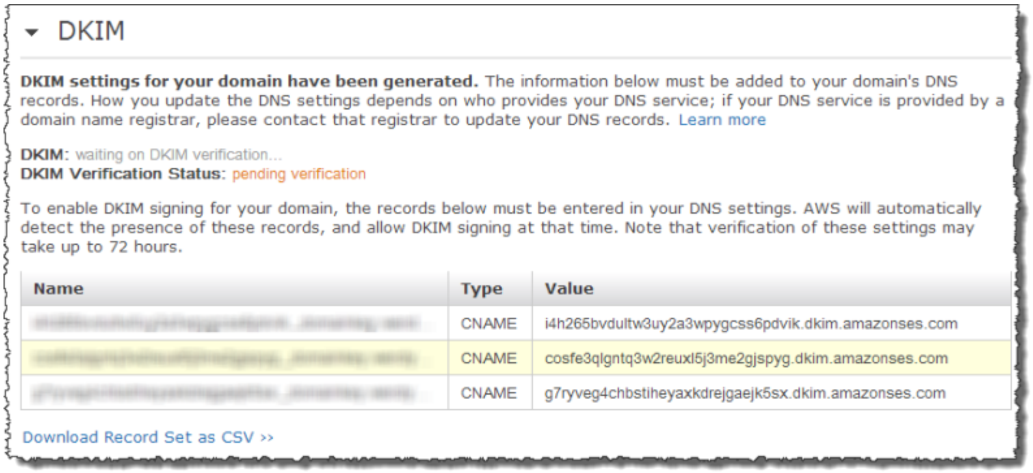
DKIM is not enabled for this domain.

Enabling DKIM for a domain is a 3 step process.

1. Your DKIM DNS setting must be generated using the button below.
2. You must update your DNS settings with the records provided by AWS.
3. Once these records are verified, you can enable DKIM for this domain. You can then disable/enable DKIM at any time.

4. Click **Generate DKIM Settings**.

   Your DKIM records will be displayed.



DKIM settings for your domain have been generated. The information below must be added to your domain's DNS records. How you update the DNS settings depends on who provides your DNS service; if your DNS service is provided by a domain name registrar, please contact that registrar to update your DNS records. Learn more

DKIM: waiting on DKIM verification...
DKIM Verification Status: pending verification

To enable DKIM signing for your domain, the records below must be entered in your DNS settings. AWS will automatically detect the presence of these records, and allow DKIM signing at that time. Note that verification of these settings may take up to 72 hours.

Name	Type	Value
[redacted]	CNAME	i4h265bvdultw3uy2a3wpygcss6pdvik.dkim.amazonses.com
[redacted]	CNAME	cosfe3qlgntq3w2reuxl5j3me2gjspyg.dkim.amazonses.com
[redacted]	CNAME	g7ryveg4chbstiheyaxkdrejgaejk5sx.dkim.amazonses.com

5. To set up DKIM, you must update your domain's DNS settings with the displayed CNAME record information. You can copy the records or click the **Download Record Set as CSV** link.

   a. If Amazon Route 53 provides the DNS service for the domain you are verifying, and you are logged in to Amazon SES console with the same email address and password you use for Amazon Route 53, then Amazon SES will give you the option of immediately updating your DNS settings for Easy DKIM. If you would like to do this, click the **Use Route 53** button.

      Next, click **Create Record Sets** in the **Use Route 53** dialog box to complete the process.

   b. If you are not using Amazon Route 53, you will need to update your DNS settings according to the procedure established by your DNS service provider. (Ask your system administrator if you are not sure who provides your DNS service.) Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours.

6. When Amazon SES has successfully detected the changes to your DNS records, the **DKIM Verification Status** for that domain in the Amazon SES console will change from *in progress* to *success*, and you will receive an Amazon SES DKIM Setup Successful confirmation email from Amazon Web Services. (Amazon Web Services emails are sent to the email address you used when you signed up for Amazon SES.)

   Important
   TO COMPLETE EASY DKIM SETUP, YOU MUST ENABLE DKIM SIGNING FOR THE VERIFIED IDENTITY IN THE NEXT STEP.

7. To sign your messages using a DKIM signature, you must enable Easy DKIM for the appropriate verified sending identity as follows:

   a. In the navigation pane, under **Identities**, click either **Email Addresses** or **Domains**, depending whether you want to enable Easy DKIM signing for an email address or a domain.
   b. Click the email address or domain for which you wish to enable Easy DKIM signing.
   c. On the Details page of the email address or domain, expand **DKIM**.
   d. In the **DKIM:** field, click **enable**.

8. You can now use Amazon SES to send email that is signed using a DKIM signature from any valid address in the verified domain. To send a test email using the Amazon SES console, check the box next to the verified domain, and then click **Send a Test Email**. View the email headers in the email you receive. Email providers typically provide this capability through an option such as **Show original** or **View message source**. Look for a header named *DKIM-Signature* with the *"d"* tag set to your domain. Note that when DKIM is enabled, there will be two *DKIM-Signature* headers added to the message: one header for your domain, and one header with *d=amazonses.com*. (Amazon SES adds a signature for *amazonses.com* automatically whether you have DKIM enabled or not. You can ignore it.) For example, for a domain called *ses-example.com*, the DKIM signature header you are looking for might look like:

```
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple;
s=xtk53kxcy4p3t6ztbrffs6d54rsrrhh6; d=ses-example.com;
t=1366720445;
h=From:To:Subject:MIME-Version:Content-Type:Content-Transfer-Encoding:Date:Mes
sage-ID;
bh=lcj/Sl5qKl6K6zwFUwb7Flgnngl892pW574kmS1hrS0=;
b=nhVMQLmSh7/DM5PW7xPV4K/PN4iVY0a5OF4YYk2L7jgUq9hHQlckopxe82TaAr64
eVTcBhHHj9Bwtzkmuk88g4G5UUN8J+AAsd/JUNGoZOBS1OofSkuAQ6cGfRGanF68Ag7
nmmEjEi+JL5JQh//u+EKTH4TVb4zdEWlBuMlrdTg=
```

Important
How you update the DNS settings depends on who provides your DNS service. DNS service may be provided by a domain name registrar such as GoDaddy or Network Solutions, or by a separate service such as Amazon Route 53.

What if Easy DKIM fails?

If your DNS settings are not correctly updated, you will first receive an *Amazon SES DKIM FAILURE* email from Amazon Web Services, and you will see a status of *failed* in the Domains area when you click on the DKIM tab.

> **Note**
> If this happens, Amazon SES will still send your email, but *it will not be signed using a DKIM signature*.

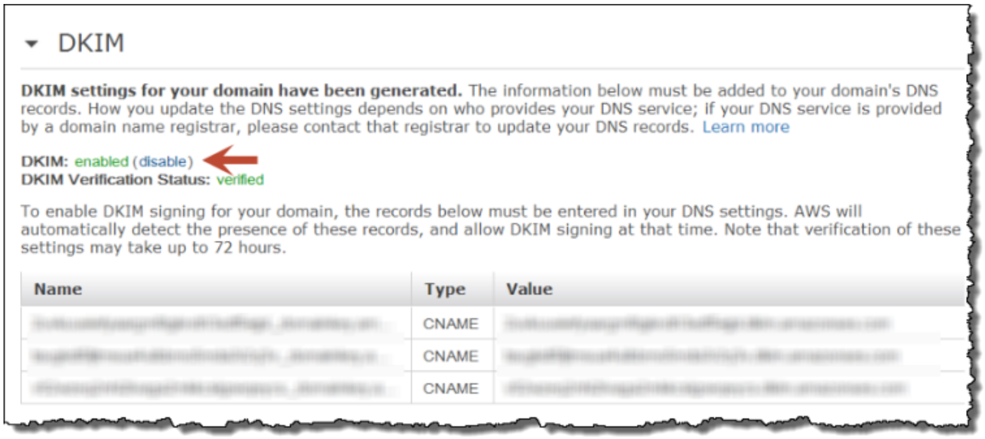
Disabling Easy DKIM in Amazon SES

If you want to temporarily stop Amazon SES from signing your messages using DKIM, you can disable Easy DKIM for your email address or domain. You can reenable it at any time.

To disable Easy DKIM signing

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the navigation pane, under **Identities**, click either **Email Addresses** or **Domains**, depending whether you want to disable Easy DKIM signing for an email address or a domain.
3. Click the email address or domain for which you wish to disable Easy DKIM signing.
4. On the Details page of the email address or domain, expand **DKIM**.
5. In the **DKIM:** field, click **disable**. Amazon SES will no longer DKIM-sign emails that you send from this identity.

   > **Note**
   > If you do not see the **disable** option as in the figure below, then DKIM is already disabled.



DKIM settings for your domain have been generated. The information below must be added to your domain's DNS records. How you update the DNS settings depends on who provides your DNS service; if your DNS service is provided by a domain name registrar, please contact that registrar to update your DNS records. Learn more

DKIM: enabled (disable)
DKIM Verification Status: verified

To enable DKIM signing for your domain, the records below must be entered in your DNS settings. AWS will automatically detect the presence of these records, and allow DKIM signing at that time. Note that verification of these settings may take up to 72 hours.

Name	Type	Value
[illegible]	CNAME	[illegible]
[illegible]	CNAME	[illegible]
[illegible]	CNAME	[illegible]

> **Note**
> If you want to *permanently* disable DKIM signing from any email address on that domain, you should also remove the CNAME records from your DNS.

DKIM Record Revocation in Amazon SES

Amazon Web Services periodically reviews DKIM DNS records, and revokes DKIM signing in cases where it is no longer valid. If Amazon Web Services is unable to detect the CNAME record information required to confirm the ownership of a domain, you will receive an *Amazon SES DKIM REVOCATION WARNING* email from Amazon Web Services. Amazon SES will continue to send your email, but it will not be signed using a DKIM signature.

If you restore the CNAME record information to your DNS settings within five days, you will receive an *Amazon SES DKIM REVOCATION CANCELLATION* email from Amazon Web Services. Amazon SES will once again sign email using a DKIM signature from a verified identity for which you have enabled Easy DKIM.

If you do not restore the CNAME record information to your DNS settings within five days, you will receive an *Amazon SES DKIM REVOCATION* email from Amazon Web Services, and email you send via Amazon SES will not be signed using a DKIM signature.

To set up Easy DKIM for a domain for which DKIM signing has been revoked, you must restart the procedure from the beginning, as if you were setting up Easy DKIM for the first time.

Other Ways to Manage Easy DKIM in Amazon SES

You can also manage Easy DKIM with the Amazon SES API. The following actions are available:

- `VerifyDomainDkim`
- `SetIdentityDkimEnabled`
- `GetIdentityDkimAttributes`

You can use these API actions to write a customized front-end application for working with Easy DKIM. For a complete description of API actions related to Easy DKIM, go to the Amazon Simple Email Service API Reference.

Creating DNS Records for DKIM Signing in Amazon SES

Unlike the Amazon SES console, the Amazon SES API does not generate fully-formed DNS records for use with DKIM. Instead, they return DKIM *tokens* — character strings that represent your domain's identity.

If you are not using the Amazon SES console, you will need to create your own `CNAME` records using the DKIM tokens returned by the API.

To create DNS records for DKIM signing

1. Obtain the DKIM tokens for your domain. To do so, if you are using the Amazon SES API, call `VerifyDomainDkim` to generate the tokens. If you already have a DKIM verified identity, call `GetIdentityDkimAttributes` to obtain the tokens.
2. In the output from the API, you will receive three DKIM tokens similar to the following:

```
vvjuipp74whm76gqoni7qmwwn4w4qusjiainivf6sf
3frqe7jn4obpuxjpwpolz6ipb3k5nvt2nhjpik2oy
wrqplteh7oodxnad7hsl4mixg2uavzneazxv5sxi2
```

3. Use these tokens to construct three `CNAME` records. For a domain named *example.com*, the records should appear similar to these:

```
vvjuipp74whm76gqoni7qmwwn4w4qusjiainivf6sf._domainkey.example.com CNAME
vvjuipp74whm76gqoni7qmwwn4w4qusjiainivf6sf.dkim.amazonses.com
3frqe7jn4obpuxjpwpolz6ipb3k5nvt2nhjpik2oy._domainkey.example.com CNAME
3frqe7jn4obpuxjpwpolz6ipb3k5nvt2nhjpik2oy.dkim.amazonses.com
wrqplteh7oodxnad7hsl4mixg2uavzneazxv5sxi2._domainkey.example.com CNAME wr
qplteh7oodxnad7hsl4mixg2uavzneazxv5sxi2.dkim.amazonses.com
```

You can now update your DNS with these records. Amazon Web Services will eventually detect that you have updated your DNS records; this detection process may take up to 72 hours. Upon successful detection, you will receive an Amazon SES DKIM Setup Successful confirmation email from Amazon Web Services. (Amazon Web Services emails are sent to the email address you used when you signed up for Amazon SES.)

Manual DKIM Signing in Amazon SES

If you prefer not to use Easy DKIM, you can still sign your email messages using a DKIM signature and send them using Amazon SES. To do this, you must use the `SendRawEmail` API and self-sign your message content according to the specifications provided at http://www.dkim.org. If you use this approach, be aware that Amazon SES does not validate the DKIM signature that you construct. If there are any errors in the signature, you will need to correct them yourself. If you DKIM-sign your own email messages, we recommend that you use keys that are at least 1024 bits.

Whether or not you DKIM-sign your messages, Amazon SES automatically adds a DKIM header with *d=amazonses.com*, which you can ignore. If you do DKIM-sign your messages, it is expected that there will be two DKIM headers: one for your domain, and one for *amazonses.com*.

Important

To ensure maximum deliverability, do *not* sign any of the following headers using a DKIM signature:

- Message-ID
- Date
- Return-Path
- Bounces-To

Note

If you are using the Amazon SES SMTP interface to send email, and your client software automatically performs DKIM signing, you should check to ensure that your client does not sign any of the headers listed above. We recommend that you check the documentation for your software to find out exactly what headers are signed with DKIM.

For more information about the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 48).

Monitoring Your Amazon SES Sending Activity

Amazon SES provides a means by which you can monitor your sending activity, and we strongly encourage you to monitor your sending activity regularly. For example, you should watch your number of bounces, complaints, and rejected emails so that you can identify and correct problems right away. Excessive bounces and complaints constitute abuse and put your AWS account at risk of termination. We also recommend that you frequently check your sending statistics to ensure that you are not close to your sending limits. If you are close to your sending limits, see Increasing Your Amazon SES Sending Limits (p. 125) for information about how to increase them. Don't wait until you reach your sending limits to consider increasing them.

You can use the Amazon SES console or the Amazon SES API, whether by calling the Query (HTTPS) interface directly or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell, to find the following kinds of information for the last 24 hours:

- Delivery attempts
- Bounces (hard bounces only)
- Complaints
- Rejected send attempts (a rejected email is an email that Amazon SES initially accepted, but later rejected because the email contained a virus. Amazon SES notifies you by email and does not send the message.)
- Sending limits (your sending quota and your maximum send rate)
- Percentage of your quota that is used

You can also be notified of bounces, complaints, and deliveries through Amazon Simple Notification Service (Amazon SNS) or through email (bounces and complaints only).

> **Important**
> Amazon SES does not support Delivery Status Notifications (DSNs) through the *message/delivery-status* content type. If you try to send an email that contains the *message/delivery-status* content type, your email will be not be accepted by Amazon SES.

The following sections describe how to monitor your sending activity:

- To receive notifications about bounces, complaints, and deliveries, see Using Notifications with Amazon SES (p. 105).

- To find the number of bounces, complaints, delivery attempts, and rejected send attempts, see Monitoring Your Amazon SES Usage Statistics (p. 121).
- To find your sending quota, maximum send rate, and the number of emails you have sent in the last 24 hours, see Monitoring Your Amazon SES Sending Limits (p. 122).

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Using Notifications with Amazon SES

Amazon SES can notify you of the status of your emails by email or through Amazon Simple Notification Service (Amazon SNS). Amazon SES supports the following three types of notifications:

- **Bounces –** The email is rejected by the recipient's ISP or rejected by Amazon SES because the email address is on the Amazon SES suppression list. For ISP bounces, Amazon SES reports only hard bounces and soft bounces that will no longer be retried by Amazon SES. In these cases, your recipient did not receive your email message, and Amazon SES will not try to resend it. Bounce notifications are available through email and Amazon SNS. You are notified of out-of-the-office (OOTO) messages through the same method as bounces, although they don't count toward your bounce statistics. To see an example of an OOTO bounce notification, you can use the Amazon SES mailbox simulator. For more information, see Testing Amazon SES Email Sending (p. 158).
- **Complaints –** The email is accepted by the ISP and delivered to the recipient, but the recipient does not want the email and clicks a button such as "Mark as spam." If Amazon SES has a feedback loop set up with the ISP, Amazon SES will send you a complaint notification. Complaint notifications are available through email and Amazon SNS.
- **Deliveries –** Amazon SES successfully delivers the email to the recipient's mail server. This notification does not indicate that the actual recipient received the email because Amazon SES cannot control what happens to an email after the receiving mail server accepts it. Delivery notifications are available only through Amazon SNS.

You can set up notifications using the Amazon SES console or the Amazon SES API.

There are several important points to know about notifications:

- **Notification settings are configured on a per-verified-identity basis –** There is no global setting, and notification settings apply only to the verified identity in the AWS region in which you configured the settings.
- **Verified domain notification settings apply to all mail sent from email addresses in that domain *except* for email addresses that are also verified –** The configurations for email addresses are separate from the configuration for the domain, so changing the domain configuration will have no effect on the email address configuration. For example, if you verify only *example.com* and configure its bounce notification settings, bounce notifications for email from *sender@example.com* will use those settings. However, if you verify both *example.com* and *sender@example.com*, *sender@example.com* will **not** use the bounce notification settings that are configured for *example.com*.
- **You must receive bounce and complaint notifications either by email or through Amazon SNS –** The default method is by email, through a feature called *email feedback forwarding*. Delivery notifications are optional and available only through Amazon SNS.
- **If you choose to receive notifications for all three types of events, then you might receive multiple notifications for one email –** For example, the receiving mail server accepts the email (triggering a delivery notification), but the recipient marks the email as spam, triggering a complaint notification.
- **Before you start sending email, make sure that you set up a process to handle bounces and complaints –** Your process needs to monitor bounces and complaints and to remove those addresses

from your mailing list. Excessive bounces and complaints put your Amazon SES account at risk of termination. You will need to analyze each bounce and complaint message that you receive to determine the cause. Bounces are usually caused by attempting to send to a nonexistent recipient; complaints arise when recipients indicate that they do not want to receive your message. In either case, we strongly recommend that you stop sending to these email addresses.

- **You can test notifications by using the Amazon SES mailbox simulator –** Emails that you send to the mailbox simulator do not affect your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 158).

The following sections describe the notification methods:

- To receive notifications by email (which applies to bounces and complaints only), see Amazon SES Notifications Through Email (p. 106).
- To receive notifications through Amazon SNS (which applies to all three notification types), see Amazon SES Notifications Through Amazon SNS (p. 108).

Amazon SES Notifications Through Email

Amazon SES can notify you of your bounces and complaints through email using a process called *email feedback forwarding* or through Amazon Simple Notification Service (Amazon SNS). This topic is about receiving notifications by email, which is the default setting. For information about setting up notifications through Amazon SNS, see Amazon SES Notifications Through Amazon SNS (p. 108). Unlike bounce and complaint notifications, delivery notifications are available only through Amazon SNS.

> **Important**
> For several important points about notifications, see Using Notifications with Amazon SES (p. 105).

The following sections describe how to receive bounce and complaint notifications through email:

- To enable bounce and complaint notifications by email, see Enabling Email Feedback Forwarding (p. 106).
- To disable bounce and complaint notifications by email, see Disabling Email Feedback Forwarding (p. 107).
- To learn the email address to which bounce and complaint notifications are sent, see Email Feedback Forwarding Destination (p. 107).

Enabling Email Feedback Forwarding

Email feedback forwarding is enabled by default. If you previously disabled it, you can enable it by using the following procedure.

To enable bounce and complaint forwarding through email using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the navigation pane, under **Verified Senders**, click **Email Addresses** or **Domains**, depending on whether you want to configure bounce and complaint notifications for an email address or domain.
3. In the list of verified senders, click the email address or domain for which you want to configure bounce and complaint notifications.
4. In the details pane of the verified sender, expand **Notifications**.
5. Click **Edit Configuration**.
6. Under **Email Feedback Forwarding**, click **Enabled**.

   > **Note**
   > Changes made to your settings on this page might take a few minutes to take effect.

You can also enable bounce and complaint notifications through email by using the Amazon SES API `SetIdentityFeedbackForwardingEnabled` action. For more information, see the Amazon Simple Email Service API Reference action.

Disabling Email Feedback Forwarding

You must receive bounce and complaint notifications through either Amazon SNS or email feedback forwarding, so you can disable email feedback forwarding only if you select an Amazon SNS topic for both bounce and complaint notifications. If you selected an Amazon SNS topic for bounces and complaints, you can disable email feedback forwarding by using the following procedure.

To disable bounce and complaint forwarding through email using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the navigation pane, under **Verified Senders**, click **Email Addresses** or **Domains**, depending on whether you want to configure bounce and complaint notifications for an email address or domain.
3. In the list of verified senders, click the email address or domain for which you want to configure bounce and complaint notifications.
4. In the details pane of the verified sender, expand **Notifications**.
5. Click **Edit Configuration**.
6. In the Edit Notification Configuration dialog box, ensure that you have selected an Amazon SNS topic for both bounces and complaints. Otherwise, you will not be able to disable email feedback forwarding in the next step.
7. Under **Email Feedback Forwarding**, click **Disabled**.
8. Click **Save Config** to save your notification configuration.

   Note
   Changes made to your settings on this page might take a few minutes to take effect.

You can also disable bounce and complaint notifications through email by using the `SetIdentityFeedbackForwardingEnabled` API. For more information, go to the Amazon Simple Email Service API Reference.

Email Feedback Forwarding Destination

When you receive notifications by email, Amazon SES rewrites the *From:* header and sends the notification to you. The address to which Amazon SES forwards the notification depends on how you sent the original message.

If you used the SMTP interface to send the message, then notifications go to the address specified in the MAIL FROM command, which overrides any *Return-Path:* header specified in the SMTP DATA.

If you used the `SendEmail` API to send the message, then the notifications are delivered as follows:

- If you specified the optional `ReturnPath` parameter of `SendEmail`, then notifications go to that address.
- Otherwise, notifications go to the address specified in the required `Source` parameter of `SendEmail`, which populates the *From:* header of the message.

If you used the `SendRawEmail` API to send the message, then the notifications are delivered as follows:

- If you specified the optional `Source` parameter of `SendRawEmail`, then notifications go to that address, overriding any *Return-Path:* header specified in the raw message.

- Otherwise, if the *Return-Path:* header was specified in the raw message, then notifications go to that address.
- Otherwise, notifications go to the address in the *From:* header of the raw message.

Amazon SES Notifications Through Amazon SNS

You can use Amazon Simple Notification Service (Amazon SNS) to receive notifications about bounces, complaints, and deliveries for emails that you send through Amazon SES. Amazon SNS notifications are in JavaScript Object Notation (JSON) format, which enables you to process them programmatically.

If you configure Amazon SNS notifications for both bounces and complaints, you can disable receiving bounce and complaint notifications by email entirely. For information about receiving bounce and complaint notifications by email, see Amazon SES Notifications Through Email (p. 106). Delivery notifications are available only through Amazon SNS.

> **Important**
> For several important points about notifications, see Using Notifications with Amazon SES (p. 105).

For information about Amazon SES bounce, complaint, and delivery notifications through Amazon SNS, see the following sections:

- To set up notifications using the Amazon SES console or the Amazon SES API, see Configuring Amazon SNS Notifications for Amazon SES (p. 108).
- For a description of the contents of a notification, see Amazon SNS Notification Contents for Amazon SES (p. 110).
- For examples of bounce, complaint, and delivery notifications, see Amazon SNS Notification Examples for Amazon SES (p. 117).

Configuring Amazon SNS Notifications for Amazon SES

Amazon SES can notify you of your bounces, complaints, and deliveries through Amazon Simple Notification Service (Amazon SNS).

> **Important**
> For several important points about notifications, see Using Notifications with Amazon SES (p. 105).

You can configure notifications by using the Amazon SES console or by using the Amazon SES API, as described in the following sections.

- To configure notifications by using the Amazon SES console, see Configuring Notifications Using the Amazon SES Console (p. 108).
- To configure notifications by using the Amazon SES API, see Configuring Notifications Using the Amazon SES API (p. 109).

Configuring Notifications Using the Amazon SES Console

To configure notifications using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the navigation pane, under **Verified Senders**, click **Email Addresses** or **Domains**, depending on whether you want to configure bounce, complaint, or delivery notifications for an email address or domain.

3. In the list of verified senders, click the email address or domain for which you want to configure notifications.
4. In the details pane of the verified sender, expand **Notifications**.
5. Click **Edit Configuration**.
6. In the Edit Notification Configuration dialog box, specify the existing Amazon SNS topics you want to use for bounces, complaints, and/or deliveries, or create a new Amazon SNS topic.

   Important
   The Amazon SNS topics you use for bounce, complaint, and delivery notifications must be within the same AWS region in which you are using Amazon SES.

   You can choose to publish bounce, complaint, and delivery notifications to the same Amazon SNS topic or to different Amazon SNS topics. If you want to use an Amazon SNS topic that you do not own, you must configure your AWS Identity and Access Management (IAM) policy to allow publishing from the Amazon Resource Name (ARN) of the Amazon SNS topic. For information about how to control access to your Amazon SNS topic through the use of access policies, see Managing Access to Your Amazon SNS Topics.
7. If you configure Amazon SNS topics for both bounces and complaints for an identity, you can disable email notifications of bounces and complaints entirely by selecting **Disabled** for **Email Feedback Forwarding**. Delivery notifications are optional and available only through Amazon SNS.
8. Click **Save Config** to save your notification configuration.

   Note
   Changes made to your settings on this page might take a few minutes to take effect.

Once you have configured your settings, you will start receiving bounce, complaint, and/or delivery notifications to your Amazon SNS topic(s). These notifications will be in the JavaScript Object Notation (JSON) format and will follow the structure described in Amazon SNS Notification Contents for Amazon SES (p. 110).

You will be charged standard Amazon SNS rates for bounce, complaint, and delivery notifications. For more information, see the Amazon SNS pricing page.

Note
If an attempt to publish to your Amazon SNS topic fails because the topic has been deleted or your AWS account no longer has permissions to publish to it, the Amazon SES configuration for that topic for the sending identity will be deleted, bounce and complaint notifications through email will be re-enabled for that identity, and you will be notified of the change through email. If you have multiple identities configured to use that topic, each identity will have its topic configuration changed when each identity experiences a failure to publish to that topic.

Configuring Notifications Using the Amazon SES API

You can also configure bounce, complaint, and delivery notifications with the Amazon SES API. The following actions are available:

- `SetIdentityNotificationTopic`
- `SetIdentityFeedbackForwardingEnabled`
- `GetIdentityNotificationAttributes`

You can use these API actions to write a customized front-end application for notifications. For a complete description of the API actions related to notifications, see the Amazon Simple Email Service API Reference.

Amazon SNS Notification Contents for Amazon SES

Bounce, complaint, and delivery notifications are published to Amazon Simple Notification Service (Amazon SNS) topics in JavaScript Object Notation (JSON) format. The top-level JSON object contains a `notificationType` string, a `mail` object, and either a `bounce` object, a `complaint` object, or a `delivery` object.

See the following sections for descriptions of the different types of objects:

- Top-level JSON object (p. 110)
- `mail` object (p. 111)
- `bounce` object (p. 112)
- `complaint` object (p. 114)
- `delivery` object (p. 116)

The following are some important notes about the contents of Amazon SNS notifications for Amazon SES:

- For a given notification type, you might receive one Amazon SNS notification for multiple recipients, or you might receive a single Amazon SNS notification per recipient. Your code should be able to parse the Amazon SNS notification and handle both cases; Amazon SES does not make ordering or batching guarantees for notifications sent through Amazon SNS. However, different Amazon SNS notification types (for example, bounces and complaints) will never be combined into a single notification.
- You might receive multiple types of Amazon SNS notifications for one recipient. For example, the receiving mail server might accept the email (triggering a delivery notification), but after processing the email, the receiving mail server might determine that the email actually results in a bounce (triggering a bounce notification). However, these will always be separate notifications because they are different notification types.
- Amazon SES reserves the right to add additional fields to the notifications. As such, applications that parse these notifications must be flexible enough to handle unknown fields.

Top-Level JSON Object

The top-level JSON object in an Amazon SES notification contains the following fields.

Field Name	Description
`notificationType`	A string that holds the type of notification represented by the JSON object. Possible values are `Bounce`, `Complaint`, or `Delivery`.
`mail`	A JSON object that contains information about the original mail to which the notification pertains. For more information, see Mail Object (p. 111).
`bounce`	This field is present only if the `notificationType` is `Bounce` and contains a JSON object that holds information about the bounce. For more information, see Bounce Object (p. 112).
`complaint`	This field is present only if the `notificationType` is `Complaint` and contains a JSON object that holds information about the complaint. For more information, see Complaint Object (p. 114).

Field Name	Description
`delivery`	This field is present only if the `notificationType` is `Delivery` and contains a JSON object that holds information about the delivery. For more information, see Delivery Object (p. 116).

Mail Object

Each bounce, complaint, or delivery notification contains information about the original email in the `mail` object. The JSON object that contains information about a `mail` object has the following fields.

Field Name	Description
`timestamp`	The time at which the original message was sent (in ISO8601 format).
`messageId`	A unique ID for the original message. This is the ID that was returned to you by Amazon SES when you sent the original message.
`source`	The email address from which the original message was sent (the envelope MAIL FROM address).
`sourceArn`	The Amazon Resource Name (ARN) of the identity that was used to send the email. In the case of sending authorization, the `sourceArn` is the ARN of the identity that the identity owner authorized the delegate sender to use to send the email. For more information about sending authorization, see Using Sending Authorization (p. 134).
`sendingAccountId`	The AWS account ID of the account that was used to send the email. In the case of sending authorization, the `sendingAccountId` is the delegate sender's account ID.
`destination`	A list of email addresses that were recipients of the original mail.

The following is an example of a `mail` object.

```
{
   "timestamp":"2012-05-25T14:59:38.623-07:00",
   "messageId":"000001378603177f-7a5433e7-8edb-42ae-af10-f0181f34d6ee-000000",

   "source":"sender@example.com",
   "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",
   "sendingAccountId":"123456789012",
   "destination":[
      "recipient1@example.com",
      "recipient2@example.com",
      "recipient3@example.com",
      "recipient4@example.com"
   ]
}
```

Bounce Object

The JSON object that contains information about bounces will always have the following fields.

Field Name	Description
`bounceType`	The type of bounce, as determined by Amazon SES. For more information, see Bounce Types (p. 113).
`bounceSubType`	The subtype of the bounce, as determined by Amazon SES. For more information, see Bounce Types (p. 113).
`bouncedRecipients`	A list that contains information about the recipients of the original mail that bounced. For more information, see Bounced Recipients (p. 113).
`timestamp`	The date and time at which the bounce was sent (in ISO8601 format). Note that this is the time at which the notification was sent by the ISP, and not the time at which it was received by Amazon SES.
`feedbackId`	A unique ID for the bounce.

Optionally, if a delivery status notification (DSN) was attached to the bounce, the following field may also be present.

Field Name	Description
`reportingMTA`	The value of the `Reporting-MTA` field from the DSN. This is the value of the Message Transfer Authority (MTA) that attempted to perform the delivery, relay, or gateway operation described in the DSN.

The following is an example of a `bounce` object.

```
{
   "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":"2012-05-25T14:59:38.605-07:00",
   "feedbackId":"000001378603176d-5a4b5ad9-6f30-4198-a8c3-b1eb0c270a1d-000000"
}
```

Bounced Recipients

A bounce notification may pertain to a single recipient or to multiple recipients. The `bouncedRecipients` field holds a list of objects—one per recipient to whom the bounce notification pertains—and will always contain the following field.

Field Name	Description
`emailAddress`	The email address of the recipient. If a DSN is available, this is the value of the `Final-Recipient` field from the DSN.

Optionally, if a DSN is attached to the bounce, the following fields may also be present.

Field Name	Description
`action`	The value of the `Action` field from the DSN. This indicates the action performed by the Reporting-MTA as a result of its attempt to deliver the message to this recipient.
`status`	The value of the `Status` field from the DSN. This is the per-recipient transport-independent status code that indicates the delivery status of the message.
`diagnosticCode`	The status code issued by the reporting MTA. This is the value of the `Diagnostic-Code` field from the DSN. This field may be absent in the DSN (and therefore also absent in the JSON).

The following is an example of an object that might be in the `bouncedRecipients` list.

```
{
    "emailAddress": "recipient@example.com",
    "action": "failed",
    "status": "5.0.0",
    "diagnosticCode": "X-Postfix; unknown user"
}
```

Bounce Types

The following bounce types are available. We recommend that you remove the email addresses that have returned bounces marked `Permanent` from your mailing list, as we do not believe that you will be able to successfully send to them in the future. `Transient` bounces are sent to you when all retry attempts have been exhausted and will no longer be retried. You may be able to successfully resend to an address that initially resulted in a `Transient` bounce.

Note
Amazon SES only reports hard bounces and soft bounces that will no longer be retried by Amazon SES. In other words, your recipient did not receive your email message, and Amazon SES will not try to resend it.

bounceType	bounceSubType	Description
`Undetermined`	`Undetermined`	Amazon SES was unable to determine a specific bounce reason.
`Permanent`	`General`	Amazon SES received a general hard bounce and recommends that you remove the recipient's email address from your mailing list.
`Permanent`	`NoEmail`	Amazon SES received a permanent hard bounce because the target email address does not exist. It is recommended that you remove that recipient from your mailing list.
`Permanent`	`Suppressed`	Amazon SES has suppressed sending to this address because it has a recent history of bouncing as an invalid address. For information about how to remove an address from the suppression list, see Removing an Email Address from the Amazon SES Suppression List (p. 187).
`Transient`	`General`	Amazon SES received a general bounce. You may be able to successfully retry sending to that recipient in the future.
`Transient`	`MailboxFull`	Amazon SES received a mailbox full bounce. You may be able to successfully retry sending to that recipient in the future.
`Transient`	`MessageTooLarge`	Amazon SES received a message too large bounce. You may be able to successfully retry sending to that recipient if you reduce the message size.
`Transient`	`ContentRejected`	Amazon SES received a content rejected bounce. You may be able to successfully retry sending to that recipient if you change the message content.
`Transient`	`AttachmentRejected`	Amazon SES received an attachment rejected bounce. You may be able to successfully retry sending to that recipient if you remove or change the attachment.

Complaint Object

The JSON object that contains information about complaints has the following fields.

Field Name	Description
`complainedRecipients`	A list that contains information about recipients that may have been responsible for the complaint. For more information, see Complained Recipients (p. 115).

Field Name	Description
`timestamp`	The date and time at which the bounce was sent (in ISO8601 format). Note that this is the time at which the notification was sent by the ISP, and not the time at which it was received by Amazon SES.
`feedbackId`	A unique ID for the complaint.

Further, if a feedback report is attached to the complaint, the following fields may be present.

Field Name	Description
`userAgent`	The value of the `User-Agent` field from the feedback report. This indicates the name and version of the system that generated the report.
`complaintFeedbackType`	The value of the `Feedback-Type` field from the feedback report received from the ISP. This contains the type of feedback.
`arrivalDate`	The value of the `Arrival-Date` or `Received-Date` field from the feedback report (in ISO8601 format). This field may be absent in the report (and therefore also absent in the JSON).

The following is an example of a `complaint` object.

```
{
   "userAgent":"Comcast Feedback Loop (V0.01)",
   "complainedRecipients":[
      {
         "emailAddress":"recipient1@example.com"
      }
   ],
   "complaintFeedbackType":"abuse",
   "arrivalDate":"2009-12-03T04:24:21.000-05:00",
   "timestamp":"2012-05-25T14:59:38.623-07:00",
   "feedbackId":"000001378603177f-18c07c78-fa81-4a58-9dd1-fedc3cb8f49a-000000"
}
```

Complained Recipients

The `complainedRecipients` field contains a list of recipients that may have submitted the complaint.

> **Important**
> Since most ISPs redact the email address of the recipient who submitted the complaint from their complaint notification, this list contains information about recipients who might have sent the complaint, based on the recipients of the original message and the ISP from which we received the complaint. Amazon SES performs a lookup against the original message to determine this recipient list.

JSON objects in this list contain the following fields.

Field Name	Description
emailAddress	The email address of the recipient.

The following is an example of a Complained Recipient object.

```
{ "emailAddress": "recipient1@example.com" }
```

Note
Because of this behavior, you can be more certain that you know which email address complained about your message if you limit your sending to one message per recipient (rather than sending one message with 30 different email addresses in the bcc line).

Complaint Types

You may see the following complaint types in the `complaintFeedbackType` field (as assigned by the reporting ISP according to http://www.iana.org/assignments/marf-parameters/marf-parameters.xml#marf-parameters-2):

- `abuse`—Indicates unsolicited email or some other kind of email abuse.
- `auth-failure`—Email authentication failure report.
- `fraud`—Indicates some kind of fraud or phishing activity.
- `not-spam`—Indicates that the entity providing the report does not consider the message to be spam. This may be used to correct a message that was incorrectly tagged or categorized as spam.
- `other`—Indicates any other feedback that does not fit into other registered types.
- `virus`—Reports that a virus is found in the originating message.

Delivery Object

The JSON object that contains information about deliveries will always have the following fields.

Field Name	Description
timestamp	The time Amazon SES delivered the email to the recipient's mail server (in ISO8601 format).
processingTimeMillis	The time in milliseconds between when Amazon SES accepted the request from the sender to passing the message to the recipient's mail server.
recipients	A list of the intended recipients of the email to which the delivery notification applies.
smtpResponse	The SMTP response message of the remote ISP that accepted the email from Amazon SES. This message will vary by email, by receiving mail server, and by receiving ISP.
reportingMTA	The host name of the Amazon SES mail server that sent the mail.

The following is an example of a `delivery` object.

```
{
   "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"
}
```

Amazon SNS Notification Examples for Amazon SES

The following sections provide examples of the three types of notifications:

- For bounce notification examples, see Amazon SNS Bounce Notification Examples (p. 117).
- For complaint notification examples, see Amazon SNS Complaint Notification Examples (p. 118).
- For delivery notification examples, see Amazon SNS Delivery Notification Example (p. 119).

Amazon SNS Bounce Notification Examples

The following is an example of a bounce notification without a delivery status notification (DSN):

```
   {
      "notificationType":"Bounce",
      "bounce":{
         "bounceType":"Permanent",
         "bounceSubType": "General",
         "bouncedRecipients":[
            {
               "emailAddress":"recipient1@example.com"
            },
            {
               "emailAddress":"recipient2@example.com"
            }
         ],
         "timestamp":"2012-05-25T14:59:38.237-07:00",
         "feedbackId":"00000137860315fd-869464a4-8680-4114-98d3-716fe35851f9-
000000"
      },
      "mail":{
         "timestamp":"2012-05-25T14:59:38.237-07:00",
         "messageId":"00000137860315fd-34208509-5b74-41f3-95c5-22c1edc3c924-
000000",
         "source":"email_1337983178237@amazon.com",
        "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",

         "sendingAccountId":"123456789012",
         "destination":[
            "recipient1@example.com",
            "recipient2@example.com",
            "recipient3@example.com",
            "recipient4@example.com"
         ]
      }
   }
```

The following is an example of a bounce notification that has a DSN:

```
   {
       "notificationType":"Bounce",
       "bounce":{
          "bounceType":"Permanent",
          "reportingMTA":"dns; email.example.com",
          "bouncedRecipients":[
             {
                "emailAddress":"username@example.com",
                "status":"5.1.1",
                "action":"failed",
                "diagnosticCode":"smtp; 550 5.1.1 <username@example.com>...
User"
             }
          ],
          "bounceSubType":"General",
          "timestamp":"2012-06-19T01:07:52.000Z",
          "feedbackId":"00000138111222aa-33322211-cccc-cccc-cccc-ddddaaaa068a-
000000"
       },
       "mail":{
          "timestamp":"2012-06-19T01:05:45.000Z",
          "source":"sender@example.com",
        "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",

          "sendingAccountId":"123456789012",
          "messageId":"00000138111222aa-33322211-cccc-cccc-cccc-ddddaaaa0680-
000000",
          "destination":[
             "username@example.com"
          ]
       }
    }
```

Amazon SNS Complaint Notification Examples

The following is an example of a complaint notification without a feedback report:

```
   {
      "notificationType":"Complaint",
      "complaint":{
         "complainedRecipients":[
            {
               "emailAddress":"recipient1@example.com"
            }
         ],
         "timestamp":"2012-05-25T14:59:38.613-07:00",
         "feedbackId":"0000013786031775-fea503bc-7497-49e1-881b-a0379bb037d3-
000000"
      },
      "mail":{
         "timestamp":"2012-05-25T14:59:38.613-07:00",
         "messageId":"0000013786031775-163e3910-53eb-4c8e-a04a-f29debf88a84-
000000",
         "source":"email_1337983178613@amazon.com",
        "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",

         "sendingAccountId":"123456789012",
         "destination":[
            "recipient1@example.com",
            "recipient2@example.com",
            "recipient3@example.com",
            "recipient4@example.com"
         ]
      }
   }
```

The following is an example of a complaint notification that has a feedback report:

```
   {
      "notificationType":"Complaint",
      "complaint":{
         "userAgent":"Comcast Feedback Loop (V0.01)",
         "complainedRecipients":[
            {
               "emailAddress":"recipient1@example.com"
            }
         ],
         "complaintFeedbackType":"abuse",
         "arrivalDate":"2009-12-03T04:24:21.000-05:00",
         "timestamp":"2012-05-25T14:59:38.623-07:00",
         "feedbackId":"000001378603177f-18c07c78-fa81-4a58-9dd1-fedc3cb8f49a-
000000"
      },
      "mail":{
         "timestamp":"2012-05-25T14:59:38.623-07:00",
         "messageId":"000001378603177f-7a5433e7-8edb-42ae-af10-f0181f34d6ee-
000000",
         "source":"email_1337983178623@amazon.com",
        "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",

         "sendingAccountId":"123456789012",
         "destination":[
            "recipient1@example.com",
            "recipient2@example.com",
            "recipient3@example.com",
            "recipient4@example.com"
         ]
      }
   }
```

Amazon SNS Delivery Notification Example

The following is an example of a delivery notification:

```
   {
      "notificationType":"Delivery",
      "mail":{
         "timestamp":"2014-05-28T22:40:59.638Z",
         "messageId":"0000014644fe5ef6-9a483358-9170-4cb4-a269-f5dcdf415321-
000000",
         "source":"sender@example.com",
        "sourceArn": "arn:aws:ses:us-west-2:888888888888:identity/example.com",
      "sendingAccountId":"123456789012",
      "destination":[
         "success@simulator.amazonses.com",
         "recipient@example.com"
      ]
   },
   "delivery":{
      "timestamp":"2014-05-28T22:41:01.184Z",
      "recipients":["success@simulator.amazonses.com"],
      "processingTimeMillis":546,
      "reportingMTA":"a8-70.smtp-out.amazonses.com",
      "smtpResponse":"250 ok:  Message 64111812 accepted"
   }
}
```

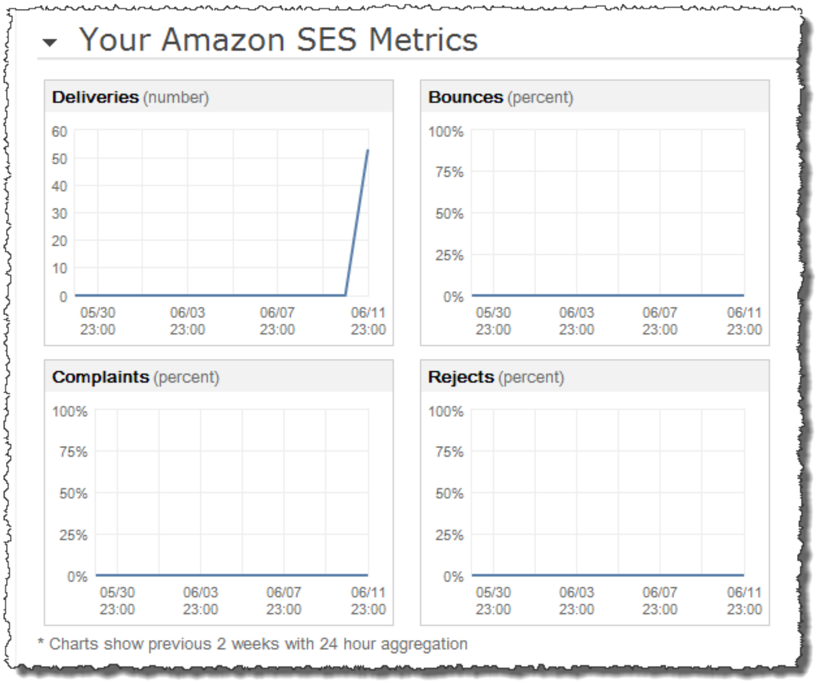
Monitoring Your Amazon SES Usage Statistics

You can monitor your usage statistics by using the Amazon SES console or through the Amazon SES API, whether by calling the Query (HTTPS) interface directly or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell.

Monitoring Your Usage Statistics Using the Amazon SES Console

The following procedure shows you how to view your usage statistics using the Amazon SES console.

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the Navigation pane, click **Dashboard**. Your usage statistics are shown under **Your Amazon SES Metrics**.



* Charts show previous 2 weeks with 24 hour aggregation

3. To view trend data for any metric, double-click the corresponding graph.
4. To update the display, click the **Refresh** button.

Monitoring Your Usage Statistics Using the Amazon SES API

The Amazon SES API provides the `GetSendStatistics` action, which returns information about your service usage. We recommend that you use `GetSendStatistics` on a regular basis, so that you can monitor your sending activity and make adjustments as needed.

When you call `GetSendStatistics`, you will receive a list of data points representing the last two weeks of your sending activity. Each data point in this list represents 15 minutes of activity and contains the following information for that period:

- Bounces (hard bounces only)
- Complaints
- Delivery attempts
- Rejected send attempts
- Timestamp

Note
For a complete description of `GetSendStatistics`, go to the Amazon Simple Email Service API Reference.

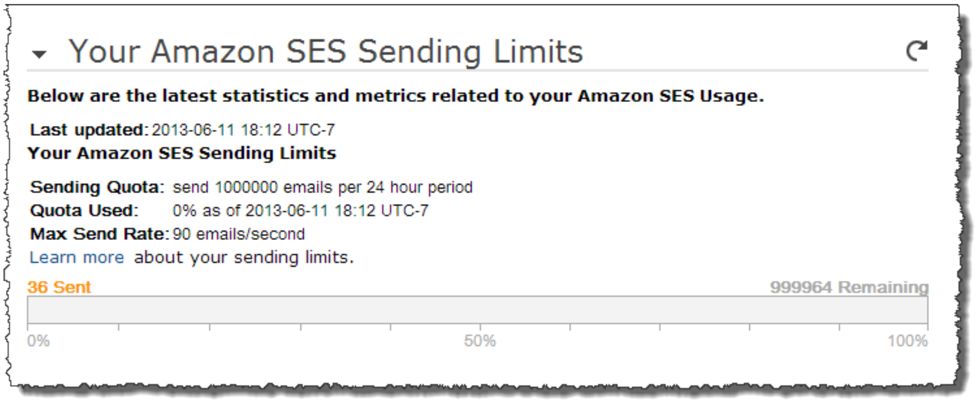
Monitoring Your Amazon SES Sending Limits

You can monitor your sending limits by using the Amazon SES console or through the Amazon SES API, whether by calling the Query (HTTPS) interface directly or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell.

Monitoring Your Sending Limits Using the Amazon SES Console

The following procedure shows you how to view your sending limits using the Amazon SES console.

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the Navigation pane, click **Dashboard**. Your sending limits are shown under **Your Amazon SES Sending Limits**.



Your Amazon SES Sending Limits

Below are the latest statistics and metrics related to your Amazon SES Usage.

Last updated: 2013-06-11 18:12 UTC-7
Your Amazon SES Sending Limits

Sending Quota: send 1000000 emails per 24 hour period
Quota Used: 0% as of 2013-06-11 18:12 UTC-7
Max Send Rate: 90 emails/second
Learn more about your sending limits.

3. To update the display, click **Refresh**.

Monitoring Your Sending Limits Using the Amazon SES API

The Amazon SES API provides the `GetSendQuota` action, which returns your sending limits. When you call `GetSendQuota` action, you receive the following information:

- Number of emails you have sent during the past 24 hours
- Sending quota for the current 24-hour period
- Maximum send rate

Note
For a complete description of `GetSendQuota`, go to the Amazon Simple Email Service API Reference.

Managing Your Amazon SES Sending Limits

Your Amazon Simple Email Service (Amazon SES) account has a set of sending limits to regulate the number of email messages that you can send and the rate at which you can send them. Sending limits benefit all Amazon SES customers because they help to maintain the trusted relationship between Amazon SES and Internet service providers (ISPs). Sending limits help you to gradually ramp up your sending activity and decrease the likelihood that ISPs will block your emails because of sudden, unexpected spikes in your email sending volume or rate.

The following are Amazon SES sending limits:

- **Sending Quota**—The maximum number of emails that you can send in a 24-hour period. The sending quota reflects a rolling time period. Every time you try to send an email, Amazon SES checks how many emails you sent in the previous 24 hours. As long as the total number of emails that you have sent is less than your quota, your send request will be accepted and your email will be sent. If you have already sent your full quota, your send request will be rejected with a throttling exception. For example, if your sending quota is 50,000, and you sent 15,000 emails in the previous 24 hours, then you can send another 35,000 emails right away. If you have already sent 50,000 emails in the previous 24 hours, you will not be able to send more emails until some of the previous sending rolls out of its 24-hour window.
- **Maximum Send Rate**—The maximum number of emails that Amazon SES can accept from your account per second. You can exceed this limit for short bursts, but not for a sustained period of time.

  Note
  The rate at which Amazon SES accepts your messages might be less than the maximum send rate.

Your Amazon SES sending limits for each AWS region are separate. For information about using Amazon SES in multiple AWS regions, see Regions and Amazon SES (p. 173).

When you are in the Amazon SES sandbox, your sending quota is 200 messages per 24-hour period and your maximum sending rate is one message per second. To increase your sending limits, you need to submit an SES Sending Limits Increase case. For more information, see Moving Out of the Amazon SES Sandbox (p. 45). After you have moved out of the sandbox and start sending emails, you can increase your sending limits further by submitting another SES Sending Limits Increase case, as discussed in Increasing Your Amazon SES Sending Limits (p. 125).

Note

Sending limits are based on recipients rather than on messages. For example, an email that has 10 recipients counts as 10 against your quota. However, we do not recommend that you send an email to multiple recipients in one call to `SendEmail` because if the call to Amazon SES fails (for example, the request is improperly formatted), the entire email will be rejected and none of the recipients will get the intended email. We recommend that you call `SendEmail` once for every recipient.

- To increase your sending limits, see Increasing Your Amazon SES Sending Limits (p. 125).
- For information about the errors your application receives when you reach your sending limits, see What Happens When You Reach Your Amazon SES Sending Limits (p. 127).
- To monitor your sending limits by using the Amazon SES console or the Amazon SES API, see Monitoring Your Amazon SES Sending Limits (p. 122).

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Increasing Your Amazon SES Sending Limits

Once your account is out of the sandbox, your sending limits will increase if you are sending high-quality content and we detect that your utilization is approaching your current limits. Often, the system will automatically increase your quota before you actually need it, and no further action is needed.

If your existing quota is not adequate for your needs and the system has not automatically increased your quota, you can open an SES Sending Limits Increase case in Support Center.

For a list of the information that you will need when you open the case, see Opening an SES Sending Limits Increase Case (p. 126).

Important

Plan ahead. Be aware of your sending limits and try to stay within them. If you anticipate needing a higher quota than the system has allocated automatically, please open an SES Sending Limits Increase case well prior to the date that you need the higher quota.

Important

If you anticipate needing to send more than one million emails per day, you must open an SES Sending Limits Increase case.

For Amazon SES to increase your quota, make sure that you use the following guidelines:

- **Send high-quality content**—Send content that recipients want and expect. For recommendations about how to send high-quality emails, see the Amazon Simple Email Service Email Sending Best Practices white paper.
- **Send real production content**—Send your actual production email. This enables Amazon SES to accurately evaluate your sending patterns, and verify that you are sending high-quality content.
- **Send near your current quota**—If your volume stays close to your quota without exceeding it, Amazon SES can detect this usage pattern and automatically increase your quota.
- **Have low bounce and complaint rates**—Try to minimize the numbers of bounces and complaints. Having high numbers of bounces and complaints can adversely affect your sending limits.

Important

Test emails that you send to your own email addresses may adversely affect your bounce and complaint metrics, or appear as low-quality content to our filters. Whenever possible, use the Amazon SES mailbox simulator to test your system. Emails that are sent to the mailbox simulator do not count toward your sending metrics or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 158).

For information about opening an SES Sending Limits Increase case, see Opening an SES Sending Limits Increase Case (p. 126).

Opening an SES Sending Limits Increase Case

To apply for higher sending limits for Amazon SES, you need to open a case in Support Center by using the following instructions.

To request higher sending limits

1. Log into the AWS Management Console.
2. Go to SES Sending Limits Increase. Alternatively, you can go to Support Center, click **Create Case**, click **Service Limit Increase**, and then select **SES Sending Limits** as the limit type.
3. In the form, provide the following information:

   - **Region:** Select the AWS region for which you are requesting a sending limit increase. Note that your Amazon SES sandbox status and sending limits are separate for each AWS region. For more information, see Regions and Amazon SES (p. 173).
   - **Limit:** Select *Desired Daily Sending Quota* or *Desired Maximum Send Rate*. Sending limits are described in Managing Your Amazon SES Sending Limits (p. 124).

     Note
     The rate at which Amazon SES accepts your messages might be less than the maximum send rate.
   - **New limit value:** Enter the amount you are requesting. **Be sure to only request the amount you think you'll need.** Keep in mind that you are not guaranteed to receive the amount you request, and the higher the limit you request, the more justification you will need to be considered for that amount.
   - **Mail type:** Select *Transactional*, *System Notifications*, *Subscription*, *Marketing*, or *Other*.
   - **Website URL.** Provide a link to your website. Although it isn't required, we highly recommend that you provide one if you have it, because it helps us evaluate your request.
   - **My email-sending complies with the AWS Service Terms and AWS Acceptable Use Policy (AUP).** Select *Yes* or *No*.
   - **I only send to recipients who have specifically requested my mail.** Select *Yes* or *No*. For tips on how to send high-quality mail and keep your recipient list clean, see Obtaining and Maintaining Your Recipient List (p. 178) and the Amazon Simple Email Service Email Sending Best Practices white paper.
   - **I have a process to handle bounces and complaints.** Select *Yes* or *No*. For information on how to monitor and handle bounces and complaints, see Processing Bounces and Complaints (p. 178).
   - **Use Case Description.** Explain your situation in as much detail as possible. For example, describe the type of emails you are sending and how email-sending fits into your business. The more information you can provide that indicates that you are sending high-quality emails to recipients who want and expect it, the more likely we are to approve your request. The higher the limit value you are requesting, the more detail you should provide.

   We will respond to the case after reviewing your request. Please allow one business day for processing.

What Happens When You Reach Your Amazon SES Sending Limits

If you attempt to send an email after you have reached your sending quota or maximum send rate, you will encounter a throttling error and the email will be dropped. The way you observe the error depends on whether you are calling the Amazon SES API directly, or accessing Amazon SES through the SMTP interface. The following sections describe both scenarios.

> **Note**
> You can send an email to multiple recipients as long as you have at least one email left before you reach your sending rate limit. Then, you have to wait until you build up the corresponding amount of sending rate quota before you can send again. For example, if your sending rate limit is one email per second, then you will be able to send an email with 10 recipients once every 10 seconds. However, we do not recommend that you send an email to multiple recipients in one call to `SendEmail`.

Reaching Sending Limits with the Amazon SES API

If your application attempts to send an email beyond your sending limits, the application will encounter a throttling error. The following are types of throttling errors that you might see:

- Daily message quota exceeded
- Maximum sending rate exceeded

A throttling error might occur because of incorrect predictions of email volume, or bursts of transactional email that are higher than expected. To handle a throttling error, program your application to wait for a random interval of between 0 and 10 minutes, and then retry the send request.

Reaching Sending Limits with SMTP

If you reach your sending limits when you are accessing Amazon SES through the SMTP interface, your SMTP client will receive one of the following errors:

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

However, SMTP clients handle these errors in various ways. For example, if you use Microsoft Outlook as your email client, and you attempt to send past your sending quota, you get a Send/Receive error that displays the following text in the status pane at the bottom of the Outlook client window:

```
Task 'andrew@example.net- Sending' reported error (0x800CCC7F): 'Establishing
an encrypted connection to your outgoing (SMTP) server failed. If this problem
continues, contact your server administrator or Internet service provider (ISP).
The server responded: 454 Throttling failure: Daily message quota exceeded.'
```

The way in which these errors are handled depends on the SMTP client that you use; some SMTP clients may not display the error code at all.

Controlling User Access to Amazon SES

You can use AWS Identity and Access Management (IAM) with Amazon Simple Email Service (Amazon SES) to specify which Amazon SES API actions an IAM user, group, or role can perform. (In this topic we refer to these entities collectively as *user*.) You can also control which email addresses the user can use for the "From", recipient, and "Return-Path" addresses of emails.

For example, you can create an IAM policy that allows users in your organization to send email, but not perform administrative actions such as checking sending statistics. As another example, you can write a policy that allows a user to send emails through Amazon SES from your account, but only if they use a specific "From" address.

To use IAM, you define an IAM policy, which is a document that explicitly defines permissions, and attach the policy to a user. To learn how to create IAM policies, see the IAM documentation. Other than applying the restrictions you set in your policy, there are no changes to how users interact with Amazon SES or in how Amazon SES carries out requests.

This topic is about controlling the access of users within the same AWS account. If you want to enable other AWS accounts access to your Amazon SES identities, see Using Sending Authorization with Amazon SES (p. 134).

> **Note**
> If you are looking for information about how to generate Amazon SES SMTP credentials for an existing IAM user, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

Creating IAM Policies for Access to Amazon SES

This section explains how you can use IAM policies specifically with Amazon SES. To learn how to create IAM policies in general, see the IAM documentation.

There are three reasons you might use IAM with Amazon SES:

- To restrict the email-sending action.
- To restrict the "From", recipient, and "Return-Path" addresses of the emails that the user sends.
- To control general aspects of API usage such as the time period during which a user is permitted to call the APIs that they are authorized to use.

Restricting the Action

To control which Amazon SES actions a user can perform, you use the `Action` element of an IAM policy. You can set the `Action` element to any Amazon SES API action by prefixing the API name with the lowercase string `ses:`. For example, you can set the `Action` to `ses:SendEmail`, `ses:GetSendStatistics`, or `ses:*` (for all actions).

Then, depending on the `Action`, specify the `Resource` element as follows:

If the `Action` element only permits access to email-sending APIs (that is, `ses:SendEmail` and/or `ses:SendRawEmail`):

- To allow the user to send from any identity in your AWS account, set `Resource` to `*`
- To limit the identities that the user can send from, set `Resource` to the ARN(s) of the identities that you are permitting the user to use.

If the `Action` element permits access to all APIs:

- If you do not want to limit the identities that the user can send from, set `Resource` to `*`
- If you **do** want to limit the identities that the user can send from, you need to create two policies (or two statements within one policy):
  - One with `Action` set to an explicit list of the permitted non-email-sending APIs and `Resource` set to `*`
  - One with `Action` set to one of the email-sending APIs (`ses:SendEmail` and/or `ses:SendRawEmail`), and `Resource` set to the ARN(s) of the identities you are permitting the user to use.

For a list of available Amazon SES actions, see the Amazon Simple Email Service API Reference. If the IAM user will be using the SMTP interface, you must allow access to `ses:SendRawEmail` at a minimum.

Restricting Email Addresses

If you want to restrict the user to specific email addresses, you can use a `Condition` block. In the `Condition` block, you specify conditions by using condition keys as described in the IAM documentation. By using condition keys, you can control the following email addresses:

> **Note**
> These email address condition keys only apply to email-sending actions (`SendEmail` and `SendRawEmail`).

Condition Key	Description
`ses:Recipients`	Restricts the recipient addresses, which include the To:, "CC", and "BCC" addresses.
`ses:FromAddress`	Restricts the "From" address.
`ses:FromDisplayName`	Restricts the "From" address that is used as the display name.
`ses:FeedbackAddress`	Restricts the "Return-Path" address, which is the address where bounces and complaints can be sent to you by email feedback forwarding. For information about email feedback forwarding, see Amazon SES Notifications Through Email (p. 106).

Restricting General API Usage

By using AWS-wide keys in conditions, you can restrict access to Amazon SES based on aspects such as the date and time that user is permitted access to APIs. Amazon SES implements only the following AWS-wide policy keys:

- `aws:CurrentTime`
- `aws:EpochTime`
- `aws:SecureTransport`
- `aws:SourceIp`
- `aws:UserAgent`

For more information about these keys, see the IAM documentation.

Example IAM Policies for Amazon SES

This topic provides examples of policies that permit a user access to Amazon SES, but only under certain conditions.

Allowing Full Access to All Amazon SES Actions

The following policy allows a user to call any Amazon SES action.

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

Allowing Access to Email-Sending Actions Only

The following policy permits a user to send email using Amazon SES, but does not permit the user to perform administrative actions such as accessing Amazon SES sending statistics.

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

Restricting the Time Period of Sending

The following policy permits a user to call Amazon SES email-sending APIs only during the month of September 2015.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
       "Effect": "Allow",
       "Action": ["ses:SendEmail", "ses:SendRawEmail"],
       "Resource":"*",
       "Condition": {
          "DateGreaterThan": {
            "aws:CurrentTime":"2015-08-31T12:00Z"
           },
          "DateLessThan": {
            "aws:CurrentTime":"2015-10-01T12:00Z"
          }
       }
     }
    ]
 }
```

Restricting the Recipient Addresses

The following policy permits a user to call the Amazon SES email-sending APIs, but only to recipient addresses in domain *example.com*.

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
       "Effect": "Allow",
       "Action": ["ses:SendEmail", "ses:SendRawEmail"],
       "Resource":"*",
           "Condition": {
             "ForAllValues:StringLike": {
               "ses:Recipients": ["*@example.com"]
             }
         }
       }
    ]
 }
```

Restricting the "From" Address

The following policy permits a user to call the Amazon SES email-sending APIs, but only if the "From" address is *marketing@example.com*.

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
       "Effect": "Allow",
       "Action": ["ses:SendEmail", "ses:SendRawEmail"],
       "Resource":"*",
           "Condition": {
             "StringEquals": {
               "ses:FromAddress": "marketing@example.com"
             }
         }
       }
    ]
}
```

Restricting the Display Name of the Email Sender

The following policy permits a user to call the Amazon SES email-sending APIs, but only if the display name of the "From" address includes *Marketing*.

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

Restricting the Destination of Bounce and Complaint Feedback

The following policy permits a user to call the Amazon SES email-sending APIs, but only if the "Return-Path" of the email is set to *feedback@example.com*.

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
      "Effect": "Allow",
      "Action": ["ses:SendEmail", "ses:SendRawEmail"],
      "Resource":"*",
      "Condition": {
        "StringEquals": {
          "ses:FeedbackAddress": "feedback@example.com"
         }
       }
    }
   ]
}
```

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Using Sending Authorization with Amazon SES

Amazon Simple Email Service (Amazon SES) enables you to authorize other users to send emails from your identities on your behalf. This feature, called *sending authorization*, lets you maintain control over your identities so that you can change or revoke the permissions at any time. For example, as a business owner, you might use sending authorization to enable an email marketing company to send marketing emails from an email address under your domain name.

If you want to authorize someone to send emails on your behalf, then you are an *identity owner*. If you are an identity owner, we recommend that you read the following sections:

- Overview of Sending Authorization (p. 134)
- Sending Authorization Policies (p. 136)
- Sending Authorization Policy Examples (p. 140)
- Identity Owner Tasks (p. 145)

If you have been authorized to send emails on behalf of someone else, then you are a *delegate sender*. If you are a delegate sender, we recommend that you read the following sections:

- Overview of Sending Authorization (p. 134)
- Delegate Sender Tasks (p. 151)

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Overview of Amazon SES Sending Authorization

This topic provides an overview of the sending authorization process and then explains how Amazon SES email-sending features, such as sending limits and notifications, work with sending authorization.

We use the following terminology:

- An *identity* is an email address or domain that Amazon SES users use to send email.

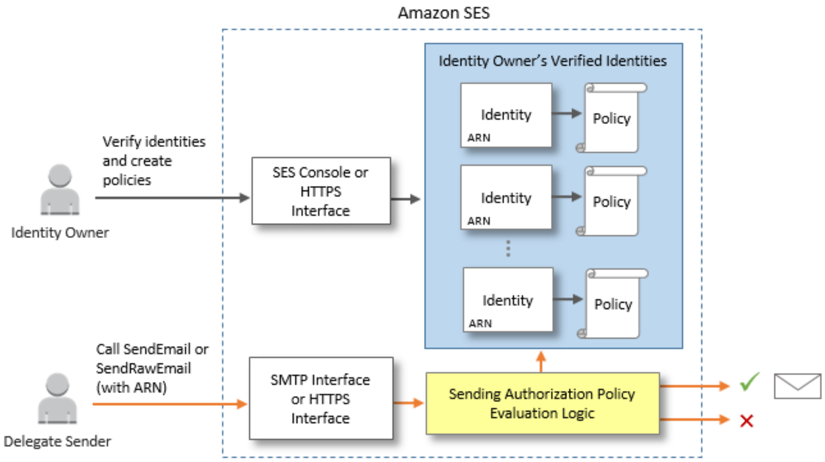
- An *identity owner* is an Amazon SES user who has verified ownership of an identity by using the procedure described in Verifying Email Addresses and Domains (p. 37).
- A *delegate sender* is an entity that is authorized to send emails from an identity it does not own. An AWS account or an Identity and Access Management (IAM) user can have this *cross-account* authority.
- A *sending authorization policy* is a document that you attach to an identity to specify who may send for that identity and under which conditions.
- An *Amazon Resource Name (ARN)* is a standardized way to uniquely identify an AWS resource across all AWS services. In the case of sending authorization, the resource is the identity that the identity owner wants the delegate sender to use. An example of an ARN is *arn:aws:ses:us-west-2:123456789012:identity/example.com*.

Sending Authorization Process

Sending authorization is based on sending authorization policies. If you want to enable a delegate sender to send on your behalf, you create a sending authorization policy and associate the policy to your identity by using the Amazon SES console or the Amazon SES API. When the delegate sender attempts to send an email through Amazon SES on your behalf, the delegate sender passes the ARN of your identity in the request or in the header of the email.

When Amazon SES receives the request to send the email, it checks your identity's policy (if present) to determine if you have authorized the delegate sender to send on the identity's behalf. If the delegate sender is authorized, Amazon SES accepts the email; otherwise, Amazon SES returns an error message.

The following diagram shows the high-level relationship between sending authorization concepts:


Amazon SES
Identity Owner's Verified Identities
Identity
ARN
Policy
Identity
ARN
Policy
Identity
ARN
Policy
Verify identities and create policies
Identity Owner
SES Console or HTTPS Interface
Call SendEmail or SendRawEmail (with ARN)
Delegate Sender
SMTP Interface or HTTPS Interface
Sending Authorization Policy Evaluation Logic
✓
×

Step by step, the sending authorization process is as follows:

1. The identity owner verifies an identity with Amazon SES by using the Amazon SES console or the Amazon SES API. For information about the verification procedure, see Verifying Email Addresses and Domains (p. 37).
2. The delegate sender gives the identity owner the AWS account ID or the ARN of an IAM user who will do the sending.
3. The identity owner creates a sending authorization policy and attaches the policy to the identity by using the Amazon SES console or the Amazon SES API.
4. The identity owner gives the delegate sender the ARN of the identity so that the delegate sender can provide the ARN to Amazon SES at the time of email sending.

5. The identity owner and delegate sender configure bounce, complaint, and delivery notifications separately by using either the Amazon SES console or the Amazon SES API. For information about setting up notifications, see Using Notifications with Amazon SES (p. 105).
6. The delegate sender attempts to send an email through Amazon SES on behalf of the identity owner by passing the ARN of the identity owner's identity in the request or in the header of the email. The delegate sender can send the email by using either the Amazon SES SMTP interface or the Amazon SES API. Upon receiving the request, Amazon SES examines any policies that are attached to the identity, and accepts the email if the delegate sender is authorized to use the specified "From" address and "Return Path" address; otherwise, Amazon SES returns an error and does not accept the message.
7. To deauthorize the delegate sender, the identity owner simply edits the sending authorization policy or deletes the policy entirely. Either action can be performed by using the Amazon SES console or the Amazon SES API.

For more information about how the identity owner or delegate sender perform those tasks, see Identity Owner Tasks (p. 145) or Delegate Sender Tasks (p. 151), respectively.

Attribution of Email-Sending Features

It is important to understand the role of the delegate sender and the identity owner with respect to Amazon SES email-sending features such as daily sending quota, bounces and complaints, DKIM signing, feedback forwarding, and so on. The attribution is the following:

- **Sending limits**—The emails count toward the delegate sender's daily sending quota.
- **Bounces and complaints**—Bounces and complaints count toward the delegate sender's bounce and complaint metrics, and therefore the delegate sender's reputation as a sender.
- **DKIM signing**—If the identity owner has enabled Easy DKIM signing for an identity, all email sent from that identity will be DKIM-signed, including email sent by a delegate sender. Only the identity owner has control of whether the emails are DKIM-signed.
- **Notifications**—The identity owner and the delegate sender set up their own Amazon SNS notifications for bounces, complaints, and deliveries independently. However, feedback forwarding by email is only available to the identity owner.
- **Verification**—Identity owners are responsible for following the procedure in Verifying Email Addresses and Domains (p. 37) to verify that they own the email addresses and domains that they are authorizing delegate senders to use. Delegate senders do not need to verify any email addresses or domains specifically for sending authorization.
- **AWS regions**—The delegate sender must send the emails from the AWS region in which the identity owner's identity is verified. The sending authorization policy that gives permission to the delegate sender must be attached to the identity in that region.

Amazon SES Sending Authorization Policies

To enable another AWS account or Identity Access and Management (IAM) user to send email through Amazon SES on your behalf, you create a *sending authorization policy*, which is a JSON document that you attach to an identity that you own. The policy explicitly lists who you are allowing to send for that identity, and under which conditions. All senders but you and the entities you explicitly grant permissions to in the policies are denied. An identity can have no policy, one policy, or multiple policies attached to it. You can also have one policy with multiple statements to achieve the effect of multiple policies.

Policies can be very simple or very detailed for fine-grained control. For example, if you owned *example.com*, you could write a simple policy to grant AWS ID 123456789012 permission to send from that domain. A more detailed policy could specify that AWS ID 123456789012 can send email only from *user@example.com* and only within a specified date range.

Policy Structure

Each sending authorization policy is a JSON document that is attached to an identity. A policy includes:

- Optional policy-wide information at the top of the document.
- One or more individual statements, each of which describes one set of permissions.

Each statement includes the core information about a single permission. If a policy includes multiple statements, Amazon SES applies a logical OR across the statements at evaluation time. Similarly, if an identity has multiple policies attached to it, Amazon SES applies a logical OR across the policies at evaluation time.

The following example shows a simple policy that allows AWS ID 123456789012 to send email from the identity *example.com* (which is under account 888888888888) but only if the "From" address is *marketing+.*@example.com*, where .* is any string that the sender wants to add after *marketing+*.

```
{
  "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"
        }
      }
    }
  ]
}
```

You can find more sending authorization policy examples at Sending Authorization Policy Examples (p. 140).

Policy Elements

This section describes the elements contained in sending authorization policies. First we describe policy-wide elements, and then we describe elements that apply only to the statement in which they are included. We follow with a discussion of how to add conditions to your statements.

For specific information about the syntax of the elements, see Grammar of the IAM Policy Language in the *IAM User Guide*.

Policy-Wide

There are two policy-wide elements: Id and Version. The following table provides information about these elements.

Name	Description	Required	Valid Values
`Id`	Uniquely identifies the policy.	No.	Any string

Name	Description	Required	Valid Values
`Version`	Specifies the policy access language version.	No, but as a best practice, we recommend that you include this field with a value of "2012-10-17".	Any string

Statements

Sending authorization policies require at least one statement. Each statement can include the elements described in the following table.

Name	Description	Required	Valid Values
`Sid`	Uniquely identifies the statement.	No.	Any string.
`Effect`	Specifies the result that you want the policy statement to return at evaluation time.	No, although a statement without an effect is useless.	"Allow" or "Deny".
`Resource`	Specifies the identity to which the policy applies. This is the email address or domain that the identity owner is authorizing the delegate sender to use.	Yes.	An identity's ARN, as specified in the Amazon SES console.
`Principal`	Specifies the AWS account or IAM user that receives the permission in the statement.	Yes.	A valid AWS account ID or the ARN of an IAM user. For examples of the format of IAM user ARNs, see the AWS General Reference.
`Action`	Specifies the email-sending action to which the statement applies.	Yes.	"ses:SendEmail", "ses:SendRawEmail" (one or both). If you use the custom policy editor, you can also set the action to "ses:*" to encompass both APIs. If your sender will access Amazon SES through the SMTP interface, you must select "ses:SendRawEmail" at a minimum (or use "ses:*").

Name	Description	Required	Valid Values
`Condition`	Specifies any restrictions or details about the permission.	No.	See the information about conditions following this table.

Conditions

A *condition* is any restriction about the permission in the statement. The part of the statement that specifies the conditions can be the most detailed of all the parts. A *key* is the specific characteristic that is the basis for access restriction, such as the date and time of the request.

You use both conditions and keys together to express the restriction. For example, if you want to restrict the delegate sender from making requests to Amazon SES on your behalf after July 30, 2015, you use the condition called `DateLessThan`. You use the key called `aws:CurrentTime` and set it to the value `2015-07-30T00:00:00Z`.

You can use any of the AWS-wide keys listed at Available Keys described in the *IAM User Guide*, or you can use one of the following keys specific to Amazon SES:

Condition Key	Description
`ses:Recipients`	Restricts the recipient addresses, which include the To:, "CC", and "BCC" addresses.
`ses:FromAddress`	Restricts the "From" address.
`ses:FromDisplayName`	Restricts the contents of the string that is used as the "From" display name (sometimes called "friendly from"). For example, the display name of "John Doe <johndoe@example.com>" is John Doe.
`ses:FeedbackAddress`	Restricts the "Return Path" address, which is the address where bounce and complaints can be sent to you by email feedback forwarding. For information about email feedback forwarding, see Amazon SES Notifications Through Email (p. 106).

It is common to use the `StringEquals` and `StringLike` conditions with the Amazon SES keys. These conditions are for case-sensitive string matching. For `StringLike`, the values can include a multi-character match wildcard (*) or a single-character match wildcard (?) anywhere in the string. For example, the following condition specifies that the delegate sender can only send from a "From" address that starts with *invoicing* and ends with *example.com*:

```
"Condition": {
    "StringLike": {
      "ses:FromAddress": "invoicing+.*@example.com"
    }
}
```

Note

When you want to disallow access to an email address, use wildcards to ensure that you are completely preventing access to all forms of that address. For example, to disallow sending from *admin@example.com*, you can prevent access to alternatives such as *"admin"@example.com* and *admin+1@example.com* by specifying the following condition:

```
"Condition": {
    "StringNotLike": {
      "ses:FromAddress": "*admin*.example.com"
    }
 }
```

For more information about how to specify conditions, see Condition in the *IAM User Guide*.

Policy Requirements

Each policy must adhere to the following requirements:

- Each policy must include at least one statement.
- Each policy must include at least one valid principal.
- Each policy must specify one resource, and that resource must be the ARN of the identity to which the policy is attached.
- Identity owners can associate up to 20 policies with each unique identity.
- Policies must not exceed 4 kilobytes (KB).
- Policy names cannot exceed 64 characters and can only include alphanumeric characters, dashes, and underscores.

Amazon SES Sending Authorization Policy Examples

Sending authorization enables you to specify the fine-grained conditions under which you allow delegate senders to send on your behalf. The following examples show you how to write policies to control different aspects of sending:

- Specifying the Delegate Sender (p. 140)
- Restricting the "From" Address (p. 141)
- Restricting the Destination of Bounce and Complaint Feedback (p. 142)
- Restricting the Time Period of Sending (p. 142)
- Restricting the Email-Sending Action (p. 143)
- Restricting the Display Name of the Email Sender (p. 143)
- Using Multiple Statements (p. 144)

Specifying the Delegate Sender

The *principal*, which is the entity to which you are granting permission, can be an AWS account or an Identity and Access Management (IAM) user. The following example policy grants AWS account ID 123456789012 permission to send from identity *example.com*.

```
{
  "Id": "ExampleAuthorizationPolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AuthorizeAccount",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
      "Principal": {"AWS": ["123456789012"]},
      "Action": ["SES:SendEmail", "SES:SendRawEmail"]
    }
  ]
}
```

The following example policy grants permission to two IAM users to send from identity *example.com*. IAM users are specified by their Amazon Resource Name (ARN).

```
{
  "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"]
    }
  ]
}
```

Restricting the "From" Address

Even if you have verified a whole domain, you might want to restrict the "From" address so that the delegate sender can send from a specified email address only. To restrict the "From" address, you set a condition on the key called *ses:FromAddress*. The following policy enables AWS account ID 123456789012 to send from identity *example.com*, but only from email address *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"
        }
      }
    }
  ]
}
```

Restricting the Destination of Bounce and Complaint Feedback

If a delegate sender is sending on your behalf and you want to ensure that bounce and complaint notifications are forwarded to you by email, you need to do two things: you must enable email feedback forwarding for the identity by using the procedure in Amazon SES Notifications Through Email (p. 106), and you must restrict the "Return Path" of the emails to an email address that you own by setting a condition on the *ses:FeedbackAddress* key.

The following sending authorization policy enables AWS account ID 123456789012 to send from the identity *example.com* as long as the "Return Path" of the email is set to *feedback@example.com*.

```
{
  "Id": "ExamplePolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ControlReturnPath",
      "Effect": "Allow",
      "Resource": "arn:aws:ses:us-east-1:888888888888:identity/example.com",
      "Principal": {"AWS": ["123456789012"]},
      "Action": ["SES:SendEmail", "SES:SendRawEmail"],
      "Condition": {
        "StringEquals": {
          "ses:FeedbackAddress": "feedback@example.com"
        }
      }
    }
  ]
}
```

Restricting the Time Period of Sending

You might want to constrain the date and time during which the delegate sender can send on your behalf. For example, if your email campaign is scheduled for the month of September 2015, the following policy enables the delegate sender to send emails on your behalf during that month only.

```
{
  "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":"2015-08-31T12:00Z"
        },
        "DateLessThan": {
          "aws:CurrentTime":"2015-10-01T12:00Z"
        }
      }
    }
  ]
}
```

Restricting the Email-Sending Action

There are two actions that senders can use to send an email with Amazon SES: `SendEmail` and `SendRawEmail`, depending on how much control the sender wants over the format of the email. Sending authorization policies enable you to restrict the delegate sender to one of those two actions. However, many identity owners leave the details of the email-sending calls up to the delegate sender by enabling both actions in their policies.

> **Note**
> If you want to enable the delegate sender to access Amazon SES through the SMTP interface, you must choose `SendRawEmail` at a minimum.

If your use case is such that you want to restrict the action, you can do so by including only one of the actions in your sending authorization policy. The following example shows you how to restrict the action to `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"]
    }
  ]
}
```

Restricting the Display Name of the Email Sender

Some email clients display the "friendly" name of the email sender (if the email header provides it), rather than the actual "From" address. For example, the display name of "John Doe <johndoe@example.com>" is John Doe. For instance, you might send emails from *user@example.com*, but you prefer that recipients see that the email is from *Marketing* rather than from *user@example.com*. The following policy enables AWS account ID 123456789012 to send from identity *example.com*, but only if the display name of the "From" address includes *Marketing*.

```
{
  "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"
          }
      }
    }
  ]
}
```

Using Multiple Statements

You can use multiple statements for fine-grained control. The following example policy has two statements. The first statement authorizes two individual AWS accounts to send from *sender@example.com* using the `SendEmail` API as long as the "From" address and the feedback address are both under the domain *example.com*. The second statement authorizes an IAM user to send email from *sender@example.com* as long as the email is sent to an email address under the domain *example.com*.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AuthorizeAWS",
        "Effect": "Allow",
        "Resource": "arn:aws:ses:us-east-1:999999999999:identity/sender@ex
ample.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@ex
ample.com",
        "Principal": {
            "AWS": "arn:aws:iam::333333333333:user/Jane"
        },
        "Action": ["SES:SendEmail", "SES:SendRawEmail"],
        "Condition": {
            "ForAllValues:StringLike": {
                "ses:Recipients": "*@example.com"
            }
        }
    }]
}
```

Identity Owner Tasks for Amazon SES Sending Authorization

This section describes all of your responsibilities as an identity owner. Your main responsibility is to create a sending authorization policy that grants permission to a delegate sender to send on your behalf, and to attach that policy to the identity that you want the delegate sender to use. There are also some small setup tasks that you need to perform.

To see where these tasks fit into the overall sending authorization process, see Overview of Sending Authorization (p. 134).

Verifying an Identity for Amazon SES Sending Authorization

As with any Amazon SES sender, you must first prove that you own the email address or domain from which your emails will be sent, even though the delegate sender will send the emails. The verification procedure is described in Verifying Email Addresses and Domains (p. 37).

You can confirm that your email address or domain is verified by looking at its status in the Verified Senders list in the Amazon SES console or by using the Amazon SES `GetIdentityVerificationAttributes` API.

Setting Up Identity Owner Notifications for Amazon SES Sending Authorization

When a delegate sender sends emails on your behalf, bounces and complaints that those emails generate count toward the delegate sender's bounce and complaint metrics rather than your own. However, if you want to be informed of these email-sending outcomes, you can set up notifications by email or by Amazon SNS notifications just as you would for any other identity. Follow the procedures in Using Notifications with Amazon SES (p. 105).

Delegate senders can set up their own Amazon SNS notifications for the identities that you have authorized them to use, although only you, the identity owner, can control whether feedback can be directly forwarded to your identity via email.

Getting Information from the Delegate Sender for Amazon SES Sending Authorization

Your sending authorization policy must specify at least one *principal*, which is the entity to which you are granting access. For Amazon SES sending authorization policies, the principal can be an AWS account or an Identity and Access Management (IAM) user.

The type of principal you choose depends on your preference, but if you want the finest grain control, ask the delegate sender to set up an IAM user so that only one delegate sender can send for you rather than any user in the delegate sender's AWS account. The delegate sender can find information about setting up an IAM user in Creating an IAM User in Your AWS Account in the *IAM User Guide*.

After you have decided whether you want to grant access to an AWS account or an IAM user, ask the delegate sender for the AWS account ID or the IAM user's Amazon Resource Name (ARN) so that you can include it in your sending authorization policy. You can refer your delegate sender to the instructions for finding this information in Providing Information to the Identity Owner (p. 151).

Creating a Policy for Amazon SES Sending Authorization

To authorize a delegate sender to send emails for one of your identities, you create a sending authorization policy and then attach that policy to the identity. Identities can have zero policies, one policy, or multiple policies. However, each policy must be associated with an identity, and one identity only.

> **Important**
> Policies attached to email address identities override policies attached to the corresponding domain identities. For example, say that you have verified *example.com* and *user@example.com*. If you create a policy for *example.com* that disallows a delegate sender, and you create a policy for *user@example.com* that allows that delegate sender, the delegate sender will be able to send from *user@example.com* if they specify the ARN of *user@example.com* in the request to send the email.

You can create a sending authorization policy in the following ways:

- **Using the Policy Generator**—You can create a simple policy by using the Policy Generator in the Amazon SES console. In addition to specifying who can send the emails, you can constrain the email-sending with conditions based on the time and date range in which emails can be sent, the "From" address, the "From" display name, the address to which bounces and complaints are sent, the recipient addresses, and the source IP. You might also want to use the Policy Generator to create the structure of a simple policy and then customize it later by editing the policy.
- **Creating a Custom Policy—**If you want to include more advanced conditions in your policy, you can create a custom policy and attach it to the identity by using the Amazon SES console or the Amazon SES API.

This topic describes both methods.

Using the Policy Generator

You can use the Policy Generator to create a simple authorization policy by using the following procedure.

To create a policy by using the Policy Generator

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, under **Identities**, choose either **Email Addresses** or **Domains**.
3. In the resource list, choose the identity for which you want to create a policy.
4. In the details pane, expand **Identity Policies**, choose **Create Policy**, and then choose **Policy Generator**.
5. In the wizard, create a policy statement by choosing values for the following fields. You can find information about these options in Sending Authorization Policies (p. 136).

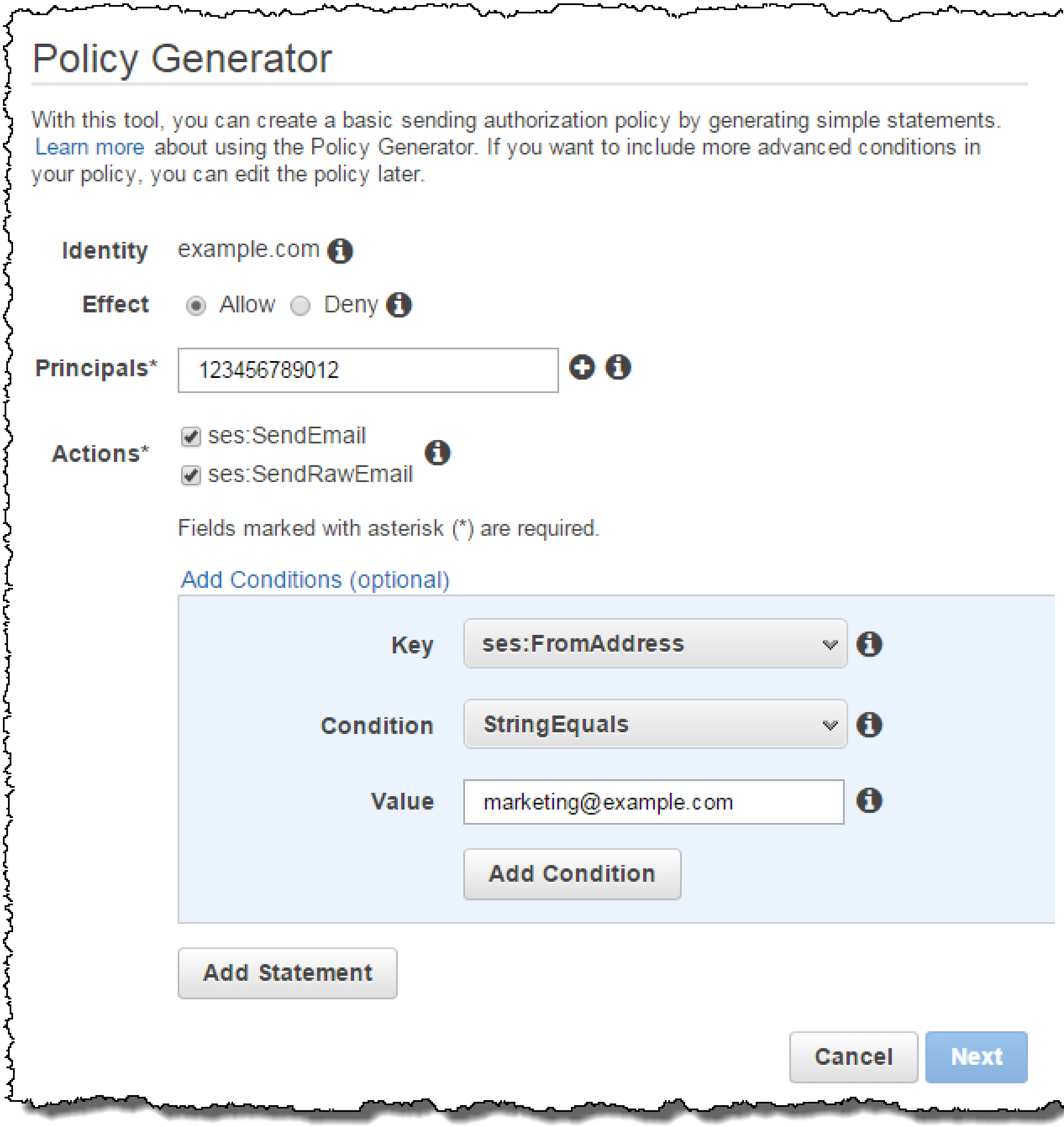
   - **Effect**—If you want to grant access, choose **Allow**; otherwise, choose **Deny**.

- **Principals**—Enter either the 12-digit AWS account ID or the ARN of an IAM user that you are allowing or denying access, and then choose **Add**. You can add more principals by repeating this step. An example of an AWS account ID is 123456789012 and an example of an IAM user ARN is *arn:aws:iam::123456789012:user/John*.
- **Actions**—Choose the email-sending access to which this policy applies. Typically, identity owners choose both options to give the delegate sender the freedom to choose how to implement the email sending. For more information, see Statements (p. 138).

6. (Optional) If you want to add restrictions to the policy, choose **Add Conditions**, and then choose the following information:

   - **Key**—This is the characteristic that is the basis for access restriction. The Policy Generator lets you choose an Amazon SES-specific key or one of a few commonly used AWS-wide keys (current time and source IP). For details, see Conditions (p. 139). If you want to specify the more advanced AWS-wide keys listed in Available Keys, you can edit the policy after you create it.
   - **Condition**—This is the type of condition that you want to specify. For example, there are string conditions, numeric conditions, date and time conditions, and so on. For a list of conditions, see the Condition Types in the *IAM User Guide*.
   - **Value**—This is the value that will be tested against the condition. For examples, see the policies in Sending Authorization Policy Examples (p. 140).

   After you choose the key, condition, and value, choose **Add Condition**. The condition appears in the **Conditions** list. You can remove conditions by choosing **Remove** next to a condition in the list. You can add another condition by choosing **Add Conditions** again.



Policy Generator

With this tool, you can create a basic sending authorization policy by generating simple statements. Learn more about using the Policy Generator. If you want to include more advanced conditions in your policy, you can edit the policy later.

Identity: example.com

Effect: ● Allow ○ Deny

Principals*: 123456789012

Actions*: ☑ ses:SendEmail ☑ ses:SendRawEmail

Fields marked with asterisk (*) are required.

Add Conditions (optional)

Key: ses:FromAddress

Condition: StringEquals

Value: marketing@example.com

7. When you are finished adding conditions (if any), choose **Add Statement**. The statement appears in the **Statements** list, where you can choose to edit or remove it. You can add additional statements by repeating steps 5-7.
8. When you are finished adding statements, choose **Next**.
9. In the **Edit Policy** dialog box, review your policy, edit it if needed, and then choose **Apply Policy**.

Creating a Custom Policy

If you want to create a custom policy and attach it to an identity, you have the following options:

- **Using the Amazon SES API**—Create a policy in a text editor and then attach the policy to the identity by using the `PutIdentityPolicy` API described in the Amazon Simple Email Service API Reference.
- **Using the Amazon SES console**—Create a policy in a text editor and attach it to an identity by pasting it into the Custom Policy editor in the Amazon SES console. The following procedure describes this method.

To create a custom policy by using the Custom Policy editor

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, under **Identities**, choose either **Email Addresses** or **Domains**.
3. In the resource list, choose the identity for which you want to create a policy.
4. In the details pane, expand **Identity Policies**, choose **Create Policy**, and then choose **Custom Policy**.
5. In the **Edit Policy** pane, paste the text of your policy and edit it as necessary.
6. Choose **Apply Policy**.

Providing the Delegate Sender with the Identity Information for Amazon SES Sending Authorization

After you create your sending authorization policy and attach it to your identity, you need to give the delegate sender the Amazon Resource Name (ARN) of the identity. The delegate sender will pass that ARN to Amazon SES in the email-sending operation or in the header of the email.

You can use the following procedure to find your identity's ARN.

To find the ARN of an identity

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, under **Identities**, choose either **Email Addresses** or **Domains**.
3. In the resource list, choose the identity to which you attached the sending authorization policy.
4. At the top of the details pane, after **Identity ARN**, you will see the identity's ARN. It will look similar to *arn:aws:ses:us-east-1:123456789012:identity/user@example.com*. Copy the entire ARN and give it to your delegate sender.

Managing Your Policies for Amazon SES Sending Authorization

In addition to creating and attaching policies to identities as explained in Creating a Policy (p. 146), you can edit, remove, list, and retrieve an identity's policies, as described in the following sections.

> **Note**
> To revoke permissions, you can either edit a policy or remove it.

Editing a Policy

The easiest way to edit a policy is to use the Amazon SES console. If you want to use the Amazon SES API instead, you can use the `GetIdentityPolicies` API to retrieve the policy, edit the policy by using a text editor, and then use the `PutIdentityPolicy` API to overwrite the older policy. These actions are explained in the Amazon Simple Email Service API Reference.

The following procedure shows you how to edit a policy by using the Amazon SES console.

To edit a policy by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, under **Identities**, choose either **Email Addresses** or **Domains**.
3. In the resource list, choose the identity that is associated with the policy that you want to edit.
4. In the details pane, expand **Identity Policies**, find the policy that you want in the Identity Policy list, and then choose **Edit Policy**.
5. In the **Edit Policy** pane, edit the policy, and then choose **Apply Policy**.
6. In the **Overwrite Existing Policy** dialog box, choose **Overwrite**.

Removing a Policy

To revoke permissions at any time, you can simply remove the policy. You can remove a policy by using the `DeleteIdentityPolicy` API, as explained in the Amazon Simple Email Service API Reference, or you can use the Amazon SES console, as described in the following procedure.

> **Important**
> After you remove a policy, there is no way to get it back. We recommend that you back up the policy by copying and pasting it into a text file before you remove the policy.

To remove a policy by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, under **Identities**, choose either **Email Addresses** or **Domains**.
3. In the resource list, choose the identity that is associated with the policy that you want to remove.
4. In the details pane, expand **Identity Policies**, find the policy that you want to remove, and then choose **Remove Policy**.
5. In the **Remove Policy** dialog box, choose **Yes, Remove Policy**.

Listing and Retrieving Policies

You can list the policies that are attached to an identity by using the `ListIdentityPolicies` API as explained in the Amazon Simple Email Service API Reference. You can also retrieve the policies themselves by using the `GetIdentityPolicies` API.

You can also jointly perform these operations in the Amazon SES console as described in the following procedure.

To list and show the policies attached to an identity by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, under **Identities**, choose either **Email Addresses** or **Domains**.

3. In the resource list, choose the identity for which you want to see policies.
4. In the details pane, expand **Identity Policies**. You will see a list of policies.
5. Find the policy that you want to view in the Identity Policy list, and then choose **Show Policy**.
6. After you are finished viewing the policy, close the **Show Policy** dialog box.

Delegate Sender Tasks for Amazon SES Sending Authorization

As a delegate sender, you are sending *cross-account* emails. This means that you are sending emails on behalf of an identity that you do not own, but are authorized to use. Even though you are sending on the identity owner's behalf, bounces and complaints count toward your bounce and complaint metrics, and the emails count toward your sending quota. You are also responsible for requesting any sending limit increases that you might need to send the identity owner's emails.

Delegate senders are responsible for the tasks described in this section. To see where these tasks fit into the overall sending authorization process, see Overview of Sending Authorization (p. 134).

- Providing Information to the Identity Owner (p. 151)
- Using Delegate Sender Notifications (p. 152)
- Sending Emails for the Identity Owner (p. 155)

Providing Information to the Identity Owner for Amazon SES Sending Authorization

As a delegate sender, you need to give your AWS account ID or the Amazon Resource Name (ARN) of the Identity and Access Management (IAM) user who will do the sending to the identity owner. You can find this information by using the following procedures.

To find the ARN of an IAM user

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam.
2. In the left navigation pane, choose **Users**.
3. In the resource list, choose the user name. The **Summary** section displays the ARN. The ARN will look something like *arn:aws:iam::123456789012:user/John*.

To find your AWS account ID

1. You can go directly to https://console.aws.amazon.com/billing/home?#/account. Alternatively, you can navigate to it by going to the AWS Management Console at https://console.aws.amazon.com/console. In the top bar, select your name, and then select **My Account**.
2. Expand **Account Settings**. The AWS account ID is at the top of this section.

Using Delegate Sender Notifications for Amazon SES Sending Authorization

As a delegate sender, you can set up Amazon Simple Notification Service (Amazon SNS) notifications to inform you of bounces, complaints, and deliveries. The format and content of these notifications are described in Amazon SES Notifications Through Amazon SNS (p. 108). Only the identity owner has the option to receive notifications by email feedback forwarding as described in Amazon SES Notifications Through Email (p. 106).

> **Important**
>
> As the delegate sender, bounces and complaints count toward *your* bounce and complaint metrics. High bounce and complaint rates put your account at risk of being shut down, so ensure that you set up notifications and have a process in place to monitor the notifications and remove recipient addresses that have bounced or complained from your mailing list. For more information, see Processing Bounces and Complaints (p. 178).

You will be charged standard Amazon SNS rates for bounce, complaint, and delivery notifications. For more information, see the Amazon SNS pricing page.

The following sections show you how to manage cross-account identity notifications.

- Setting Up a Notification Configuration (p. 152)
- Editing a Notification Configuration (p. 153)
- Viewing a Notification Configuration (p. 154)
- Removing a Notification Configuration (p. 154)

Setting Up an Amazon SES Cross-Account Identity Notification Configuration

Before you set up notifications, you need to know the Amazon Resource Name (ARN) of the identity that the identity owner has authorized you to use, and for which you want to configure notifications. For example, the ARN for identity *user@example.com* would look similar to *arn:aws:ses:us-east-1:123456789012:identity/user@example.com*. If the identity owner has not given you the identity's ARN, refer them to the procedure in Providing the Delegate Sender with the Identity Information (p. 149).

The easiest way to configure notifications is to use the Amazon SES console. If you want to use the Amazon SES API instead, you can use the `SetIdentityNotificationTopic` API and pass the identity's ARN as the `Identity` parameter. This action is explained in the Amazon Simple Email Service API Reference. The following procedure shows you how to set up notifications by using the Amazon SES console.

To set up Amazon SNS bounce, complaint, and/or delivery notifications by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, choose **Cross-Account Notifications**.
3. Choose **Add Notification Config**.
4. In the **Edit Notification Configuration** dialog box, enter the ARN of the identity that the identity owner has authorized you to use, and for which you want to configure notifications. The identity cannot belong to the account that is currently logged in. If you want to configure notifications for your own identities, see Configuring Amazon SNS Notifications for Amazon SES (p. 108).

5. Specify the existing Amazon SNS topics that you want to use for bounces, complaints, and/or deliveries, or create a new Amazon SNS topic.

   > **Important**
   > The Amazon SNS topics that you use for Amazon SES notifications must be within the same AWS region in which you are using Amazon SES.

   You can choose to publish bounce, complaint, and delivery notifications to the same Amazon SNS topic or to different Amazon SNS topics. If you want to use an Amazon SNS topic that you do not own, then the owner of that topic must configure an Amazon SNS access policy that allows your account to call the `SNS:Publish` action on their topic. For information about how to control access to your Amazon SNS topic through the use of IAM policies, see Managing Access to Your Amazon SNS Topics.
6. Choose **Save Config** to save your notification configuration. Changes might take a few minutes to take effect.

   After you have configured your settings, you will start receiving bounce, complaint, and/or delivery notifications to your Amazon SNS topic(s). These notifications will follow the structure described in Amazon SNS Notification Contents for Amazon SES (p. 110).

Editing an Amazon SES Cross-Account Notification Configuration

The easiest way to edit notification configurations is to use the Amazon SES console. If you want to use the Amazon SES API instead, you can use the `SetIdentityNotificationTopic` API and pass the identity's ARN as the `Identity` parameter. This action is explained in the Amazon Simple Email Service API Reference.

The following procedure shows you how to edit a cross-account notification configuration by using the Amazon SES console.

To edit a cross-account notification configuration by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, choose **Cross-Account Notifications**.

   The cross-account identities for which you have set up notifications will be listed in the **Cross-Account Notifications** details pane.
3. Choose the ARN of the identity for which you want to view the notification configuration.
4. Edit the notification settings, and then choose **Save Config**.

   > **Note**
   > Setting all notifications to **No SNS Topic** is the equivalent of removing the identity's notification configuration entirely. In this case, the ARN of the cross-account identity will disappear from your list of cross-account identity ARNs in the Amazon SES console. This does not mean that you cannot continue to send for that identity; it just means that you are no longer set up to receive bounce, complaint, and/or delivery notifications for it. If you want to re-enable notifications, you need to repeat the notification setup procedure described in Setting Up a Notification Configuration (p. 152).

Viewing Your Amazon SES Cross-Account Identity Notifications

The easiest way to view your notification configurations is to use the Amazon SES console. If you want to use the Amazon SES API instead, you can use the `GetIdentityNotificationAttributes` API and pass the identity's ARN as the `Identity` parameter. This action is explained in the Amazon Simple Email Service API Reference.

> **Note**
> The only cross-account identities that you will find in the cross-account identity list are the identities for which you have configured notifications by using the procedure described in Setting Up a Notification Configuration (p. 152).

To view your cross-account notification configurations by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, choose **Cross-Account Notifications**.

   The cross-account identities for which you have set up notifications will be listed in the **Cross-Account Notifications** details pane.
3. Choose the ARN of an identity.

   The **Edit Configuration Notification** dialog box will display the identity's settings.

Removing an Amazon SES Cross-Account Identity Notification Configuration

The easiest way to remove a notification configuration is to use the Amazon SES console. If you want to use the Amazon SES API instead, you can use the `SetIdentityNotificationTopic` API, pass the identity's ARN as the `Identity` parameter, and pass in null for the `SnsTopic` parameter. This action is explained in the Amazon Simple Email Service API Reference. To completely remove the notification configuration, you must perform this operation for each type of notification type (bounce, complaint, and/or delivery) that was set.

> **Note**
> When you remove a notification configuration, the ARN of the cross-account identity will disappear from your list of cross-account identity ARNs in the Amazon SES console. This does not mean that you cannot continue to send for that identity; it just means that you are no longer set up to receive bounce, complaint, and/or delivery notifications for it. If you want to re-enable notifications, you need to repeat the notification setup procedure described in Setting Up a Notification Configuration (p. 152).

The following procedure shows you how to remove a cross-account notification configuration by using the Amazon SES console.

To remove a cross-account notification configuration by using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the left navigation pane, choose **Cross-Account Notifications**.

   The cross-account identities for which you have set up notifications will be listed in the **Cross-Account Notifications** details pane.
3. Choose the box to the left of the cross-identity that you want to remove, and then choose **Remove**.

4. In the **Remove Cross-Account Notification Config** dialog box, choose **Delete Notification config**.

   The ARN of the cross-account identity will no longer appear in the list of cross-account identity ARNs. This does not mean that you cannot send for the identity, just that you no longer have configured notifications for it.

Sending Emails for the Identity Owner for Amazon SES Sending Authorization

As a delegate sender, you send emails the same way that other Amazon SES senders do, except that you provide the ARN of the identity that the identity owner has authorized you to use. When you call Amazon SES to send the email, Amazon SES checks to see if the identity that you specified has a policy that authorizes you to send for it.

There are different ways that you can specify the identity's ARN when you send an email. The method that you can use depends on whether you send the email by using the Amazon SES API (`SendEmail` or `SendRawEmail`) or the Amazon SES SMTP interface.

> **Important**
> To successfully send an email on behalf of an identity owner's identity, you must connect to the Amazon SES endpoint of the AWS region in which the identity is verified. The sending authorization policy that grants you permission must be attached to the identity in that region.

Using the Amazon SES API

As with any Amazon SES email sender, if you access Amazon SES through the Amazon SES API (either directly through HTTPS or indirectly through an AWS SDK), you can choose between one of two email-sending actions: `SendEmail` and `SendRawEmail`. The Amazon Simple Email Service API Reference describes the details of these APIs, but we provide an overview of the sending authorization parameters here.

SendRawEmail

If you want to use `SendRawEmail` so that you can control the format of your emails, you can specify the cross-account identity in one of two ways:

- **Pass optional parameters to the `SendRawEmail` API**—These parameters are as follows:

Parameter	Description
`SourceArn`	The ARN of the identity that is associated with the sending authorization policy that permits you to send for the email address specified in the `Source` parameter of `SendRawEmail`. **Note** For the most common use case, we recommend that you specify the `SourceArn` and do not specify either the `FromArn` or `ReturnPathArn`. If you only specify the `SourceArn`, Amazon SES will simply set the "From" address and the "Return Path" addresses to the identity specified in `SourceArn`.

Parameter	Description
`FromArn`	The ARN of the identity that is associated with the sending authorization policy that permits you to specify a particular "From" address in the header of the raw email.
`ReturnPathArn`	The ARN of the identity that is associated with the sending authorization policy that permits you to use the email address specified in the `ReturnPath` parameter of `SendRawEmail`.

- **Include X-headers in the email**— X-headers are custom headers that you can use in addition to standard email headers. Amazon SES has three X-headers that you can use to specify sending authorization parameters. If you include multiple instances of any of the X-headers, Amazon SES will use the first instance. In all cases, Amazon SES removes all X-headers from the email before sending it. The following table shows you the three X-headers that you can use with Amazon SES for sending authorization.

  ### Important

  Do not include these X-headers in the DKIM signature, because they are removed by Amazon SES before sending the email.

X-Header	Description
`X-SES-SOURCE-ARN`	Corresponds to the `SourceArn`.
`X-SES-FROM-ARN`	Corresponds to the `FromArn`.
`X-SES-RETURN-PATH-ARN`	Corresponds to the `ReturnPathArn`.

The following example shows an email that includes sending authorization X-headers:

```
X-SES-SOURCE-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com
X-SES-FROM-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com
X-SES-RETURN-PATH-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com

From: sender@example.com
To: recipient@example.com
Return-Path: feedback@example.com
Subject: subject
Content-Type: multipart/alternative;
 boundary="----=_boundary"

------=_boundary
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 7bit

body
------=_boundary
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: 7bit

body
------=_boundary--
```

SendEmail

If you want to use `SendEmail` so that Amazon SES formats your emails for you, you can specify the cross-account identity by passing in the optional parameters below. You cannot use the X-header method because when you use `SendEmail`, Amazon SES assembles the message for you.

Parameter	Description
`SourceArn`	The ARN of the identity that is associated with the sending authorization policy that permits you to send for the email address specified in the `Source` parameter of `SendRawEmail`.
`ReturnPathArn`	The ARN of the identity that is associated with the sending authorization policy that permits you to use the email address specified in the `ReturnPath` parameter of `SendRawEmail`.

Using the Amazon SES SMTP interface

If you are using the Amazon SES SMTP interface for cross-account sending, the only method you can use is to include the X-headers as `SendRawEmail` described earlier.

Testing Amazon SES Email Sending

Amazon Simple Email Service (Amazon SES) provides a mailbox simulator that you can use to test how your application handles various email sending scenarios without affecting your sending quota or your bounce and complaint metrics. The Amazon SES mailbox simulator is a set of test email addresses. Each email address represents a specific scenario. You can send emails to the mailbox simulator when you want to:

- Test your application without having to create test "To" addresses.
- Test how your email sending program handles bounces, complaints, and out-of-the-office (OOTO) responses.
- See what happens when you email an address that is on the Amazon SES suppression list.
- Generate a bounce without putting a valid email address on the suppression list.
- Find your system's maximum throughput without using up your daily sending quota.
- Send test emails without affecting your email deliverability metrics for bounces and complaints.

To use the mailbox simulator, email the addresses and observe how your setup responds to the simulated scenarios. The following table lists each simulated scenario and the corresponding email address that you would use. The email addresses are not case-sensitive.

Simulated scenario	Mailbox simulator email address
Success—The recipient's ISP accepts your email. If you have set up delivery notifications as described in Using Notifications with Amazon SES (p. 105), Amazon SES sends you a delivery notification through Amazon Simple Notification Service (Amazon SNS). Otherwise, you will not receive any confirmation about this successful delivery other than the API return value.	success@simulator.amazonses.com

Simulated scenario	Mailbox simulator email address
Bounce—The recipient's ISP rejects your email with an SMTP 550 5.1.1 response code ("Unknown User"). Amazon SES generates a bounce notification and sends it to you via email or by using an Amazon SNS notification, depending on how you set up your system. This mailbox simulator email address will not be placed on the Amazon SES suppression list as one normally would when an email hard bounces. The bounce response that you receive from the mailbox simulator is compliant with RFC 3464. For information about how to receive bounce feedback, see Using Notifications with Amazon SES (p. 105).	bounce@simulator.amazonses.com
Out of the Office—The recipient's ISP accepts your email and delivers it to the recipient's inbox. The ISP sends an out-of-the-office (OOTO) message to Amazon SES. Amazon SES then forwards the OOTO message to you via email or by using an Amazon SNS notification, depending on how you set up your system. The OOTO response that you receive from the Mailbox Simulator is compliant with RFC 3834. For information about how to set up your system to receive OOTO responses, follow the same instructions for setting up how Amazon SES sends you notifications in Using Notifications with Amazon SES (p. 105).	ooto@simulator.amazonses.com
Complaint—The recipient's ISP accepts your email and delivers it to the recipient's inbox. The recipient, however, does not want to receive your message and clicks "Mark as Spam" within an email application that uses an ISP that sends a complaint response to Amazon SES. Amazon SES then forwards the complaint notification to you via email or by using an Amazon SNS notification, depending on how you set up your system. The complaint response that you receive from the mailbox simulator is compliant with RFC 5965. For information about how to receive bounce feedback, see Using Notifications with Amazon SES (p. 105).	complaint@simulator.amazonses.com
Address on Suppression List—Amazon SES treats your email as a hard bounce because the address you are sending to is on the Amazon SES suppression list.	suppressionlist@simulator.amazonses.com

Important

If you send an email to a mailbox simulator address other than the test addresses listed above, the unlisted address will be placed on the suppression list.

The mailbox simulator provides typical bounce, complaint, and OOTO responses. In the bounce scenario, multiple bounces from the same sending request are gathered into a single response. In practice, the response varies by ISP. To reduce your bounce and complaint rates, see the Amazon Simple Email Service Email Sending Best Practices white paper.

When you send emails to the mailbox simulator, you will be limited by your maximum send rate. You will also be billed for your emails. However, emails to the mailbox simulator will not affect your email deliverability metrics for bounces and complaints or count against your sending quota.

The mailbox simulator supports labeling, which enables you to send emails to the same mailbox simulator address in multiple ways, or to test your support for Variable Envelope Return Path (VERP). For example, you can send an email to bounce+label1@simulator.amazonses.com and bounce+label2@simulator.amazonses.com to test how your setup matches a bounce message with the undeliverable address that caused the bounce. For more information about VERP, see http://en.wikipedia.org/wiki/Variable_envelope_return_path.

You can send emails to the mailbox simulator even if you are in the sandbox.

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Logging Amazon SES API Calls By Using AWS CloudTrail

Amazon SES is integrated with CloudTrail, a service that captures API calls made by or on behalf of Amazon SES in your AWS account and delivers the log files to an Amazon S3 bucket that you specify. CloudTrail captures API calls made from the Amazon SES console or from the Amazon SES API. Using the information collected by CloudTrail, you can determine what request was made to Amazon SES, the source IP address from which the request was made, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

Amazon SES Information in CloudTrail

When CloudTrail logging is enabled in your AWS account, API calls made to a subset of Amazon SES actions are tracked in log files. Amazon SES records are written together with other AWS service records in a log file. CloudTrail determines when to create and write to a new file based on a time period and file size.

The following actions are supported:

- DeleteIdentity
- DeleteVerifiedEmailAddress
- GetIdentityDkimAttributes
- GetIdentityNotificationAttributes
- GetIdentityVerificationAttributes
- GetSendQuota
- GetSendStatistics
- ListIdentities
- ListVerifiedEmailAddresses
- SetIdentityDkimEnabled
- SetIdentityFeedbackForwardingEnabled
- SetIdentityNotificationTopic
- VerifyDomainDkim
- VerifyDomainIdentity

- VerifyEmailAddress
- VerifyEmailIdentity

Every log entry contains information about who generated the request. The user identity information in the log helps you determine whether the request was made with root or IAM user credentials, with temporary security credentials for a role or federated user, or by another AWS service. For more information, see the **userIdentity** field in the CloudTrail Event Reference.

You can store your log files in your bucket for as long as you want, but you can also define Amazon S3 lifecycle rules to archive or delete log files automatically. By default, your log files are encrypted by using Amazon S3 server-side encryption (SSE).

You can choose to have CloudTrail publish Amazon SNS notifications when new log files are delivered if you want to take quick action upon log file delivery. For more information, see Configuring Amazon SNS Notifications.

You can also aggregate Amazon SES log files from multiple AWS regions and multiple AWS accounts into a single Amazon S3 bucket. For more information, see Aggregating CloudTrail Log Files to a Single Amazon S3 Bucket.

Understanding Amazon SES Log File Entries

CloudTrail log files contain one or more log entries where each entry is made up of multiple JSON-formatted events. A log entry represents a single request from any source and includes information about the requested action, any parameters, the date and time of the action, and so on. The log entries are not guaranteed to be in any particular order. That is, they are not an ordered stack trace of the public API calls.

The following example shows a CloudTrail log.

```
{
   "Records": [
      {
        "awsRegion": "us-west-2",
        "eventID": "0ffa308d-1467-4259-8be3-c749753be325",
        "eventName": "DeleteIdentity",
        "eventSource": "ses.amazonaws.com",
        "eventTime": "2015-02-02T21:34:50Z",
        "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": "17bb827a-dc8c-4156-90b1-c214e1d135c9",
      "eventName": "DeleteVerifiedEmailAddress",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-04T00:57:15Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "c29fb5c1-ac08-11e4-8ff5-a56a3119e253",
      "requestParameters": {
          "emailAddress": "user@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"
      }
    },
    {
      "awsRegion": "us-west-2",
      "eventID": "0b311e38-b5c6-43b3-9a39-5fbf0c2d0d99",
      "eventName": "GetIdentityDkimAttributes",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-02T21:34:50Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "50f92e80-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": {
          "identities": [
          "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"
      }
    },
    {
      "awsRegion": "us-west-2",
      "eventID": "bf695be8-1c67-45b0-8f10-fd56afee09dd",
      "eventName": "GetIdentityNotificationAttributes",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-02T21:34:50Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "5133ed92-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": {
          "identities": [
          "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"
      }
    },
    {
      "awsRegion": "us-west-2",
      "eventID": "8f9aed63-b03a-4d30-a880-33ae0c6b7786",
      "eventName": "GetIdentityVerificationAttributes",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-04T00:57:16Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "c2d23773-ac08-11e4-8ff5-a56a3119e253",
      "requestParameters": {
          "identities": [
              "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"
      }
    },
    {
      "awsRegion": "us-west-2",
      "eventID": "60ef4f01-9826-4fb4-828e-8c36dda81f40",
      "eventName": "GetSendQuota",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-04T01:03:27Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "a0760648-ac09-11e4-8ff5-a56a3119e253",
      "requestParameters": null,
      "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": "0fe5eef3-0c28-4480-808e-307b21404a78",
      "eventName": "GetSendStatistics",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-02T21:34:51Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "51644c64-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": null,
      "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": "6eb8178e-69c3-4a93-8af0-2a5a0f5f209e",
      "eventName": "ListIdentities",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-04T01:03:27Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "a0a4de7a-ac09-11e4-8ff5-a56a3119e253",
      "requestParameters": {
          "identityType": "Domain",
          "maxItems": 10
      },
      "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": "a18a9745-d06a-43e9-aad0-8eee4de50f48",
      "eventName": "ListVerifiedEmailAddresses",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-02T21:34:51Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "51ad8a66-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": null,
      "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": "da975f45-e68b-4499-8e3f-31a89140e0c9",
      "eventName": "SetIdentityDkimEnabled",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-04T01:01:24Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "5731c4ab-ac09-11e4-8ff5-a56a3119e253",
      "requestParameters": {
          "dkimEnabled": true,
          "identity": "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"
      }
    },
    {
      "awsRegion": "us-west-2",
      "eventID": "5d817126-dadb-436f-b480-f9843289f487",
      "eventName": "SetIdentityFeedbackForwardingEnabled",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-02T21:34:51Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "51dd4cf8-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": {
          "forwardingEnabled": true,
          "identity": "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"
      }
    },
    {
      "awsRegion": "us-west-2",
      "eventID": "1a31fd43-55ba-4ce7-b3fe-55659e8144c0",
      "eventName": "SetIdentityNotificationTopic",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-04T00:59:21Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "0d553aac-ac09-11e4-8ff5-a56a3119e253",
      "requestParameters": {
          "identity": "example.com",
          "notificationType": "Bounce",
          "snsTopic": "arn:aws:sns:us-west-2:123456789100:MyTopic"
      },
      "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": "aec73edb-6dac-4503-81bb-cca1102f959e",
      "eventName": "VerifyDomainDkim",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-02T21:34:52Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "52215ada-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": {
          "domain": "example.com"
      },
      "responseElements": {
          "dkimTokens": [
          "3r2ultrqtelopya3v2apjulcvz7z5n5o",
          "yexya47xmy5f3j3e7vgm6pcrcmayu6nu",
          "wtlduqduorhmb2vdt2m53yqlcj2m6tpw"
          ]
      },
      "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": "33b3e2eb-7ba3-460b-a127-a5f4cedb4469",
      "eventName": "VerifyDomainIdentity",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-04T00:59:21Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "0d9c2ebe-ac09-11e4-8ff5-a56a3119e253",
      "requestParameters": {
          "disableEmailNotifications": false,
          "domain": "example.com"
      },
      "responseElements": {
         "verificationToken": "pmBGN/7MjnfhTKUZ06Enqq1PeGUaOkw8lGhcfwefcHU="

      },
      "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": "eb2e1616-2b7b-4cd2-b6dc-29f83fc1789f",
      "eventName": "VerifyEmailAddress",
      "eventSource": "ses.amazonaws.com",
      "eventTime": "2015-02-02T21:34:53Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.02",
      "recipientAccountId": "111122223333",
      "requestID": "5265ddec-ab23-11e4-9106-5b36376f9d12",
      "requestParameters": {
          "emailAddress": "user@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"
        }
      },
      {
        "awsRegion": "us-west-2",
        "eventID": "5613b0ff-d6c6-4526-9b53-a603a9231725",
        "eventName": "VerifyEmailIdentity",
        "eventSource": "ses.amazonaws.com",
        "eventTime": "2015-02-04T01:05:33Z",
        "eventType": "AwsApiCall",
        "eventVersion": "1.02",
        "recipientAccountId": "111122223333",
        "requestID": "eb2ff803-ac09-11e4-8ff5-a56a3119e253",
        "requestParameters": {
            "emailAddress": "user@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"
      }
     }
   ]
}
```

Using Credentials With Amazon SES

To interact with Amazon Simple Email Service (Amazon SES), you use security credentials to verify who you are and whether you have permission to interact with Amazon SES. There are different types of credentials, and the credentials you use depend on what you want to do. For example, you use AWS access keys when you send an email using the Amazon SES API, and SMTP credentials when you send an email using the Amazon SES SMTP interface.

The following table lists the types of credentials you might use with Amazon SES, depending on what you are doing.

If you want to access the...	Use these credentials	What the credentials consist of	How to get the credentials
Amazon SES API (You might access the Amazon SES API directly, or indirectly through an AWS SDK, the AWS Command Line Interface, or the AWS Tools for Windows PowerShell.)	AWS access keys	Access key ID and secret access key	See the *Access keys* section of How Do I Get Security Credentials? in the *AWS General Reference*. **Note** For security best practice, use AWS Identity and Access Management (IAM) user access keys instead of AWS account access keys. Your AWS account credentials grant full access to all your AWS resources, so you should store them in a safe place and instead use IAM user credentials for day-to-day interaction with AWS. For more information, see Root Account Credentials vs. IAM User Credentials in the *AWS General Reference*.
Amazon SES SMTP interface	SMTP credentials	User name and password	See Obtaining Your Amazon SES SMTP Credentials (p. 50). **Note** Although your Amazon SES SMTP credentials are different than your AWS access keys and IAM user access keys, Amazon SES SMTP credentials are actually a type of IAM credentials. An IAM user can create Amazon SES SMTP credentials, but the root account owner must ensure that the IAM user's policy gives them permission to access the following IAM actions: "iam:ListUsers", "iam:CreateUser", "iam:CreateAccessKey", and "iam:PutUserPolicy".

If you want to access the...	Use these credentials	What the credentials consist of	How to get the credentials
Amazon SES console	IAM user name and password OR Email address and password	IAM user name and password OR Email address and password	See the *IAM user name and password* and *Email address and password* sections of How Do I Get Security Credentials? in the *AWS General Reference*. **Note** For security best practice, use an IAM user name and password instead of an email address and password. The email address and password combination are for your AWS account, so you should store them in a safe place instead of using them for day-to-day interaction with AWS. For more information, see Root Account Credentials vs. IAM User Credentials in the *AWS General Reference*.

For more information about different types of AWS security credentials (except for SMTP credentials, which are used only for Amazon SES), see Types of Security Credentials in the *AWS General Reference*.

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Regions and Amazon SES

When you use Amazon Simple Email Service (Amazon SES), you connect to a URL that provides an endpoint for the Amazon SES API or SMTP interface. Amazon SES has endpoints in multiple AWS regions. To reduce network latency, it's a good idea to choose an endpoint closest to your application.

The following table lists the AWS regions in which Amazon SES is available, and the corresponding endpoints for the Amazon SES API and SMTP interface.

Region name	API (HTTPS) endpoint	SMTP endpoint
US East (N. Virginia)	email.us-east-1.amazonaws.com	email-smtp.us-east-1.amazonaws.com
US West (Oregon)	email.us-west-2.amazonaws.com	email-smtp.us-west-2.amazonaws.com
EU (Ireland)	email.eu-west-1.amazonaws.com	email-smtp.eu-west-1.amazonaws.com

This topic contains information you need to know when you use Amazon SES endpoints in multiple AWS regions. It discusses the following subjects:

For general information about AWS regions, see Regions and Endpoints in the *AWS General Reference.*

Selecting a Region to Use with Amazon SES

The following sections describe how to select a region depending on which method you use to call Amazon SES.

Amazon SES API

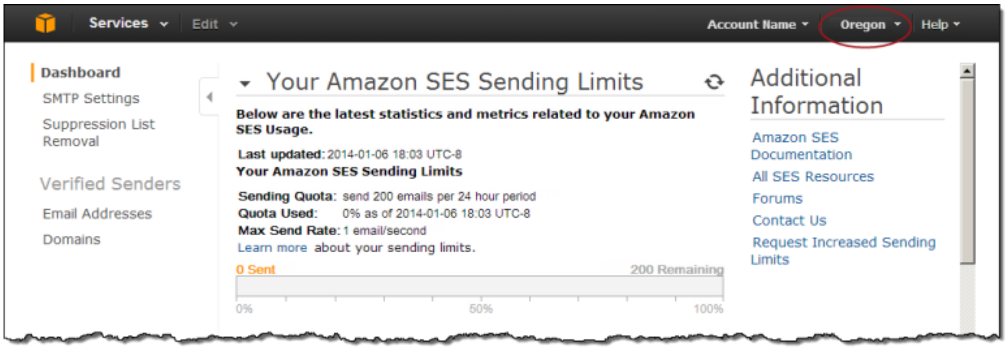
When you use the Amazon SES API, you specify an endpoint in the Query request. That endpoint determines the AWS region you are using. For more information, see Query Requests and Amazon SES (p. 84).

Amazon SES SMTP Interface

When you use the SMTP interface, the SMTP endpoint you specify in your code or configuration settings determines the AWS region you are using. For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 53).

Amazon SES Console

When you use the Amazon SES console, you can change the endpoint by clicking the region name in the upper right corner of the navigation bar, as shown in the following screenshot.



Your Amazon SES Sending Limits

Below are the latest statistics and metrics related to your Amazon SES Usage.

Last updated: 2014-01-06 18:03 UTC-8

Your Amazon SES Sending Limits

Sending Quota: send 200 emails per 24 hour period
Quota Used: 0% as of 2014-01-06 18:03 UTC-8
Max Send Rate: 1 email/second
Learn more about your sending limits.

0 Sent 200 Remaining

Sandbox and Sending Limit Increases

Sandbox status and sending limits apply on a per-region basis. You must request sending limit increases for each region individually. When you open an SES Sending Limits case in Support Center, the form has a menu you use to select the AWS region for which you are requesting a sending limit increase. For more information on increasing your sending limits, see Opening an SES Sending Limits Increase Case (p. 126).

Verification

Before you send email using Amazon SES, you must verify that you own your email address or domain with Amazon SES. Verification status for each region is separate, as described in the following sections.

Email Address Verification

You must verify each sender's email address separately for each region you want to use. For example, if you verify an email address in the US West (Oregon) region, you will be able to send from it when you connect to an Amazon SES endpoint in the US West (Oregon) region, but you will not be able to send from it using an endpoint in the US East (N. Virginia) region until you verify that email address in the US East (N. Virginia) region. For more information about verifying email addresses, see Verifying Email Addresses in Amazon SES (p. 37).

Domain Verification

Like email address verification, domain verification applies to each region separately. You must perform the domain verification procedure for each region in which you want to send from a given domain. For example, if you want to send email from *example.com* from both the US West (Oregon) region endpoint and the US East (N. Virginia) region endpoint, you must add two TXT records to your DNS settings — one record for each region. You generate these records by using the Amazon SES console with the appropriate region selected, or the Amazon SES API endpoint that corresponds to the region you want. For more information about verifying domains, see Verifying Domains in Amazon SES (p. 40).

Easy DKIM Setup

You must perform the Easy DKIM setup procedure for each region in which you want to use Easy DKIM. That is, for each region, you must use the Amazon SES console or the Amazon SES API to generate TXT records, add the TXT records to your DNS settings, and then use the Amazon SES API or the Amazon SES console to enable DKIM signing for your chosen sending identity (email address or domain) within that region. For more information about setting up Easy DKIM, see Easy DKIM in Amazon SES (p. 94).

Suppression List

Although each region has a separate suppression list, if you remove an address from the suppression list of one region, the address is removed from the suppression list of all regions. You remove addresses from the suppression list by using the Amazon SES console. For more information about the suppression list, see Removing an Email Address from the Amazon SES Suppression List (p. 187).

Feedback Notifications

There are two important points to note about setting up feedback notifications in multiple regions:

- Verified identity settings, such as whether you receive feedback by email or through Amazon Simple Notification Service (Amazon SNS), apply only to the region in which you set them. For example, if you verify user@example.com in the US West (Oregon) and US East (N. Virginia) regions and you want to receive bounced emails via Amazon SNS notifications, you must use the Amazon SES API or the Amazon SES console to set up Amazon SNS feedback notifications for user@example.com in both regions.
- Amazon SNS topics you use for feedback forwarding must be within the same region in which you are using Amazon SES.

SMTP Credentials

You can use the same set of SMTP credentials in all regions. For more information about SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 50).

Sending Authorization

The delegate sender must send the emails from the AWS region in which the identity owner's identity is verified. The sending authorization policy that gives permission to the delegate sender must be attached to the identity in that region. For more information about sending authorization, see Using Sending Authorization with Amazon SES (p. 134).

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Best Practices with Amazon SES

This section contains the following topics on best practices for sending email using Amazon Simple Email Service (Amazon SES):

- For tips on how to improve the chances that your emails will be delivered to your recipients' inboxes, see Improving Deliverability with Amazon SES (p. 177).
- For ways to keep your mailing list from containing invalid addresses and recipients who do not want your mail, see Obtaining and Maintaining Your Recipient List (p. 178).
- For guidance on how to handle bounces and complaints, see Processing Bounces and Complaints (p. 178).
- For factors to consider when you send email through Amazon SES using multiple AWS accounts, see Using Multiple Amazon SES Accounts (p. 179).

Improving Deliverability with Amazon SES

The following recommendations can help improve your deliverability when you use Amazon SES.

- **Only send email to recipients who have requested it**—Collect recipients' email addresses yourself, and with the recipients' permission. Do not buy mailing lists from third parties. Keep your mailing lists up-to-date and provide a mechanism for recipients to unsubscribe. If your mailing list is associated with a discussion group, consider unsubscribing recipients who have not interacted with you for a long period of time (for example, 180 days).
- **Keep your number of bounces and complaints low**—High numbers of bounces indicate to ISPs that you do not know your recipients very well. High numbers of complaints indicate that recipients do not want to receive your emails. If an email bounces or is marked as spam by a recipient, make sure to remove that recipient from your list. For information about how to be notified of bounces and complaints, see Using Notifications with Amazon SES (p. 105).
- **Authenticate your email**—Authentication is a way to show ISPs that your emails are genuine and have not been modified in transit. For more information, see Authenticating Email in Amazon SES (p. 93).
- **Send high-quality email**—High-quality email is email that your recipients expect and find valuable. Value means different things to different recipients and can come in the form of offers, order confirmations, receipts, newsletters, etc. Inform your recipients of what you plan to send and understand what your recipients expect from an email program.
- **Check your sending statistics**—Regularly monitor your number of delivery attempts, bounces, complaints, and rejected emails so that you can identify and correct problems right away. To check your sending statistics, see Monitoring Your Amazon SES Usage Statistics (p. 121).

- **Watch your sending limits**—If you attempt to exceed your sending limits, your calls to the Amazon SES API will fail. Check the Amazon SES console or call `GetSendQuota`. If you need to raise your sending limits, see Increasing Your Amazon SES Sending Limits (p. 125).
- **Watch for upward trends in rejected emails.** Amazon SES will generate a *MessageRejected* error for any message that it does not accept; if you see a large number of rejections, make sure that none of your applications are trying to send the same rejected message repeatedly.

For a more in-depth discussion of these and other best practices, see the Amazon Simple Email Service Email Sending Best Practices white paper.

Obtaining and Maintaining Your Recipient List

Ultimately, you want to make sure that the recipient addresses on your mailing list are valid and that your recipients want and expect your mail. Emails to invalid recipient addresses will bounce, and if valid recipients do not want your mail, they may mark your email as spam in their email client. High bounce and complaint rates put your account at risk of being shut down.

The following list includes, but is not limited to, ways that will help you keep your recipient list clean. For more detailed information, see the Amazon Simple Email Service Email Sending Best Practices white paper.

- Set up a process to monitor bounces and complaints, and when a recipient address bounces or complains, remove it from your mailing list. For more information, see Processing Bounces and Complaints (p. 178).
- Do not buy email lists.
- Only send emails to recipients who have interacted with your site recently (for example, within the last 180 days).
- When a recipient signs up for your list, make it clear what type of mail they are signing up for, and do not send them other types of mail. For example, recipients who sign up to receive notifications about particular events might not appreciate your marketing mail.
- You can use double opt-in to ensure that you don't repeatedly send email to a bad address. With double opt-in, a subscriber must first request to be subscribed to your list. Then, the subscriber receives a verification email. They must click on the link in the email to confirm that they want to be subscribed.
- One way to prevent bots from signing up for your mailing list is to use CAPTCHA during your sign-up process. CAPTCHA is an automated challenge-response test that is designed to verify that a human, rather than a computer, is entering the information. For more information, see http://www.captcha.net.
- Do not use Amazon SES as a way to clean your recipient list.

Processing Bounces and Complaints

High bounce and complaint rates put your account at risk of being shut down, so you need to make sure that you have a process in place to remove recipient addresses that have bounced or complained from your recipient list. For tips on preventing the inclusion of invalid email addresses on your list, see Obtaining and Maintaining Your Recipient List (p. 178). The following guidelines pertain to handling bounces and complaints. For more detailed information, see the Amazon Simple Email Service Email Sending Best Practices white paper.

- Monitor your bounces and complaints and remove any bounced or complained recipient addresses from your mailing list. You can be notified of bounces and complaints in one of two ways: by email or by Amazon Simple Notification Service (Amazon SNS) notifications. For more information, see Using Notifications with Amazon SES (p. 105).

- You can remove bounced and complained addresses from your list manually by physically looking at the notifications and then removing the addresses from your list, but if your recipient list is large, you should probably set up an automated process. For .NET example code on how you might manage your email list using the information in Amazon SNS notifications, see Handling Bounces and Complaints on the Amazon SES blog.
- Treat suppression list bounces like any other hard bounce. Although it is possible to remove addresses from the suppression list by using the Amazon SES console, only do that if you are 100% sure that the email address is valid. In most cases, the email address is not valid, and you should remove it from your list.
- If you need to test your bounce and complaint handling process, use the Amazon SES mailbox simulator. Emails that you send to the mailbox simulator do not affect your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 158).

Using Multiple Amazon SES Accounts

When you need to send distinctly different streams of email, you can send emails through Amazon SES using multiple AWS accounts. For example, you might send marketing emails from one account and transactional emails from another account, or you might use separate accounts to send email on behalf of different clients. When you use multiple AWS accounts to send emails through Amazon SES, keep the following in mind:

- Make sure to monitor the emails you receive at the email address associated with each AWS account you are using. We send notifications about the status of your accounts, such as Amazon SES probation and suspension notices, to the email address associated with each particular AWS account. It is important to pay attention to the status of all of your accounts, because the suspension of one account puts the other accounts at risk of suspension.
- If you are sending through multiple accounts, make sure that you send (and continue to send) different types of email through your different accounts. Using multiple accounts to send very similar content can be indicative of sending spam, and puts your accounts at risk of suspension.

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Troubleshooting Amazon SES

When you use Amazon Simple Email Service (Amazon SES), you might encounter problems when you attempt to send email. The most common problems are parsing errors; however, there could be other reasons why the service cannot accept your request, or you may not be able to reach your maximum send rate. Even if your request is successful, it's still possible that your email will not be delivered due to circumstances beyond the control of Amazon SES.

This section contains the following topics that may help you when you encounter problems:

- For a list of common delivery problems that you might encounter when you send email, along with corrective actions that you can take, see Amazon SES Delivery Problems (p. 181).
- For a description of issues recipients may see when they receive an email that was sent through Amazon SES, see Problems with Emails Received from Amazon SES (p. 181).
- For a list of errors that can occur when you send an email with Amazon SES, see Amazon SES Email Sending Errors (p. 182).
- For information about domain verification problems that you might encounter, see Amazon SES Domain Verification Problems (p. 183).
- For solutions to Easy DKIM issues, see Amazon SES DKIM Problems (p. 185).
- For solutions to problems with bounce, complaint, and delivery notifications, see Amazon SES Notification Problems (p. 186).
- For information about how to remove an email address from the suppression list, see Removing an Email Address from the Amazon SES Suppression List (p. 187).
- For tips on how to increase your email sending speed when you make multiple calls to Amazon SES using either the API or the SMTP interface, see Increasing Throughput with Amazon SES (p. 188).
- For solutions to common problems that you might encounter when you use Amazon SES through its Simple Mail Transfer Protocol (SMTP) interface, see Amazon SES SMTP Issues (p. 189).
- For a list of SMTP response codes that a client application can receive from Amazon SES, see SMTP Response Codes Returned by Amazon SES (p. 191).
- For a list of error codes that are returned by the Amazon SES Query (HTTPS) API, see API Error Codes Returned by Amazon SES (p. 193).

If you are calling the Amazon SES API directly, see Amazon Simple Email Service API Reference for the HTTP errors that you might receive.

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Amazon SES Delivery Problems

After you make a successful request to Amazon SES, your message is often sent immediately. At other times, there might be a short delay. In any case, you can be assured that your email will be sent.

When Amazon SES sends your message, however, several factors can prevent it from being delivered successfully, and in some cases you will become aware that delivery failed only when the message you send does not arrive. Use the following process to resolve this situation.

If an email does not arrive, try the following:

- Verify that you made a `SendEmail` or `SendRawEmail` request for the email in question and that you received a successful response. (See Structure of a Successful Response (p. 91) for an example.) If you are making these requests programmatically, check your software logs to ensure that the program made the request and received a successful response.
- Read the blog article Three places where your email could get delayed when sending through SES because the problem might actually be a delay rather than a nondelivery.
- Check the sender's email address (the "From" address) to verify that it is valid. Also check the Return-Path address, which is where bounce messages are sent. If your mail bounced, there will be an explanatory error message there.
- Check the AWS Service Health Dashboard at http://status.aws.amazon.com to confirm that there is not a known problem with Amazon SES.
- Contact the email recipient or the recipient's ISP. Verify that the recipient is using the correct email address, and inquire whether there have been any known delivery problems with the recipient's ISP. Also, determine whether the email did arrive but was filtered as spam.
- If you have signed up for a paid AWS Support Plan, you can open a new technical support case. In your correspondence with us, please provide any relevant recipient addresses, along with any request IDs or message IDs returned from the `SendEmail` or `SendRawEmail` responses.
- Wait to see if the problem is actually a delay, not a permanent delivery failure. To combat spammers, some ISPs temporarily reject incoming messages from unknown sending mail servers. This process, called *greylisting*, can cause a delay in delivery. Amazon SES will retry these messages. If greylisting is the issue, the ISP should accept the email on one of these retry attempts.

Problems with Emails Received from Amazon SES

The following issue can arise when a recipient receives an email sent through Amazon SES. If you are looking for troubleshooting information that talks about when a recipient does not receive an email at all, see Amazon SES Delivery Problems (p. 181).

- **A recipient's email client displays "sent via amazonses.com" as the source of the email**—Some email clients display the "via" domain when the sender's domain does not match the domain that the email was actually sent from (in this case, amazonses.com). For more information on why, see this explanation from Google. As a workaround, you can set up Domain Keys Identified Mail (DKIM), which is good practice anyway. When you authenticate your emails using DKIM, email clients will typically not show the "via" domain because the DKIM signature shows that the email is from the domain it claims to be from. For information about how to set up DKIM, see Authenticating Email with DKIM in Amazon SES (p. 94).
- **Your email is not displaying correctly in a recipient's email client**
  - If your email contains non-ASCII characters, you must construct the email in Multipurpose Internet Mail Extensions (MIME) format and send it using the `SendRawEmail` API. For more information, see Sending Raw Email Using the Amazon SES API (p. 80).

- Your email might contain improperly formatted MIME. Ensure that it complies with RFC 2047. For example, it must use appropriate header fields and message body encoding.
- The recipient's email server or email client might impose limitations on the rendered content.

Amazon SES Email Sending Errors

This topic reviews the types of email sending-specific errors that you may encounter when you send an email through Amazon SES. If you try to send an email through Amazon SES and the call to Amazon SES fails, Amazon SES returns an error message to your application and does not send the email. The way that you observe this error message depends on the way that you call Amazon SES.

- If you call the Amazon SES API directly, the Query action will return an error. The error may be `MessageRejected` or one of the errors specified in the Common Errors topic of the Amazon SES API Reference.
- If you call Amazon SES using an AWS SDK that uses a programming language that supports exceptions, Amazon SES may throw an exception. The type of exception depends on the SDK and on the error. For example, the exception could be an Amazon SES `MessageRejectedException` (the actual name may vary depending on the SDK) or a general AWS exception. Regardless of the type of exception, the error type and the error message in the exception will give you more information.
- If you call Amazon SES through its SMTP interface, the way that you experience the error depends on the application. Some applications may display a specific error message, some may not. For a list of SMTP response codes, see SMTP Response Codes Returned by Amazon SES (p. 191).

> **Note**
> When your call to Amazon SES to send an email fails, you are not billed for that email.

The following are the types of Amazon SES-specific problems that can cause Amazon SES to return an error when you try to send an email. These errors are in addition to general AWS errors like `MalformedQueryString` as specified in the Common Errors topic of the Amazon SES API Reference.

- **Email address is not verified**—Your account is in the sandbox and one of the recipient email addresses has not been verified. This might apply to "Sender", "Return-Path", or "From" addresses.

  If your account is still in the sandbox, you must verify *every* recipient email address except for the recipients provided by the Amazon SES mailbox simulator. You must also verify your own "From" address. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 37) and Testing Amazon SES Email Sending (p. 158).
- **Customer is suspended**—Your AWS account has been blocked from sending email using Amazon SES. You can still access the Amazon SES console and perform any activity (e.g., view your metrics) except for email sending; if you attempt to send an email, you will receive this error message.

  If this happens, you should have received an email from Amazon SES to the email address associated with your AWS account informing you of the problem. To appeal your suspension and reinstate email sending privileges, follow the instructions in the email. You will need to explain in detail why you believe that the suspension itself was an error, or the changes you have made to ensure that the same problem does not occur again.
- **Throttling**—Amazon SES is limiting the rate at which you can send messages. Your application may be trying to send too much email, or to send email at too fast a rate. In these cases, the error may be similar to the following:
  - **Daily message quota exceeded**—You have sent the maximum number of messages that you are permitted in a 24-hour period. If you have exceeded your daily quota, you will have to wait until the next 24-hour period before you can send more email.
  - **Maximum sending rate exceeded**—You are attempting to send more emails per second than is permitted by your maximum send rate. If you have exceeded your sending rate, you can continue

to send email, but will need to reduce your send rate. For more information, see How to handle a "Throttling - Maximum sending rate exceeded" error on the Amazon SES blog.

You should regularly monitor your sending activity to see how close you are to your sending limits. For more information, see Monitoring Your Amazon SES Sending Limits (p. 122). For general information about sending limits, see Managing Your Amazon SES Sending Limits (p. 124). For information about how to increase your sending limits, see Increasing Your Amazon SES Sending Limits (p. 125).

> **Important**
> If the error text that explains the throttling error is not related to you exceeding your daily quota or maximum send rate, then there might be a system-wide problem that is causing reduced sending capabilities. For information about the service status, go to the AWS Service Health Dashboard at http://status.aws.amazon.com.

- **There are no recipients specified**—No recipients were provided.
- **You encounter problems when sending an email with a non-ASCII character in the email address**—By default, the email address string must be 7-bit ASCII. If you want it to contain any other characters, then you must use MIME encoded-word syntax (RFC 2047) instead of a literal string. To use MIME-encoded word syntax, you need to use `SendRawEmail` as described in Using MIME (p. 80). MIME encoded-word syntax uses the following form: *=?charset?encoding?encoded-text?=*. For more information, see RFC 2047.

Amazon SES Domain Verification Problems

To verify a domain with Amazon SES, you initiate the process using either the Amazon SES console or the Amazon SES API, and then publish a TXT record to your DNS server as described in Verifying Domains in Amazon SES (p. 40). This section contains the following topics that might help you if you encounter problems:

- To verify that the TXT record is correctly published to your DNS server, see How to Check Domain Verification Settings (p. 183).
- For some common problems you may encounter when you attempt to verify your domain with Amazon SES, see Common Domain Verification Problems (p. 184).

How to Check Domain Verification Settings

You can check that your Amazon SES domain verification TXT record is published correctly to your DNS server by using the following procedure. This procedure uses the nslookup tool, which is available for Windows and Linux. On Linux, you can also use dig.

The commands in these instructions were executed on Windows 7, and the example domain we use is *ses-example.com*.

In this procedure, you first find the DNS servers that serve your domain, and then query those servers to view the TXT records. You query the DNS servers that serve your domain because those servers contain the most up-to-date information for your domain, which can take time to propagate to other DNS servers.

To verify that your domain verification TXT record is published to your DNS server

1. Find the name servers for your domain by taking the following steps.

   a. Go to the command line. To get to the command line on Windows 7, click **Start** and then type **cmd**. On Linux-based operating systems, open a terminal window.

   b. At the command prompt, type the following, where <*domain*> is your domain. This will list all of the name servers that serve your domain.

      ```
      nslookup -type=NS <domain>
      ```

      If your domain was *ses-example.com*, this command would look like:

      ```
      nslookup -type=NS ses-example.com
      ```

      The command's output will list the name servers that serve your domain. You will query one of these servers in the next step.

2. Verify that the TXT record is correctly published by taking the following steps.

   a. At the command prompt, type the following, where <*domain*> is your domain, and <*name server*> is one of the name servers you found in step 1.

      ```
      nslookup -type=TXT  _amazonses.<domain> <name server>
      ```

      In our *ses-example.com* example, if a name server that we found in step 1 was called *ns1.name-server.net*, we would type the following:

      ```
      nslookup -type=TXT  _amazonses.ses-example.com ns1.name-server.net
      ```

   b. In the output of the command, verify that the string that follows `text =` matches the TXT value you see when you click the domain in the Verified Senders list of the Amazon SES console.

      In our example, we are looking for a TXT record under *_amazonses.ses-example.com* with a value of `fmxqxT/icOYx4aA/bEUrDPMeax9/s3frblS+niixmqk=`. If the record is correctly published, we would expect the command to have the following output:

      ```
      _amazonses.ses-example.com text = "fmxqxT/icOYx4aA/bEUrDP
      Meax9/s3frblS+niixmqk="
      ```

Common Domain Verification Problems

If you attempt to verify a domain using the procedure in Verifying Domains in Amazon SES (p. 40) and you encounter problems, review the possible causes and solutions below.

- **Your DNS provider does not allow underscores in TXT record names**—You can omit the *_amazonses* from the TXT record name.
- **You want to verify the same domain multiple times and you can't have multiple TXT records with the same name**— You might need to verify your domain more than once because you're sending in different regions or you're sending from multiple AWS accounts from the same domain in the same region. If your DNS provider does not allow you to have multiple TXT records with the same name, there are two workarounds. The first workaround, if your DNS provider allows it, is to assign multiple values to the TXT record. For example, if your DNS is managed by Amazon Route 53, you can set up multiple values for the same TXT record as follows:

1. In the Amazon Route 53 console, click the _*amazonses* TXT record you added when you verified your domain in the first region.
2. In the **Value** box, press Enter after the first value.
3. Add the value for the additional region, and save the record set.

The other workaround is that if you only need to verify your domain twice, you can verify it once with _*amazonses* in the TXT record name and the other time you can omit _*amazonses* from the record name entirely. We recommend the previous solution as a best practice, however.

- **Your email address is provided by a web-based email service you do not have control over**—You cannot successfully verify a domain that you do not own. For example, if you want to send email through Amazon SES from a gmail address, you need to verify that email address specifically; you cannot verify gmail.com. For information about individual email address verification, see Verifying Email Addresses in Amazon SES (p. 37).
- **Amazon SES reports that domain verification failed**—You receive a "Domain Verification Failure" email from Amazon SES, and the domain displays a status of "failed" in the **Domains** tab of the Amazon SES console. This means that Amazon SES cannot find the necessary TXT record on your DNS server. Verify that the required TXT record is correctly published to your DNS server by using the procedure in How to Check Domain Verification Settings (p. 183), and look for the following possible errors:
  - **Your DNS provider appended the domain name to the end of the TXT record**—Adding a TXT record that already contains the domain name (such as _*amazonses.example.com*) may result in the duplication of the domain name (such as _*amazonses.example.com.example.com*). To avoid duplication of the domain name, add a period to the end of the domain name in the TXT record. This will indicate to your DNS provider that the record name is fully qualified (that is, no longer relative to the domain name), and prevent the DNS provider from appending an additional domain name.
- **You receive an email from Amazon SES that says your domain verification has been (or will be) revoked**— Amazon SES can no longer find the required TXT record on your DNS server. The notification email will inform you of the length of time in which you must re-publish the TXT record before your domain verification status is revoked.

  Note
  You can review the required TXT record information in the Amazon SES console by using the following instructions. In the navigation pane, under **Identities**, click **Domains**. In the list of domains, click (not just expand) the domain to display the domain verification settings, which include the TXT record name and value.

  If your domain verification status is revoked, you must restart the verification procedure in Verifying Domains in Amazon SES (p. 40) from the beginning, just as if the revoked domain were an entirely new domain. After you publish the TXT record to your DNS server, verify that the TXT record is correctly published by using How to Check Domain Verification Settings (p. 183).

Amazon SES DKIM Problems

If you attempt to set up Easy DKIM using the procedure in Easy DKIM in Amazon SES (p. 94) and you encounter problems, review the possible causes and solutions below.

- **You set up Easy DKIM successfully, but your messages are not being DKIM-signed**—Possible problems are:
  - Make sure that you have enabled Easy DKIM for the appropriate verified sending identity. To enable Easy DKIM for a verified sender in the Amazon SES console, click the email address or domain in the Verified Senders list. On the Details page for the email address or domain, expand **DKIM**, and then click **Enable** to enable DKIM.
  - You could be sending from an individually verified email address that does not have DKIM-signing enabled. If you set up Easy DKIM for a domain, it will apply to all email addresses in that domain *except* for email addresses that you individually verified. Individually verified email addresses use separate settings. If this is your issue, either remove the email address from your verified identity list

(its settings will then be inherited from the verified domain's settings) or enable Easy DKIM for the email address as explained above.

  - If you are using Amazon SES in multiple regions or with multiple AWS accounts, you must perform the Easy DKIM set-up procedure described in Easy DKIM in Amazon SES (p. 94) for each region and account for which you want to use Easy DKIM. Amazon SES will generate a unique set of DNS records for each domain/account/region combination. You will need to add all of these records to your DNS server and enable DKIM-signing for the appropriate verified sending identities. If you remove the necessary DNS records for a specific region or account, Amazon SES will disable DKIM signing only for that account in that region, and notify you by email so that you can take action.

- **Your domain's DKIM details in the Amazon SES console show *DKIM: waiting on sender verification... DKIM Verification Status: pending verification***—Your DKIM status is pending, which means that Amazon SES has not yet detected the required CNAME records on your DNS server, which you should have published during the Easy DKIM setup procedure (Easy DKIM in Amazon SES (p. 94)). If your DKIM status is pending, see the following articles on the Amazon SES blog:
  - DKIM Troubleshooting Series: Your DKIM Status is Pending
  - DKIM Troubleshooting Series: Your DKIM Status is Still Pending
- **When queried, your DNS servers successfully return the Amazon SES DKIM CNAME records, but return SERVFAIL for the TXT records**—Your DNS provider might have problems redirecting CNAME records. Note that Amazon SES and ISPs query for TXT records. To comply with the DKIM specification, your DNS servers must be able to respond to TXT record queries as well as CNAME record queries. If your DNS provider cannot respond to TXT record queries, an alternative is to use Amazon Route 53 for your DNS hosting.
- **Your emails are being DKIM-signed, but the DKIM signature is not validating**—See DKIM Troubleshooting Series: Why is My Signature Not Validating? on the Amazon SES blog.
- **You receive an email from Amazon SES that says your DKIM setup has been (or will be) revoked**—This means that Amazon SES can no longer find the required CNAME records on your DNS server. The notification email will inform you of the length of time in which you must re-publish the CNAME records before your DKIM setup status is revoked and DKIM signing is disabled. If your DKIM setup is revoked, you must restart the DKIM setup procedure in Easy DKIM in Amazon SES (p. 94) from the beginning.
- **You did not enable DKIM, yet your message headers contain a DKIM signature**—If you did not enable DKIM, your messages are not DKIM-signed. The DKIM signature you are seeing has *d=amazonses.com* and is automatically added by Amazon SES.
- **Your emails contain two DKIM signatures**—The extra DKIM signature, which has *d=amazonses.com*, is automatically added by Amazon SES. You can ignore it.

Amazon SES Notification Problems

If you encounter a problem with bounce, complaint, or delivery notifications, review the possible causes and solutions below.

- **You receive bounce notifications via Amazon SNS, but you don't know which recipients the notifications correspond to.** In the future, to associate a bounce notification with a given recipient, you have the following options:
  - Since Amazon SES doesn't retain any custom message IDs that you have added, store a mapping between an identifier and the Amazon SES message ID that Amazon SES passes back to you when it accepts the email.
  - In each call to Amazon SES, send to a single recipient, rather than sending a single message to multiple recipients.
  - You can enable feedback forwarding via email, which will forward the full bounce message to you.
- **You receive complaint notifications via Amazon SNS or email feedback forwarding, but you don't know which recipients the notifications correspond to.** Some ISPs redact the complained

recipient's email address before passing the complaint notification to Amazon SES. To enable you to find the recipient's email address, your best option is to store your own mapping between an identifier and the Amazon SES message ID that Amazon SES passes back to you when it accepts the email. Note that Amazon SES does not retain any custom message IDs that you add.

- **You want to set up notifications to go to an Amazon SNS topic you don't own.** The owner of that topic must configure an Amazon SNS access policy that allows your account to call the `SNS:Publish` action on their topic. For information about how to control access to your Amazon SNS topic through the use of IAM policies, see Managing Access to Your Amazon SNS Topics.

Removing an Email Address from the Amazon SES Suppression List

Amazon SES maintains a suppression list (formerly called a blacklist) of recipient email addresses that have caused a hard bounce for any Amazon SES customer within the past 14 days. If you try to send an email through Amazon SES to an address that is on the suppression list, the call to Amazon SES succeeds, but Amazon SES treats the email as a hard bounce instead of attempting to send it. Like any hard bounce, suppression list bounces count towards your sending quota and your bounce rate.

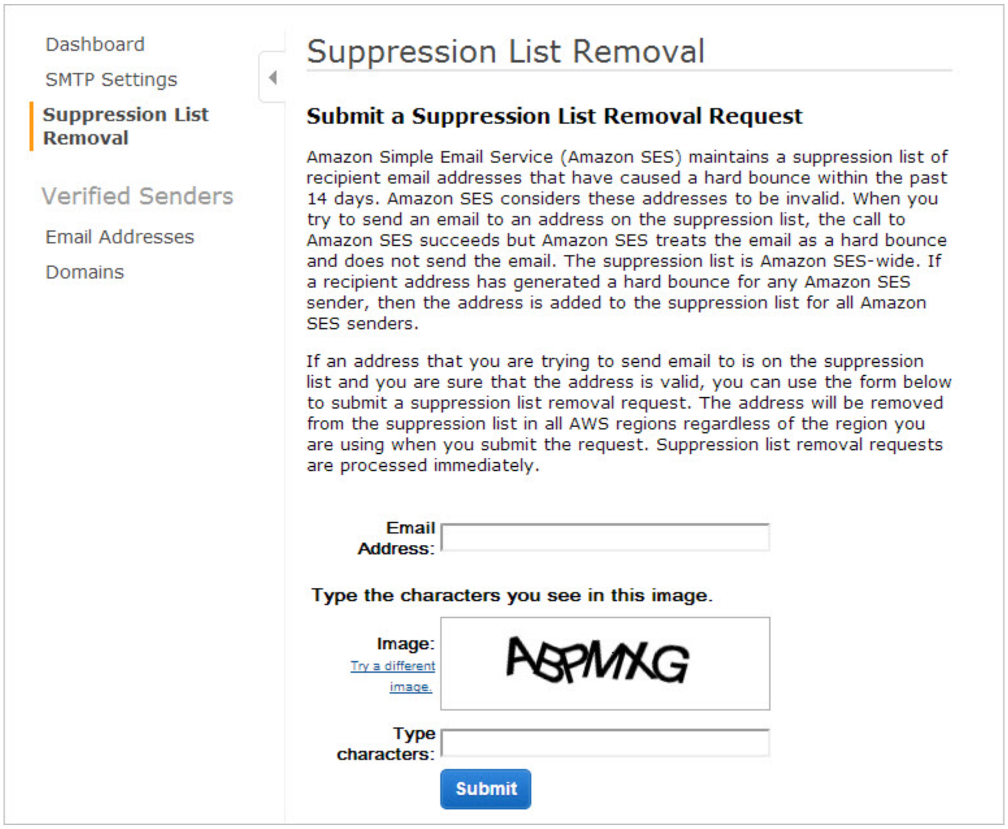
The only way you will know if an address is on the suppression list is that you will receive a suppression list bounce when you send to it. There is no way to query the suppression list in advance.

> **Important**
> As with any email address that hard bounces, you should remove addresses that cause a suppression list bounce from your mailing list unless you are absolutely sure the address is valid, because suppression list bounces count towards your bounce rate and a high bounce rate puts your account at risk of being shut down. If you remove an address from the suppression list when it is indeed undeliverable, then the next time you or another Amazon SES customer sends an email to that address, it will hard bounce and the address will go back on the suppression list.

If you are sure that an address on the suppression list is valid, you can remove it from the list by using the following procedure. Although each AWS region has a separate suppression list, if you remove an address from the suppression list of one region, the address is removed from the suppression list of all regions.

To remove an email address from the suppression list

1. Sign in to the AWS Management Console and open the Amazon SES console at https://console.aws.amazon.com/ses.
2. In the Navigation pane, click **Suppression List Removal**.



Suppression List Removal

Submit a Suppression List Removal Request

Amazon Simple Email Service (Amazon SES) maintains a suppression list of recipient email addresses that have caused a hard bounce within the past 14 days. Amazon SES considers these addresses to be invalid. When you try to send an email to an address on the suppression list, the call to Amazon SES succeeds but Amazon SES treats the email as a hard bounce and does not send the email. The suppression list is Amazon SES-wide. If a recipient address has generated a hard bounce for any Amazon SES sender, then the address is added to the suppression list for all Amazon SES senders.

If an address that you are trying to send email to is on the suppression list and you are sure that the address is valid, you can use the form below to submit a suppression list removal request. The address will be removed from the suppression list in all AWS regions regardless of the region you are using when you submit the request. Suppression list removal requests are processed immediately.

Email Address:

Type the characters you see in this image.

Image: Try a different image.

Type characters:

3. In the **Email Address** field, type the email address that you want to remove from the suppression list.
4. In the **Type characters** field, type the characters that you see in the image above it.
5. Click **Submit**.

After you submit the form, you can fill out the form for another email address. Suppression list removal requests are processed immediately.

Increasing Throughput with Amazon SES

When you send emails, you can call Amazon SES as frequently as your maximum send rate allows. (For more information about your maximum send rate, see Managing Your Amazon SES Sending Limits (p. 124).) However, each call to Amazon SES takes time to complete.

If you are making multiple calls to Amazon SES using the Amazon SES API or the SMTP interface, you may want to consider the following tips to help you improve your throughput:

- **Measure your current performance to identify bottlenecks**—A possible performance test involves sending multiple test emails as quickly as possible within a code loop in your application. Measure the round-trip latency of each `SendEmail` request. Then, incrementally launch additional instances of the application on the same machine, and watch for any impact on network latency. You may also want to run this test on multiple machines and on different networks to help pinpoint any possible machine resource bottlenecks or network bottleneck that may exist.

- **(API only) Consider using persistent HTTP connections**—Rather than incurring the overhead of establishing a separate new HTTP connection for each API request, use persistent HTTP connections. That is, reuse the same HTTP connection for multiple API requests.
- **Consider using multiple threads**—When an application uses a single thread, the application code calls the Amazon SES API and then synchronously waits for an API response. Sending emails is typically an I/O-bound operation, and doing the work from multiple threads provides better throughput. You can send concurrently using as many threads of execution as you wish.
- **Consider using multiple processes**—Using multiple processes can help increase your throughput because you will have more concurrent active connections to Amazon SES. For example, you can segment your intended emails into multiple buckets, and then run multiple instances of your email sending script simultaneously.
- **Consider using a local mail relay**—Your application can quickly transmit messages to your local mail server, which can then help to buffer the messages and asynchronously transmit them to Amazon SES. Some mail servers support delivery concurrency, which means that even if your application is generating emails to the mail server in a single-threaded fashion, the mail server will use multiple threads when sending to Amazon SES. For more information, see Integrating Amazon SES with Your Existing Email Server (p. 58).
- **Consider hosting your application closer to the Amazon SES API endpoint**—You may wish to consider hosting your application in a data center close to the Amazon SES API endpoint, or on an Amazon EC2 instance in the same AWS Region as the Amazon SES API endpoint. This may help to decrease network latency between your application and Amazon SES, and improve throughput. For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 173).
- **Consider using multiple machines**—Depending on the system configuration on your host machine, there may be a limit on the number of simultaneous HTTP connections to a single IP address, which may limit the benefits of parallelism once you exceed a certain number of concurrent connections on a single machine. If this is a bottleneck, you may wish to consider making concurrent Amazon SES requests using multiple machines.
- **Consider using the Amazon SES query API instead of the SMTP endpoint**—Using the Amazon SES query API enables you to submit the email send request using a single network call, whereas interfacing with the SMTP endpoint involves an SMTP conversation which consists of multiple network requests (for example, EHLO, MAIL FROM, RCPT TO, DATA, QUIT). For more information about the Amazon SES query API, see Using the Amazon SES API to Send Email (p. 79).
- **Use the Amazon SES mailbox simulator to test your maximum throughput**—To test any changes you may implement, you can use the mailbox simulator. The mailbox simulator can help you to determine your system’s maximum throughput without using up your daily sending quota. For information about the mailbox simulator, see Testing Amazon SES Email Sending (p. 158).

If you are accessing Amazon SES through its SMTP interface, see Amazon SES SMTP Issues (p. 189) for specific SMTP-related issues that may affect throughput.

Amazon SES SMTP Issues

If you are having problems sending email through the Amazon SES Simple Mail Transfer Protocol (SMTP) interface, review the possible causes and solutions below. For general information about sending email through the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 48).

- **You are unable to connect to the Amazon SES SMTP endpoint**
  - Verify that you are using the right credentials. Your SMTP credentials are different than your AWS credentials. To obtain your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 50). For more information about credentials, see Using Credentials With Amazon SES (p. 170).

  - Your network might be blocking outbound connections over the port you're trying to send email from. Try the following command: `telnet email-smtp.us-west-2.amazonaws.com <port>`, where <port> is the port you're trying to use (typically 25, 465, 587, or 2587). If that works, and you are trying to connect to Amazon SES using TLS Wrapper or STARTTLS, try the openssl commands shown in Using the Command Line to Send Email Through the Amazon SES SMTP Interface (p. 76). If you cannot connect to the Amazon SES SMTP endpoint using telnet or openssl, then something in your network (for example, a firewall) is blocking outbound connections over the port you're trying to use. Work with your network administrator to diagnose and fix the problem.
- **You are sending to Amazon SES from an Amazon EC2 instance via port 25 and you cannot reach your Amazon SES sending limits or you are receiving time outs**—Amazon EC2 imposes default sending limits on email sent via port 25 and throttles outbound connections if you attempt to exceed those limits. To remove these limits, submit a Request to Remove Email Sending Limitations. You can also connect to Amazon SES via port 465 or port 587, neither of which is throttled.
- **Network errors are causing dropped emails**—Ensure that your application uses retry logic when it connects to the Amazon SES SMTP endpoint, and that your application can detect and retry message delivery in case of a network error. SMTP is a verbose protocol and submitting an email using this protocol requires several network round trips. Because of the nature of this protocol, the potential of transient network errors increases. A message is accepted by Amazon SES for delivery only when Amazon SES responds with an Amazon SES message ID.
- **You lose connection with the SMTP endpoint**
  - If you receive a time-out error message, the maximum transmission unit (MTU) size on the network interface of the computer you're using to connect to the Amazon SES SMTP interface might be too large. To mitigate this, you can try setting the MTU size on that computer to 1500. For instructions on how to set the MTU size on Microsoft Windows, Linux, and Mac OS X operating systems, see Queries Appear to Hang in the Client and Do Not Reach the Cluster in the Amazon Redshift documentation. Users connecting to Amazon SES from an Amazon EC2 instance can alternatively try the workaround described in Security Group Rules for Path MTU Discovery in the Amazon EC2 documentation.
  - Do not attempt to maintain long-lived connections with the Amazon SES SMTP endpoint. The Amazon SES SMTP endpoint runs on a fleet of Amazon EC2 instances behind an Elastic Load Balancer (ELB). In order to ensure that the system is up-to-date and fault tolerant, active Amazon EC2 instances are periodically terminated and replaced with new instances. Because your application connects to an Amazon EC2 instance through the ELB, the connection becomes invalid when the Amazon EC2 instance is terminated. You should establish a new SMTP connection after you have delivered a fixed number of messages via a single SMTP connection, or if the SMTP connection has been active for some amount of time. You will need to experiment to find appropriate thresholds depending on where your application is hosted and how it submits email to Amazon SES.
- **You want to know the IP addresses of the Amazon SES SMTP mail servers so that you can whitelist the IP addresses with your network**—We are unable to provide a specific set of IP addresses for the Amazon SES SMTP endpoints because they reside behind load balancers and the IP addresses can change frequently. We recommend that you only whitelist based on DNS and not static IP addresses.
- **You are integrating Amazon SES with a Sendmail or Postfix mail server using the instructions in Integrating Amazon SES with Your Existing Email Server (p. 58), and your mail server cannot authenticate with the Amazon SES SMTP endpoint because the hostname does not match.**—In this case, try the following steps.
  - **Sendmail**—In Step 1 of Integrating Amazon SES with Sendmail (p. 63), put the following additional line in */etc/mail/authinfo*, depending on the AWS region of the Amazon SES endpoint you are using. Note that you must replace USERNAME and PASSWORD with your SMTP user name and password.

Region name	Add this line to */etc/mail/authinfo*
US East (N. Virginia)	`AuthInfo:ses-smtp-prod-335357831.us-east-1.elb.amazonaws.com "U:root" "I:USERNAME" "P:PASSWORD" "M:LOGIN"`

Region name	Add this line to */etc/mail/authinfo*
US West (Oregon)	`AuthInfo:ses-smtp-us-west-2-prod-14896026.us-west-2.elb.amazonaws.com "U:root" "I:USERNAME" "P:PASSWORD" "M:LOGIN"`
EU (Ireland)	`AuthInfo:ses-smtp-eu-west-1-prod-345515633.eu-west-1.elb.amazonaws.com "U:root" "I:USERNAME" "P:PASSWORD" "M:LOGIN"`

In Step 4 of Integrating Amazon SES with Sendmail (p. 63), add the following to */etc/mail/access*:

Region name	Add this line to */etc/mail/access*
US East (N. Virginia)	`Connect:ses-smtp-prod-335357831.us-east-1.elb.amazonaws.com RELAY`
US West (Oregon)	`Connect:ses-smtp-us-west-2-prod-14896026.us-west-2.elb.amazonaws.com RELAY`
EU (Ireland)	`Connect:ses-smtp-eu-west-1-prod-345515633.eu-west-1.elb.amazonaws.com RELAY`

- **Postfix**—In Step 3 of Integrating Amazon SES with Postfix (p. 60), put the following additional line in */etc/postfix/sasl_passwd*, depending on the AWS region of the Amazon SES endpoint you are using. Note that you must replace USERNAME and PASSWORD with your SMTP user name and password.

Region name	Add this line to */etc/postfix/sasl_passwd*
US East (N. Virginia)	`ses-smtp-prod-335357831.us-east-1.elb.amazonaws.com:25 USERNAME:PASSWORD`
US West (Oregon)	`ses-smtp-us-west-2-prod-14896026.us-west-2.elb.amazonaws.com:25 USERNAME:PASSWORD`
EU (Ireland)	`ses-smtp-eu-west-1-prod-345515633.eu-west-1.elb.amazonaws.com:25 USERNAME:PASSWORD`

SMTP Response Codes Returned by Amazon SES

SMTP response codes that Amazon SES returns are listed in the following table. Note that 4xx errors indicate a temporary issue.

Note

The way in which errors are handled depends on the SMTP client that you use; some SMTP clients may not display error codes at all.

Description	Response code	More information
Authentication successful	235 Authentication successful	N/A
Successful delivery	250 Ok <*Message ID*>	<*Message ID*> is a string of characters that Amazon SES uses to uniquely identify a message.
Daily sending quota exceeded	454 Throttling failure: Daily message quota exceeded	You have exceeded the maximum number of emails that Amazon SES permits you to send in a 24-hour period. For more information, see Managing Your Amazon SES Sending Limits (p. 124).
Maximum send rate exceeded	454 Throttling failure: Maximum sending rate exceeded	You have exceeded the maximum number of emails that Amazon SES permits you to send per second. For more information, see Managing Your Amazon SES Sending Limits (p. 124).
Amazon SES issue when validating SMTP credentials	454 Temporary authentication failure	Possible reasons include, but are not limited to: • There is a problem with the encryption between your email-sending application and Amazon SES. Note that you need to use an encrypted connection when you connect to Amazon SES. For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 53). • Amazon SES could be experiencing an issue. Check the Service Health Dashboard for updates.
Problem receiving the request	454 Temporary service failure	Amazon SES did not successfully receive the request and therefore did not send the message. Please retry the request.
Incorrect credentials	530 Authentication required	Your email-sending application did not attempt to authenticate with Amazon SES when it tried to connect to the Amazon SES SMTP interface. For an example of how to set up an email-sending application to authenticate with Amazon SES, see Configuring Email Clients to Send Through Amazon SES (p. 54).

Description	Response code	More information
Authentication Credentials Invalid	535 Authentication Credentials Invalid	Your email-sending application did not provide the correct SMTP credentials to Amazon SES. Note that your SMTP credentials are not the same as your AWS credentials. For more information, see Obtaining Your Amazon SES SMTP Credentials (p. 50).
Account not subscribed to Amazon SES	535 Account not subscribed to SES	The AWS account that owns the SMTP credentials is not signed up for Amazon SES. To sign up, go to Manage Your Account.
User not authorized to call the Amazon SES SMTP endpoint	554 Access denied: User <User ARN> is not authorized to perform ses:SendRawEmail on resource <Identity ARN>	The AWS Identity and Access Management (IAM) policy or the Amazon SES sending authorization policy of the user who owns the SMTP credentials is not allowed to call the Amazon SES SMTP endpoint. For information about how to get SMTP credentials for an existing IAM user, see Obtaining Amazon SES SMTP Credentials by Converting AWS Credentials (p. 51).
Unverified email address	554 Message rejected: Email address is not verified	You are trying to send email from an email address or domain that you have not verified with Amazon SES. If your account is still in the sandbox, you also need to verify the recipient address. For more information, see Verifying Email Addresses in Amazon SES (p. 37). Note that Amazon SES has endpoints in multiple AWS regions, and email address verification status is separate for each AWS region. You must complete the verification process for each sender in the AWS region(s) you want to use. For information about using Amazon SES in multiple AWS regions, see Regions and Amazon SES (p. 173).

API Error Codes Returned by Amazon SES

Error codes that are returned by the Amazon SES Query (HTTPS) API are listed in the following table. For more information about the Amazon SES API, see Amazon Simple Email Service API Reference.

Error	Description	HTTPS Status Code	Actions That Return This Code
MessageRejected	Indicates that the action failed, and the message could not be sent. Check the error stack for a description of what caused the error. For more information about problems that can cause this error, see Amazon SES Email Sending Errors (p. 182).	400	`SendEmail`, `SendRawEmail`
IncompleteSignature	The request signature does not conform to AWS standards.	400	All
InternalFailure	The request processing has failed because of an unknown error, exception, or failure.	500	All
InvalidAction	The requested action or operation is invalid. Verify that the action is typed correctly.	400	All
InvalidClientTokenId	The X.509 certificate or AWS access key ID provided does not exist in our records.	403	All
InvalidParameterCombination	Parameters that must not be used together were used together.	400	All
InvalidParameterValue	An invalid or out-of-range value was supplied for the input parameter.	400	All
InvalidQueryParameter	The AWS query string is malformed, does not adhere to AWS standards.	400	All
MalformedQueryString	The query string contains a syntax error.	404	All
MissingAction	The request is missing an action or a required parameter.	400	All
MissingAuthenticationToken	The request must contain either a valid (registered) AWS access key ID or X.509 certificate.	403	All

Error	Description	HTTPS Status Code	Actions That Return This Code
MissingParameter	A required parameter for the specified action is not supplied.	400	All
OptInRequired	The AWS access key ID needs a subscription for the service.	403	All
RequestExpired	The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.	400	All
ServiceUnavailable	The request failed due to a temporary failure of the server.	503	All
Throttling	The request was denied due to request throttling.	400	All

Amazon SES Enforcement FAQs

If the emails you send result in excessive bounces, complaints, or other issues, your sending abilities might be placed on probation or suspended. This process is called *enforcement*. In these cases, you will receive a notification at the email address associated with your AWS account.

This section contains FAQs on the following enforcement-related topics:

- Probations (p. 195)
- Suspensions (p. 198)
- Bounces (p. 199)
- Complaints (p. 201)
- Spamtraps (p. 204)
- Manual Investigations (p. 205)

Amazon SES Probation FAQ

Q1. I received a probation notice. What does that mean?

We have detected a significant issue with the sending on your account and we're giving you time to fix it. You can still send normally for now, but if you don't fix the problem in the allotted timeframe or sending allowance, your Amazon SES sending privileges might be suspended.

Q2. Will I always be notified if I am put on probation?

Yes. You will receive a notification at the email address of the AWS account associated with the Amazon SES probation.

Q3. Will the Amazon SES probation affect my use of other AWS services?

No.

Q4. What should I do if I'm on probation?

You should do the following:

- If your situation allows it, stop sending mail until you fix the problem. Although probation does not affect your ability to send mail through Amazon SES, if you continue to send mail without first making changes, you are putting your continued sending at risk.
- Look at the email you received from us for a summary of the issue.
- Investigate your sending to determine what aspect of your sending specifically triggered the issue.
- Once you have made your fixes, reply to the probation notification telling us about the fixes you made (see Q6. How do I submit an appeal? (p. 196)). If you cannot find the probation notification, send your appeal to *ses-enforcement@amazon.com* from the email address associated with your AWS account. Note that you should appeal your probation only after you've made your changes—do not submit an appeal outlining changes you plan to make. If you do, we will ask you to contact us again once the fixes are actually in place. If we find that you have fixed the problem, we'll take you off probation.
- Be sure to provide any information we specifically request. We need this information to evaluate your case.

Q5. What is an appeal?

An appeal is when you reply to a probation or suspension notification (or email *ses-enforcement@amazon.com* from the email address associated with your AWS account) and provide the specific information we need to determine whether we can remove the probation or suspension. For a list of information to provide, see Q6. How do I submit an appeal? (p. 196).

Q6. How do I submit an appeal?

Just reply to the probation notification. If you cannot find the probation notification, send your appeal to *ses-enforcement@amazon.com* from the email address associated with your AWS account. In your appeal, you should explain in as much detail as possible the following three things:

- An explanation of how and why you think the problem occurred.
- A list of changes you have already made to address the issue (not changes you plan to make).
- An explanation of why you believe these changes will prevent the problem from happening again.

Please read the FAQ specific to your issue (for example, bounces) to see if there is additional information you need to provide in your appeal.

> **Note**
> Failure to provide this information will delay the appeal process because we will request the remaining information before we can make a decision. In addition, be sure to provide any additional information we specifically request during the appeal correspondence.

Q7. What if my appeal isn't accepted?

We will respond to you explaining why your appeal wasn't accepted, and you will often have the option of appealing again after you address the issue. For example, we might ask you for more information, and once you provide the information, your appeal might be accepted. As another example, if you tell us how you will fix the problem and haven't actually fixed it, we'll ask you to contact us again once you've actually fixed the issue.

Q8. Can you help me diagnose the problem?

Typically we can give you only a high-level overview of your issue (for example, that you have a problem with bounces). You will need to investigate the root cause on your end.

Q9. How will I know if I'm off probation?

We will provide this information in our response to your appeal, or in some cases you will receive an automated notification at the email address associated with your AWS account. The notification will indicate either that you're off probation, or that your account has been suspended because you haven't fixed the problem.

Q10. Will I always have a probation period if there's a problem?

No. There are two cases in which you might not be provided a probation period:

- If your sending problem is very severe, you might be immediately suspended. As with any suspension, we will send you a notification at that time.
- If you show a pattern of repeated probations, eventually you might be suspended rather than being put on probation again. For this reason, it is important to address the underlying problem rather than just the specific incident that caused a specific probation. For instance, if a particular campaign triggers a probation, you must do more than simply stop that campaign. You need to determine which properties of the campaign were problematic and ensure that you have processes in place so that your future campaigns won't have the same issue.

Q11. What if I make my fixes shortly before the probation is due to expire?

Contact us through the appeal process to let us know that you fixed the problem.

Q12. Can I get help from my AWS representative or Premium Support?

If you are actively working with an AWS account representative, we will contact him or her regarding your probation, and he or she might be able to help you to better understand the problem. Feel free to contact your representative directly as well. If you are signed up for Premium Support, you might also want to engage that team to help with this process.

Amazon SES Suspension FAQ

Q1. I received a suspension notice. What does that mean?

We had to shut down your account due to a critical issue with your sending, so you can no longer send emails. There are three scenarios in which a suspension can occur:

- You are put on probation for a sending problem (for example, bounces), the probation expires, and the issue has not been resolved.
- Your problem is so severe that you were immediately suspended without a probation period.
- You have a history of repeated probations for a particular issue, and the issue reoccurred.

Q2. Will I always be notified if I am suspended?

Yes. You will receive a notification at the email address of the AWS account associated with the Amazon SES suspension.

Q3. Will the Amazon SES suspension affect my use of other AWS services?

No.

Q4. What should I do if my account has been suspended?

You should do the following:

- Look at the email you received from us for a summary of the issue.
- Investigate your sending to determine what aspect of your sending specifically triggered the issue.
- Once you have fixed the issue, reply to the suspension notification telling us about the fixes you made (see Q6. How do I submit an appeal? (p. 198)). If you cannot find the suspension notification, send your appeal to *ses-enforcement@amazon.com* from the email address associated with your AWS account. Note that you should appeal your suspension only after you've made your changes—do not submit an appeal outlining changes you plan to make. If you do, we will ask you to contact us again once the fixes are actually in place.
- Be sure to provide any information we specifically request. We need this information to evaluate your case.

Q5. What is an appeal?

An appeal is when you reply to a probation or suspension notification (or email *ses-enforcement@amazon.com* from the email address associated with your AWS account) and provide the specific information we need to determine whether we can remove the probation or suspension. For a list of information to provide, see Q6. How do I submit an appeal? (p. 198).

Q6. How do I submit an appeal?

Just reply to the suspension notification. If you cannot find the suspension notification, send your appeal to *ses-enforcement@amazon.com* from the email address associated with your AWS account. In your appeal, you should explain in as much detail as possible the following three things:

- An explanation of how and why you think the problem occurred.
- A list of changes you have already made to address the issue (not changes you plan to make).

- An explanation of why you believe these changes will prevent the problem from happening again.

Read the FAQ specific to your issue (for example, bounces) to see if there is additional information you need to provide in your appeal.

> **Note**
> Failure to provide this information will delay the appeal process because we will request the remaining information before we can make a decision. In addition, be sure to provide any additional information we specifically request during the appeal correspondence.

Q7. What if my appeal isn't accepted?

We will respond to you explaining why your appeal wasn't accepted, and you will often have the option of appealing again after you address the issue. For example, we might ask you for more information, and once you provide the information, your appeal might be accepted. As another example, if you tell us how you will fix the problem and haven't actually fixed it, we'll ask you to contact us again once you've actually fixed the issue.

Q8. Can you help me diagnose the problem?

Typically we can give you only a high-level overview of your issue (for example, that you have a problem with bounces). You will need to investigate the root cause on your end.

Q9. How will I know if my account has been reinstated?

We will provide this information in our response to your appeal, or in some cases you will receive an automated notification at the email address associated with your AWS account. You can also try sending an email to yourself through Amazon SES (for example, using the Amazon SES console). If the attempt is successful, then you have been reinstated.

Q10. Can I get help from my AWS representative or Premium Support?

If you are actively working with an AWS account representative, we will contact him or her regarding your probation, and he or she might be able to help you to better understand the problem. Feel free to contact your representative directly as well. If you are signed up for Premium Support, you might also want to engage that team to help with this process.

Amazon SES Bounce FAQ

Q1. Why do you care about my bounces?

High bounce rates are often used by entities such as ISPs, mailbox providers, and anti-spam organizations as indicators that senders are engaging in low-quality email-sending practices and their email should be blocked or sent to the spam folder.

Q2. What should I do if I receive a probation or suspension notice for my bounce rate?

Fix the underlying problem and appeal to get your case reevaluated. For information about the appeal process, see the FAQs on probation and suspension. In your appeal, in addition to the information requested in the probation and suspension FAQs, tell us the following:

- The method you use to track your bounces

- How you ensure that the email addresses of new recipients are valid prior to sending to them. For example, which of the recommendations are you following in Q11. What can I do to minimize bounces? (p. 201)

Q3. What types of bounces count toward my bounce rate?

Your bounce rate includes only hard bounces to domains you have not verified. We do not include emails to your verified domains in the bounce rate calculation.

Q4. Do you disclose the Amazon SES bounce rate limits that trigger probation and suspension?

No, but you can find general bounce rate guidelines and tips on how to avoid bounces in the Amazon Simple Email Service Email Sending Best Practices white paper.

Q5. Over what period of time is my bounce rate calculated?

We don't calculate your bounce rate based on a fixed period of time, because different senders send at different rates. Instead, we look at what is called a *representative volume*, which represents a reasonable amount of mail with which to evaluate your sending practices. To be fair to both high-volume and small-volume senders, the representative volume is different for each user and changes as the user's sending patterns change.

Q6. Can I calculate my own bounce rate by using the information from the Amazon SES console or the GetSendStatistics API?

No. The bounce rate is calculated using representative volume (see Q5. Over what period of time is my bounce rate calculated? (p. 200)). Depending on your sending rate, your bounce rate can stretch farther back in time than the Amazon SES console or GetSendStatistics can retrieve. In addition, only emails to non-verified domains are considered when calculating your bounce rate. However, if you regularly monitor your bounce rates using those methods, you should still have a good indicator that you can use to catch problems before they get to levels that trigger a probation or suspension.

Q7. How can I find out which email addresses bounced?

Examine the bounce notifications that Amazon SES sends you. The email address to which Amazon SES forwards the notifications depends on how you sent the original messages, as described at Amazon SES Notifications Through Email (p. 106). You can also set up bounce notifications through Amazon Simple Notification Service (SNS), as described at Using Notifications with Amazon SES (p. 105). Note that simply removing bounced addresses from your list without any additional investigation might not solve the underlying problem. For information about what you can do to reduce bounces, see Q11. What can I do to minimize bounces? (p. 201).

Q8. If I haven't been monitoring my bounces, can you give me a list of addresses that have bounced?

No. We cannot give you a comprehensive list. You should be regularly monitoring your bounces by email or through Amazon SNS.

Q9. How should I handle bounces?

You need to remove bounced addresses from your mailing list and stop sending mail to them immediately. If you are a small sender, it might be sufficient to simply monitor bounces through email and manually remove bounced addresses from your mailing list. If your volume is higher, you will probably want to set up automation for this process, either by programmatically processing the mailbox where you receive bounces, or by setting up bounce notifications through Amazon SNS. For more information, see Using Notifications with Amazon SES (p. 105).

Q10. Could my emails be bouncing because I've reached my sending limits?

No. Bounces have nothing to do with sending limits. If you try to exceed your sending limits, you will receive an error from the Amazon SES API or SMTP interface when you try to send an email.

Q11. What can I do to minimize bounces?

First, be sure that you are aware of your bounces (see Q7. How can I find out which email addresses bounced? (p. 200)). Then follow these guidelines:

- Do not buy, rent, or share email addresses. Use only addresses that specifically requested your mail.
- Remove bounced email addresses from your list.
- When you ask for email addresses, ask twice and require a match to minimize typos.
- Use double opt-in to sign up new users. That is, when a new users sign up, send them a confirmation email that they need to click before receiving any additional mail. This prevents people from signing up other people as well as accidental signups.
- If you must send to addresses that you haven't mailed lately (and thus you can't be confident that the addresses are still valid), do so only with a small portion of your overall sending. For more information, see our blog post Never send to old addresses, but what if you have to?.
- Ensure that you are not structuring signups to encourage people to use fictional addresses. Either do not provide any value to new users until their email address is verified, or ask for their address only when they specifically sign up to receive mail.
- If you have an "email a friend" feature, be sure you use CAPTCHA or a similar mechanism to discourage automated use of the feature, and do not allow arbitrary content to be inserted by the user. For more information about CAPTCHA, see http://www.captcha.net/.
- If you are using Amazon SES for system notifications, ensure that you are sending the notifications to real addresses that can receive mail. Also consider turning off notifications that you do not need.
- If you are testing a new system, be sure you are either sending to real addresses that can receive email, or you are using the Amazon SES mailbox simulator. For more information, see Testing Amazon SES Email Sending (p. 158).

Amazon SES Complaint FAQ

Q1. What is a complaint?

A complaint occurs when a recipient does not want to receive an email and clicks the "This is spam" button, or equivalent, in his or her email client.

Q2. Why do you care about my complaints?

High complaint rates are often used by entities such as ISPs, mailbox providers, and anti-spam organizations as indicators that a sender is sending to recipients who did not specifically sign up to receive emails, or that the sender is sending content that is different from the type that recipients signed up for.

Q3. What should I do if I receive a probation or suspension notice for my complaint rate?

Review your list acquisition process and the content of your emails to try to understand why your recipients might not appreciate your email. Once you have determined the cause, fix the underlying problem and appeal to get your case reevaluated. For information about the appeal process, see the FAQs on probation and suspension.

Q4. Do all email providers send complaint information to Amazon SES?

No. A number of large email providers return complaint information, but some providers do not. Nevertheless, the complaint rate from domains that provide feedback is likely to be representative of the rest of your sending as well.

Q5. Do you disclose the Amazon SES complaint rate limits that trigger probation and suspension?

No, but you can find general complaint rate guidelines and tips on how to avoid complaints in the Amazon Simple Email Service Email Sending Best Practices white paper.

Q6. Over what period of time is my complaint rate calculated?

We don't calculate your complaint rate based on a fixed period of time, because different senders send at different rates. Instead, we look at what is called a *representative volume*, which represents a reasonable amount of mail with which to evaluate your sending practices. To be fair to both high-volume and small-volume senders, the representative volume is different for each user and changes as the user's sending patterns change. Additionally, the complaint rate isn't calculated based on every email. It is calculated as the percentage of complaints on mail sent to domains that send complaint feedback to Amazon SES.

Q7. Can I calculate my own complaint rate by using metrics from the Amazon SES console or the GetSendStatistics API?

No. There are two primary reasons for this:

- The complaint rate is calculated using representative volume (see Q6. Over what period of time is my complaint rate calculated? (p. 202)). Depending on your sending rate, your complaint rate can stretch farther back in time than the Amazon SES console or GetSendStatistics can retrieve. However, if you regularly monitor your complaint rates using those methods, you should still have a good indicator that you can use to catch problems before they get to levels that trigger a probation or suspension.
- When calculating complaint rate, not every email counts. Complaint rate is calculated as the percentage of complaints on mail sent to domains that send complaint feedback to Amazon SES.

Q8. How can I find out which email addresses complained?

Examine the complaint notifications that Amazon SES sends you. The process for monitoring complaints is essentially the same as for monitoring bounces. The email address to which Amazon SES forwards the notifications depends on how you sent the original messages, as described at Amazon SES Notifications Through Email (p. 106). You can also set up complaint notifications through Amazon Simple Notification Service (SNS), as described at Using Notifications with Amazon SES (p. 105).

Q9. If I haven't been monitoring my complaints, can you give me a list of addresses that have complained?

Unfortunately, we can't give you a comprehensive list. However, you can monitor future complaints by email or through Amazon SNS.

Q10. How should I handle complaints?

You need to do the following:

- Remove addresses that generated complaints from your mailing list and stop sending mail to them immediately. Do not even send an email that says you have received the request to unsubscribe. If you are a small sender, it may be sufficient to simply monitor complaints through email and manually remove complained addresses from your mailing list. If your volume is higher, you will probably want to set up automation for this process, either by programmatically processing the mailbox where you receive complaints, or by setting up complaint notifications through Amazon SNS. For more information, see Using Notifications with Amazon SES (p. 105).
- Take a close look at your sending to determine why your recipients do not appreciate the mail you are sending, and address that underlying problem. For ideas, see Q11. What can I do to minimize complaints? (p. 203). For every person who complains, there are potentially dozens who didn't appreciate your mail who did not (or were not able to) complain. If all you do is remove the recipients who actually complain, you are not addressing the underlying problem with your sending.

Q11. What can I do to minimize complaints?

First, be sure that you are aware of your complaints (see Q8. How can I find out which email addresses complained? (p. 203)). Then follow these guidelines:

- Do not buy, rent, or share email addresses. Use only addresses that specifically requested your mail.
- Use double opt-in to sign up new users. That is, when users sign up, send them a confirmation email that they need to click before receiving any additional mail. This prevents people from signing up other people as well as accidental signups.
- Monitor engagement with the mail you send and stop sending to recipients who do not open or click your messages.
- When new users sign up, be clear about the type of email they will receive from you, and ensure that you send only the type of mail that they signed up for. For example, if users sign up for news updates, do not send them advertisements.
- Ensure that your mail is well-formatted and looks professional.
- Ensure that your mail is clearly from you and cannot be confused for something else.
- Provide users an obvious and easy way to unsubscribe from your mail.

Amazon SES Spamtrap FAQ

Q1. What are spamtraps?

A spamtrap is a special email address maintained by an email provider, ISP, or anti-spam organization that is guaranteed not to have a human being behind it. Because that address will never legitimately be signed up to receive email, the organizations that maintain these spamtraps know that anyone who sends mail to any of these addresses is likely engaging in questionable email practices.

Q2. How are spamtraps set up?

Spamtrap addresses can be set up in multiple ways. They can be converted from addresses that were once valid, but have been unused (and bouncing) for an extended period of time. They can also be addresses that were set up just to be spamtraps. They can be unusual addresses that are hard to guess, and sometimes they are addresses that are close to real addresses (for example, introducing a typo into a common domain name). Often, but not always, spamtraps are "seeded" into the world by putting them on the internet in a variety of ways.

Q3. How does Amazon SES know if I am sending to spamtraps?

Certain organizations that operate spamtraps send Amazon SES notifications when their spamtraps are hit by Amazon SES senders.

Q4. How does Amazon SES use the spamtrap reports?

We review the reports, and if we find enough evidence that you have a problem with sending to spamtraps, we will put you on probation and ask you to fix the underlying problem. If you do not fix the problem in the probation period, your account will be suspended. Also, if your spamtrap problem is very severe, you might be immediately suspended without a probation period. As with any suspension, we will send you a notification at that time.

Q5. What should I do if I receive a probation or suspension notice for sending to spamtraps?

Fix the underlying problem and appeal to get your case reevaluated. For information about the appeal process, see the FAQs on probation and suspension. Due to the way spamtrap sending is reported, it will take a minimum of three weeks before we can confirm that a fix you have put in place has succeeded.

Q6. How many spamtrap hits can I have before I am put on probation or suspended?

Spamtrap hits are a very negative sign, so it takes only a small number of them to indicate that you are engaging in questionable sending practices.

Q7. Do you disclose the spamtrap addresses?

No. Spamtrap organizations disclose only the occurrence of spamtrap hits, not the actual spamtrap addresses. This is one of the measures they take to keep spamtrap addresses confidential and effective.

Q8. What can I do to avoid sending to spamtraps?

To reduce the risk of sending to spamtraps, follow these guidelines:

- Do not buy, rent, or share email addresses. Use only addresses that specifically requested your mail.
- Ensure that you ask for the email address twice to reduce the chance of typos.
- Ensure that you remove addresses that hard bounce from your list, so that they are removed long before they are converted to spamtraps.
- Ensure that you are monitoring engagement by your recipients, and stop sending to recipients who have not engaged with your emails or website recently. Time frames for what an "engaged user" is depend on your use case, but generally speaking if users haven't opened or clicked your emails in several months, you should consider removing them unless you have evidence that they do want your mail.
- Be very careful with reengagement campaigns where you intentionally contact people who have not interacted with you recently. These efforts tend to be highly risky, and can often cause problems not only with spamtrap sending, but also with bounces and complaints.

Amazon SES Manual Investigation FAQ

Q1. I received a probation or shutdown notice for a manual investigation. What does that mean?

An Amazon SES investigator has identified a significant problem with your sending. Typical problems include, but are not limited to, the following:

- Your sending violates the AWS Acceptable Use Policy (AUP).
- Your emails appear to be unsolicited.
- Your content is associated with a use case that Amazon SES does not support.

If the problem is correctable, your account is put on probation and you are given a certain amount of time (rather than a certain volume of mail, as with bounces and complaints) to correct the problem. If the problem is uncorrectable, your account is suspended without a probation period.

Q2. Why would you do a manual investigation?

There are a variety of reasons. These include, but are not limited to, the following:

- Recipients contact Amazon SES to complain about your emails.
- We detect a significant change in your sending patterns.
- The spam filters of Amazon SES flag a significant portion of your emails.

The probation or suspension notification indicates the issue at a high level. For some problems, we are able to provide more specific details.

Q3. What are "unsolicited" emails?

Unsolicited emails are emails that the recipient did not specifically sign up for. This includes cases in which a recipient signs up for a certain type of mail (for example, notifications), and instead is sent a different type of mail (for example, advertisements). If the probation or suspension notice indicates that unsolicited sending is your problem, you should provide the following information in your appeal:

- Are all the messages that you send specifically requested by the recipient, and do they comply with the AUP?
- Have you acquired email addresses in any way other than a customer specifically interacting with you or your website and requesting emails from it? You should explain how you accumulated your mailing list.
- How do your subscribe and unsubscribe processes work? You should include your opt-in and opt-out links.

Also, feel free to provide any other information you might have to assure us that your email list was accumulated and is managed using the practices described in the Amazon Simple Email Service Email Sending Best Practices white paper.

Q4. What should I do if I receive a probation or suspension notice for a manual investigation?

As with any probation or suspension, fix the underlying problem that is causing the issue specified in the probation or suspension notice, and then appeal to get your case reevaluated. For information about the appeal process, see the FAQs on probation and suspension.

Q5. What types of problems do you view as "correctable?"

Generally, we believe the situation is correctable if you have a history of good sending practices, and if there are steps you can take to eliminate the problematic sending while continuing the bulk of your sending. For example, if you are sending three different types of email and only one type is problematic, you might be able to simply stop the problematic sending and continue with the rest of your sending.

Q6. What if I cannot find the source of the problem?

You can respond to the notification (or email *ses-enforcement@amazon.com* from the email address associated with your AWS account) and request a sample of the mail that caused the issue.

Limits in Amazon SES

The following tables list limits within Amazon Simple Email Service (Amazon SES).

Sending Limits

Note
Sending limits are based on recipients rather than on messages.

Limit	Description
Sending limits in the sandbox environment	• Sending quota: 200 emails per 24-hour period. • Maximum send rate: 1 email per second. **Note** The rate at which Amazon SES accepts your messages might be less than the maximum send rate. To increase your sending limits, open an SES Sending Limit case in Support Center. For more information, see Moving Out of the Amazon SES Sandbox (p. 45).

Message Limits

Limit	Description
Maximum message size (including attachments)	10 MB per message (after base64 encoding).
Accepted header fields	Amazon SES accepts any email headers that follow the format described in RFC 822.
Accepted attachment types	Amazon SES accepts all file attachment types *except* for attachments with file extensions listed in Appendix: Unsupported Attachment Types (p. 214).

Sender and Recipient Limits

Limit	Description
Sender address	Both in and out of the sandbox, you are required to verify "From" email addresses and domains (including "Return-Path," although *not* "Reply-To").
Recipient address	• In the sandbox environment, all "To" addresses except for Amazon SES mailbox simulator addresses must be verified. If you don't want to verify your "To" addresses, open an SES Sending Limit case in Support Center. For more information, see Moving Out of the Amazon SES Sandbox (p. 45).
Maximum number of recipients per message	50 recipients per message. A recipient is any "To", "CC", or "BCC" address.
Maximum number of identities you can verify	1000 identities (domains or email addresses in any combination) per AWS account.

Amazon SES API Limits

Limit	Description
Rate at which you can call Amazon SES API actions	All actions (except for `SendEmail` and `SendRawEmail`) are throttled at one request per second. For more information about the Amazon SES API, go to the Amazon Simple Email Service API Reference.

Amazon EC2-Related Limits

Limit	Description
Email sending over port 25	Amazon EC2 throttles email traffic over port 25 by default. To remove this throttle, fill out a Request to Remove Email Sending Limitations.

Free Tier Limits

Limit	Description
Amazon SES free tier eligibility	You are eligible for the free tier with Amazon SES if you use Amazon Elastic Compute Cloud (Amazon EC2). For more information, go to Amazon SES Pricing.

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Amazon SES Resources

The following table lists resources that you may find useful as you work with Amazon Simple Email Service (Amazon SES).

Resource	Description
Amazon Simple Email Service API Reference	The Amazon SES API Reference. Contains complete descriptions of the API actions, parameters, and data types, and a list of errors that the service returns.
Amazon Simple Email Service Email Sending Best Practices	A white paper about Amazon SES best practices.
Amazon SES Pricing	Pricing information for Amazon SES.
SES Sending Limits Increase	The Support Center form to request an increase in your sending limits and move out of the sandbox.
Request to Remove Email Sending Limitations	The form to request to remove the default Amazon EC2 sending limits.
Amazon SES Discussion Forum	The forum in which Amazon SES users can post questions and discuss various Amazon SES topics.
Amazon SES Blog	The blog that contains blog posts and announcements by the Amazon SES team.
AWS Developer Tools	Links to developer tools and resources that provide documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
AWS Support Center	The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
AWS Glossary	The AWS Glossary. Contains definitions of common terms used in Amazon SES and other AWS services.

Resource	Description
Conditions of Use	Amazon Web Services Acceptable Use Policy. Describes email abuse and other prohibited uses of the web services offered by Amazon Web Services, Inc.

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Amazon SES Developer Guide Appendix

This appendix contains supplementary information about sending emails through Amazon Simple Email Service (Amazon SES).

- For the header field requirements for emails that you send through Amazon SES, see Appendix: Header Fields (p. 212).
- For a list of attachment types that Amazon SES does not accept, see Appendix: Unsupported Attachment Types (p. 214).
- For a list of Amazon SES Perl scripts that you can use to perform Amazon SES tasks, see Appendix: Amazon SES Scripts (p. 214).

  Note
  Amazon SES no longer maintains these scripts. For command-line access to Amazon SES, you can use the AWS Command Line Interface or the AWS Tools for Windows PowerShell. To integrate your mail server with Amazon SES, you can use the Amazon SES Simple Mail Transfer Protocol (SMTP) interface. For more information, see Integrating Amazon SES with Your Existing Email Server (p. 58).

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.

Appendix: Header Fields

Amazon SES accepts any email headers that follow the format described in RFC 822.

The following fields cannot appear more than once in a header:

- Accept-Language
- acceptLanguage (**Note:** This field is nonstandard. If possible, use Accept-Language instead.)
- 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 (**Note:** Amazon SES overrides any Date header you provide with the time that Amazon SES accepts the message. The time zone of the Date header is UTC.)
- Delivered-To
- Disposition-Notification-Options
- Disposition-Notification-To
- DKIM-Signature
- DomainKey-Signature
- Errors-To
- From
- Importance
- In-Reply-To
- Keywords
- List-Archive
- List-Help
- List-Id
- List-Owner
- List-Post
- List-Subscribe
- List-Unsubscribe
- Message-Context
- Message-ID (**Note:** Amazon SES overrides any Message-ID header you provide.)
- MIME-Version
- Organization
- Original-From
- Original-Message-ID
- Original-Recipient
- Original-Subject
- Precedence
- Priority
- References
- Reply-To
- Return-Path (**Note:** Amazon SES removes any Return-Path header you provide.)

- Return-Receipt-To
- Sender
- Solicitation
- Sensitivity
- Subject
- Thread-Index
- Thread-Topic
- User-Agent
- VBR-Info

Appendix: Unsupported Attachment Types

You can send messages with attachments through Amazon SES by using the Multipurpose Internet Mail Extensions (MIME) standard. Amazon SES accepts all file attachment types *except* for attachments with the file extensions in the following list.

> **Note**
> Some ISPs have further limitations (e.g., regarding archived attachments), so we recommend testing your email sending through major ISPs before you send your production email.

Unsupported Attachment Types

.ade	.fxp	.mag	.msc	.prg	.url
.adp	.gadget	.mam	.msh	.reg	.vb
.app	.hlp	.maq	.msh1	.scf	.vbe
.asp	.hta	.mar	.msh2	.scr	.vbs
.bas	.inf	.mas	.mshxml	.sct	.vps
.bat	.ins	.mat	.msh1xml	.shb	.vsmacros
.cer	.isp	.mau	.msh2xml	.shs	.vss
.chm	.its	.mav	.msi	.sys	.vst
.cmd	.js	.maw	.msp	.ps1	.vsw
.com	.jse	.mda	.mst	.ps1xml	.vxd
.cpl	.ksh	.mdb	.ops	.ps2	.ws
.crt	.lib	.mde	.pcd	.ps2xml	.wsc
.csh	.lnk	.mdt	.pif	.psc1	.wsf
.der	.mad	.mdw	.plg	.psc2	.wsh
.exe	.maf	.mdz	.prf	.tmp	.xnk

Appendix: Amazon SES Scripts

Amazon SES provides several Perl scripts for performing various tasks. You can download and run the scripts as they are, modify them, or use them as examples for building your own solutions.

> **Note**
> Amazon SES no longer maintains these scripts. For command-line access to Amazon SES, you can use the AWS Command Line Interface or the AWS Tools for Windows PowerShell. To integrate your mail server with Amazon SES, you can use the Amazon SES Simple Mail Transfer Protocol (SMTP) interface. For more information, see Integrating Amazon SES with Your Existing Email Server (p. 58).

Downloading the Scripts

You can download all of the Amazon SES scripts at http://aws.amazon.com/developertools/Amazon-SES. Be sure to consult the README file that comes with these scripts, because this file contains important information about how to prepare the scripts for use.

Using Your AWS Credentials

You must provide your AWS credentials whenever you run any of the Amazon SES scripts. For information about AWS credentials, see Using Credentials With Amazon SES (p. 170). Because it is not secure to provide these credentials on the command line or in a file, these scripts get your AWS access key ID and secret access key from environment variables. The procedure for setting environment variables depends on your operating system.

To specify your AWS credentials

1. Create an environment variable called AWS_ACCESS_KEY_ID and set it to your AWS access key ID. Your AWS access key ID will look something like: `AKIAIOSFODNN7EXAMPLE`.
2. Create an environment variable called AWS_SECRET_ACCESS_KEY and set it to your AWS secret access key. Your AWS secret access key will look something like: `wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY`.

Verifying Email Addresses

> **Important**
> Warning: *ses-verify-email-address.pl* is deprecated. Please use *ses-verify-identity.pl* instead.

You can use the *ses-verify-email-address.pl* script to verify your email address.

Syntax

```
perl ses-verify-email-address.pl [--help] [-e URL] [--verbose] -v EMAIL | -l |
-d EMAIL
```

Description

This script verifies an email address, prints a list of verified email addresses, or deletes an email address from your list of verified addresses.

Options

Name	Description	Required
`--help`	Displays a usage summary for this script.	No
`-e`	Uses the specified Amazon SES endpoint. The default is *email*.	No
`--verbose`	Displays detailed information about the endpoint response.	No
`-v`	Requests verification of an email address. Condition: You must specify one and only one of `-v`, `-l`, or `-d`	Conditional

Name	Description	Required
`-l`	Lists the email addresses that you have already verified. Condition: You must specify `-v`, `-l`, or `-d`	Conditional
`-d`	Deletes a verified email address. Condition: You must specify one and only one of `-v`, `-l`, or `-d`	Conditional

Output

The *ses-verify-email-address.pl* script does not generate any output except for error messages.

Examples

Following are some examples of how to use *ses-verify-email-address.pl*. In each example, the $ character represents the operating system command line prompt.

Example

This example shows how to list your verified email addresses:

```
$ perl ses-verify-email-address.pl -l
```

Example

This example shows how to request verification:

```
$ perl ses-verify-email-address.pl -v user@example.com
```

When you run this script, Amazon SES sends an email to *user@example.com*. When you click the link in this email, your web browser displays a confirmation page to indicate that Amazon SES has verified this email address.

Example

This example shows how to delete an email address from your verified list:

```
$ perl ses-verify-email-address.pl -d user@example.com
```

Managing Identities

You can verify email addresses or domains using *ses-verify-identity.pl*.

Syntax

```
perl ses-verify-identity.pl [--help] [-e URL] [--verbose] -v IDENTITY | -l |
-d IDENTITY | -a IDENTITY
```

Description

The *ses-verify-identity.pl* script verifies an email address or domain, prints a list of email addresses and domains that are verified or pending verification, deletes an email address or domain from your list of verified addresses, or returns the verification status of an email address or domain.

Options

Name	Description	Required
`--help`	Displays a usage summary for this script.	No
`-e`	Uses the specified Amazon SES endpoint. The default is *email*.	No
`--verbose`	Displays detailed information about the endpoint response.	No
`-v`	Requests verification of an email address or domain. Condition: You must specify one and only one of `-v`, `-l`, `-d`, or `-a`	Conditional
`-l`	Lists the email addresses and domains that you have already verified. Condition: You must specify one and only one of `-v`, `-l`, `-d`, or `-a`	Conditional
`-d`	Deletes a verified email address or domain. Condition: You must specify one and only one of `-v`, `-l`, `-d`, or `-a`	Conditional
`-a`	Returns the verification status of an email address or domain. Condition: You must specify one and only one of `-v`, `-l`, `-d`, or `-a`	Conditional

Output

The *ses-verify-identity.pl* script has no output for deleting verified identities or for requesting verification of an email address. Requesting verification of a domain will output the required token, listing identities will output a list of identities, and returning the verification status of an identity will output that status.

Examples

Following are some examples of how to use *ses-verify-identity.pl*. In each example, the $ character represents the operating system command line prompt.

Example

This example shows how to list your verified email addresses and domains:

```
$ perl ses-verify-identity.pl -l
```

The output from the script looks similar to this:

```

username@example.com
example.com
username2@example.com
```

Example

This example shows how to request email address verification:

```
$ perl ses-verify-identity.pl -v user@example.com
```

When you run this script, Amazon SES sends an email to *user@example.com*. When you click the link in this email, your web browser displays a confirmation page to indicate that Amazon SES has verified this email address.

Example

This example shows how to request domain verification:

```
$ perl ses-verify-identity.pl -v example.com
```

The output from the script looks similar to this:

```

EXAMPLETOKENEXAMPLETOKENEXAMPLETOKEN=
```

In order to complete verification of `example.com`, create a TXT record in the DNS settings for the domain with the following values:

- Record Type: TXT
- TXT Name: _amazonses.example.com
- TXT Value: <token in the output from the script>

Example

This example shows how to request the status of a domain:

```
$ perl ses-verify-identity.pl -a example.com
```

The output of this script looks similar to this for a pending domain verification:

```

Identity,Status,VerificationToken
example.com,Pending,EXAMPLETOKENEXAMPLETOKENEXAMPLETOKEN=
```

`Status` has two possible values: Pending or Success.

Example

This example shows how to request the status of an email address:

```
$ perl ses-verify-identity.pl -a user@example.com
```

The output of this script looks similar to this for a pending email address verification:

```


Identity,Status,VerificationToken
username@example.com,Pending,
```

`Status` has two possible values: Pending or Success.

Example

This example shows how to delete an email address from your list:

```
$ perl ses-verify-identity.pl -d user@example.com
```

This example shows how to delete a domain from your list:

```
$ perl ses-verify-identity.pl -d example.com
```

Managing Bounce and Complaint Notifications

You can configure bounce and complaint notifications using *ses-setup-notifications.pl*.

> **Note**
> You can test bounce and complaint notifications by using the Amazon SES mailbox simulator. For more information, see Testing Amazon SES Email Sending (p. 158).

Syntax

```
perl ses-setup-notifications.pl [--help] [-e URL] [--verbose] -a
IDENTITY[,IDENTITY]... | -i IDENTITY -f FORWARD | -i IDENTITY -n NOTIFICATION
-t [TOPIC]
```

Description

The *ses-setup-notifications.pl* script prints the notification settings that apply to a verified email address or domain, modifies whether bounces and complaints should be sent via email for a verified email address or domain, or configures Amazon SNS topics to which bounce and complaint notifications should be sent.

Options

Name	Description	Required
`--help`	Displays a usage summary for this script.	No
`-e`	Uses the specified Amazon SES endpoint. The default is *email*.	No
`--verbose`	Displays detailed information about the endpoint response.	No

Name	Description	Required
`-a`	Retrieves notification attributes for an email address or domain. Condition: You must specify one and only one of `-a` or `-i`	Conditional
`-i`	Modifies notification attributes for an email address or domain. Condition: You must specify one and only one of `-a` or `-i`	Conditional
`-n`	Designates the notification type for which the settings apply. Possible values are `Bounce` and `Complaint`. Condition: You must use this option only with `-i`	Conditional
`-t`	Designates the Amazon Resource Name (ARN) of the SNS topic to use for sending notifications. If the Amazon SNS ARN is not specified with this option, the SNS topic will be disassociated from the identity and Amazon SES will stop publishing notifications to that topic. Condition: You must use this option only with `-n`	Conditional
`-f`	Modifies whether feedback should continue to be forwarded by email. Possible values are `true` and `false`. Condition: You must use this option only with `-i`	Conditional

Output

Returning the notification configuration of a verified identity will output its status. Modifying the email forwarding configuration or Amazon SNS topic for a verified email address or domain has no output.

If you attempt to disable email forwarding and Amazon SNS topics are not configured to receive both bounce notifications and complaint notifications, you will receive the following error message: *"Feedback notification topic not set."*

If you attempt to modify notification attributes for an unverified email address or domain, you will receive the following error message: *"Invalid identity. Must be a verified email address or domain."*

Examples

Following are some examples of how to use *ses-setup-notifications.pl*. In each example, the $ character represents the operating system command line prompt.

Example

This example shows how to get notification settings for your verified email addresses and domains:

```
$ perl ses-setup-notifications.pl -a example.com,username@example.com
```

The output from the script looks similar to this:

```
Identity,ForwardingEnabled,BounceTopic,ComplaintTopic
username@example.com,false,arn:aws:sns:us-east-1:999555999500:username-sns-
bounce,arn:aws:sns:us-east-1:999555999501:username-sns-complaint
example.com,true,arn:aws:sns:us-east-1:999555999502:domain-sns-
bounce,arn:aws:sns:us-east-1:999555999503:domain-sns-complaint
```

Example

This example shows how to disable email forwarding for bounces and complaints:

```
$ perl ses-setup-notifications.pl -i example.com -f false
```

Example

This example shows how to enable email forwarding for bounces and complaints:

```
$ perl ses-setup-notifications.pl -i example.com -f true
```

Example

This example shows how to setup an Amazon SNS topic to receive bounce notifications:

```
$ perl ses-setup-notifications.pl -i example.com -n Bounce -t arn:aws:sns:us-
east-1:999555999500:username-sns-bounce
```

Amazon SES will publish a notification of type `AmazonSnsSubscriptionSucceeded` to the Amazon SNS topic. You can process this notification to confirm the topic is setup successfully.

Example

This example shows how to stop sending bounce notifications to an Amazon SNS topic:

```
$ perl ses-setup-notifications.pl -i user@example.com -n Bounce -t
```

Managing DKIM Settings

You can configure DKIM settings using *ses-setup-dkim.pl*.

Syntax

```
perl ses-setup-dkim.pl [--help] [-e URL] [--verbose] -a IDENTITY[,IDENTITY]...
| -i IDENTITY -s SIGN | -v DOMAIN
```

Description

The *ses-setup-dkim.pl* script prints the DKIM settings that apply to a verified email address or domain, verifies the domain to use for DKIM signed emails, or modifies if emails sent from a verified email address or domain should be DKIM signed.

Options

Name	Description	Required
`--help`	Displays a usage summary for this script.	No
`-e`	Uses the specified Amazon SES endpoint. The default is *email*.	No
`--verbose`	Displays detailed information about the endpoint response.	No
`-a`	Retrieves DKIM signing attributes for an email address or domain. Condition: You must specify one and only one of `-a`, `-v`, or `-i`	Conditional
`-i`	Modifies DKIM signing for an email address or domain. Condition: You must specify one and only one of `-a`, `-v`, or `-i`	Conditional
`-s`	Indicates if email from a DKIM verified email address or domain must be DKIM signed. Possible values are `true` and `false`. Condition: You must use this option only with `-i`	Conditional
`-v`	Verifies the domain to use for DKIM signed emails. Condition: You must specify one and only one of `-a`, `-v`, or `-i`	Conditional

Output

You can return the DKIM configuration of a verified identity to output its status. If you modify the DKIM signing configuration for a verified domain, no output will be returned. If you verify a domain for DKIM signing, DKIM tokens generated by Amazon SES will be returned in the output.

If you attempt to enable DKIM signing for a domain that has not been verified with Amazon SES, you will receive the following error message: *"Domain DOMAINNAME ownership not confirmed."*

If you attempt to enable DKIM signing for a domain that has not been verified for DKIM signing with Amazon SES, you will receive the following error message: *"Domain DOMAINNAME is not verified for DKIM signing."*

If you attempt to verify DKIM settings for an email address, you will receive the following error message: *"Invalid domain name IDENTITYNAME."*

Examples

The following are some examples of how to use *ses-setup-dkim.pl*. In each example, the $ character represents the operating system command-line prompt.

Example

This example shows how to retrieve DKIM settings for your verified email addresses and domains:

```
$ perl ses-setup-dkim.pl -a example.com,username@example.com
```

The output from the script looks similar to this:

```
Identity,DkimVerificationStatus,DkimEnabled,DkimToken1,DkimToken2,DkimToken3
example.com,Pending,false,sesdnsdkimtoken1,sesdnsdkimtoken2,sesdnsdkimtoken3
username@example.com,NotStarted,false
```

Possible values for `DkimVerificationStatus` are: `NotStarted`, `Failed`, `TemporaryFailure`, `Pending`, and `Success`.

Example

This example shows how to enable DKIM signing for email sent through Amazon SES for *example.com*:

```
$ perl ses-setup-dkim.pl -i example.com -s true
```

Email sent via Amazon SES after this command is successful and the setting takes effect will be DKIM signed.

Example

This example shows how to disable DKIM signing for email sent through Amazon SES for *example.com*:

```
$ perl ses-setup-dkim.pl -i example.com -f false
```

Email sent via Amazon SES after this command is successful and the setting takes effect will not be DKIM signed.

Example

This example shows how to verify a domain for DKIM signing through Amazon SES:

```
$ perl ses-setup-dkim.pl -v example.com
```

The output from the script looks similar to this:

```
DkimToken1,DkimToken2,DkimToken3
sesdnsdkimtoken1,sesdnsdkimtoken2,sesdnsdkimtoken3
```

You must create CNAME records with these DNS tokens before Amazon SES can successfully verify the domain.

Sending Formatted Email

You can send a formatted email message using *ses-send-email.pl*.

Syntax

```
perl ses-send-email.pl [--help] [-e URL] [--verbose] -s SUBJECT -f FROM_EMAIL
[-c CC_EMAIL[,CC_EMAIL]...] [-b BCC_EMAIL[,BCC_EMAIL]...] TO_EMAIL[,TO_EMAIL]...
```

Description

The *ses-send-email.pl* script sends a request to Amazon SES to compose and send a multipart MIME email message.

Options

Name	Description	Required
`--help`	Displays a usage summary for this script.	No
`-e`	Uses the specified Amazon SES endpoint. The default is *email*.	No
`--verbose`	Displays detailed information about the endpoint response.	No
`-s`	The subject of the message.	Yes
`-f`	The email address of the sender, followed immediately by the email address of the recipient. You can specify a list of recipients, separated by commas.	Yes
`-c`	The email addresses of copy recipients (Cc). You can specify multiple recipients, separated by commas.	No
`-b`	The email addresses of blind-carbon-copy recipients (Bcc). You can specify multiple recipients, separated by commas.	No
`to-addr`	The email addresses of the primary recipients. You can specify multiple recipients, separated by commas.	Yes
`body`	A file containing the body of the message. Use the left angle bracket (<) character to read from this file, rather than from standard input.	No

Important
The total size of the message cannot exceed 10 MB.

Note
Amazon SES has a limit on the total number of recipients per message: The combined number of To:, CC: and BCC: email addresses cannot exceed 50. If you need to send an email message to a larger audience, you can divide your recipient list into groups of 50 or fewer, and then call Amazon SES repeatedly to send the message to each group.
For every message that you send, the total number of recipients (To:, CC: and BCC:) is counted against your *sending quota* — the maximum number of emails you can send in a 24-hour period. For information about your sending quota, see Managing Your Amazon SES Sending Limits (p. 124).

Output

The *ses-send-email.pl* script does not generate any output, except for error messages.

Examples

Following are some examples of how to use *ses-send-email.pl* to send a formatted email message. In each example, the perl $ character represents the operating system command line prompt.

This example shows how to send an email message to a single recipient.

```
$ perl ses-send-email.pl -s "Greetings" -f user@example.com andrew@example.com
Hello. I hope you are having a good day.
^D
```

The script waits for you to type the body of the message. You can use as many lines as you want, followed by an end-of-file character (EOF). On many systems, you can enter an EOF character by pressing CTRL+D or CTRL+Z. Then press <Enter>.

This example shows how to send to multiple recipients. The input is redirected from a file *body.txt*; the content of this file is used for the message body.

```
$ perl ses-send-email.pl -s "Greetings" -f user@example.com "andrew@ex
ample.com,bob@example.com" < body.txt
```

Sending Raw Email

You can send a raw email message using *ses-send-email.pl*. This is the same script that is described in Sending Formatted Email (p. 223). Use the *-r* parameter to specify raw email.

Syntax

```
perl ses-send-email.pl [--help] [-e URL] [--verbose] [-f FROM_EMAIL] -r
TO_EMAIL[,TO_EMAIL]...
```

Description

When you use the script *ses-send-email.pl* with the `-r` parameter, you must compose the message yourself. The script then sends a request to Amazon SES to send the message.

Options

Name	Description	Required
`--help`	Displays a usage summary for this script.	No
`-e`	Uses the specified Amazon SES endpoint. The default is *email*.	No
`--verbose`	Displays detailed information about the endpoint response.	No
`-f`	The email address of the sender, followed immediately by the email address of the recipient. You can specify multiple recipients, separated by commas.	No
`-r`	Specifies that this is a raw email message. You are responsible for ensuring that the raw message contains the correct email headers, and that it is encoded properly.	No

Name	Description	Required
`-b`	The email addresses of blind copy recipients (Bcc). You can specify multiple recipients, separated by commas.	No
`to-addr`	The email addresses of the primary recipients. You can specify multiple recipients, separated by commas.	Yes
`body`	A file containing the body of the message. Use the left angle bracket (<) character to read from this file, rather than from standard input.	No

You can specify the To: address as a command line option, or include it in the raw email header.

> **Important**
> The total size of the message cannot exceed 10 MB. This includes any attachments that are part of the message.

> **Note**
> Amazon SES has a limit on the total number of recipients per message: The combined number of To:, CC: and BCC: email addresses cannot exceed 50. If you need to send an email message to a larger audience, you can divide your recipient list into groups of 50 or fewer, and then call Amazon SES repeatedly to send the message to each group.
> For every message that you send, the total number of recipients (To:, CC: and BCC:) is counted against your *sending quota* — the maximum number of emails you can send in a 24-hour period. For information about your sending quota, see Managing Your Amazon SES Sending Limits (p. 124).

Output

The *ses-send-email.pl* script does not generate any output, except for error messages.

Example

This example shows how to send a MIME-encoded email, using the -*r* parameter.

```
$ perl ses-send-email.pl -r < mimefile.txt
```

The raw email message is contained in a file named *mimefile.txt*, which might look like the following.

```
From:  user@example.com
To:  andrew@example.com
Subject: MIME email test
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==boundary=="

This is a message with multiple parts in MIME format.
--==boundary==
Content-Type: text/plain

This is part of the message.
--==boundary==
Content-Type: text/html

<h1>This is another part of the message.</h1>
--==boundary==--
```

Important
If you use the `-r` parameter, then the raw message must conform to Internet email standards—Amazon SES cannot send it otherwise.

Monitoring Your Sending Limits

You can run the *ses-get-stats.pl* script with the `-q` switch to check your sending quota and maximum send rate.

Syntax

```
perl ses-get-stats.pl [--help] [-e URL] [--verbose] -s | -q
```

Description

Displays your sending quota and maximum send rate.

Options

Name	Description	Required
`--help`	Displays a usage summary for this script.	No
`-e`	Uses the specified Amazon SES endpoint. The default is *email*.	No
`--verbose`	Displays detailed information about the endpoint response.	No
`-q`	Displays sending quota and maximum send rate.	Yes

Output

The *ses-get-stats.pl* script displays the number of emails you have sent during the past 24 hours, your sending quota for the current 24-hour period, and your maximum send rate.

Example

Here is an example of how to use *ses-get-stats.pl* to display sending activity information. The $ character represents the operating system command line prompt.

```
$ perl ses-get-stats.pl -q
```

The output from the script looks similar to this:

```

                    SentLast24Hours Max24HourSend   MaxSendRate
                    4542            50000           65
```

Monitoring Your Usage Statistics

You can run the *ses-get-stats.pl* script with the `-s` switch to display statistics about email messages that have been sent.

Syntax

```
perl ses-get-stats.pl [--help] [-e URL] [--verbose] -s | -q
```

Description

When used with the `-s` option, the *ses-get-stats.pl* script displays statistics about email messages that have been sent.

Options

Name	Description	Required
`--help`	Displays a usage summary for this script.	No
`-e`	Uses the specified Amazon SES endpoint. The default is *email*.	No
`--verbose`	Displays detailed information about the endpoint response.	No
`-s`	Displays information about prior sending activity.	Yes

Output

The *ses-get-stats.pl* script displays the number of delivery attempts, rejects, bounces, and complaints for the past two weeks. The data is updated every 10 minutes and is displayed for 15-minute intervals.

Example

Here is an example of how to use *ses-get-stats.pl* to display sending statistics. The $ character represents the operating system command line prompt.

```
$ perl ses-get-stats.pl -s
```

The output from the script looks similar to this:

```

                    Timestamp               DeliveryAttempts        Rejects
Bounces  Complaints
                    2010-12-17T22:49:00Z    149                     4       13
      0
                    2010-12-17T23:04:00Z    138                     3       19
      0
                    2010-12-17T23:19:00Z    41                      23      2
      0
                    2010-12-17T23:34:00Z    2                       2       0
      0
                    2010-12-17T23:49:00Z    127                     2       2
      0
                    2010-12-18T00:04:00Z    140                     1       20
      0
                    2010-12-18T00:19:00Z    41                      21      2
      0
                    2010-12-18T00:34:00Z    2                       0       0
      0
```

```
                    2010-12-18T00:49:00Z    76                      0       1
      0
                    2010-12-18T01:04:00Z    157                     0       3
      0
                    2010-12-18T01:19:00Z    19                      1       7
      0
                    2010-12-18T01:34:00Z    34                      6       1
      0
                    ...remaining output deleted...
```

Integrating with Your Existing Email Server

If you currently administer your own email server, you can adapt the *ses-send-email.pl* script to redirect all of your outgoing email to Amazon SES. When you do this, there is no need to modify your existing email clients and applications; the changeover to Amazon SES will be transparent to them.

You can configure the *ses-send-email.pl* script to work with several mail transfer agents (MTAs). For information about downloading and configuring this script, see Appendix: Amazon SES Scripts (p. 214).

The following instructions provide general guidance on how to configure the Postfix and Sendmail MTAs to send email using Amazon SES.

Integrating Amazon SES with Postfix

To integrate *ses-send-email.pl* with Postfix

1. On your mail server, open the *master.cf* file. On many systems, this file resides in the */etc/postfix* directory.
2. Configure a new mail transport: Add the following two lines to the *master.cf* file, modifying them to reflect your particular situation, and then save the file.

   ```
   aws-email  unix  -       n       n       -       -       pipe
   ```

   ```
     flags=R user=mailuser argv=/opt/third-party/amazon/ses-send-email.pl -r
   -e https://email.us-west-2.amazonaws.com -f ${sender} ${recipient}
   ```

   Note the following about these lines:

   - The *flags* line begins with at least one space.
   - The *user* parameter must specify a non-*root* user (i.e., a nonadministrative user). The user "mailuser" in the previous example is for illustrative purposes only.
   - The *ses-send-email.pl* script and the credentials file reside on your mail server in the */opt/third-party/amazon* directory.
   - For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 173).

3. Open the *main.cf* file in the same directory as *master.cf*.
4. Look for the *default_transport* line. If it exists, modify it as follows. If it doesn't exist, add the following line.

   ```
   default_transport = aws-email
   ```

5. When you are done, save the file.
6. Restart Postfix. At the command line, type the following command and press ENTER.

   ```
   sudo /etc/init.d/postfix restart
   ```

   Note
   This command may not be exactly the same on your particular server.

From this point on, your outgoing email is sent via Amazon SES. To verify that this change was successful, send an email message through your Postfix server, and then verify that it arrives at its destination. If the message is not delivered, check your system's mail log for errors. On many systems, the log file is */var/log/mail.log*.

Integrating Amazon SES with Sendmail

These instructions were tested on a 64-bit Amazon EC2 instance using the following Amazon Machine Image (AMI):

- Amazon Linux AMI 2012.09 (ami-1624987f), which runs Linux 3.2

For more information about AMIs, see Amazon Machine Images (AMIs).

Prerequisites

Before you perform the following procedures, verify the following:

- Sendmail is installed on your computer, and you are able to successfully send an email using Sendmail without Amazon SES.

  Note
  If you are sending email from an Amazon EC2 instance with or without Amazon SES, you may need to assign an Elastic IP Address to your Amazon EC2 instance for the receiving ISP to accept your email. For more information, see Amazon EC2 Elastic IP Addresses.
- In addition to the Sendmail package, the following packages are installed on your computer: Sendmail-cf and m4.
- The Amazon SES Perl scripts are installed and set up on your computer, and as a test, you are able to successfully send an email using the *ses-send-email.pl* script. For more information, see Appendix: Amazon SES Scripts (p. 214) and Sending Formatted Email (p. 223).
- You have verified your "From" address and, if your account is still in the sandbox, you have also verified your "To" addresses. For more information, see Verifying Email Addresses in Amazon SES (p. 37).
- (Optional) If you are sending email through Amazon SES from an Amazon EC2 instance, you can fill out a Request to Remove Email Sending Limitations to remove the additional sending limit restrictions that are applied to port 25 by default.

To integrate *ses-send-email.pl* with Sendmail

1. On your mail server, save a back-up copy of */etc/mail/sendmail.mc* and */etc/mail/sendmail.cf*.
2. Open the */etc/mail/sendmail.mc* file.
3. To configure a new mailer, add the following lines to *sendmail.mc* before any MAILER() lines. If you add the lines after a MAILER() line, when you run `m4` in a subsequent step you will get the following error: `"ERROR: FEATURE() should be before MAILER()."`.

   Modify the text to reflect your particular situation.

```
MAILER_DEFINITIONS
Maws-email, P=/opt/third-party/amazon/ses-send-email.pl, F=mDFMuXn,
U=mailuser, S=EnvFromSMTP/HdrFromSMTP, R=EnvToSMTP, A=ses-send-email.pl -r
 -e   https://email.us-west-2.amazonaws.com   -f $f $u
MASQUERADE_AS(YOUR_DOMAIN)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(masquerade_entire_domain)dnl
```

Note the following about this example:

- The *U* parameter must specify a non-*root* user (i.e., a nonadministrative user). The user "mailuser" is for illustrative purposes only.
- The *ses-send-email.pl* script and the credentials file reside in the */opt/third-party/amazon* directory.
- For a list of Amazon SES endpoints, see Regions and Amazon SES (p. 173).
- In the line that starts with `MASQUERADE_AS`, *YOUR_DOMAIN* must be replaced with the domain name from which you are sending your email. By adding this masquerade, you are making email from this host appear to be sent from your domain. Otherwise, the email will appear as if the email is being sent from the host name of the mail server, and you will get an "Email address not verified" error when you try to send an email.

You should modify these parameters as appropriate.

4. Verify that the *mailertable* feature is enabled. Look for the following line:

```
FEATURE(`mailertable')dnl
```

If this line does not exist, you will need to add it.

5. Save *sendmail.mc*.
6. Open or create the */etc/mail/mailertable* file.
7. Add the following line to */etc/mail/mailertable* to send all email using the *aws-email* mailer:

```
.<TAB>aws-email:%0
```

Note
Replace `<TAB>` with an actual TAB character.

8. Build the *mailertable* database.

```
sudo makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
```

9. At a command prompt, type the following command to make *sendmail.cf* writeable:

```
sudo chmod 666 /etc/mail/sendmail.cf
```

10. At a command prompt, type the following command to recreate *sendmail.cf*:

```
sudo m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

11. At a command prompt, type the following command to reset the permissions of *sendmail.cf* to read only:

```
sudo chmod 664 /etc/mail/sendmail.cf
```

12. At a command prompt, type the following command to restart Sendmail:

```
sudo /etc/init.d/sendmail restart
```

Tip
If you encounter errors when restarting Sendmail, you can restart Sendmail in debug mode:

```
sudo /etc/init.d/sendmail restart -bD -O LogLevel=20 -X
/tmp/sendmail.log
```

13. To verify your configuration, generate a deliverability report:

    ```
    sudo sendmail -bv user@example.com
    ```

14. In this report, look for the following output:

    ```
    user@example.com... deliverable: mailer awsemail, host example.com, user
    user@example.com
    ```

15. At a command prompt, type the following to send a test email:

    ```
    sudo /usr/bin/sendmail -f from@example.com to@example.com
    This email is sent through Amazon SES.
    ^D
    ```

    Replace *from@example.com* with your "From" email address, which you must have verified with Amazon SES. Replace *to@example.com* with your "To" address. If your account is still in the sandbox, the "To" address must be verified also. Then, press <Enter>. Type the body of the message, pressing <Enter> after each line. When you are finished typing the email, press CTRL-D to send the email.

16. Check the recipient email's client for the email. If you cannot find the email, check the Junk box in the recipient's email client. If you still cannot find the email, look at the Sendmail log on the mail server. The log is typically in */var/spool/mail/<user>*.

    Tip

    If your mail does not arrive, look in the mail log for the following error:

    `Can't locate SES.pm in @INC`

    This message indicates that the *SES.pm* Perl module does not exist or cannot be found on your system. For more information, refer to the README file that comes with the Amazon SES scripts.

Amazon SES Developer Guide Document History

The following table describes the important changes to the documentation since the last release of Amazon Simple Email Service (Amazon SES).

- **API version:** 2010-12-01
- **Latest documentation update:** August 20, 2015

Change	Description	Date Changed
New Feature	Updated for sending authorization.	July 8, 2015
New Feature	Updated for AWS CloudTrail logging.	May 7, 2015
Service update	Updated to reflect the consolidation of the Amazon SES limit increase forms and removed "production access" terminology.	April 8, 2015
Service update	Updated with new requirements for domain verification TXT records.	February 25, 2015
Documentation update	Added Enforcement FAQ.	December 15, 2014
New Feature	Updated for delivery notifications.	June 23, 2014
New Feature	Updated for subdomain support.	March 19, 2014
New Feature	Updated for Amazon SES expansion to the US West (Oregon) region.	January 29, 2014
New Feature	Updated for Amazon SES expansion to the EU (Ireland) region.	January 15, 2014
New Feature	Updated to reflect the changes in validation of Header Fields and MIME Types.	November 6, 2013
Documentation update	Removed content on Sender ID.	August 22, 2013

Change	Description	Date Changed
New feature	Updated to reflect the Amazon SES console redesign.	June 19, 2013
New feature	Replaced the blacklist with the suppression list.	May 8, 2013
New feature	Updated for the blacklist removal feature.	March 4, 2013
Documentation update	Added MIME types.	February 4, 2013
Documentation update	Included a Getting Started section to replace the stand-alone Getting Started guide, restructured the Table of Contents, and updated the Sendmail integration instructions.	January 21, 2013
Documentation update	Added troubleshooting sections on increasing throughput and SMTP issues.	December 12, 2012
Documentation update	Restructured the information on sending limits.	November 9, 2012
New Feature	Updated for the Amazon SES mailbox simulator.	October 3, 2012
New Feature	Updated for using a DKIM signature to sign email from a verified identity.	July 17, 2012
New Feature	Updated for receiving bounce and complaint feedback notifications through Amazon Simple Notification Service (Amazon SNS).	June 26, 2012
New Feature	Updated for domain verification.	May 15, 2012
New Feature	Updated to reflect additional header and attachment types.	April 25, 2012
New Feature	Updated for the STARTTLS extension to SMTP.	March 7, 2012
New Feature	Updated for Variable Envelope Return Path (VERP).	February 22, 2012
New Feature	Updated for SMTP support.	December 13, 2011
New Feature	Updated for AWS Management Console support.	November 17, 2011
New Feature	Updated for attachment support.	July 18, 2011
Initial Release	This is the first release of the *Amazon Simple Email Service Developer Guide*.	January 25, 2011

For technical discussions about various Amazon SES topics, visit the Amazon SES blog. To browse and post questions, see the Amazon SES forum.