Amazon Simple Email Service

Developer Guide

API Version 2010-12-01

Amazon Web Services

Amazon Simple Email Service: Developer Guide

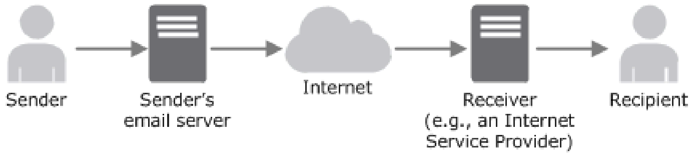
Amazon Web Services

What Is Amazon SES?

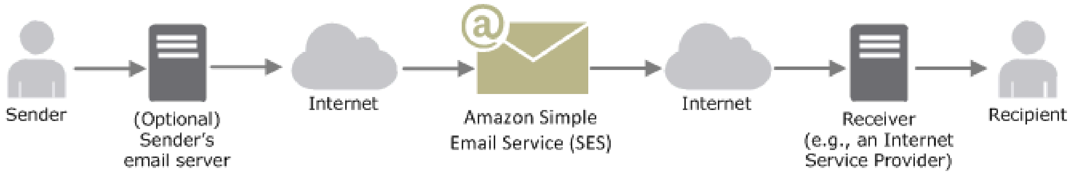
Welcome to the Amazon Simple Email Service (Amazon SES) Developer Guide. Amazon SES is an outbound-only email-sending service that provides an easy, cost-effective way for you to send email. You can use Amazon SES to send marketing emails such as special offers, transactional emails such as order confirmations, 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 via 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, you can use the AWS Command Line Interface, or you can use an AWS SDK, which wraps 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 Java, PHP, .NET, Android, iOS, 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 Elastic Compute Cloud (EC2) instance by using the AWS Software Development Kits (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 AWS 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 have bounced or produced a complaint.
- 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:

- Getting Started with Amazon SES (p. 15)—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.
- Email Deliverability and Amazon SES (p. 4)—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. 33)	Shows you how to sign up for AWS, get your AWS access keys, download an AWS SDK, and verify email addresses or domains so that you can start sending emails with Amazon SES.
Using the Amazon SES SMTP Interface to Send Email (p. 44)	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. 62)	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.

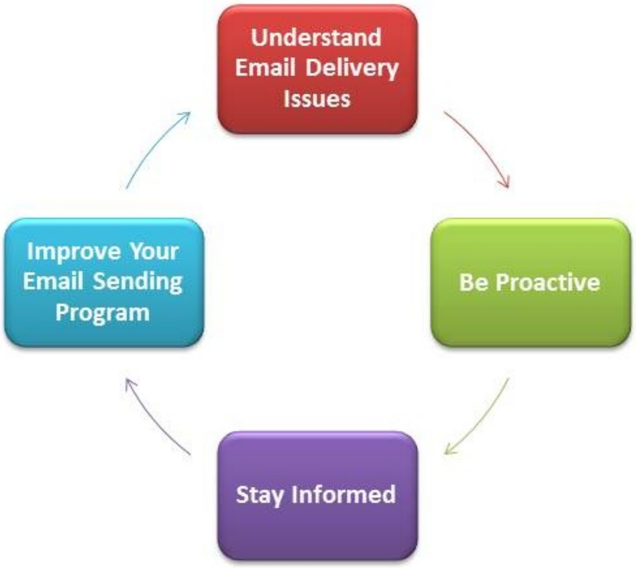
Section	Description
Authenticating Email in Amazon SES (p. 81)	Shows you how to use Sender Policy Framework (SPF), Sender ID, and DKIM with Amazon SES to show ISPs that you own the account you are sending from and your emails have not been modified in transit.
Monitoring Your Amazon SES Sending Activity (p. 92)	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 notifications of your bounces and complaints by email or by setting up Amazon SNS notifications.
Managing Your Amazon SES Sending Limits (p. 108)	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. 112)	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. 113)	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.
Testing Amazon SES Email Sending (p. 115)	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 blacklisted.
Limits in Amazon SES (p. 127)	Lists limits within Amazon SES.
Troubleshooting Amazon SES (p. 118)	Explains common causes of delivery problems and provides descriptions of common Amazon SES exceptions and SMTP response codes.

For more information about Amazon SES terms, see the AWS glossary.

Email Deliverability and Amazon SES

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 blacklisting 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 forwards the bounced email to you or sends you a notification via Amazon Simple Notification Service (Amazon SNS), depending on how you have your system set up. For more information, see Bounce and Complaint Notifications in Amazon SES (p. 93).

There are *hard bounces* and *soft bounces*. A hard bounce occurs when there is a persistent email delivery failure such as when a mailbox does not exist. A soft bounce is a temporary email delivery failure such as when mailbox is full or the ISP is too busy to handle the request. If a soft bounce is retried multiple times and the email still cannot be delivered, then it becomes a hard bounce. We recommend that you do not make repeated delivery attempts to email addresses that hard bounce.

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 Bounce and Complaint Notifications in Amazon SES (p. 93). We recommend that you do not make repeated delivery attempts to email addresses that generate complaints.

Blacklist

The term *blacklist* can refer to different, but related, things. If an ISP suspects that an IP address or email address is the source of spam, it might add the address to a blacklist and block all incoming emails from it.

In the context of Amazon SES, a blacklist can also refer to a list of recipient email addresses that Amazon SES considers to be invalid because the addresses have caused a hard bounce within the past 14 days. Amazon SES rejects requests to send emails to addresses on the blacklist. If you are sure that the email address that you're trying to send to is valid, you can submit a blacklist removal request. For more information, see Removing an Email Address from the Amazon SES Blacklist (p. 121).

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. 34) and Verifying Domains in Amazon SES (p. 37).

If you have not yet requested production access, you also need to verify all recipient addresses except for addresses provided by the Amazon SES mailbox simulator. For information about getting production access, see Requesting Production Access to Amazon SES (p. 42). For more information about the mailbox simulator, see Testing Amazon SES Email Sending (p. 115).

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 provides three methods of authentication that you can use: Sender Policy Framework (SPF), Sender ID, and DomainKeys Identified Mail (DKIM). For more information, see Authenticating Email in Amazon SES (p. 81).

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 you can send per second). 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 Extended Access Request 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. 108).

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. 115).

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. 112) and the Amazon SES Email Sending Best Practices white paper.

Stay Informed

When deliveries fail, or when recipients complain about your emails, Amazon SES helps you to track down the issue by providing feedback notifications and by enabling you to easily monitor your usage statistics.

Feedback Notifications for Bounces and Complaints

When an email hard bounces, the ISP notifies Amazon SES, and Amazon SES forwards the notification to you. 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 forwards the bounce and complaint messages to you in one of two ways: it can send you notifications by email, or you can set your account up to receive feedback notifications via Amazon Simple Notification Service (Amazon SNS). For more information, see Bounce and Complaint Notifications in Amazon SES (p. 93).

Usage Statistics

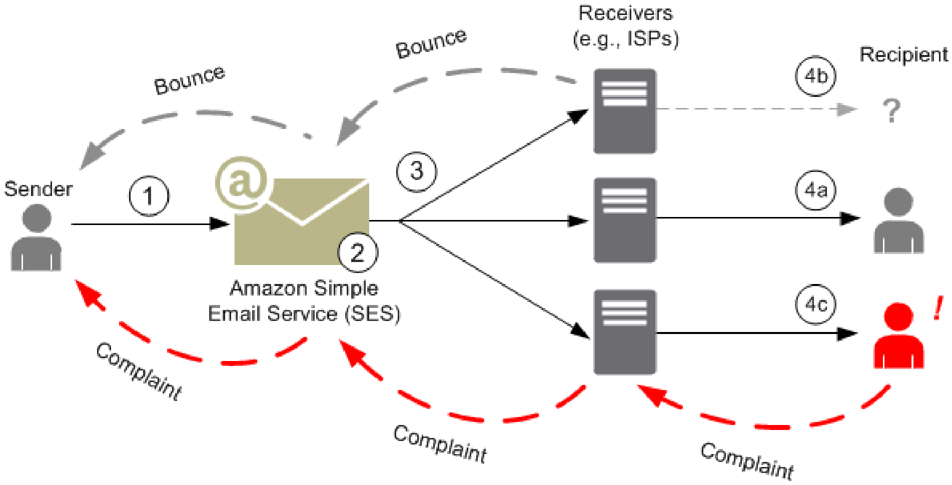
Amazon SES provides usage statistics so that you can view your failed deliveries to determine and resolve the root causes. You can view your usage statistics by using the Amazon SES console or by calling the Amazon SES API. You can view how many deliveries, bounces, complaints, and virus-infected rejected 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. 112) 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 assembles an email based on the request parameters and then queues it for delivery.
3. The message is routed over the Internet to the recipient's receiver. In this example, the receiver is an Internet service provider (ISP).

4. 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, queues it for delivery, and sends it out over the Internet using Simple Mail Transfer Protocol (SMTP). Your message is usually sent immediately with the first delivery attempt normally occurring 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 via email or via 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:

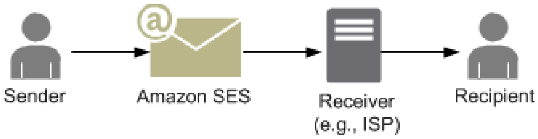
- If you are calling Amazon SES through the Query (HTTPS) API (`SendEmail` or `SendRawEmail`), the actions will return an error. For more information, see the Amazon Simple Email Service API Reference.
- If you are using an AWS SDK for a programming language that uses exceptions, the call to Amazon SES 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. 119).

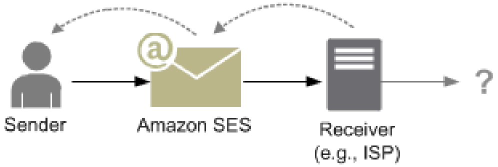
After Amazon SES Sends an Email

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.

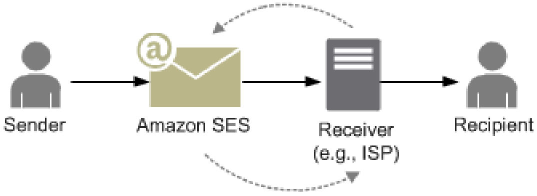


- **Hard bounce—**The email is rejected by the ISP because of a persistent condition. For example, a hard bounce can occur because the recipient's address or domain name is invalid. A hard bounce notification is sent from the ISP back to Amazon SES, which forwards it to the sender via email or through Amazon Simple Notification Service (Amazon SNS), depending on the sender's setup. The path of a hard bounce is shown in the following figure.

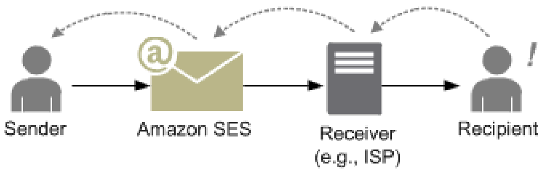


- **Soft bounce—**The ISP cannot deliver the email to the recipient because of a temporary condition. For example, the ISP is too busy to handle the request or the recipient's mailbox is full. The ISP sends a soft bounce notification back to Amazon SES. Amazon SES retries sending the email for a length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

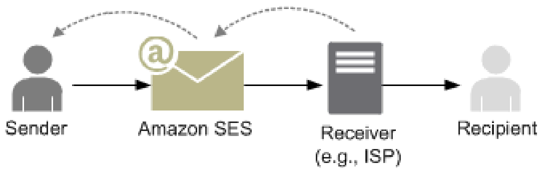
  If the ISP can deliver the email to the recipient during a retry, the delivery is successful. If the ISP cannot deliver the email to the recipient by the time Amazon SES stops retrying, then the soft bounce turns into a hard bounce and Amazon SES forwards the hard bounce notification to the sender via email or by an Amazon SNS notification. 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 address blacklisted response. For more information, see Testing Amazon SES Email Sending (p. 115).

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. 134). 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. 64).
- 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. 66).

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. 62).

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.

For more information about sending email using the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 44).

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).

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. 16). Then, send an email in one of the following ways. You can also watch our Getting Started with Amazon SES video.

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. 16).

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. 18).
- 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. 24).

For introductory information on both SMTP sending methods, see Sending an Email Through Amazon SES Using SMTP (p. 17).

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. 25).

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. 34).
- **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. 34).
- **(Optional) Authorize Amazon SES to send from your domain—**If you manage own DNS server, and are already publishing Sender Policy Framework (SPF) or Sender ID records to your Domain Name Service (DNS) you need to authenticate the Amazon AWS domain or recipients might not receive your email. For information, see Authorizing Amazon SES Sending From Your Domain (p. 41).

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. 33).

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.

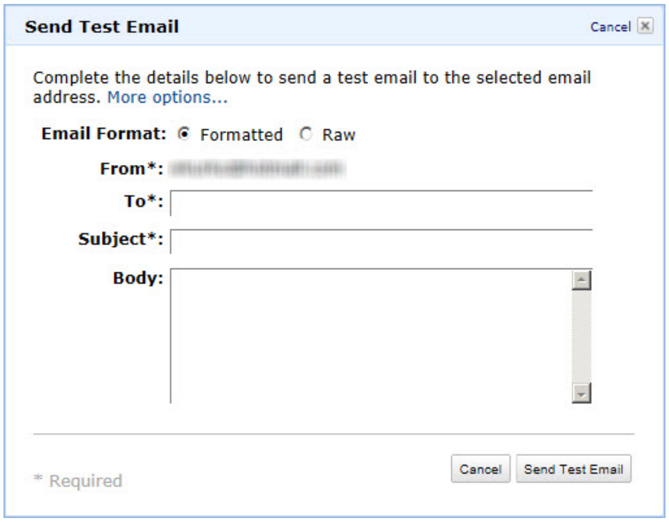
> **Important**
> In this getting started tutorial, you send an email to yourself so that you can check to see if you received it. For further experimentation or load testing, use the Amazon SES mailbox simulator whenever possible. Emails that you send to the mailbox simulator do not count toward your sending quota or your bounce and complaint rates. For more information, see Testing Amazon SES Email Sending (p. 115).

Before you follow these steps, make sure you review the setup instructions in Before You Begin with Amazon SES (p. 16).

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, click **Verified Senders** to view the email address that you verified in Verifying Email Addresses in Amazon SES (p. 34).
3. In the **Verified Senders** pane, select the checkbox of the email address that you have verified.
4. Click **Send a Test Email**.
5. In the **Send Test Email** dialog box, fill out the **To**, **Subject**, and **Body** fields. Make sure that you have verified the address in the **To** field. For more information, see Verifying Email Addresses in Amazon SES (p. 34).



Send Test Email

Complete the details below to send a test email to the selected email address. More options...

Email Format: ● Formatted ○ Raw

From*: [redacted]

To*:

Subject*:

Body:

* Required

6. Click **Send Test Email**.
7. Sign in to the email client of the address you sent the email to. You should find the email message that you sent.

Please Provide Feedback

Your input is important to help make our documentation helpful and easy to use. Please tell us about your experience getting started with Amazon SES by completing our Getting Started Survey.

Thank you.

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. 16). 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. 45).

  Important
  Your SMTP credentials are different from your AWS credentials.

- The Amazon SES SMTP hostname, which is *email-smtp.us-east-1.amazonaws.com*.

- 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. 46).

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. 18)
- Configuring Your Existing Email Server or SMTP-Enabled Application to Send Email Through Amazon SES (p. 24)

For more information about the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email (p. 44).

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. 17) and then select one of the following tutorials:

- Sending an Email Through the Amazon SES SMTP Interface with C# (p. 18)
- Sending an Email Through the Amazon SES SMTP Interface with Java (p. 21)

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. 16) and Sending an Email Through Amazon SES Using SMTP (p. 17).

> **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. 115).

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

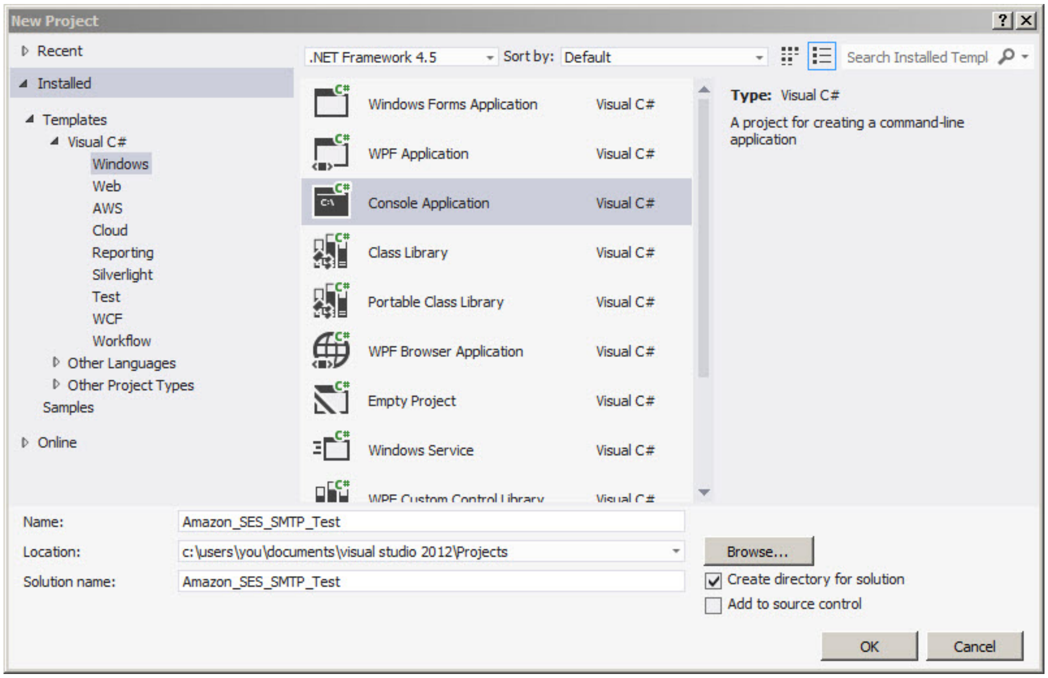
1. Create a console project in Visual Studio by performing the following steps:

   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 `Amazon_SES_SMTP_Test`. The dialog box should look similar to the following figure.



Type: Visual C#

A project for creating a command-line application

Name: Amazon_SES_SMTP_Test

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

Solution name: Amazon_SES_SMTP_Test

☑ 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 Amazon_SES_SMTP_Test
{
    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 you have not yet requested
                                                        // production access,
 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 credential.
            const String SMTP_PASSWORD = "YOUR_SMTP_PASSWORD";  // Replace
 with your SMTP password.

            // Amazon SES SMTP host name.
            const String HOST = "email-smtp.us-east-1.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. 34).

- RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If you have not yet requested production access, you must verify this address before you use it. For more information, see Requesting Production Access to Amazon SES (p. 42).

4. In Program.cs, replace the following SMTP credentials with the values that you obtained in Obtaining Your Amazon SES SMTP Credentials (p. 45):

   Important
   Your SMTP credentials are different from your AWS Access Key ID and Secret Key.

   - 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.

5. Save Program.cs.
6. To build the project, click **Build** and then click **Build Solution**.
7. To run the program, click **Debug** and then click **Start Debugging**.
8. Review the program's console output to verify that the sending was successful. (You should see "Email sent!")
9. 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.

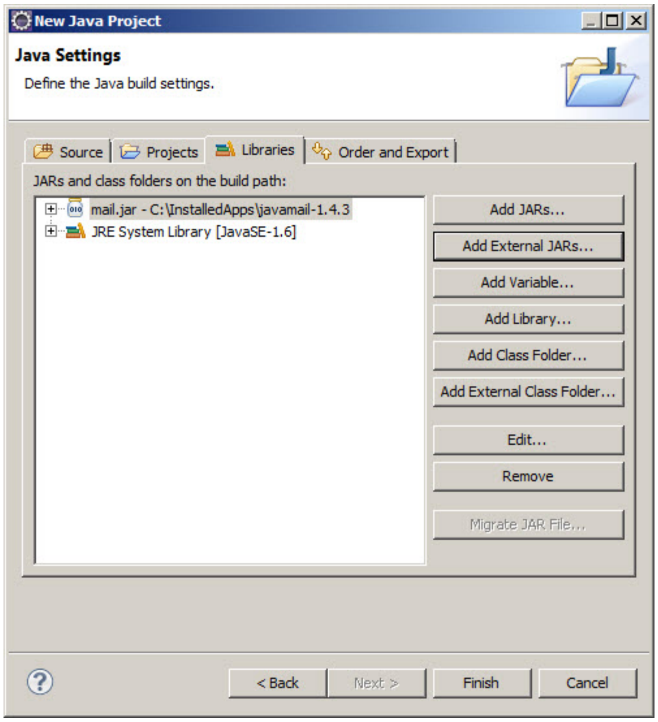
Before you perform the following procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 16) and Sending an Email Through Amazon SES Using SMTP (p. 17).

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. 115).

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:



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 `amazon_ses_smtp_test` and then click **Finish**.
5. Replace the entire contents of amazon_ses_smtp_test.java with the following code:

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

public class amazon_ses_smtp_test {

    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 you have not yet requested
                                                      // production access,
 this address 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 credential.
```

```
    static final String SMTP_PASSWORD = "YOUR_SMTP_PASSWORD";  // Replace
with your SMTP password.

    // Amazon SES SMTP host name.
    static final String HOST = "email-smtp.us-east-1.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 amazon_ses_smtp_test.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. 34).
   - RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If you have not yet requested production access, you must verify this address before you use it. For more information, see Requesting Production Access to Amazon SES (p. 42).

7. In amazon_ses_smtp_test.java, replace the following SMTP credentials with the values that you obtained in Obtaining Your Amazon SES SMTP Credentials (p. 45):

   Important
   Your SMTP credentials are different from your AWS Access Key ID and Secret Key.

   - 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. Save amazon_ses_smtp_test.java.
9. To build the project, click **Project** and then click **Build Project**. (If this option is disabled, then you may have automatic building enabled.)
10. To start the program and send the email, click **Run** and then click **Run** again.
11. Review the program's console output to verify that the sending was successful. (You should see "Email sent!")
12. 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. 17). 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. 53)
- Integrating Amazon SES with Sendmail (p. 55)
- Integrating Amazon SES with Exim (p. 58)

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. 47).

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. 49).

Please Provide Feedback

Your input is important to help make our documentation helpful and easy to use. Please tell us about your experience getting started with Amazon SES by completing our Getting Started Survey.

Thank you.

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. 16). 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. 41).
- **Get your AWS credentials—**To access Amazon SES programmatically, you need your AWS Access Key ID and Secret Access Key. For more information, see Getting Your AWS Access Keys (p. 40).

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. 25)
- Sending an Email Through Amazon SES Using AWS SDK for Java (p. 29)

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

The following procedure shows you how to use 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.

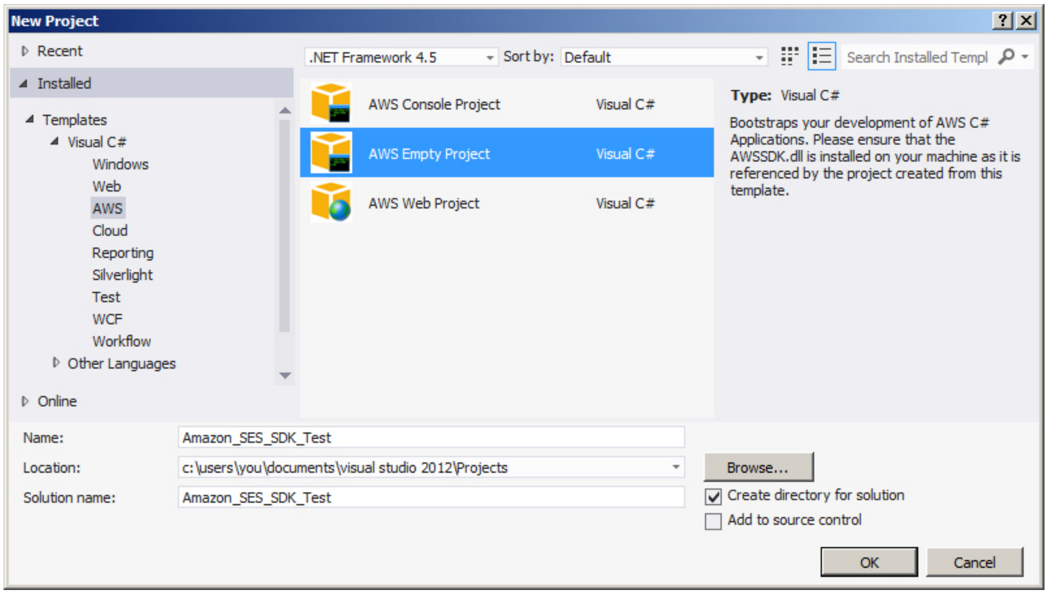
Before you begin this procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 16) and Sending an Email Through Amazon SES Using an AWS SDK (p. 25).

> **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. 115).

To send an email using the AWS SDK for .NET

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 `Amazon_SES_SDK_Test`. The dialog box should look similar to the following figure.



.NET Framework 4.5 Sort by: Default

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: Amazon_SES_SDK_Test

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

Solution name: Amazon_SES_SDK_Test

☑ Create directory for solution

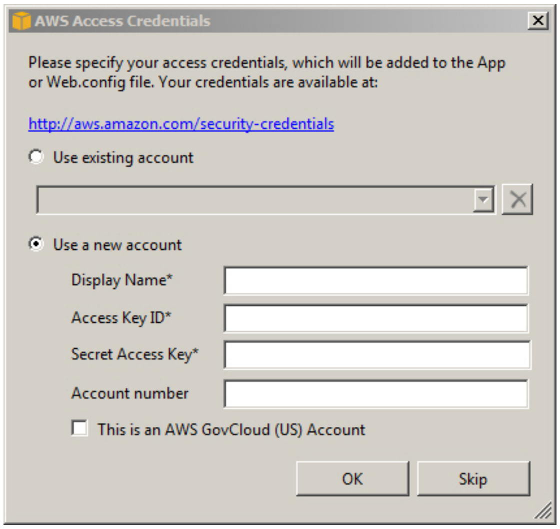
☐ Add to source control

   g. Click **OK**.

2. In the **AWS Access Credentials** dialog box, 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. 40).
   - **Secret Access Key**—Enter the AWS Secret Access Key that you obtained in Getting Your AWS Access Keys (p. 40).

- **Account Number—**(Optional) Enter your AWS account number. To find your AWS account number, go to https://portal.aws.amazon.com/gp/aws/securityCredentials#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.



AWS Access Credentials

Please specify your access credentials, which 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

3. Click **OK**.

   Note
   Your Access Key ID and Secret Access Key are added to the App.config file within your project. When you instantiate an `AmazonSimpleEmailServiceClient` object with no parameters, the constructor looks in App.config for your security credentials by default.

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 Amazon_SES_SDK_Test
{
    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 you have not yet requested
                                                      // production access,
 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().WithToAddresses(new
 List<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).With
Body(body);

            // Assemble the email.
            SendEmailRequest request = new SendEmailRequest().With
Source(FROM).WithDestination(destination).WithMessage(message);

           // Instantiate an Amazon SES client, which will make the service
 call. Since we are instantiating an
            // AmazonSimpleEmailServiceClient object with no parameters,
the constructor looks in App.config for
            // your AWS credentials by default. When you created your new
AWS project in Visual Studio, the AWS
            // credentials you entered were added to App.config.
          AmazonSimpleEmailServiceClient client = new AmazonSimpleEmailSer
viceClient();

            // 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 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. 34).

- RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If you have not yet requested production access, you must verify this address before you use it. For more information, see Requesting Production Access to Amazon SES (p. 42).

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 to create an AWS SDK project and modify the Java code to send an email through Amazon SES.

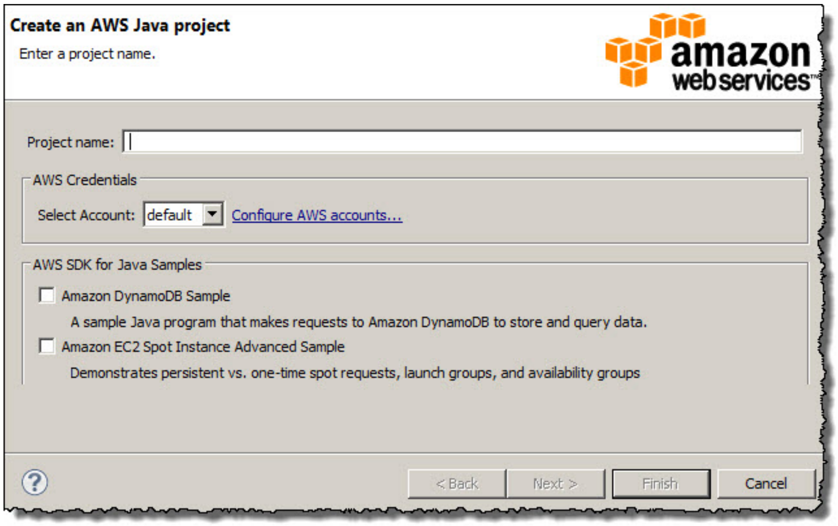
Before you begin this procedure, complete the setup tasks described in Before You Begin with Amazon SES (p. 16) and Sending an Email Through Amazon SES Using an AWS SDK (p. 25).

> **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. 115).

To send an email using the AWS SDK for Java

1. 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**.
   c. In the **Create an AWS Java Project** dialog box, type a project name.



Create an AWS Java project

Enter a project name.

Project name:

AWS Credentials

Select Account: default

AWS SDK for Java Samples

□ Amazon DynamoDB Sample

A sample Java program that makes requests to Amazon DynamoDB to store and query data.

□ Amazon EC2 Spot Instance Advanced Sample

Demonstrates persistent vs. one-time spot requests, launch groups, and availability groups

d. Under **AWS Credentials**, click **Configure AWS accounts**.

e. In the **AWS Toolkit** dialog box, under **Account Details**, enter the following information:

- **Account Name—**Type a name that identifies your account, or leave the value at the default. Next time you create an AWS project in Eclipse, you will be able to select this account so you will 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. 40).
- **Secret Access Key—**Enter the AWS Secret Access Key that you obtained in Getting Your AWS Access Keys (p. 40).

> **Note**
>
> Your Access Key ID and Secret Access Key are added to the AwsCredentials.properties file within your project.

f. Click **OK**.

g. Click **Next**.

h. 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 **Java Class** dialog box, in the **Name** field, type `amazon_ses_sdk_test` and then click **Finish**.
5. Replace the entire contents of amazon_ses_sdk_test.java with the following code:

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

public class amazon_ses_sdk_test {

    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 you have not yet requested
                                                     // production access, 
 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 {

        // Your AWS credentials are stored in the AwsCredentials.properties 
 file within the project.
       // You entered these AWS credentials when you created a new AWS Java 
 project in Eclipse.
        PropertiesCredentials credentials = new PropertiesCredentials(
          amazon_ses_sdk_test.class
                       .getResourceAsStream("AwsCredentials.properties"));
        // Retrieve the AWS Access Key ID and Secret Key from AwsCreden
tials.properties.
        credentials.getAWSAccessKeyId();
        credentials.getAWSSecretKey();

        // 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 with the supplied AWS credentials.
          AmazonSimpleEmailServiceClient client = new AmazonSimpleEmailSer
viceClient(credentials);

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

6. In amazon_ses_sdk_test.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. 34).
   - RECIPIENT@EXAMPLE.COM—Replace with your "To" email address. If you have not yet requested production access, you must verify this address before you use it. For more information, see Requesting Production Access to Amazon SES (p. 42).

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

Please Provide Feedback

Your input is important to help make our documentation helpful and easy to use. Please tell us about your experience getting started with Amazon SES by completing our Getting Started Survey.

Thank you.

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. 34).
- Before you send email through Amazon SES, you need to verify that you own the "From" address. If you do not have production access yet, 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. 34).
- If you manage your own DNS server and are already publishing Sender Policy Framework (SPF) or Sender ID records to your Domain Name Service (DNS), you need to authenticate the domain that Amazon SES uses as its envelope-FROM address or recipients might not receive your email. For more information, see Authorizing Amazon SES Sending From Your Domain (p. 41).

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, you need to obtain your AWS Access Key ID and Secret Access Key. For more information, see Getting Your AWS Access Keys (p. 40).
- 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. 41).
- 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. 42).
- When you first sign up for Amazon SES, you are in the sandbox. In the sandbox, you can 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 Requesting Production Access to Amazon SES (p. 42).

For definitions of Amazon SES terms, see the AWS Glossary.

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. 34).
- 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 getting production access, see Requesting Production Access to Amazon SES (p. 42).

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**
> Regardless of whether you are in the sandbox or have production access, you always 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 you do not have production access yet, 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.

- For information about verifying email addresses, see Verifying Email Addresses in Amazon SES (p. 34).
- For information about verifying entire domains, see Verifying Domains in Amazon SES (p. 37).

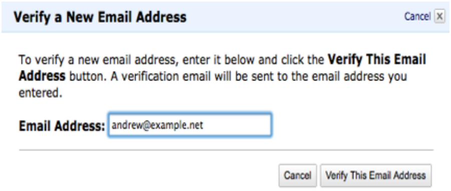
Verifying Email Addresses in Amazon SES

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. 37).

> **Important**
>
> With the exception of addresses containing labels (see below), you must verify each email address that will be used as a "From" or "Return-Path" address for your messages.
>
> 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. 37).)
>
> Until you are granted production access to Amazon SES, 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. 115). For more information about production access, see Requesting Production Access to Amazon SES (p. 42).

To verify an email address

1. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. In the navigation pane, click **Verified Senders**.
3. On the **Email Addresses** tab, click **Verify a New Email Address**.
4. 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: andrew@example.net

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

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

7. The status of the email address on the **Email Addresses** tab will change from "pending verification" to "verified".
8. 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, then click **Send a Test Email**.

You will need to verify each email address or domain from which you want to send messages. You can verify as many as 1000 identities (domains and email addresses, in any combination) per AWS account.

If you include a "Return-Path" header, you must verify the email address or domain contained in that header.

Verifying an email address also allows you to set the "From" or "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.net*, with user-specified text between the plus sign (+) and the at sign (@).

For example, if you verify *andrew@example.net*, you can also send email from *andrew+recipient1@example.net*, *andrew+recipient2@example.net*, 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.net* does not allow you to send from *andrew@example.net*, *andrew+recipient2@example.net*, or *andrew+recipient1+recipient2@example.net*.

You can send an email message to yourself or to the mailbox simulator immediately after verification; however, to send email to other "To" addresses, you must verify each of them. This restriction affects only Amazon SES users who have not requested production access.

To send email to other addresses, without needing to verify them, you will need to request production access to Amazon SES. For more information, see Requesting Production Access to Amazon SES (p. 42).

> **Note**
> 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. 66).

To view your verified email addresses

1. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. In the navigation pane, click **Verified Senders** and choose the **Email Addresses** tab.

To remove verified email addresses

1. Go to the AWS Management Console, and then log in with the email address and password you used when you signed up for Amazon SES.
2. In the navigation pane, click **Verified Senders** and choose the **Email Addresses** tab.
3. Check the box beside each email address that you want to remove, 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.

Using an Amazon SES Script

At the command line, you can use the *ses-verify-identity.pl* script to verify your email address. For information about downloading and configuring this script, see Appendix: Amazon SES Scripts (p. 144).

Verifying Domains in Amazon SES

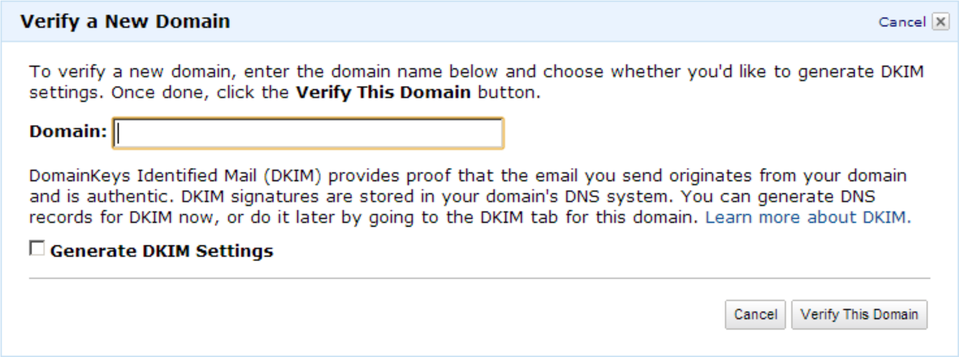
Amazon SES requires that you verify your email address or domain, to confirm that you own it and to prevent others from using it. When you verify an entire domain, you are verifying all email addresses from that domain, so you don't need to verify email addresses from that domain individually. This section discusses verifying entire domains. For example, if you verify the domain *example.com*, you can send email from *sender1@example.com*, *sender2@example.com*, *sender3@example.com*, or any other user at *example.com*. Domain names are case-insensitive. If you verify *example.com*, you can send from *EXAMPLE.com* also.

For individual email address verification, see Verifying Email Addresses in Amazon SES (p. 34).

> **Important**
> Amazon SES only verifies fully qualified domain names (FQDNs). Even if you verify a domain, you have to verify subdomains of that domain. For example, if you want to send email from both *example.com* and *newyork.example.com*, you need to verify each of these FQDNs separately.

To verify a domain

1. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. In the navigation pane, click **Verified Senders**.
3. In the content pane, choose the **Domains** tab.
4. Click the **Verify a New Domain** button.
5. In the **Verify a New Domain** dialog box, enter the domain name, 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.

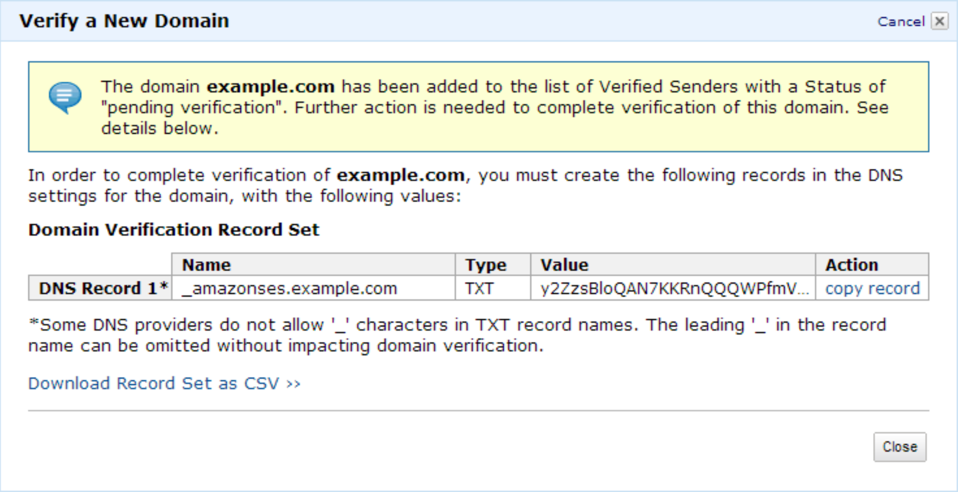
Domain:

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**

> **Note**
> If you want to set up DKIM signing for this domain, select the **Generate DKIM Settings** option. For more information about DKIM signing, see Authenticating Email with DKIM in Amazon SES (p. 83).

6. 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 from the **Domains** tab 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	Action
DNS Record 1*	_amazonses.example.com	TXT	y2ZzsBloQAN7KKRnQQQWPfmV...	copy record

*Some DNS providers do not allow '_' characters in TXT record names. The leading '_' in the record name can be omitted without impacting domain verification.

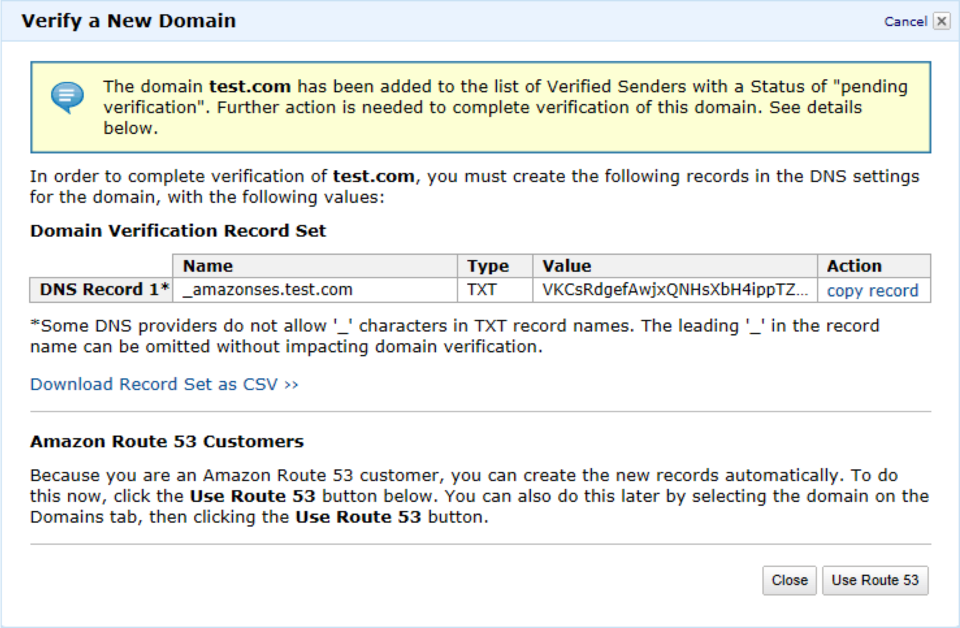
Download Record Set as CSV »

7. To complete domain verification, you must add a TXT record with the displayed **Name** and **Value** to your domain's DNS settings. 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.

   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.

   > **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.

   If Amazon Route 53 provides the DNS service for the domain you are verifying, and you are logged in to Amazon SES with the same email address and password 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.



The domain **test.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 **test.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	Action
DNS Record 1*	_amazonses.test.com	TXT	VKCsRdgefAwjxQNHsXbH4ippTZ...	copy record

*Some DNS providers do not allow '_' characters in TXT record names. The leading '_' in the record name can be omitted without impacting domain verification.

Amazon Route 53 Customers

Because you are an Amazon Route 53 customer, you can create the new records automatically. To do this now, click the **Use Route 53** button below. You can also do this later by selecting the domain on the Domains tab, then clicking the **Use Route 53** button.

Otherwise, 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.

8. If you are not using Route 53, Amazon Web Services needs to verify that a TXT record with the specified **Name** and **Value** have been added to your DNS settings. This may take up to 72 hours.

   When verification is complete, the domain's Status on the **Domains** tab will change from "pending verification" to "verified", and you will receive an **Amazon SES Domain Verification SUCCESS** 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.)
9. 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, then click **Send a Test Email**.

What if domain verification fails?

If the DNS settings are not correctly updated, you will receive an **Amazon SES Domain Verification FAILURE** email from Amazon Web Services, and the domain will display a status of "failed" in the **Domains** tab.

If this happens, please click the "retry" link next to the "failed" status notification. This will reinitiate the domain verification process. Add the new TXT record information to your DNS settings, and check with your DNS service provider to ensure that you have entered the TXT record information correctly.

To view your verified domains

1. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. In the navigation pane, click **Verified Senders**.
3. In the content pane, choose the **Domains** tab.

To remove a verified domain

1. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.

2. In the navigation pane, click **Verified Senders**.
3. In the content pane, choose the **Domains** tab.
4. Check the box beside each domain that you want to remove, then click **Remove**.
5. You will no longer be able to send email from the removed domain.

Domain revocation

Amazon Web Services periodically reviews domain verification status, and revokes verification in cases where it is no longer valid. If Amazon Web Services 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 Web Services.

If you restore the TXT record information to your DNS settings within 72 hours, you will receive an **Amazon SES Domain Verification REVOCATION CANCELLATION** email from Amazon Web Services.

If you do not restore the TXT record information to your DNS settings within 72 hours, you will receive an **Amazon SES Domain Verification REVOCATION** email from Amazon Web Services, 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.

Using the Amazon SES API

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

- `ListIdentities`
- `VerifyDomainIdentity`
- `DeleteIdentity`
- `GetIdentityVerificationAttributes`

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

At the command line, you can use the *ses-verify-identity.pl* script to verify your domains. For information about downloading and configuring this script, see Appendix: Amazon SES Scripts (p. 144).

Getting Your AWS Access Keys

After you've signed up for Amazon SES, you'll need to obtain your AWS access key identifiers if you want to send email by using the Amazon SES API either directly or by using an AWS SDK. These identifiers consist of an Access Key ID and a Secret Access Key. You will use these keys to access AWS SES programmatically.

> **Important**
> Your Secret Access Key is confidential and only you and AWS should know it. To protect your account, it is important to keep it secret. Never include it in your requests to AWS, and never email it to anyone. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your Secret Access Key.

To get your AWS access keys

1. Go to http://aws.amazon.com.
2. In the upper right corner, click **My Account/Console**, and then click **Security Credentials**.
3. If the **Sign In or Create an AWS Account** window appears, sign into your AWS account. (If the window doesn't appear, then you are already signed in.)
4. Navigate to the **Access Credentials** section of the page, and then click the **Access Keys** tab.
5. Copy your Access Key ID and Secret Access Key to a text file and put them in a safe place. To do so:

   - **Access Key ID—**On the **Access Keys** tab, under **Your Access Keys**, under **Access Key ID**, highlight your Access Key ID, press Ctrl-C to copy it, and then paste it into a TXT file.
   - **Secret Access Key—**On the **Access Keys** tab, under **Your Access Keys**, under **Secret Access Key**, click **Show**. In the **Secret Access Key** box that appears, highlight the Secret Access Key, press Ctrl-C, and then paste it into the TXT file.

   Important
   Do not share the Secret Access Key with anyone.

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 are available for the following programming languages:

- Java
- PHP
- .NET
- Android
- iOS
- 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. 25).

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

Authorizing Amazon SES Sending From Your Domain

If you manage your own DNS server and publish Sender Policy Framework (SPF) or Sender ID records in your domain's Domain Name Service (DNS), recipients might not receive email you send through

Amazon SES. To ensure reliable delivery, you need to authorize Amazon SES to send from your domain. To authorize Amazon SES, add the following to any SPF or Sender ID records:

```
include:amazonses.com
```

We recommend that all Amazon SES users publish both SPF and Sender ID records to their DNS. For more information, see Authenticating Email in Amazon SES (p. 81).

Many domain registrars provide self-service tools for administering DNS records. If such tools are unavailable, contact your domain registrar for assistance.

Getting Your SMTP Credentials for Amazon SES

To use the Amazon SES SMTP interface, you must first create an SMTP user name and password. 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.

To get your SMTP Credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 45).

Requesting Production Access to Amazon SES

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 the Amazon SES API so you can test and evaluate the service; however, the following restrictions are in effect:

- Emails can be sent only from verified email addresses or domains.
- Emails can be sent only to the Amazon SES mailbox simulator and to verified email addresses or domains.
- You can send a maximum of 200 messages per 24-hour period.
- You can send a maximum of one message per second.

To request production access to Amazon SES

1. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.

   Important
   You must sign in as the AWS account holder, not as an AWS Identity and Access Management (IAM) user. Otherwise, you will see "Permission Denied."

2. Go to http://aws.amazon.com/ses/fullaccessrequest and complete our brief request form.

   We will contact you by email after reviewing your request. Please allow one business day for processing.

The following are three ways to determine whether you have received production access:

- You received a confirmation email from AWS.
- 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. 106).

If you are granted production access, 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 submit an Extended Access Request if the gradual increase does not meet your needs. For more information, see Managing Your Amazon SES Sending Limits (p. 108).

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.

- 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. 44).
- 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. 62).

Before you send emails, see Setting up Amazon SES (p. 33).

> **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. For example, if a "BCC" address is on the blacklist, then the call will fail and the email is not sent to any of the "To", "CC", or "BCC" addresses. We therefore recommend that you send an email to one recipient at a time.

For definitions of Amazon SES terms, see the AWS Glossary.

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. 62). 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. 123).

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. 34).
- The SMTP interface hostname and port number. The hostname is *email-smtp.us-east-1.amazonaws.com*. The port number varies with the connection method. For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 46).
- An SMTP user name and password that you obtain from the Amazon SES console. For more information, see Obtaining Your Amazon SES SMTP Credentials (p. 45).
- Client software that can communicate using Transport Layer Security (TLS). For more information, see Connecting to the Amazon SES SMTP Endpoint (p. 46).
- An email address that you have verified with Amazon SES. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 34).
- Amazon SES production access, if you want to send large quantities of email. For more information, see Requesting Production Access to Amazon SES (p. 42).

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. 47).
- 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. 49).
- To program an application to send email through Amazon SES, including an example in PHP, see Sending Email Through Amazon SES From Your Application (p. 51).
- 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. 52).
- 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. 60).

For a list of SMTP response codes, see SMTP Response Codes Returned by Amazon SES (p. 123).

Obtaining Your Amazon SES SMTP Credentials

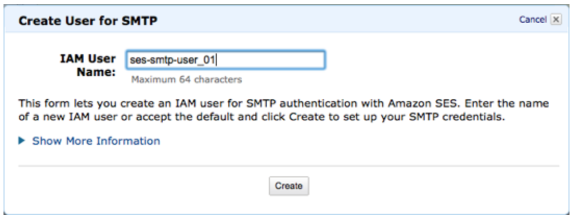
To use the Amazon SES SMTP interface, you must first create an SMTP user name and password.

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.

To create your SMTP credentials

1. Go to the AWS Management Console at http://aws.amazon.com/console and then log in with the email address and password you used when you signed up for Amazon SES.
2. Open the **Amazon SES** console.
3. In the navigation pane, click **SMTP Settings**.
4. In the content pane, click **Create My SMTP Credentials**.
5. 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**.

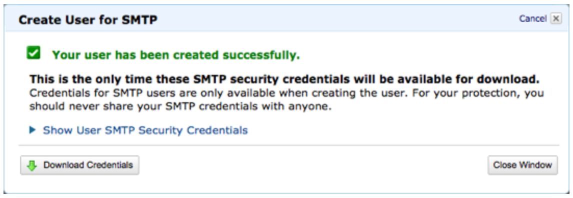


IAM User Name: ses-smtp-user_01

Maximum 64 characters

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.

6. 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 user has 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.

7. Click **Close Window** to dismiss the dialog box.

Amazon SES uses AWS Identity and Access Management (IAM) to create SMTP credentials. If you want to delete your SMTP credentials, go to the AWS IAM Dashboard in the AWS Management Console and delete the IAM user name that corresponds with your SMTP credentials. To learn more, go to the "Deleting a User from Your AWS Account" section of the *Using IAM* guide.

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

Connecting to the Amazon SES SMTP Endpoint

The Amazon SES SMTP endpoint (*email-smtp.us-east-1.amazonaws.com*) requires that all connections be encrypted using Transport Layer Security (TLS).

> **Note**
> TLS is often referred to by the name of its predecessor protocol, SSL.

Amazon SES supports two mechanisms for establishing an 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**
> Elastic Compute Cloud (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.

> **Important**
> Clients are responsible for verifying that they are using the correct server certificate. For the Amazon SES SMTP endpoint, the correct certificate information is:
>
> - Common Name: `email-smtp.us-east-1.amazonaws.com`
> - Serial Number: `2bbebf7bdeebb9a5931ff13206050840`
> - Thumbprint: `3e 86 aa a8 ca 0a 86 7b ec 56 f3 7e 00 7d b9 e2 30 a6 65 9c`
>
> If your client does not properly verify this information, the integrity of the connection cannot be guaranteed.

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. 52).

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).

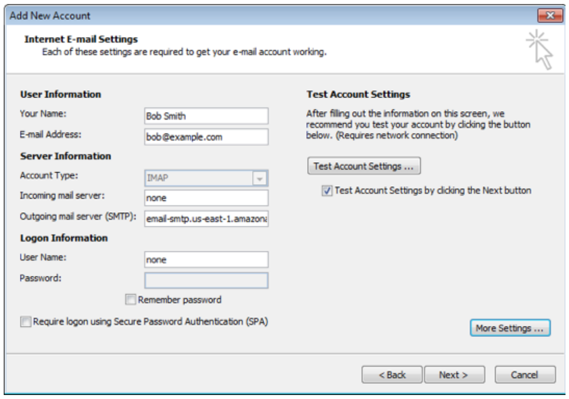
To configure your email client, you must provide the Amazon SES SMTP interface hostname (*email-smtp.us-east-1.amazonaws.com*) and port number, along with your SMTP user name and password. Here are some examples to help you get started.

Most email clients can send email using an SMTP server. 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. 34).
   c. **Account Type**—IMAP
   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)**—`email-smtp.us-east-1.amazonaws.com`
   f. **User Name**—type the word *none*. (You will configure your credentials later in this procedure.)



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

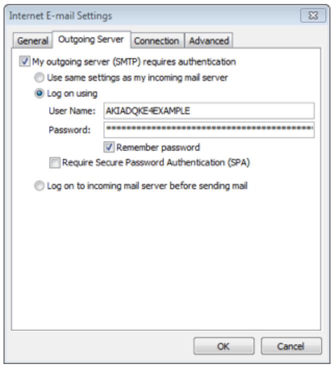
Server Information
Account Type: IMAP
Incoming mail server: none
Outgoing mail server (SMTP): email-smtp.us-east-1.amazon

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

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.
   d. **Password**—enter your SMTP password.
   e. **Remember Password**—check this box.



☑ 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)

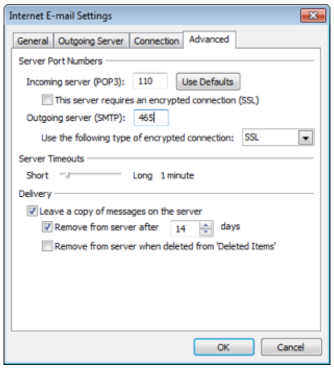
○ Log on to incoming mail server before sending mail

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.



Incoming server (POP3): 110

□ This server requires an encrypted connection (SSL)

Outgoing server (SMTP): 465

Use the following type of encrypted connection: SSL

Short — Long 1 minute

☑ 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. If the test message that Outlook sends through Amazon SES arrives successfully, click **Next**.
11. Click **Finish**.

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, login 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. 34).

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

   d. **Host Name**—`email-smtp.us-east-1.amazonaws.com`

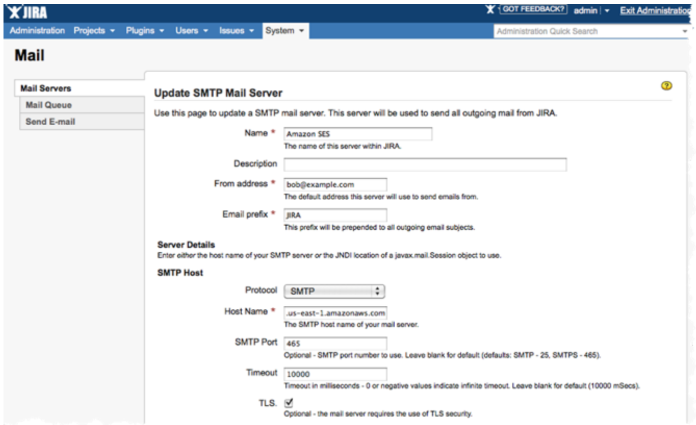
   e. **SMTP Port**—25, 587, or 2587 (to connect using STARTTLS), or 465 or 2465 (to connect using TLS Wrapper).

   f. **TLS**—Select this check box.

   g. **Username**—Your SMTP username.

   h. **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.

An example for PHP is provided below. For examples in C# and Java, see Sending an Email Through the Amazon SES SMTP Interface Programmatically (p. 18) in the Getting Started section.

Example : PHP and MySQL

The following code example is a PHP program that connects to a MySQL database and iterates through a table containing email addresses. For each address, the program sends an email message using the SMTP email capability of PHP.

This example was created on an Amazon EC2 instance with Apache, PHP, MySQL and the Postfix mail server. Postfix was configured to relay email through the Amazon SES SMTP interface. For more information on this topic, go to Integrating Amazon SES with Postfix (p. 53).

```
<?php

 mysql_connect("localhost", "mailapp", "mailpass")
     or die(mysql_error());
 mysql_select_db("test")
     or die(mysql_error());

 $result = mysql_query("select * from customers")
     or die(mysql_error());

 while ($row = mysql_fetch_array($result)) {

     $headers = "From: bob@example.com\n";

     $subject = "Thank you " . $row['first_name'] . "!";

     $message = "Dear " . $row['first_name'] . ",\n\n";
     $message .= "Thank you for being a loyal customer. " .
                "Please contact us at 555-4321 for a special discount offer.";


     mail($row['email_address'], $subject, $message, $headers);

     echo "Sent email to:  " .
                $row['first_name'] . " " . $row['last_name'] .
                "(" . $row['email_address'] . ")" . "<br>";

     sleep(5);
 }

?>
```

The `sleep` function causes the program to pause for five seconds between messages. You can adjust the number of seconds to keep the program from exceeding your maximum send rate with Amazon SES.

For more information about your maximum send rate, go to Managing Your Amazon SES Sending Limits (p. 108).

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. 53).
- To configure Sendmail to send email through Amazon SES, see Integrating Amazon SES with Sendmail (p. 55).
- To configure Exim to send email through Amazon SES, see Integrating Amazon SES with Exim (p. 58).

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. 52).

Setting Up a Secure Tunnel to Connect to Amazon SES

The Amazon SES SMTP endpoint (email-smtp.us-east-1.amazonaws.com) 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):

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

To set up a secure tunnel 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.

   ```
   [smtp-tls-wrapper]
   accept = 2525
   client = yes
   connect = email-smtp.us-east-1.amazonaws.com:465
   ```

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 conceived as an alternative to the widely used Sendmail MTA. It is designed to be fast, easy to administer, and secure.

For information about Postfix, go to http://www.postfix.org.

To configure integration using STARTTLS

1. On your mail server, open the *main.cf* file. On many systems, this file resides in the */etc/postfix* folder.
2. Add the following lines to the *main.cf* file, modifying them to reflect your particular situation, and then save the file.

   ```

   relayhost = email-smtp.us-east-1.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
   ```

3. 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.

   ```

   email-smtp.us-east-1.amazonaws.com:25 USERNAME:PASSWORD
   ses-smtp-prod-335357831.us-east-1.elb.amazonaws.com:25 USERNAME:PASSWORD
   ```

4. Save the *sasl_passwd* file.
5. At a command prompt, issue the following command to create an encrypted file containing your SMTP credentials.

   ```
   sudo postmap hash:/etc/postfix/sasl_passwd
   ```
6. Remove the */etc/postfix/sasl_passwd* file.
7. Tell Postfix where to find the CA certificate (needed to verify the SES server certificate).

   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'
   ```

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. 47). 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, modifying them to reflect your particular situation, and then save the 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
   ```

4. 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.

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

5. Save the *sasl_passwd* file.
6. At a command prompt, issue the following command to create an encrypted file containing your SMTP credentials.

   ```
   sudo postmap hash:/etc/postfix/sasl_passwd
   ```
7. Remove the */etc/postfix/sasl_passwd* file.

When you have finished updating the configuration, 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.

When you have completed this procedure, your outgoing email will be sent via the Amazon SES SMTP interface. 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

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

For more information about AMIs, 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 and m4.
- You have verified your "From" address and, if you do not yet have production access, you have also verified your "To" addresses. For more information, see Verifying Email Addresses in Amazon SES (p. 34).
- (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 Amazon SES using STARTTLS

1. Create or edit a file called */etc/mail/authinfo*. Add the following lines to the file, replacing *USERNAME* and *PASSWORD* with your Amazon SES SMTP user name and password.

   Important
   Your SMTP credentials are different from your AWS Access Key ID and Secret Key. For information about how to get your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 45).

```
AuthInfo:email-smtp.us-east-1.amazonaws.com "U:root" "I:USERNAME" "P:PASSWORD"
 "M:LOGIN"
```

```
AuthInfo:ses-smtp-prod-335357831.us-east-1.elb.amazonaws.com "U:root"
"I:USERNAME" "P:PASSWORD" "M:LOGIN"
```

2. Save the *authinfo* file.
3. At a command prompt, type the following command to generate */etc/mail/authinfo.db*:
   `sudo makemap hash /etc/mail/authinfo.db < /etc/mail/authinfo`
4. Open the */etc/mail/access* file and include support for relaying to the Amazon SES SMTP endpoint by adding the following lines:

```
Connect:email-smtp.us-east-1.amazonaws.com RELAY
Connect:ses-smtp-prod-335357831.us-east-1.elb.amazonaws.com RELAY
```

   Save the file.
5. Save a back-up copy of */etc/mail/sendmail.mc* and */etc/mail/sendmail.cf*.
6. 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
   Be sure to use the ` character and the apostrophe exactly as shown.

```

define(`SMART_HOST', `email-smtp.us-east-1.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
```

7. 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.
8. Save the *sendmail.mc* file.
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 regenerate *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`
13. 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 you have not yet applied for production access, 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.

14. 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. 52). 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. Create or edit a file called */etc/mail/authinfo*. Add the following lines to the file, replacing *USERNAME* and *PASSWORD* with your SMTP user name and password.

   Important
   Your SMTP credentials are different from your AWS Access Key ID and Secret Key. For information about how to get your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 45).

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

3. Save the *authinfo* file.
4. At a command prompt, type the following command:
   `sudo makemap hash /etc/mail/authinfo.db < /etc/mail/authinfo`
5. 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.
6. 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
   ```

7. 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
```

8. Save and close the *sendmail.mc* file.
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 regenerate *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
    ```
13. 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 you have not yet applied for production access, 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.

14. 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 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 using STARTTLS

1. Edit the */etc/exim/exim.conf* file. If the file does not exist, create it. In this file, 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-east-1.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:

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

2. 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. 47). 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. Edit the */etc/exim/exim.conf* file. If the file does not exist, create it. In this file, 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
```

3. 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://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 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-east-1.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-east-
1.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`

```
AUTH LOGIN

334 VXNlcm5hbWU6
c2VzLXNtdHAtdXNlcEXAMPLE
334 UGFzc3dvcmQ6
SkFYTVpaM3k0U2paVEYwOFpLEXAMPLE
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. 34).

```
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. 44). This section describes how to send email by using the API.

The Amazon SES API has a Query (HTTPS) interface. 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. 71).
- **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 Java, PHP, .NET, Android, iOS, and Ruby.
- **Use the AWS Command Line Interface (CLI)—**If you prefer to use a command-line interface that supports Amazon SES and other AWS operations such as request authentication, you can use the AWS CLI. The AWS CLI is Python-based. For more information, see AWS Command Line Interface.
- **Use Amazon SES scripts—**You can use Perl scripts to perform Amazon SES operations on the command line. The Perl scripts provide Amazon SES operations only. For more information about the Perl scripts, see Appendix: Amazon SES Scripts (p. 144). If you want command-line access to other AWS functionality in addition to Amazon SES operations, use the AWS Command Line Interface.

Regardless of whether you access the Amazon SES API by using raw Query requests, an AWS SDK, the CLI, or scripts, 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. 64).
- **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. 66).

Sending Formatted Email Using the Amazon SES API

You can send a formatted email using the AWS Management Console, calling the Amazon SES API from an application program, or using a command line script.

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
For tips on how to increase your email sending speed when you make multiple calls to `SendEmail`, see Increasing Throughput with Amazon SES (p. 122).

Example

The following code sample shows how to compose a formatted message. This code sample makes use of the AWS SDK for Java, which is available for download at http://aws.amazon.com/sdkforjava.

```
        package com.amazonaws.test;

        import java.util.ArrayList;
        import java.util.List;

        import com.amazonaws.AmazonClientException;
        import com.amazonaws.auth.BasicAWSCredentials;
        import com.amazonaws.services.simpleemail.*;
        import com.amazonaws.services.simpleemail.model.*;

        public class SendEmailSample {

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


          SendEmailRequest request = new SendEmailRequest()
              .withSource("bob@example.com");

          List<String> toAddresses = new ArrayList<String>();
          toAddresses.add("andrew@example.com");
          Destination dest = new Destination().withToAddresses(toAddresses);
          request.setDestination(dest);

          Content subjContent = new Content().withData("Test of Amazon SES");
          Message msg = new Message().withSubject(subjContent);

                // Include a body in both text and HTML formats.
                Content textContent = new Content().withData("Hello - I hope
you're having a good day.");
          Content htmlContent = new Content().withData("<h1>Hello - I hope
you're having a good day.</h1>");
          Body body = new Body().withHtml(htmlContent).withText(textContent);
          msg.setBody(body);

          request.setMessage(msg);

                // Set AWS access credentials.
                AmazonSimpleEmailServiceClient client =
                    new AmazonSimpleEmailServiceClient(
                        new BasicAWSCredentials(
                            "Access_key_ID_goes_here",
                            "Secret_key_goes_here"));

          // Call Amazon SES to send the message.
          try {
           client.sendEmail(request);
          } catch (AmazonClientException e) {
           System.out.println(e.getMessage());
          } catch (Exception e) {
           e.printStackTrace();
          }
         }
        }
```

At the command line, you can use the *ses-send-email.pl* script to send formatted email. For information about downloading and configuring this script, see Appendix: Amazon SES Scripts (p. 144).

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: MIME Types (p. 136).

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. 122).

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:

```
            package com.amazonaws.test;

            import java.io.ByteArrayOutputStream;
            import java.io.IOException;
            import java.io.PrintStream;
            import java.nio.ByteBuffer;
            import java.util.Properties;

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

            import com.amazonaws.AmazonClientException;
            import com.amazonaws.AmazonServiceException;
            import com.amazonaws.auth.BasicAWSCredentials;

           import com.amazonaws.services.simpleemail.AmazonSimpleEmailService
Client;
            import com.amazonaws.services.simpleemail.model.RawMessage;
          import com.amazonaws.services.simpleemail.model.SendRawEmailRequest;


            public class SendRawEmailSample {

                public static void main(String[] args)
                    throws AmazonClientException, AmazonServiceException,
                    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("andrew@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--
```

From here, it's a matter of calling the `SendRawEmail` action. The following code snippet continues where the previous program left off. This time, however, instead of writing the message to the console, the program sends it using Amazon SES:

```
// Capture the raw message
ByteArrayOutputStream out = new ByteArrayOutputStream();
msg.writeTo(out);

RawMessage rm = new RawMessage();
rm.setData(ByteBuffer.wrap(out.toString().getBytes()));

// Set AWS access credentials
AmazonSimpleEmailServiceClient client =
    new AmazonSimpleEmailServiceClient(
        new BasicAWSCredentials(
            "Access_key_ID_goes_here",
            "Secret_key_goes_here"));

// Call Amazon SES to send the message
try {
    client.sendRawEmail(
        new SendRawEmailRequest().withRawMessage(rm));
} catch (Exception e) {
    e.printStackTrace();
}
```

Feel free to adapt this sample code to your requirements.

At the command line, you can use the *ses-send-email.pl* script to send raw email. For information about downloading and configuring this script, see Appendix: Amazon SES Scripts (p. 144).

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. 71).
- For information about request authentication, see Request Authentication and Amazon SES (p. 74).
- For examples of GET and POST requests, see GET and POST Examples for Amazon SES (p. 77).
- For information about Query responses, see Query Responses and Amazon SES (p. 79).

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.
- **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 Amazon SES.

```
https://email.us-east-1.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-east-1.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 HMAC-SHA Signatures (p. 75).

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 HMAC-SHA Signatures (p. 75))

5. Send the completed request.

```
POST / HTTP/1.1
Date: Thu, 26 May 2011 06:49:50 GMT
Host: email.us-east-1.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 HMAC-SHA Signatures (p. 75)).

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

The topics in this section describe how Amazon Simple Email Service authenticates your requests. In this section you can learn about the basics of authentication, how your AWS account and identifiers are used to support authentication, and how to create an HMAC-SHA signature. This section also covers the request authentication requirements for Query requests.

What Is Authentication?

Authentication is a process for identifying and verifying who is sending a request.

General Process of Authentication

1	The sender obtains the necessary credential.
2	The sender sends a request with the credential to the recipient.
3	The recipient uses the credential to verify that the sender truly sent the request.
4	If so, the recipient processes the request. If no, the recipient rejects the request and responds accordingly.

During authentication, AWS verifies both the identity of the sender and whether the sender is registered to use services offered by AWS. If either test fails, the request is not processed.

The subsequent sections describe how Amazon SES implements authentication to protect your data.

Your AWS Account

To access any services offered by AWS, you must first create an AWS account at http://aws.amazon.com. An AWS account is simply an Amazon.com account that is enabled to use AWS products. You can use an existing Amazon.com account email address and password to create the AWS account.

Alternately, you could create a new AWS-enabled Amazon.com account by using a new email address and password. The address you provide must be valid. You'll be asked to provide a credit card or other payment method to cover the charges for any AWS products you use.

From your AWS account you can view your account activity, view usage reports, and manage your AWS account access identifiers.

To create an AWS account

1. Go to http://aws.amazon.com, and then click **Sign Up Now**.
2. Follow the on-screen instructions.

Your AWS Access Keys

When you create an AWS account, AWS assigns you a pair of related access keys:

- *Access Key ID* (a 20-character, alphanumeric sequence)

  For example: AKIAIOSFODNN7EXAMPLE
- *Secret Access Key* (a 40-character sequence)

  For example: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

> **Caution**
> Your Secret Access Key is a secret, and only you and AWS should know it. Never include your secret key in your requests to AWS, and never email it to anyone. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your Secret Access Key.

Viewing Your AWS Access Credentials

Your Access Key ID and Secret Access Key are displayed to you when you create your AWS Account. They are not emailed to you. If you need to see them again, you can view them at any time from your AWS account.

To view your AWS access credentials

1. Go to the Amazon Web Services website at http://aws.amazon.com.
2. Under **Your Account**, click **Security Credentials**.
3. In the spaces provided, type your user name and password, and then click **Sign in using our secure server**.
4. Under **Access Credentials**, on the **Access Keys** tab, your access ID key is displayed. To view your secret key, under **Secret Access Key**, **click Show**

HMAC-SHA Signatures

The Access Key ID is associated with your AWS account. You include it in AWS service requests to identify yourself as the sender of the request.

The Access Key ID is not a secret, and anyone could use your Access Key ID in requests to AWS. To provide proof that you truly are the sender of the request, you must also include a digital signature. For all requests, you calculate the signature using your Secret Access Key. AWS uses the Access Key ID in the request to look up your Secret Access Key and then calculates a digital signature with the key. If the signature AWS calculates matches the signature you sent, the request is considered authentic. Otherwise, the request fails authentication and is not processed.

The topics in this section describe how Amazon Simple Email Service uses HMAC-SHA signatures to authenticate query requests.

In order to access Amazon SES, you must provide the following items so the request can be authenticated:

- **AWSAccessKeyId**—Your AWS account is identified by your *Access Key ID*, which AWS uses to look up your Secret Access Key.
- **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.
- **Algorithm**—Identify which HMAC hash algorithm is being used to calculate your signature, either SHA256 or SHA1.

These items are used to construct an `X-Amzn-Authorization` HTTP header, which must be sent with every request. For information about HMAC, go to http://www.faqs.org/rfcs/rfc2104.html.

To authenticate a request to AWS, you create a request signature, which you place in an `X-Amzn-Authorization` HTTP header. This header must be included in the request that you send to AWS.

When AWS 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 AWS matches the one you sent in the request, AWS handles the request. If the comparison fails, the request is discarded, and AWS returns an error response.

You can send Query API requests to Amazon SES over HTTPS (Hypertext Transfer Protocol Secure). You must calculate an HMAC-SHA signature to be sent with every request.

The signature forms part of the *X-Amzn-Authorization* HTTP header, which must be sent with each request. The method used to construct the signature is known as *signature version 3*.

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-east-1.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-east-1.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-east-1.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 three authentication mechanisms ISPs use—Sender Policy Framework (SPF), Sender ID, 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. 81).
- To learn about Sender ID, an authentication mechanism similar to SPF, see Authenticating Email with Sender ID in Amazon SES (p. 82).
- 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. 83).

For the best delivery rates, and to help prevent spoofing and phishing, we recommend that all Amazon SES users maintain both SPF records (*v=spf1*) *and* Sender ID records (*spf2.0/pra*) in their DNS servers.

For definitions of Amazon SES terms, see the AWS Glossary.

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.

For more information about SPF, go to http://www.openspf.net and RFC 4408.

Domains with Preexisting SPF Records

If your "From" domain already has an SPF record, then you will need to add the following mechanism to it:

```
include:amazonses.com
```

Important
If you have an existing SPF record, then you *must* add this mechanism—otherwise, ISPs that examine "From:" headers might reject email that you send using Amazon SES.

Adding a New SPF Record

If your "From" domain does not have an SPF record, we recommend that you add one to ensure that ISPs do not reject your email. The following is an example TXT record that you can publish to enable SPF:

```
"v=spf1 include:amazonses.com -all"
```

Important
If you use "-all" as shown in the example above, ISPs may block email from IP addresses that are not listed in your SPF record. You therefore must add a record for every IP address that you send email from. As a debugging aid, you can use "~all" instead. When you use "~all", ISPs will typically accept email from IP addresses that are not listed. However, they may flag it. To maximize deliverability, use "-all" and add a record for each IP address.

Authenticating Email with Sender ID in Amazon SES

Sender ID is an authentication system similar to SPF. Like SPF, Sender ID relies on cooperation between senders and ISPs to verify that an email message can be traced back to the system from which it was sent.

To be Sender ID-compliant, an email sender publishes one or more DNS records to establish the sending domain's identity. For more information about Sender ID, go to http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx and RFC 4406.

Domains with Preexisting Sender ID Records

If your "From" domain already has a Sender ID record, then you must add the following mechanism to it:

```
include:amazonses.com
```

Important
If you have an existing Sender ID record, then you *must* add this mechanism—otherwise, ISPs that look for "From:" headers in message headers and bodies might reject email that you send using Amazon SES.

Adding a New Sender ID Record

If your *From* domain does not have a Sender ID record, we recommend that you add one to ensure that ISPs do not reject your email. You can publish the following example TXT record to enable Sender ID:

```
"spf2.0/pra include:amazonses.com -all"
```

Important
If you use "-all" as shown in the example above, ISPs may block email from IP addresses that are not listed in your Sender ID record. You therefore must add a record for every IP address that you send email from. As a debugging aid, you can use "~all" instead. When you use "~all", ISPs will typically accept email from IP addresses that are not listed. However, they may flag it. To maximize deliverability, use "-all" and add a record for each IP address.

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. 83).
- To add your own DKIM signature to any email that you send using the `SendRawEmail` API, see Manual DKIM Signing in Amazon SES (p. 90).

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 AWS Management 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.

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`).

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.

- To set up Easy DKIM while you verify a new domain, see Setting Up Easy DKIM for a New Domain (p. 84).
- 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. 86).

> **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.

Setting Up Easy DKIM for a New Domain

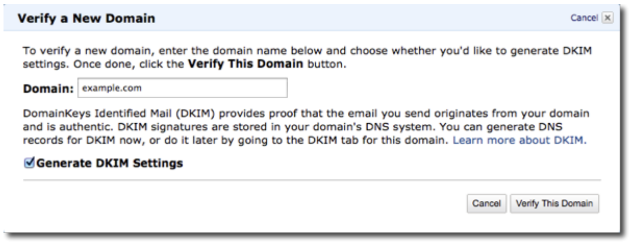
When you use the AWS Management console to verify a new domain, you can also set up Easy DKIM at the same time.

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. 86).

> **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 the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. In the navigation pane, click **Verified Senders**.
3. In the content pane, choose the **Domains** tab.
4. Click the **Verify a New Domain** button.
5. In the **Verify a New Domain** dialog box, enter your domain name, select the **Generate DKIM settings** check box, and click **Verify This Domain**.



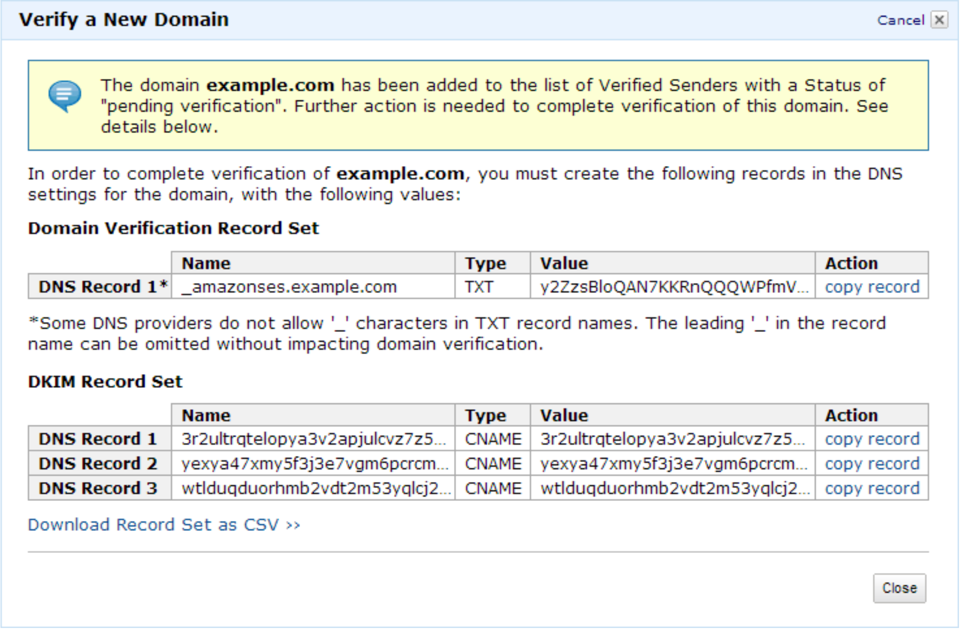
To verify a new domain, enter the domain name below and choose whether you'd like to generate DKIM settings. Once done, click the **Verify This Domain** button.

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. (After you close the dialog box, this information remains available to you in the **Details** section on the **Domains** tab.)



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	Action
DNS Record 1*	_amazonses.example.com	TXT	y2ZzsBloQAN7KKRnQQQWPfmV...	copy record

*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	Action
DNS Record 1	3r2ultrqtelopya3v2apjulcvz7z5...	CNAME	3r2ultrqtelopya3v2apjulcvz7z5...	copy record
DNS Record 2	yexya47xmy5f3j3e7vgm6pcrcm...	CNAME	yexya47xmy5f3j3e7vgm6pcrcm...	copy record
DNS Record 3	wtlduqduorhmb2vdt2m53yqlcj2...	CNAME	wtlduqduorhmb2vdt2m53yqlcj2...	copy record

Download Record Set as CSV ››

6. 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.)

   Use the **copy record** action to obtain the complete record, or select **Download Record Set CSV** to download all of the records.

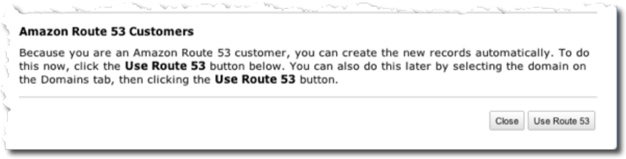
   Important

   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.

7. 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.

   Use the **copy record** action to obtain the CNAME records, or select **Download Record Set 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.



Amazon Route 53 Customers

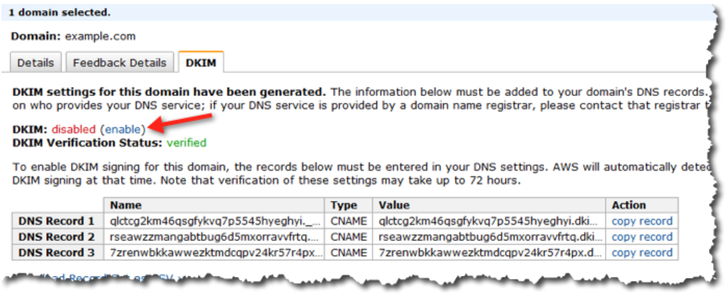
Because you are an Amazon Route 53 customer, you can create the new records automatically. To do this now, click the **Use Route 53** button below. You can also do this later by selecting the domain on the Domains tab, then clicking the **Use Route 53** button.

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 on the **Domains** tab 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** on the **DKIM** tab for your domain will change from *in progress* to *success*, and you will receive an *Amazon SES DKIM Setup Successful* confirmation email from Amazon Web Services.

8. To sign your messages using a DKIM signature, you must enable Easy DKIM for the appropriate verified sending identity. Go back into the **Verified Senders** area of the SES tab on the AWS Management Console and click the identity you wish to enable. Click the **DKIM** tab below, and click *enable* to enable DKIM.



1 domain selected.

Domain: example.com

DKIM settings for this domain have been generated. The information below must be added to your domain's DNS records. on who provides your DNS service; if your DNS service is provided by a domain name registrar, please contact that registrar t

DKIM: disabled (enable)
DKIM Verification Status: verified

To enable DKIM signing for this domain, the records below must be entered in your DNS settings. AWS will automatically detec DKIM signing at that time. Note that verification of these settings may take up to 72 hours.

	Name	Type	Value	Action
DNS Record 1	qlctcg2km46qsgfykvq7p5545hyeghyi._...	CNAME	qlctcg2km46qsgfykvq7p5545hyeghyi.dki...	copy record
DNS Record 2	rseawzzmangabtbug6d5mxorravvfrtq....	CNAME	rseawzzmangabtbug6d5mxorravvfrtq.dki...	copy record
DNS Record 3	7zrenwbkkawwezktmdcqpv24kr57r4px...	CNAME	7zrenwbkkawwezktmdcqpv24kr57r4px.d...	copy record

9. You can now use Amazon SES to send email that is signed using a DKIM signature from any enabled address in the verified domain. To send a test email, check the box next to the verified domain, then click **Send a Test Email**.

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.

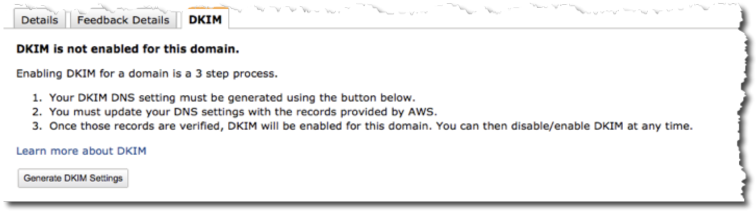
These instructions are for adding DKIM signing to a domain that has already been verified. If you are verifying a new domain and want to set up Easy DKIM at the same time, see Setting Up Easy DKIM for a New Domain (p. 84).

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 the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. In the navigation pane, click **Verified Senders**.
3. In the content pane, click the verified identity for which you would like to set up Easy DKIM.
4. Click the **DKIM** tab and click the **Generate DKIM Settings** button.



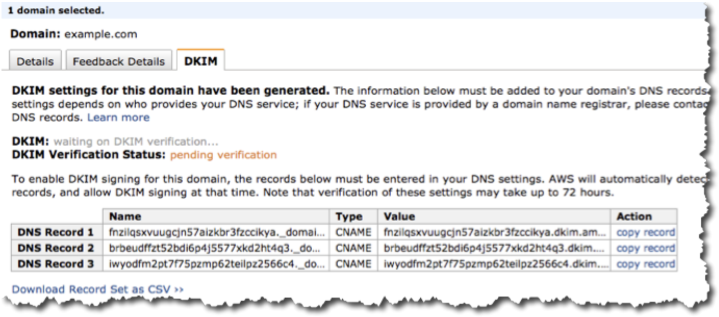
DKIM is not enabled for this domain.

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 those records are verified, DKIM will be enabled for this domain. You can then disable/enable DKIM at any time.

Learn more about DKIM

Your DKIM records will be displayed on the **DKIM** tab.



1 domain selected.

Domain: example.com

DKIM settings for this domain have been generated. The information below must be added to your domain's DNS records
settings depends on who provides your DNS service; if your DNS service is provided by a domain name registrar, please contac
DNS records. Learn more

DKIM: waiting on DKIM verification...
DKIM Verification Status: pending verification

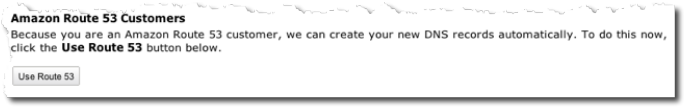
To enable DKIM signing for this domain, the records below must be entered in your DNS settings. AWS will automatically detec
records, and allow DKIM signing at that time. Note that verification of these settings may take up to 72 hours.

	Name	Type	Value	Action
DNS Record 1	fnzilqsxvuugcjn57aizkbr3fzccikya._domai...	CNAME	fnzilqsxvuugcjn57aizkbr3fzccikya.dkim.am...	copy record
DNS Record 2	brbeudffzt52bdi6p4j5577xkd2ht4q3._do...	CNAME	brbeudffzt52bdi6p4j5577xkd2ht4q3.dkim....	copy record
DNS Record 3	iwyodfm2pt7f75pzmp62teilpz2566c4._do...	CNAME	iwyodfm2pt7f75pzmp62teilpz2566c4.dkim....	copy record

Download Record Set as CSV ››

5. To set up DKIM, you must update your domain's DNS settings with the CNAME record information from the DKIM tab. 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.



Amazon Route 53 Customers

Because you are an Amazon Route 53 customer, we can create your new DNS records automatically. To do this now, click the **Use Route 53** button below.

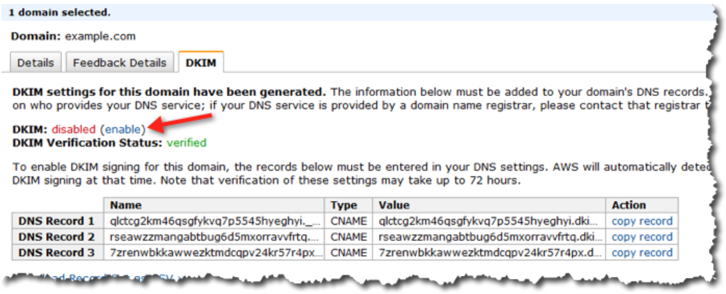
   Next, click **Create Record Sets** in the **Use Route 53** dialog box to complete the process.

   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 Easy DKIM is set up, the **DKIM Verification Status** on the **DKIM** tab for that domain 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.)

7. To sign your messages using a DKIM signature, you must enable Easy DKIM for the appropriate verified sending identity. Go back to the **Verified Senders** area of the *SES* tab of the AWS Management Console and click on the identity you want to enable. Click the **DKIM** tab below, and click *enable* to enable DKIM.



Domain: example.com

DKIM settings for this domain have been generated. The information below must be added to your domain's DNS records. on who provides your DNS service; if your DNS service is provided by a domain name registrar, please contact that registrar t

DKIM: disabled (enable)
DKIM Verification Status: verified

To enable DKIM signing for this domain, the records below must be entered in your DNS settings. AWS will automatically deted DKIM signing at that time. Note that verification of these settings may take up to 72 hours.

	Name	Type	Value	Action
DNS Record 1	qlctcg2km46qsgfykvq7p5545hyeghyi._...	CNAME	qlctcg2km46qsgfykvq7p5545hyeghyi.dki...	copy record
DNS Record 2	rseawzzmangabtbug6d5mxorravvfrtq....	CNAME	rseawzzmangabtbug6d5mxorravvfrtq.dki...	copy record
DNS Record 3	7zrenwbkkawwezktmdcqpv24kr57r4px...	CNAME	7zrenwbkkawwezktmdcqpv24kr57r4px.d...	copy record

8. You can now use Amazon SES to send email signed using a DKIM signature from any enabled address in the verified domain. To send a test email, check the box next to the verified domain, then click **Send a Test Email**.

> **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.

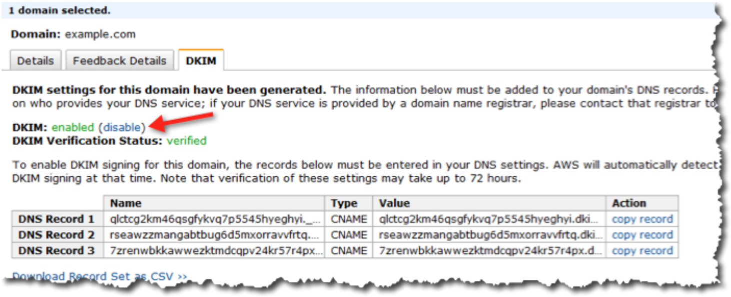
> **Note**
> If this happens, Amazon SES will still send your email, but *it will not be signed using a DKIM signature*.

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 domain. You can reenable it at any time.

To disable Easy DKIM signing

1. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. In the navigation pane, click **Verified Senders**.
3. Select the email address or domain for which you wish to disable Easy DKIM signing.
4. Click on the **DKIM** tab in the bottom pane.
5. In the **DKIM:** field that specifies pending, enabled, or failed, click the **disable** link next to the status.



1 domain selected.

Domain: example.com

DKIM settings for this domain have been generated. The information below must be added to your domain's DNS records. F
on who provides your DNS service; if your DNS service is provided by a domain name registrar, please contact that registrar to

DKIM: enabled (disable)
DKIM Verification Status: verified

To enable DKIM signing for this domain, the records below must be entered in your DNS settings. AWS will automatically detect
DKIM signing at that time. Note that verification of these settings may take up to 72 hours.

	Name	Type	Value	Action
DNS Record 1	qlctcg2km46qsgfykvq7p5545hyeghyi._...	CNAME	qlctcg2km46qsgfykvq7p5545hyeghyi.dki...	copy record
DNS Record 2	rseawzzmangabtbug6d5mxorravvfrtq....	CNAME	rseawzzmangabtbug6d5mxorravvfrtq.dki...	copy record
DNS Record 3	7zrenwbkkawwezktmdcqpv24kr57r4px...	CNAME	7zrenwbkkawwezktmdcqpv24kr57r4px.d...	copy record

6. In the dialog box that appears, select *Yes, Disable DKIM*. Amazon SES will no longer sign email that you send from this identity using a DKIM signature.

   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.

At the command line, you can use the *ses-setup-dkim.pl* script to set up Easy DKIM. For information about downloading and configuring this script, see Appendix: Amazon SES Scripts (p. 144).

Creating DNS Records for DKIM Signing in Amazon SES

Unlike the AWS Management Console, neither the Amazon SES API nor the *ses-setup-dkim.pl* script generates fully-formed DNS records for use with DKIM. Instead, the API and the script return DKIM *tokens* — character strings that represent your domain's identity.

If you are not using the Console, you will need to create your own `CNAME` records using the DKIM tokens returned by the API or the script..

To create DNS records for DKIM signing

1. Obtain the DKIM tokens for your domain.

   a. 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.

   b. If you are using the *ses-setup-dkim.pl* script, use the `-v` option to generate the tokens. If you already have a DKIM verified identity, use the `-a` option to obtain the tokens.

2. In the output from either the API or the script, 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.

> **Note**
> If you DKIM-sign your own email messages, we recommend that you use keys that are at least 1024 bits.

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.

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. 44).

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. 109) 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, the Amazon SES API, the AWS Command Line Interface, or an Amazon SES script to find the following kinds of information, for the last 24 hours:

- Delivery attempts
- Hard bounces
- Complaints
- Rejected send attempts
- Sending limits: Your sending quota and your maximum send rate
- Percentage of your quota that is used

You can also be notified of bounces and complaints by email or by Amazon Simple Notification Service (Amazon SNS).

The following sections describe how to monitor your sending activity:

- To receive notifications about bounces and complaints either by email or by Amazon Simple Notification Service (Amazon SNS), see Bounce and Complaint Notifications in Amazon SES (p. 93).
- To find the number of bounces, complaints, delivery attempts, and rejected send attempts, see Monitoring Your Amazon SES Usage Statistics (p. 105).
- 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. 106).

For definitions of Amazon SES terms, see the AWS Glossary.

Bounce and Complaint Notifications in Amazon SES

If you send an email that results in a bounce, the intended recipient's ISP sends a notification message to Amazon SES. Some ISPs also report complaints. In either case, Amazon SES forwards the notification to you either by email or by Amazon Simple Notification Service (Amazon SNS). Excessive bounces and complaints put your Amazon SES account at risk of termination, so you should monitor them and correct problems right away.

The following sections describe these notification methods:

- To receive notifications by email, see Feedback Notifications via Email in Amazon SES (p. 93).
- To receive notifications via Amazon SNS, see Amazon SES Feedback Notifications via Amazon SNS (p. 94).

You can test bounce and complaint notifications by using the Amazon SES mailbox simulator. For more information, see Testing Amazon SES Email Sending (p. 115).

> **Note**
> In terms of bounces, Amazon SES only notifies you of hard bounces. A hard bounce indicates a persistent delivery failure (for example, the mailbox does not exist). Amazon SES does not retry hard bounces.

Feedback Notifications via Email in Amazon SES

If you send an email message that results in a bounce or a complaint, the intended recipient's ISP will send a notification message to Amazon SES. When this happens, Amazon SES rewrites the *From:* header and forwards the notification to you.

How 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 SMTP's required MAIL FROM command, which overrides any *Return-Path:* header specified in the SMTP DATA.

If you used the **SendEmail API action** to send the message, then:

- If you specified `SendEmail`'s optional `ReturnPath` parameter, then notifications go to the specified address.
- Otherwise, notifications go to the address specified in `SendEmail`'s required `Source` parameter, which populates the *From:* header of the message.

If you used the **SendRawEmail API action** to send the message, then:

- If you specified `SendRawEmail`'s optional `Source` parameter, 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.

You will need to analyze each feedback message that you receive to determine the cause. Bounces are usually caused by attempting to send to a nonexistent recipient; complaints arise when the recipient indicates that they do not want to receive your message. In either case, we recommend that you stop sending to these email addresses.

Amazon SES Feedback Notifications via Amazon SNS

You can use Amazon Simple Notification Service (Amazon SNS) to receive notifications about bounces and complaints for emails that you send through Amazon SES. Once configured, you will receive bounce and complaint feedback notifications for emails through Amazon SNS in a machine-readable format, allowing you to process them programmatically. You can also disable feedback forwarding via email entirely if you configure Amazon SNS feedback notifications for both bounces and complaints.

> **Note**
> You can publish to any Amazon SNS topic to which your login has permissions—you do not need to be the topic owner in order to publish to it.

For information about Amazon SNS feedback notifications for Amazon SES, see the following sections:

- To set up feedback notifications using the Amazon SES console or the Amazon SES API, see Configuring Amazon SES Feedback Notifications via Amazon SNS (p. 94).
- For a description of the contents of a feedback notification, see Feedback Notification Contents for Amazon SES via Amazon SNS (p. 96).
- For examples of bounce and complaint notifications, see Examples of Amazon SES Bounce and Complaint Feedback Notifications (p. 102).

Configuring Amazon SES Feedback Notifications via Amazon SNS

Feedback notification settings are configured on a per-verified-identity basis (each email address and domain that you verify has configurable feedback notification settings). You must specify separate feedback notification configurations for different senders, as there is no global setting.

If a configuration is specified for a domain, it will apply to all mail sent from email addresses in that domain except for those email addresses that are also verified. The configurations for email addresses are completely separate from the configuration for the domain, so changing the domain configuration will have no effect on the email address configuration.

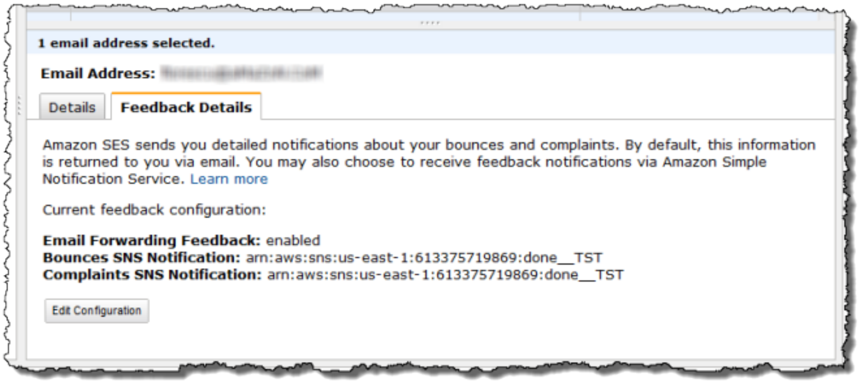
For example, if you verify only example.com and set its feedback notification topics, feedback for email from sender@example.com will use those topics. If you verify both example.com and sender@example.com, sender@example.com will NOT use feedback settings for example.com, whether or not they are set. Each verified identity must have its own configuration.

You can configure feedback notifications by using the AWS Management Console or by using the Amazon SES API, as described in the following sections.

Configuring Feedback Notifications Using the AWS Management Console

To configure feedback notifications using the AWS Management 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 **Verified Senders**.
3. In the Verified Senders pane, select the sender for whom you wish to configure feedback notifications.
4. In the bottom section of the Verified Senders pane, click the **Feedback Details** tab.



1 email address selected.

Email Address: [redacted]

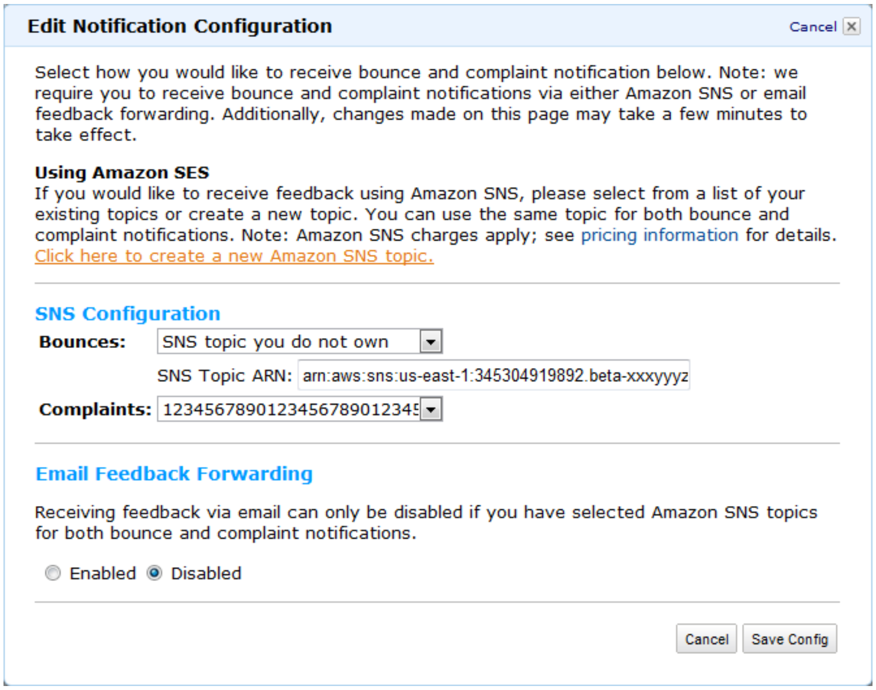
Amazon SES sends you detailed notifications about your bounces and complaints. By default, this information is returned to you via email. You may also choose to receive feedback notifications via Amazon Simple Notification Service. Learn more

Current feedback configuration:

Email Forwarding Feedback: enabled
Bounces SNS Notification: arn:aws:sns:us-east-1:613375719869:done__TST
Complaints SNS Notification: arn:aws:sns:us-east-1:613375719869:done__TST

5. To configure feedback notifications, click the **Edit Configuration** button.
6. In the dialog box, specify the existing Amazon SNS topics for bounces and complaints you want to use, or create a new SNS topic. You can choose to publish bounces and complaints to the same SNS topic or to different SNS topics. If you want to use an SNS topic that you do not own, you must configure your AWS Identity and Access Management (IAM) policy to allow publishing from the SNS topic's Amazon Resource Name (ARN). For information about how to control access to your SNS topic through the use of access policies, see Managing Access to Your Amazon SNS Topics.

   Note
   Changes made to your settings on this page will take a few minutes to take effect.



Edit Notification Configuration

Select how you would like to receive bounce and complaint notification below. Note: we require you to receive bounce and complaint notifications via either Amazon SNS or email feedback forwarding. Additionally, changes made on this page may take a few minutes to take effect.

Using Amazon SES
If you would like to receive feedback using Amazon SNS, please select from a list of your existing topics or create a new topic. You can use the same topic for both bounce and complaint notifications. Note: Amazon SNS charges apply; see pricing information for details.
Click here to create a new Amazon SNS topic.

SNS Configuration

Bounces: SNS topic you do not own
SNS Topic ARN: arn:aws:sns:us-east-1:345304919892.beta-xxxyyyz
Complaints: 12345678901234567890123456

Email Feedback Forwarding

Receiving feedback via email can only be disabled if you have selected Amazon SNS topics for both bounce and complaint notifications.

○ Enabled ● Disabled

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 **Feedback Forwarding**.
8. Click **Save Config** to save your notification configuration.

Once configured, you will start receiving bounce and complaint notifications to your Amazon SNS topic. These notifications will be in the JavaScript Object Notation (JSON) format and will follow the structure described in Feedback Notification Contents for Amazon SES via Amazon SNS (p. 96).

Note that you will be charged standard Amazon SNS rates for bounce and complaint 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, feedback forwarding will be re-enabled for that identity, and you will be notified of the change via 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 Feedback Notifications Using the Amazon SES API

You can also configure feedback 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 feedback notification. For a complete description of the API actions related to feedback notification, go to the Amazon Simple Email Service API Reference.

Feedback Notification Contents for Amazon SES via Amazon SNS

Feedback notifications are published to Amazon Simple Notification Service (Amazon SNS) topics in JavaScript Object Notation (JSON) format. The top-level JSON object contains a notification type, a `Mail` object, and either a `Bounce` object or a `Complaint` object.

- For a description of the top-level JSON object, see Top-Level JSON Object (p. 96).
- For a description of the `Mail` Object, see Mail Object (p. 97).
- For a description of the `Bounce` Object, see Bounce Object (p. 98).
- For a description of the `Complaint` Object, see Complaint Object (p. 100).

> **Note**
> You may receive multiple bounces in a single notification or a single notification per bounce. Your code should be able to handle both cases; Amazon SES does not make ordering or batching guarantees for notifications sent via Amazon SNS. The same is true for complaints. However, bounces and complaints will never be combined into a single notification, because they are different notification types.

Top-Level JSON Object

The top-level JSON object in an Amazon SES feedback 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 or Complaint.
mail	A JSON object that contains information about the original mail to which the notification pertains. For more information, see Mail Object (p. 97).
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. 98).
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. 100).

Mail Object

Each bounce or complaint 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 the 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).
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",
   "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. 99).
`bounceSubType`	The subtype of the bounce, as determined by Amazon SES. For more information, see Bounce Types (p. 99).
`bouncedRecipients`	A list that contains information about the recipients of the original mail that bounced. For more information, see Bounced Recipients (p. 99).
`timestamp`	The date and time at which the bounce was sent (in ISO8601 format). Note that this is the time at which the feedback 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 a Bounced Recipient object.

```
{
    "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.

> **Note**
> Amazon SES only reports hard bounces back to you. A hard bounce indicates a persistent delivery failure (e.g., mailbox does not exist). 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.
`Transient`	`General`	Amazon SES received a general hard 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. 101).
`timestamp`	The date and time at which the bounce was sent (in ISO8601 format). Note that this is the time at which the feedback 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 field:

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`—Indicates unsolicited email or some other kind of email abuse.
- `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.

Examples of Amazon SES Bounce and Complaint Feedback Notifications

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",
         "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",
          "messageId":"00000138111222aa-33322211-cccc-cccc-cccc-ddddaaaa0680-
000000",
          "destination":[
             "username@example.com"
          ]
       }
    }
```

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",
         "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",
         "destination":[
            "recipient1@example.com",
            "recipient2@example.com",
            "recipient3@example.com",
            "recipient4@example.com"
         ]
      }
   }
```

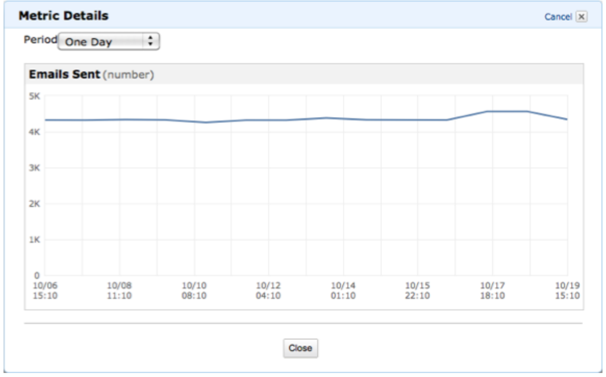
Monitoring Your Amazon SES Usage Statistics

You can monitor your usage statistics by using the Amazon SES console, the Amazon SES API, the AWS Command Line Interface, or an Amazon SES script.

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. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. Within the AWS Management Console, go to the Amazon SES console.
3. In the navigation pane, click **Dashboard**. Your usage statistics are shown under **Your Amazon SES Metrics**.
4. To view trend data for any metric, double-click the corresponding graph. The following example shows a close-up of the **Emails Sent** metric.



Metric Details

Period: One Day

5. 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:

- Hard bounces
- 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 Usage Statistics Using the AWS Command Line Interface (CLI)

At a command prompt, run *get-send-statistics*. For information about how to use the AWS CLI, see AWS Command Line Interface.

Monitoring Your Usage Statistics Using an Amazon SES Script

At a command prompt, you can run the *ses-get-stats.pl* script with the `-s` switch to check your sending quota and maximum send rate. For information about downloading and configuring this script, see Appendix: Amazon SES Scripts (p. 144).

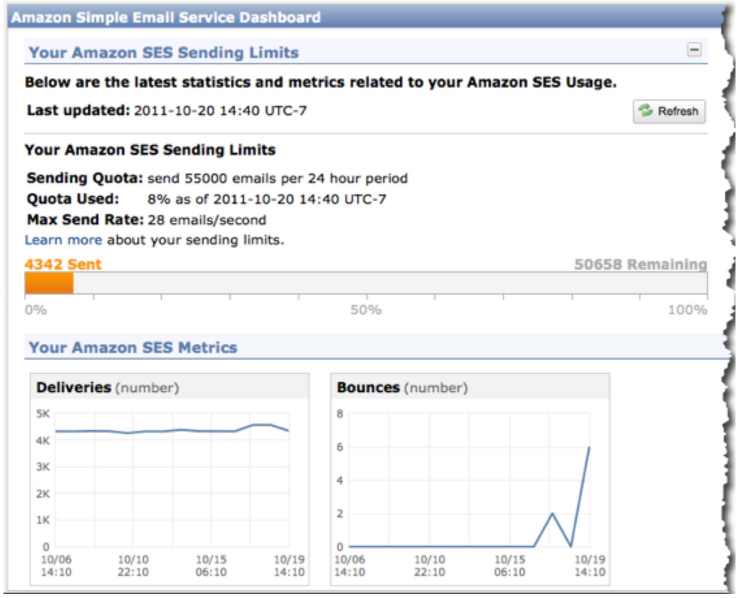
Monitoring Your Amazon SES Sending Limits

You can monitor your sending limits by using the Amazon SES console, the Amazon SES API, the AWS Command Line Interface, or an Amazon SES script.

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. Go to the AWS Management Console and log in with the email address and password you used when you signed up for Amazon SES.
2. Within the AWS Management Console, go to the Amazon SES console.
3. In the navigation pane, click **Dashboard**.



Amazon Simple Email Service Dashboard

Your Amazon SES Sending Limits

Below are the latest statistics and metrics related to your Amazon SES Usage.

Last updated: 2011-10-20 14:40 UTC-7

Your Amazon SES Sending Limits

Sending Quota: send 55000 emails per 24 hour period
Quota Used: 8% as of 2011-10-20 14:40 UTC-7
Max Send Rate: 28 emails/second
Learn more about your sending limits.

4342 Sent
50658 Remaining

Your Amazon SES Metrics

Your sending limits are shown under **Your Amazon SES Daily Volume Quota**. 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.

Monitoring Your Sending Limits Using the AWS Command Line Interface (CLI)

At a command prompt, run *get-send-quota*. For information about how to use the AWS CLI, see AWS Command Line Interface.

Monitoring Your Sending Limits Using an Amazon SES Script

At a command prompt, run the *ses-get-stats.pl* script with the `-q` switch to check your sending quota and maximum send rate. For information about downloading and configuring this script, see Appendix: Amazon SES Scripts (p. 144).

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 you can send per second. You can exceed this limit for short bursts, but not for a sustained period of time.

When you are in the 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 request production access. For more information, see Requesting Production Access to Amazon SES (p. 42). After you request production access and start sending emails, you can increase your sending limits further by following the guidance in the Increasing Your Amazon SES Sending Limits (p. 109) section.

> **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 one of the addresses is on the blacklist, 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. 109).

- 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. 110).
- To monitor your sending limits by using the console, the Amazon SES API, or a script, see Monitoring Your Amazon SES Sending Limits (p. 106).

For definitions of Amazon SES terms, see the AWS Glossary.

Increasing Your Amazon SES Sending Limits

Once you have been granted production access, 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 submit an Extended Access Request to explain in detail your need for a higher limit, and we will evaluate your request. For a list of the information that you will need when you submit the request, see the Submitting an Amazon SES Extended Access Request (p. 110) section.

> **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 submit an Extended Access Request 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 submit an Extended Access Request.

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. 115).

For information about submitting an Extended Access Request, see Submitting an Amazon SES Extended Access Request (p. 110).

Submitting an Amazon SES Extended Access Request

To submit an Extended Access Request, log into your AWS account, go to Extended Access Request, and then provide the following information:

- First name
- Last name
- Contact email address
- Telephone number
- Website address
- Link to your opt-in page
- Link to your opt-out policy
- Link to your privacy policy
- Email message type: Marketing, Transactional, Subscription, and/or System Notifications
- Anticipated maximum daily sending volume
- Use case detail

We highly recommend that you fill in the website address, opt-in, opt-out, and privacy page fields. However, these fields are not required (you can enter “None”).

In the use case detail field, explain your situation in as much detail as possible to support the case that the emails you send will be high quality. The more information you can provide that indicates that your sending will be high quality, the more likely it is that your request will be approved. The higher the jump you are requesting from your existing quota, the more detail you should provide.

Extended Access Requests are generally processed within one business day, but plan ahead and don’t wait until your situation is critical to submit the request.

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.

Improving Deliverability with Amazon SES

The following recommendations can help improve your deliverability when you use Amazon Simple Email Service (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 Bounce and Complaint Notifications in Amazon SES (p. 93).
- **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. 81).
- **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. 105).
- **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. 109).
- **Watch for upward trends in rejected emails.** Amazon SES will generate a *MessageRejected* error for any message that cannot be queued for delivery; 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.

Controlling User Access to Amazon SES

Amazon Simple Email Service (Amazon SES) integrates with AWS Identity and Access Management (IAM) so that you can specify which Amazon SES API actions a user can perform. For example, you can create an IAM policy that allows users in your organization to send email but not perform administrative actions. IAM manages which users are authorized to perform which tasks, but there are no changes to how you interact with Amazon SES or in how Amazon SES carries out requests.

When you use IAM, you define policies. A policy is a document that provides a formal statement of one or more permissions. You assign a policy to an entity, so that the entity must abide by the permissions stated in the policy. You can assign multiple policies to an entity.

> **Important**
> When you generate your SMTP credentials using the Amazon SES console, a new IAM user with the appropriate permissions and security credentials is created automatically. If you want to create an IAM user that has access to Amazon SES SMTP and other AWS services, you need to create the IAM user from the Amazon SES SMTP console and then apply additional IAM policies to this user. For more information about generating your SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials (p. 45).

For more information about IAM, see the IAM Documentation. For an example of a policy that covers Amazon SES actions, see Example IAM Policies for Amazon SES (p. 114).

Using IAM with Amazon SES

The following is information about using IAM with Amazon SES:

- You can't specify a particular Amazon SES resource in an IAM policy. You only control access to Amazon SES actions. Therefore, Amazon SES does not use Amazon Resource Names (ARNs), which identify resources in a policy. When you write a policy to control access to Amazon SES actions, you use * as the resource. For more information about ARNs, see Identifiers for IAM Entities in the IAM documentation.
- All Amazon SES API actions can be referred to in an IAM policy. Each action name must be prefixed with the lowercase string `ses:`. For example: `ses:SendEmail`, `ses:GetSendStatistics`, or `ses:*` (for all actions). For a list of available actions, see the Amazon SES API Reference.

- Amazon SES implements only the following AWS-wide policy keys. For more information about policy keys, see Element Descriptions in the IAM documentation.
  - `aws:CurrentTime` (for date/time conditions)
  - `aws:EpochTime` (the date in epoch or UNIX time, for use with date/time conditions)
  - `aws:SecureTransport` (Boolean representing whether the request was sent using SSL) For services that use only SSL, such as Amazon RDS and Amazon Route 53, the `aws:SecureTransport` key has no meaning.
  - `aws:SourceIp` (the requester's IP address, for use with IP address conditions) If you use `aws:SourceIp` and the request comes from an Amazon EC2 instance, we evaluate the instance's public IP address to determine if access is allowed.
  - `aws:UserAgent` (information about the requester's client application, for use with string conditions)

  The key names are case insensitive. For example, `aws:CurrentTime` is equivalent to `AWS:currenttime`.

Example IAM Policies for Amazon SES

This section shows a simple policy for controlling user access to Amazon SES.

> **Note**
> In the future, Amazon SES might add new actions that should logically be included in one of the following policies, based on the policy’s stated goals.

Example 1: Allow users to send email but not to perform administrative actions

This policy permits the group it is attached to (for example, the AllUsers group) to send email using Amazon SES. Users in this group cannot perform administrative actions, such as accessing Amazon SES sending statistics.

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

Example 2: Allow full access to all Amazon SES actions

This policy allows the group it is attached to (for example, the EmailAdmins group) to call any Amazon SES action.

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

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 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 has been blacklisted by Amazon SES.
- Generate a bounce without putting a valid email address on the blacklist.
- 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 and delivers it to the recipient's inbox. You will not receive any confirmation or other type of feedback 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 Simple Notification Service (Amazon SNS) notification, depending on how you set up your system. This mailbox simulator email address will not be blacklisted by Amazon SES 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 Bounce and Complaint Notifications in Amazon SES (p. 93).	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 Bounce and Complaint Notifications in Amazon SES (p. 93).	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 Bounce and Complaint Notifications in Amazon SES (p. 93).	complaint@simulator.amazonses.com
Address Blacklisted—Your attempt to send the email fails with a *MessageRejected* error that contains an "Address Blacklisted" error message. Amazon SES treats this address as if it will never accept emails and blocks your send attempt.	blacklist@simulator.amazonses.com

Important

If you send an email to a mailbox simulator address other than the addresses listed above, the unlisted address will be blacklisted.

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 in every scenario except for “Address Blacklisted.” 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 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.

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. 118).
- For a list of errors that can occur when you send an email with Amazon SES, see Amazon SES Email Sending Errors (p. 119).
- For information about how to remove an email address from the blacklist, see Removing an Email Address from the Amazon SES Blacklist (p. 121).
- 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. 122).
- 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. 123).
- 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. 123).
- For a list of error codes that are returned by the Amazon SES SES Query (HTTPS) API, see API Error Codes Returned by Amazon SES (p. 124).

If you are calling the Amazon SES API directly, see Amazon Simple Email Service API Reference for the HTTP errors that you might receive.

Amazon SES Delivery Problems

After you make a successful request to Amazon SES, the service queues your email message for sending. Your message is often sent immediately. At other times, there might be a short delay while the service sends other messages that are queued in front of yours. 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.

Process for Troubleshooting Email that Does Not Arrive

1	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. 79) 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.
2	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.
3	Verify that your SPF records are set up correctly. For more information, see Authenticating Email in Amazon SES (p. 81). Missing or misconfigured SPF records can cause deliverability problems for your email.
4	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.
5	Check the AWS Service Health Dashboard at http://status.aws.amazon.com to confirm that there is not a known problem with Amazon SES.
6	If you have signed up for AWS Premium Support, 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.

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. 123).

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**—One of the recipient email addresses has not been verified. This might apply to "Sender", "Return-Path", or "From" addresses.

  If you have not requested production access to Amazon SES, 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. 34) and Testing Amazon SES Email Sending (p. 115).
- **Address is blacklisted**—One of the recipient email addresses has caused a hard bounce within the past 14 days. This error refers to a recipient address, not a sender address. When a recipient address generates a hard bounce, Amazon SES considers that address to be temporarily unreachable, and blocks anyone from sending email to it for a period of 14 days. For example, suppose Sender A sends an email to *andrew@example.com*, but there is no user named *andrew* at the domain *example.com*. This results in a hard bounce, and the address *andrew@example.com* is added to the Amazon SES address blacklist. For the next 14 days, any email to *andrew@example.com* from Sender B, Sender C, etc., will be blocked, and all these senders will get "Address blacklisted" error messages.

  > **Note**
  > If you send an email to multiple recipients (a recipient is any "To", "CC", or "BCC" address) and one recipient's address is on the blacklist, the whole message is rejected and it doesn't get delivered to any of the recipients.

  Address blacklisting is Amazon SES-wide. If a recipient address has generated a hard bounce for *any* Amazon SES sender, then you will get this error message if you send email to that address, even if you have never sent email to it before.

  If you are sure that the email address that you're trying to send to is valid, you can submit a blacklist removal request. For more information, see Removing an Email Address from the Amazon SES Blacklist (p. 121).
- **Customer is blacklisted**—Your AWS account has been blocked from accessing Amazon SES. This may be due to repeated attempts to send low-quality content or violation of the Amazon Web Services Customer Agreement. If this happens, you should have received an email from Amazon SES informing you of the problem. To appeal your suspension, 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. 106). For general information about sending limits, see Managing Your Amazon SES Sending Limits (p. 108). For information about how to increase your sending limits, see Increasing Your Amazon SES Sending Limits (p. 109).

  > **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.

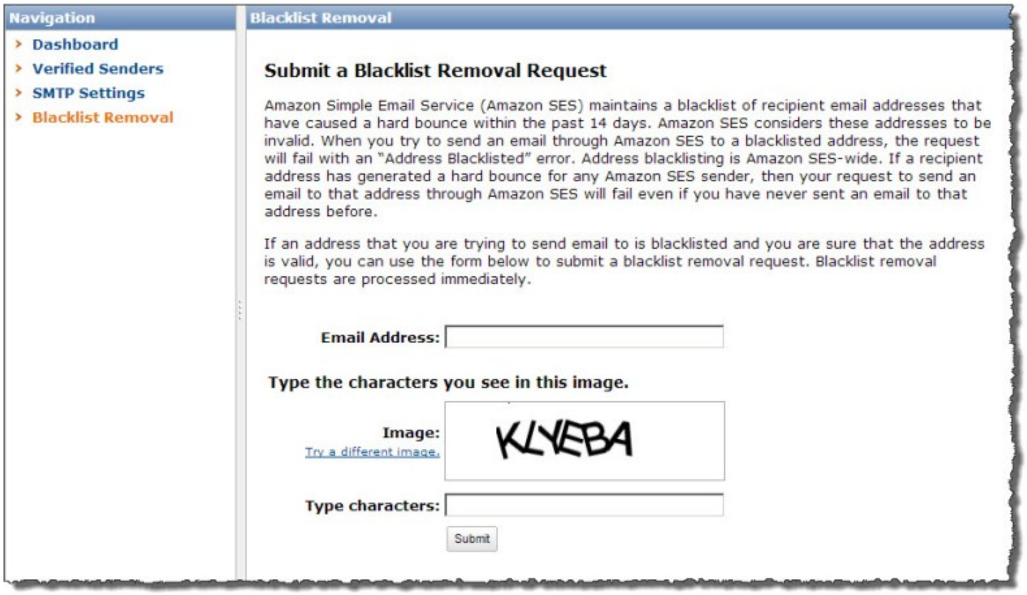
Removing an Email Address from the Amazon SES Blacklist

Amazon SES maintains a blacklist of recipient email addresses that have caused a hard bounce within the past 14 days. If you try to send an email through Amazon SES to an address on the blacklist, the call will fail and none of the recipients (including "To", "CC", and "BCC" addresses) will receive the email. If you are sure that the email address that you are trying to send to is valid, you can submit a blacklist removal request using the following procedure.

> **Note**
> If an address that you remove from the blacklist is indeed undeliverable, then the next time anyone sends an email to that address, it will hard bounce and the address will go back on the blacklist.

To remove an email address from the blacklist

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 **Blacklist Removal**.



Submit a Blacklist Removal Request

Amazon Simple Email Service (Amazon SES) maintains a blacklist 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 through Amazon SES to a blacklisted address, the request will fail with an "Address Blacklisted" error. Address blacklisting is Amazon SES-wide. If a recipient address has generated a hard bounce for any Amazon SES sender, then your request to send an email to that address through Amazon SES will fail even if you have never sent an email to that address before.

If an address that you are trying to send email to is blacklisted and you are sure that the address is valid, you can use the form below to submit a blacklist removal request. Blacklist removal requests are processed immediately.

3. In the **Email Address** field, type the email address that you want to remove from the blacklist.
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. Blacklist 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. 108).) 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. 52).
- **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.

  Note
  As of December 2012, Amazon SES is only available in the US East (Northern Virginia) Region. The following page lists AWS Regions with Amazon SES API endpoints: http://aws.amazon.com/about-aws/globalinfrastructure/regional-product-services/.

- **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. 62).
- **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. 115).

If you are accessing Amazon SES through its SMTP interface, see Amazon SES SMTP Issues (p. 123) 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. 44).

- **You cannot 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. 45).
- **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 SES 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.
- **You are unable to connect to the Amazon SES SMTP endpoint**—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-east-1.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. 60). 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.
- **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**—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.

SMTP Response Codes Returned by Amazon SES

SMTP response codes that Amazon SES returns are listed in the following table.

> **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
Successful delivery	250 Ok <*Message ID*> where <*Message ID*> is a string of characters that Amazon SES uses to uniquely identify the message.
Authentication successful	235 Authentication successful
Authentication Credentials Invalid	535 Authentication Credentials Invalid
Incorrect credentials	530 Authentication required.
Recipient address is blacklisted	554 Message rejected: Address blacklisted.
Unverified email address	554 Message rejected: Email address is not verified.
Daily sending quota exceeded	454 Throttling failure: Daily message quota exceeded.
Maximum send rate exceeded	454 Throttling failure: Maximum sending rate exceeded.

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. 119).	400	`SendEmail`, `SendRawEmail`
IncompleteSignature	The request signature does not conform to AWS standards.	400	All

Error	Description	HTTPS Status Code	Actions That Return This Code
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
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

Error	Description	HTTPS Status Code	Actions That Return This Code
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

Limits in Amazon SES

The following tables list limits within Amazon Simple Email Service (Amazon SES).

Sending Limits

Limit	Description
Sending limits in the sandbox environment (that is, without production access)	• Sending quota: 200 emails per 24-hour period. • Maximum send rate: 1 email per second. To increase your sending limits, apply for production access. For more information, see Requesting Production Access to Amazon SES (p. 42).
Starting sending limits in the production environment	• Sending quota: 10,000 emails per 24-hour period. • Maximum send rate: 5 emails per second. To increase your sending limits, you can wait for Amazon SES to increase your limits automatically, or you can apply for extended access. For more information about both methods, see Increasing Your Amazon SES Sending Limits (p. 109).

Message Limits

Limit	Description
Maximum message size (including attachments)	10 MB per message.
Accepted header fields	For a list of accepted header fields, see Appendix: Header Fields (p. 134).

Limit	Description
Accepted content types	For a list of accepted content types, see Appendix: MIME Types (p. 136).
Accepted attachment types	For a list of accepted attachment types, see Appendix: MIME Types (p. 136).

Sender and Recipient Limits

Limit	Description
Sender address	In both the sandbox and production environments, you are required to verify "From" email addresses and domains (including "Return-Path" and "Reply-To").
Recipient address	• 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, apply for production access. For more information, see Requesting Production Access to Amazon SES (p. 42). • Production environment: "To" addresses do not need to be verified.
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.

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.
AWS Glossary	The AWS Glossary. Contains definitions of common terms used in Amazon SES and other AWS services.
Production Access Request	The form to request production access to Amazon SES to increase your sending limits and eliminate the need to verify "To" addresses.
Extended Access Request	The form to request an increase to your sending limits.
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 Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support (if you have subscribed to this program).

Resource	Description
AWS Premium Support Information	The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
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.

Amazon SES Developer Guide Document History

This documentation is associated with the 2010-12-01 release of Amazon Simple Email Service (Amazon SES). This guide was last updated on March 20, 2013.

The following table describes the major changes to the documentation since the last release of Amazon SES.

Change	Description	Date Changed
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

Change	Description	Date Changed
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

Amazon SES Developer Guide Appendix

This appendix contains supplementary information about sending emails with Amazon Simple Email Service (Amazon SES).

- For a list of header fields that Amazon SES accepts, see Appendix: Header Fields (p. 134).
- For a list of Multipurpose Internet Mail Extensions (MIME) types that you can use with Amazon SES, see Appendix: MIME Types (p. 136).
- For a list of command-line scripts that you can use to perform Amazon SES tasks, see Appendix: Amazon SES Scripts (p. 144).

Appendix: Header Fields

Amazon SES accepts the following email header fields.

Note
Amazon SES also accepts user-defined X-headers, which must begin with "X-" or "x-". X-headers are defined in RFC 822.

- Accept-Language
- acceptLanguage
- Archived-At
- Authentication-Results
- Auto-Submitted
- Bcc
- Bounces-To
- Cc
- Comments
- Content-Alternative
- Content-Class
- Content-Description
- Content-Disposition

- Content-Duration
- Content-Features
- Content-ID
- Content-Language
- Content-Length
- Content-Location
- Content-MD5
- Content-Transfer-Encoding
- Content-Type
- Date

  Note

  Amazon SES ignores the timestamp that you specify in this header and replaces it with the timestamp of the server. It does so to prevent problems with time synchronization between client systems and Amazon SES. If you do not include this header, Amazon SES will add it automatically.
- 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
- MIME-Version
- Organization
- Original-From
- Original-Message-ID
- Original-Recipient
- Original-Subject
- Precedence
- Priority
- PICS-Label
- Received
- Received-SPF
- References
- Reply-To

- Return-Path
- Return-Receipt-To
- Sender

  Note
  You must verify the email address that you use in the *Sender* header. For more information, see Verifying Email Addresses and Domains in Amazon SES (p. 34)

- Sensitivity
- Solicitation
- Subject
- Thread-Index
- Thread-Topic
- To
- User-Agent
- VBR-Info

Appendix: MIME Types

Amazon SES accepts the Multipurpose Internet Mail Extensions (MIME) types shown in the following tables.

General

Content Type	Extension
application/ics	ics
application/ms-tnef	dat

Content Type	Extension
application/octet-stream	ai, asc, avi, bak, bmp, bpm, conf, css, csv, dcm, dicom, diff, doc, docx, dotx, gif, gpx, htm, html, ical, icalendar, ico, ics, ifb, img, jp2, jpe, jpeg, jpg, jpm, jpx, kml, kmz, mov, mp3, mp4, mpe, mpg, msg, p10, p7c, p7n, p7s, pdf, pem, pkpass, png, ppsx, ppt, pptx, psd, pst, qt, rpt, rtf, rtx, sgml, sig, svg, targa, text, tfx, tgatif, tiff, tsv, txt, vcard, vcf, vsd, wav, wbmp, wml, xls, xlsx, xltx, xml
application/msoutlook	msg
application/vnd.apple.pkpass	pkpass
application/vnd.google-earth.kmz	kmz
application/vnd.ms-outlook	msg
application/x-bittorrent	torrent
application/x-msg	msg
application/x-x509-ca-cert	pem
message/rfc822	N/A
multipart/*	N/A

Text

Content Type	Extension
text/calendar	ical, icalendar, ics, ifb
text/comma-separated-values	csv
text/css	css

Content Type	Extension
text/csv	csv
text/html	htm, html
text/plain	bak, conf, diff, log, text, txt, rpt
text/richtext	rtf, rtx
text/rtf	rtf, rtx
text/sgml	sgml
text/tab-separated-values	tsv
text/vnd.wap.wml	wml, xml
text/x-vcard	vcard, vcf
text/xml	nzb, xml

Image

Content Type	Extension
application/acad	dwf, dwg, dxf
application/autocad_dwg	dwg
application/dicom	dcm, dicom, img
application/dwf	dwf
application/dwg	dwg
application/dxf	dxf
application/octet-stream	dwf, dwg, dxf, odg, pdd, psb
application/vnd.oasis.opendocument.graphics	odg
application/x-acad	dwf, dwg, dxf
application/x-autocad	dwf, dwg, dxf
application/x-dwf	dwf
application/x-dwg	dwg
application/x-dxf	dxf
application/x-vnd.oasis.opendocument.graphics	odg
drawing/dwg	dwg
drawing/x-dwf	dwf
drawing/x-dxf	dxf

Content Type	Extension
image/bmp	bmp
image/gif	gif
image/jp2	jp2
image/jpeg	jpe, jpeg, jpg
image/jpg	jpe, jpeg, jpg
image/jpm	jpm
image/jpx	jpx
image/photoshop	psd
image/pjpeg	pjpg
image/png	png
image/psd	psd
image/svg+xml	svg
image/targa	targa, tga
image/tiff	tif, tiff
image/tiff-fx	tfx
image/vnd.dwf	dwf
image/vnd.dwg	dwg
image/vnd.dxf	dxf
image/vnd.microsoft.icon	ico
image/vnd.wap.wbmp	wbmp
image/x-autocad	dxf
image/x-bmp	bmp
image/x-dicom	dcm, dicom
image/x-dwf	dwf
image/x-dwg	dwg
image/x-dxf	dxf
image/x-ms-bmp	bmp
image/x-photoshop	psd
image/x-targa	targa, tga
image/x-tga	targa, tga

Encryption

Content Type	Extension
application/pgp-encrypted	(Any)
application/pgp-keys	asc
application/pgp-signature	asc, sig
application/pkcs10	p10
application/pkcs7-mime	p7m, p7c
application/pkcs7-signature	p7s
application/x-pkcs7-mime	p7m, p7c
application/x-pkcs7-signature	p7s

Documents

Content Type	Extension
application/acrobat	pdf
application/epub+zip	epub
application/illustrator	ai
application/msword	doc, dot
application/octet-stream	aeh, fb2, iba, kf8, odf, odm, odp, ods, odt, otg, oth, otp, ots, ott, smf, stc, std, sti, stw, sxc, sxd, sxg, sxi, sxm, sxw, twb, twbx, xps
application/oleobject	ods
application/pdf	pdf
application/photoshop	psd
application/postscript	ai, eps, ps
application/psd	psd
application/rtf	rtf
application/sgml	sgml

Content Type	Extension
application/soffice	smf, sti, stw, sxc, sxd, sxg, sxi, sxm, sxw
application/visio	vsd
application/visio.drawing	vsd
application/vnd.amazon.ebook	azw
application/vnd.google-earth.kml+xml	kml
application/vnd.ms-excel	xla, xls, xlt
application/vnd.ms-excel.addin.macroEnabled.12	xlam
application/vnd.ms-excel.sheet.binary.macroEnabled.12	xlsb
application/vnd.ms-excel.sheet.macroEnabled.12	xlsm
application/vnd.ms-excel.template.macroEnabled.12	xltm
application/vnd.ms-powerpoint	pot, ppa, pps, ppt
application/vnd.ms-powerpoint.addin.macroEnabled.12	ppam
application/vnd.ms-powerpoint.presentation.macroEnabled.12	pptm
application/vnd.ms-powerpoint.slideshow.macroEnabled.12	ppsm
application/vnd.ms-powerpoint.template.macroEnabled.12	potm
application/vnd.ms-word.document.macroEnabled.12	docm
application/vnd.ms-word.template.macroEnabled.12	dotm
application/vnd.ms-xpsdocument	xps
application/vnd.msword	doc
application/vnd.oasis.opendocument.formula	odf
application/vnd.oasis.opendocument.graphics-template	otg
application/vnd.oasis.opendocument.presentation	odp
application/vnd.oasis.opendocument.presentation-template	otp
application/vnd.oasis.opendocument.spreadsheet	ods
application/vnd.oasis.opendocument.spreadsheet-template	ots
application/vnd.oasis.opendocument.text	odt
application/vnd.oasis.opendocument.text-master	odm
application/vnd.oasis.opendocument.text-web	oth
application/vnd.oasis.opendocument.text-template	ott
application/vnd.openxmlformats-officedocument.presentationml.presentation	pptx

Content Type	Extension
application/vnd.openxmlformats-officedocument.presentationml.slideshow	ppsx
application/vnd.openxmlformats-officedocument.presentationml.template	potx
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	xlsx
application/vnd.openxmlformats-officedocument.spreadsheetml.template	xltx
application/vnd.openxmlformats-officedocument.wordprocessingml.document	docx
application/vnd.openxmlformats-officedocument.wordprocessingml.template	dotx
application/vnd.stardivision.math	smf
application/vnd.sun.xml.calc	sxc
application/vnd.sun.xml.draw	sxd
application/vnd.sun.xml.calc.template application/x-soffice application/soffice	stc
application/vnd.sun.xml.draw.template application/x-soffice application/soffice	std
application/vnd.sun.xml.impress	sxi
application/vnd.sun.xml.math	sxm
application/vnd.sun.xml.impress.template	sti
application/vnd.sun.xml.writer	sxw
application/vnd.sun.xml.writer.global	sxg
application/vnd.sun.xml.writer.template	stw
application/vnd.visio	vsd
application/vsd	vsd
application/x-mobipocket-ebook	mobi
application/x-ms-reader	lit
application/x-msworks-db	wdb
application/x-msworks-wp	wps
application/x-obak	lit
application/x-soffice	smf, sti, stw, sxc, sxd, sxg, sxi, sxm, sxw
application/x-rpt	rpt
application/x-visio	vsd
application/x-vnd.oasis.opendocument.formula	odf
application/x-vnd.oasis.opendocument.graphics-template	otg
application/x-vnd.oasis.opendocument.presentation	odp

Content Type	Extension
application/x-vnd.oasis.opendocument.presentation-template	otp
application/x-vnd.oasis.opendocument.spreadsheet	ods
application/x-vnd.oasis.opendocument.spreadsheet-template	ots
application/x-vnd.oasis.opendocument.text	odt
application/x-vnd.oasis.opendocument.text-master	odm
application/x-vnd.oasis.opendocument.text-template	ott
application/x-vnd.oasis.opendocument.text-web	oth
application/x-vsd	vsd
application/xhtml+xml	htm, html, xhtml
application/xml	xml
image/pdf	pdf
text/xml	twb

Audio and Video

Content Type	Extension
application/octet-stream	3g2, 3gp, aif, aiff, flac, m4r, ogg, pss
application/vnd.rn-realmedia	rm
audio/aiff	aif, aiff
audio/midi	mid, midi
audio/mp4a-latm	m4a, m4b, m4p
audio/mpeg	mp3
audio/mpeg3	mp3
audio/wav	wav
audio/x-aiff	aif, aiff
audio/x-flac	flac
audio/x-mpeg	mp3
audio/x-mpeg-3	mp3
audio/x-ms-wma	wma
audio/x-pn-realaudio	ra, ram
audio/x-wav	wav
video/3gpp	3gp

Content Type	Extension
video/3gpp2	3g2
video/avi	avi
video/mp4	mp4
video/mpeg	mpe, mpg
video/msvideo	avi
video/ogg	ogg
video/quicktime	mov, qt
video/x-m4v	m4v
video/x-ms-asf	asf, asx
video/x-ms-wmv	wmv

Appendix: Amazon SES Scripts

Amazon SES provides several command line 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
These scripts are optional. They are provided as a convenience only, and you are not required to use the scripts in order to use Amazon SES.

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. Because it is not secure to provide these credentials on the command line, these scripts use an alternative way of accessing your AWS access key ID and secret access key.

To provide an Amazon SES script with your AWS credentials

1. Create a new text file with the name of your choice—*aws-credentials*, for example.
2. Add the following lines to the file, and then save the file. Replace the key values shown with your actual AWS credentials:

```
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE

AWSSecretKey=wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY
```

3. Change the permissions on the file so that only you have access. The procedure for this depends on your operating system.

Important
Nobody else should ever see your secret access key.

After you have created the credential file, you can use these scripts in one of the following ways:

- Use the *-k* parameter, followed by the name of your credential file.
- Set an environment variable named *AWS_CREDENTIALS_FILE*, whose value is the name of the file containing your AWS credentials. The procedure for this depends on your operating system. Here is an example of how you can do this on a Linux system, using the *bash* shell:

```
export AWS_CREDENTIALS_FILE=aws-credentials
```

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

```
ses-verify-email-address.pl [--help] [-e URL] [-k FILE] [--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 *https://email.us-east-1.amazonaws.com*.	No
`-k`	Specifies a file containing your AWS credentials. For more information, see Using Your AWS Credentials (p. 144).	Yes
`--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
`-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:

```
$ ses-verify-email-address.pl -k aws-credentials -l
```

Example

This example shows how to request verification:

```
$ ses-verify-email-address.pl -k aws-credentials -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:

```
$ ses-verify-email-address.pl -k aws-credentials -d user@example.com
```

Managing Identities

You can verify email addresses or domains using *ses-verify-identity.pl*.

Syntax

```
ses-verify-identity.pl [--help] [-e URL] [-k FILE] [--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

Name	Description	Required
`-e`	Uses the specified Amazon SES endpoint. The default is *https://email.us-east-1.amazonaws.com*.	No
`-k`	Specifies a file containing your AWS credentials. For more information, see Using Your AWS Credentials (p. 144).	Yes
`--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:

```
$ ses-verify-identity.pl -k aws-credentials -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:

```
$ ses-verify-identity.pl -k aws-credentials -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:

```
$ ses-verify-identity.pl -k aws-credentials -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:

```
$ ses-verify-identity.pl -k aws-credentials -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:

```
$ ses-verify-identity.pl -k aws-credentials -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:

```
$ ses-verify-identity.pl -k aws-credentials -d user@example.com
```

This example shows how to delete a domain from your list:

```
$ ses-verify-identity.pl -k aws-credentials -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. 115).

Syntax

```
ses-setup-notifications.pl [--help] [-e URL] [-k FILE] [--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 *https://email.us-east-1.amazonaws.com*.	No

Name	Description	Required
`-k`	Specifies a file containing your AWS credentials. For more information, see Using Your AWS Credentials (p. 144).	Yes
`--verbose`	Displays detailed information about the endpoint response.	No
`-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:

```
$ ses-setup-notifications.pl -k aws-credentials -a example.com,username@ex
ample.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:

```
$ ses-setup-notifications.pl -k aws-credentials -i example.com -f false
```

Example

This example shows how to enable email forwarding for bounces and complaints:

```
$ ses-setup-notifications.pl -k aws-credentials -i example.com -f true
```

Example

This example shows how to setup an Amazon SNS topic to receive bounce notifications:

```
$ ses-setup-notifications.pl -k aws-credentials -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:

```
$ ses-setup-notifications.pl -k aws-credentials -i user@example.com -n Bounce
-t
```

Managing DKIM Settings

You can configure DKIM settings using *ses-setup-dkim.pl*.

Syntax

```
ses-setup-dkim.pl [--help] [-e URL] [-k FILE] [--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 *https://email.us-east-1.amazonaws.com*.	No
`-k`	Specifies a file containing your AWS credentials. For more information, see Using Your AWS Credentials (p. 144).	Yes
`--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:

```
$ ses-setup-dkim.pl -k aws-credentials -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*:

```
$ ses-setup-dkim.pl -k aws-credentials -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*:

```
$ ses-setup-dkim.pl -k aws-credentials -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:

```
$ ses-setup-dkim.pl -k aws-credentials -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

```
ses-send-email.pl [--help] [-e URL] [-k FILE] [--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 *https://email.us-east-1.amazonaws.com*.	No
`-k`	Specifies a file containing your AWS credentials. For more information, see Using Your AWS Credentials (p. 144).	Yes
`--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. 108).

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 $ character represents the operating system command line prompt.

This example shows how to send an email message to a single recipient.

```
$ ses-send-email.pl -k aws-credentials -s "Greetings" -f user@example.com an
drew@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 this by pressing CTRL+D.

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.

```
$ ses-send-email.pl -k aws-credentials -s "Greetings" -f user@example.com "an
drew@example.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. 153). Use the *-r* parameter to specify raw email.

Syntax

```
ses-send-email.pl [--help] [-e URL] [-k FILE] [--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 *https://email.us-east-1.amazonaws.com*.	No
`-k`	Specifies a file containing your AWS credentials. For more information, see Using Your AWS Credentials (p. 144).	Yes
`--verbose`	Displays detailed information about the endpoint response.	No

Name	Description	Required
`-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
`-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. 108).

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.

```
$ 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

```
ses-get-stats.pl [--help] [-e URL] [-k FILE] [--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 *https://email.us-east-1.amazonaws.com*.	No
`-k`	Specifies a file containing your AWS credentials. For more information, see Using Your AWS Credentials (p. 144).	Yes
`--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.

```
$ ses-get-stats.pl -k aws-credentials -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

```
ses-get-stats.pl [--help] [-e URL] [-k FILE] [--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 *https://email.us-east-1.amazonaws.com*.	No
`-k`	Specifies a file containing your AWS credentials. For more information, see Using Your AWS Credentials (p. 144).	Yes
`--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.

```
$ ses-get-stats.pl -k aws-credentials -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. 144).

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
-k /opt/third-party/amazon/aws-credentials -e https://email.us-east-
1.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.
- The endpoint for communicating with Amazon SES is always *https://email.us-east-1.amazonaws.com*.

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. 144) and Sending Formatted Email (p. 153).

- You have verified your "From" address and, if you do not yet have production access, you have also verified your "To" addresses. For more information, see Verifying Email Addresses in Amazon SES (p. 34).
- (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
    -k /opt/third-party/amazon/aws-credentials -e   https://email.us-east-
   1.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.
   - The endpoint for communicating with Amazon SES is *https://email.us-east-1.amazonaws.com*.
   - 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/mailmailertable* file.
7. Add the following line to */etc/mailmailertable* 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 you have not yet applied for production access, 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.