Amazon SNS FAQs

## **Overview**

Q: What is Amazon Simple Notification Service (Amazon SNS)?

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from [the cloud](https://aws.amazon.com/what-is-cloud-computing/). It provides developers with a highly scalable, flexible, and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or other applications. It is designed to make web-scale computing easier for developers. Amazon SNS follows the [“publish-subscribe” (pub-sub) messaging paradigm](https://aws.amazon.com/pub-sub-messaging/), with notifications being delivered to clients using a “push” mechanism that eliminates the need to periodically check or “poll” for new information and updates. With simple APIs requiring minimal up-front development effort, no maintenance or management overhead and pay-as-you-go pricing, Amazon SNS gives developers an easy mechanism to incorporate a powerful notification system with their applications.

Q: How can I get started using Amazon SNS?

You can create an Amazon SNS topic and publish messages in a few steps by completing our 10-minute tutorial, [Send Fanout Notifications](https://aws.amazon.com/getting-started/tutorials/send-fanout-event-notifications/).

For additional detail, see the [Amazon SNS Developer Guide](http://docs.aws.amazon.com/sns/latest/dg/), and [sample code in the Resource Center](http://aws.amazon.com/code/Amazon-SNS?browse=1).

Q: What are the benefits of using Amazon SNS?

Amazon SNS offers several benefits making it a versatile option for building and integrating loosely-coupled, distributed applications:

* Instantaneous, push-based delivery (no polling)
* Simple APIs and easy integration with applications
* Flexible message delivery over multiple transport protocols
* Inexpensive, pay-as-you-go model with no up-front costs
* Web-based AWS Management Console offers the simplicity of a point-and-click interface

Q: What are some example uses for Amazon SNS notifications?

The Amazon SNS service can support a wide variety of needs including event notification, monitoring applications, workflow systems, time-sensitive information updates, mobile applications, and any other application that generates or consumes notifications. For example, Amazon SNS can be used in workflow systems to relay events among distributed computer applications, move data between data stores or update records in business systems. Event updates and notifications concerning validation, approval, inventory changes and shipment status are immediately delivered to relevant system components as well as end-users. A common pattern is to use SNS to publish messages to [Amazon SQS](https://aws.amazon.com/sqs/) message queues to reliably send messages to one or many system components asynchronously. Another example use for Amazon SNS is to relay time-critical events to mobile applications and devices. Since Amazon SNS is both highly reliable and scalable, it provides significant advantages to developers who build applications that rely on real-time events.

Q: How does Amazon SNS work?

It is very easy to get started with Amazon SNS. Developers must first create a “topic” which is an “access point” – identifying a specific subject or event type – for publishing messages and allowing clients to subscribe for notifications. Once a topic is created, the topic owner can set policies for it such as limiting who can publish messages or subscribe to notifications, or specifying which notification protocols will be supported (i.e. HTTP/HTTPS, email, SMS). Subscribers are clients interested in receiving notifications from topics of interest; they can subscribe to a topic or be subscribed by the topic owner. Subscribers specify the protocol and end-point (URL, email address, etc.) for notifications to be delivered. When publishers have information or updates to notify their subscribers about, they can publish a message to the topic – which immediately triggers Amazon SNS to deliver the message to all applicable subscribers.

Q: How is Amazon SNS different from Amazon SQS?

Amazon Simple Queue Service (SQS) and Amazon SNS are both messaging services within AWS, which provide different benefits for developers. Amazon SNS allows applications to send time-critical messages to multiple subscribers through a “push” mechanism, eliminating the need to periodically check or “poll” for updates. Amazon SQS is a [message queue service](https://aws.amazon.com/message-queue/) used by distributed applications to exchange messages through a polling model, and can be used to decouple sending and receiving components. Amazon SQS provides flexibility for distributed components of applications to send and receive messages without requiring each component to be concurrently available.

A common pattern is to use SNS to publish messages to Amazon SQS queues to reliably send messages to one or many system components asynchronously.

Q: How is Amazon SNS different from Amazon MQ?

Amazon MQ, Amazon SQS, and Amazon SNS are messaging services that are suitable for anyone from startups to enterprises. If you're using messaging with existing applications, and want to move your messaging to the cloud quickly and easily, we recommend you consider [Amazon MQ](https://aws.amazon.com/amazon-mq/). It supports industry-standard APIs and protocols so you can switch from any standards-based message broker to Amazon MQ without rewriting the messaging code in your applications. If you are building brand new applications in the cloud, we recommend you consider [Amazon SQS](https://aws.amazon.com/sqs/) and Amazon SNS. Amazon SQS and SNS are lightweight, fully managed message queue and topic services that scale almost infinitely and provide simple, easy-to-use APIs. You can use Amazon SQS and SNS to decouple and scale microservices, distributed systems, and serverless applications, and improve reliability.

Q: How can I get started using Amazon SNS?

To sign up for Amazon SNS, click the “Sign up for Amazon SNS” button on the Amazon SNS detail page. You must have an Amazon Web Services account to access this service; if you do not already have one, you will be prompted to create one when you begin the Amazon SNS sign-up process. After signing up, please refer to the Amazon SNS documentation and Getting Started Guide to begin using Amazon SNS. Using the AWS Management Console, you can easily create topics, add subscribers, send notifications, and edit topic policies – all from your browser.

Q: Is Amazon SNS supported in the AWS Management Console?

Amazon SNS is supported in the [AWS Management Console](http://aws.amazon.com/console) which provides a point-and-click, web-based interface to access and manage Amazon SNS. Using the AWS Management Console, you can create topics, add subscribers, and send notifications – all from your browser. In addition, the AWS Management Console makes it easy to publish messages to your endpoint of choice (HTTP, SQS, Lambda, mobile push, email, or SMS) and edit topic policies to control publisher and subscriber access.

Q: What are the Amazon SNS service access points in each region?

Please refer to the [AWS Regions and Endpoints](http://docs.aws.amazon.com/general/latest/gr/rande.html#sns_region) section of the AWS documentation for the latest list of all Amazon SNS service access points.

Q: Can I get a history of SNS API calls made on my account for security analysis and operational troubleshooting purposes?

Yes. SNS supports [AWS CloudTrail](http://aws.amazon.com/cloudtrail/), a web service that records AWS API calls for your account and delivers log files to you. With CloudTrail, you can obtain a history of such information as the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by SNS.

SNS currently supports CloudTrail auditing for authenticated calls only. CloudTrail Audit logs for unauthenticated ConfirmSubscription and Unsubscribe calls are not available at this time. For more information, see the [CloudTrail section of the SNS Developer Guide](http://docs.aws.amazon.com/sns/latest/dg/logging-using-cloudtrail.html).

To receive a history of SNS API calls made on your account, simply [turn on AWS CloudTrail in the AWS Management Console](https://console.aws.amazon.com/cloudtrail). To learn more about AWS CloudTrail, [click here](http://aws.amazon.com/cloudtrail/).

## **Billing**

Q: How much does Amazon SNS cost?

With Amazon SNS, there is no minimum fee and you pay only for what you use. Users pay $0.50 per 1 million Amazon SNS Requests, $0.06 per 100,000 notification deliveries over HTTP, and $2.00 per 100,000 notification deliveries over email. For SMS messaging, users can send 100 free notification deliveries, and for subsequent messages charges vary by destination country.

Amazon SNS also includes a Free Tier, where users can get started with Amazon SNS for free. Each month, Amazon SNS customers incur no charges for the first 1 million Amazon SNS requests, no charges for the first 100,000 notifications over HTTP, no charges for the first 100 notifications over SMS, and no charges for the first 1,000 notifications over email.

Please refer to the [Amazon SNS Details page](https://aws.amazon.com/sns/details/) for additional details on pricing and data transfer costs.

Q: How will I be charged and billed for my use of Amazon SNS?

There are no set-up fees to begin using the service. At the end of the month, your credit card will automatically be charged for that month’s usage. You can view your charges for the current billing period at any time on the Amazon Web Services web site by logging into your Amazon Web Services account and clicking “Account Activity” under “Your Web Services Account”.

Q: When does billing of my Amazon SNS use begin and end?

Your Amazon SNS billing cycle begins on the first day of each month and ends on the last day of each month. Your monthly charges will be totalled at the end of each month.

Q: Do your prices include taxes?

Except as otherwise noted, our prices are exclusive of applicable taxes and duties, including VAT and applicable sales tax. For customers with a Japanese billing address, use of AWS services is subject to Japanese Consumption Tax. [Learn more](https://aws.amazon.com/c-tax-faqs/).

## **Features and Functionality**

Q: What is the format of an Amazon SNS topic?

Topic names are limited to 256 characters. Alphanumeric characters plus hyphens (-) and underscores (\_) are allowed. Topic names must be unique within an AWS account. After you delete a topic, you can reuse the topic name. When a topic is created, Amazon SNS will assign a unique ARN (Amazon Resource Name) to the topic, which will include the service name (SNS), region, AWS ID of the user and the topic name. The ARN will be returned as part of the API call to create the topic. Whenever a publisher or subscriber needs to perform any action on the topic, they should reference the unique topic ARN.

The following is the ARN for a topic named “mytopic” created by a user with the AWS account ID “123456789012” and hosted in the US East region:

arn:aws:sns:us-east-1:1234567890123456:mytopic Note: Users should NOT attempt to build the topic ARN from its separate components – they should always use the name returned from the API call to create the topic.

Q: What are the available operations for Amazon SNS and who can perform these operations?

Amazon SNS provides a set of simple APIs to enable event notifications for topic owners, subscribers and publishers.

Owner operations:

* CreateTopic – Create a new topic.
* DeleteTopic – Delete a previously created topic.
* ListTopics – List of topics owned by a particular user (AWS ID).
* ListSubscriptionsByTopic – List of subscriptions for a particular topic
* SetTopicAttributes – Set/modify topic attributes, including setting and modifying publisher/subscriber permissions, transports supported, etc.
* GetTopicAttributes – Get/view existing attributes of a topic
* AddPermission – Grant access to selected users for the specified actions
* RemovePermission – Remove permissions for selected users for the specified actions

Subscriber operations:

* Subscribe – Register a new subscription on a particular topic, which will generate a confirmation message from Amazon SNS
* ConfirmSubscription – Respond to a subscription confirmation message, confirming the subscription request to receive notifications from the subscribed topic
* UnSubscribe – Cancel a previously registered subscription
* ListSubscriptions – List subscriptions owned by a particular user (AWS ID)

Publisher operations:

* Publish: Publish a new message to the topic.

Q: Why are there two different APIs to list subscriptions?

The two APIs to list subscriptions perform different functions and return different results:

* The ListSubscriptionsByTopic API allows a topic owner to see the list of all subscribers actively registered to a topic.
* The ListSubscriptions API allows a user to get a list of all their active subscriptions (to one or more topics).

Q: What are the different delivery formats/transports for receiving notifications?

In order for customers to have broad flexibility of delivery mechanisms, Amazon SNS supports notifications over multiple transport protocols. Customers can select one the following transports as part of the subscription requests:

* “HTTP”, “HTTPS” – Subscribers specify a URL as part of the subscription registration; notifications will be delivered through an HTTP POST to the specified URL.
* ”Email”, “Email-JSON” – Messages are sent to registered addresses as email. Email-JSON sends notifications as a JSON object, while Email sends text-based email.
* “SQS” – Users can specify an SQS standard queue as the endpoint; Amazon SNS will enqueue a notification message to the specified queue (which subscribers can then process using SQS APIs such as ReceiveMessage, DeleteMessage, etc.). Note that FIFO queues are not currently supported.
* “SMS” – Messages are sent to registered phone numbers as SMS text messages.

Q: Can topic owners control the transports that are allowed on topics they create/own?

Topic owners can configure specific transports on their topics by setting the appropriate permissions through access control policies.

Q: How does an owner set Access Control policies?

Please refer to the Amazon SNS Getting Started Guide for an overview of setting access control policies.

Q: Can a single topic support subscriptions over multiple protocols/transports?

Subscribers to an Amazon SNS topic can receive notifications on any transport supported by the topic. A topic can support subscriptions and notification deliveries over multiple transports.

Q: Can subscribers selectively receive only a subset of messages published to a topic?

Yes, you can use message filtering on Amazon Simple Notification Service (SNS) to build simpler and more streamlined pub/sub architectures. Message filtering enables Amazon SNS topic subscribers to selectively receive only a subset of the messages they are interested in, as opposed to receiving all messages published to a topic. For more details, try our 10-minute tutorial, [Filter Messages Published to Topics](https://aws.amazon.com/getting-started/tutorials/filter-messages-published-to-topics/), or see the [documentation](https://aws.amazon.com/documentation/sns/).

Q: Can Amazon SNS be used with other AWS services?

Amazon SNS can be used with other AWS services such as Amazon SQS, Amazon EC2 and Amazon S3. Here is an example of how an order processing workflow system uses Amazon SNS with Amazon EC2, SQS, and SimpleDB. In this workflow system, messages are sent between application components whenever a transaction occurs or an order advances through the order processing pipeline. When a customer initially places an order, the transaction is first recorded in Amazon SimpleDB and an application running on Amazon EC2 forwards the order request to a payment processor which debits the customer’s credit card or bank account. Once approved, an order confirmation message is published to an Amazon SNS topic. In this case, the topic has various subscribers over Email/HTTP – merchant, customer and supply chain partners – and notifications sent by Amazon SNS for that topic can instantly update all of them that payment processing was successful. Notifications can also be used to orchestrate an order processing system running on EC2, where notifications sent over HTTP can trigger real-time processing in related components such as an inventory system or a shipping service. By integrating Amazon SNS with Amazon SQS, all notifications delivered are also persisted in an Amazon SQS queue where they are processed by an auditing application at a future time.

Q: Is Amazon SNS available in all regions where AWS services are available?

Please refer to the [AWS Regions and Endpoints](http://docs.aws.amazon.com/general/latest/gr/rande.html#sns_region) section of the AWS documentation for the most up to date information on Amazon SNS availability.

Q: How soon can customers recreate topics with previously used topic names?

Topic names should typically be available for reuse approximately 30-60 seconds after the previous topic with the same name has been deleted. The exact time will depend on the number of subscriptions which were active on the topic – topics with a few subscribers will be available instantly for reuse, topics with larger subscriber lists may take longer.

## **Transports**

Q: How would a user subscribe for notifications to be delivered over email?

To receive email notifications for a particular topic, a subscriber should specify “Email” or “Email-JSON” as the protocol and provide a valid email address as the end-point. This can be done using the [AWS Management Console](http://aws.amazon.com/console) or by calling the Amazon SNS API directly. Amazon SNS will then send an email with a confirmation link to the specified email address, and require the user monitoring the email address to explicitly opt-in for receiving email notifications from that particular topic. Once the user confirms the subscription by clicking the provided link, all messages published to the topic will be delivered to that email address.

Q: Why does Amazon SNS provide two different transports to receive notifications over email?

The two email transports are provided for two distinct types of customers/end-users. “Email-JSON” sends notifications as a JSON object, and is meant for applications to programmatically process emails. The ”Email” transport is meant for end-users/consumers and notifications are regular, text-based messages which are easily readable.

Q: Can a user change the Subject and Display name for notifications sent over Email/Email-JSON?

Amazon SNS allows users to specify the Subject field for emails as a parameter passed in to the Publish API call and can be different for every message published. The Display name for topics can be set using the SetTopicAttributes API – this name applies to all emails sent from this topic.

Q: Do subscribers need to specifically configure their email settings to receive notifications from Amazon SNS?

In most cases, users should be able to receive subscription confirmations and notifications from Amazon SNS without doing anything specific. However, there could be cases where the email provider’s default settings or other user-specific configurations mistakenly redirect the emails to the junk/spam folder. To ensure that users see confirmation messages and notifications sent from Amazon SNS, users can add “no-reply@sns.amazonaws.com” to their contact lists and check their junk/spam folders for messages from Amazon SNS.

Q: In the case of passing in an SQS queue as an endpoint, will users need to create the queue prior to subscribing? What permissions will the queue require?

Using the SQS console, users should create the SQS queue prior to subscribing it to a Topic. Select this queue on the console, and from the ‘Queue Actions’ in the menu bar, select ‘Subscribe Queue to SNS Topic’ from the drop-down list. In the subscribe dialog box, select the topic from the ‘Choose a Topic’ drop-down list, and click the ‘Subscribe’ button. For complete step-by-step instructions, please refer to the [Amazon SNS documentation](https://aws.amazon.com/documentation/sns/).

Q: Are Amazon SQS FIFO queues compatible with Amazon Simple Notification Service (SNS)?

Amazon SNS does not currently support forwarding messages to Amazon SQS FIFO queues. You can use SNS to forward messages to standard queues.

Q: How would a developer setup an Amazon SQS queue to receive Amazon SNS notifications?

To have Amazon SNS deliver notifications to an SQS queue, a developer should subscribe to a topic specifying “SQS” as the transport and a valid SQS standard queue as the end-point. In order to allow the SQS queue to receive notifications from Amazon SNS, the SQS queue owner must subscribe the SQS queue to the Topic for Amazon SNS to successfully deliver messages to the queue.

If the user owns both the Amazon SNS topic being subscribed to and the SQS queue receiving the notifications, nothing further is required. Any message published to the topic will automatically be delivered to the specified SQS queue. If the user owning the SQS queue is not the owner of the topic, Amazon SNS will require an explicit confirmation to the subscription request.

Please refer to the [Amazon SNS documentation](https://aws.amazon.com/documentation/sns/) for further details on subscribing an SQS queue to a topic and setting access control policies for SQS queues.

Q: How can I fanout identical messages to multiple SQS queues?

Create an SNS topic first using SNS. Then create and subscribe multiple SQS standard queues to the SNS topic. Now whenever a message is sent to the SNS topic, the message will be fanned out to the SQS queues, i.e. SNS will deliver the message to all the SQS queues that are subscribed to the topic.

Q: What is the format of structured notification messages sent by Amazon SNS?

The notification message sent by Amazon SNS for deliveries over HTTP, HTTPS, Email-JSON and SQS transport protocols will consist of a simple JSON object, which will include the following information:

* MessageId: A Universally Unique Identifier, unique for each notification published.
* Timestamp: The time (in GMT) at which the notification was published.
* TopicArn: The topic to which this message was published
* Type: The type of the delivery message, set to “Notification” for notification deliveries.
* UnsubscribeURL: A link to unsubscribe the end-point from this topic, and prevent receiving any further notifications.
* Message: The payload (body) of the message, as received from the publisher.
* Subject: The Subject field – if one was included as an optional parameter to the publish API call along with the message.
* Signature: Base64-encoded “SHA1withRSA” signature of the Message, MessageId, Subject (if present), Type, Timestamp, and Topic values.
* SignatureVersion: Version of the Amazon SNS signature used.

Notification messages sent over the “Email” transport only contain the payload (message body) as received from the publisher.

Q: How would a user subscribe for notifications to be delivered over SMS?

Please refer to the 'SMS Related Question' section below.

## **Security**

Q: How can users secure the messages sent to my topics?

All API calls made to Amazon SNS are validated for the user’s AWS Id and the signature. In addition, we recommend that users secure their data over the wire by connecting to our secure SSL end-points.

Q: Who can create a topic?

Topics can only be created by users with valid AWS IDs who have signed up for Amazon SNS. The easiest way to create a topic is to use the AWS Management Console. It can also be created through the CreateTopic API. The AWS Management Console is available at: <http://aws.amazon.com/console>

Q: Can multiple users publish to a single topic?

A topic owner can set explicit permissions to allow more than one user (with a valid AWS ID) to publish to a topic. By default, only topic owners have permissions to publish to a topic.

Q: How can the owner grant/revoke publish or subscribe permissions on a topic?

The AddPermission and RemovePermission APIs provide a simple interface for developers to add and remove permissions for a topic. However, for conditional access and more advanced use cases, users should use access control policies to manage permissions. The easiest way to manage permissions is to use the AWS Management Console. The AWS Management Console is available at: <http://aws.amazon.com/console>

Q: How does a topic owner give access to subscribers? Do subscribers have to have valid AWS IDs?

Amazon SNS makes it easy for users with and without AWS IDs to receive notifications. The owner of the topic can grant/restrict access to subscribers by setting appropriate permissions for the topic using Access Control policies. Users can receive notifications from Amazon SNS in two ways:

* Users with AWS IDs: Subscribers with valid AWS IDs (please refer to this link for details on obtaining AWS IDs) can subscribe to any topic directly – as long as the topic owner has granted them permissions to do so. The AWS IDs will be validated as part of the subscription registration.
* Other users: Topic owners can subscribe and register end-points on behalf of users without AWS IDs.

In both cases, the owner of the subscription endpoint needs to explicitly opt-in and confirm the subscription by replying to confirmation message sent by Amazon SNS.

Q: How will Amazon SNS authenticate API calls?

All API calls made to Amazon SNS will validate authenticity by requiring that requests be signed with the secret key of the AWS ID account and verifying the signature included in the requests.

Q: How does Amazon SNS validate a subscription request to ensure that notifications will not be sent to users as spam?

As part of the subscription registration, Amazon SNS will ensure that notifications are only sent to valid, registered subscribers/end-points. To prevent spam and ensure that a subscriber end-point is really interested in receiving notifications from a particular topic, Amazon SNS requires an explicit opt-in from subscribers using a 2-part handshake:

i. When a user first calls the Subscribe API and subscribes an end-point, Amazon SNS will send a confirmation message to the specified end-point.

ii. On receiving the confirmation message at the end-point, the subscriber should confirm the subscription request by sending a valid response. Only then will Amazon SNS consider the subscription request to be valid. If there is no response to the challenge, Amazon SNS will not send any notifications to that end-point. The exact mechanism of confirming the subscription varies by the transport protocol selected:

* For HTTP/HTTPS notifications, Amazon SNS will first POST the confirmation message (containing a token) to the specified URL. The application monitoring the URL will have to call the ConfirmSubscription API with the token included token.
* For Email and Email-JSON notifications, Amazon SNS will send an email to the specified address containing an embedded link. The user will need to click on the embedded link to confirm the subscription request.
* For SQS notifications, Amazon SNS will enqueue a challenge message containing a token to the specified queue. The application monitoring the queue will have to call the ConfirmSubscription API with the token.

Note: The explicit “opt-in” steps described above are not required for the specific case where you subscribe your Amazon SQS queue to your Amazon SNS topic – and both are “owned” by the same AWS account.

Q: How long will subscription requests remain pending, while waiting to be confirmed?

Token included in the confirmation message sent to end-points on a subscription request are valid for 3 days.

Q: Who can change permissions on a topic?

Only the owner of the topic can change permissions for that topic.

Q: How can users verify that notification messages are sent from Amazon SNS?

To ensure the authenticity of the notifications, Amazon SNS will sign all notification deliveries using a cryptographically secure, asymmetric mechanism (private-public key pair based on certificates). Amazon SNS will publish its certificate to a well-known location (e.g. <http://sns.us-east-1.amazonaws.com/SimpleNotificationService.pem> for the US East region) and sign messages with the private key of that certificate. Developers/applications can obtain the certificate and validate the signature in the notifications with the certificate’s public key, to ensure that the notification was indeed sent out by Amazon SNS. For further details on certificate locations, please refer to the Amazon SNS details page.

Q: Do publishers have to sign messages as well?

Amazon SNS requires publishers with AWS IDs to validate their messages by signing messages with their secret AWS key; the signature is then validated by Amazon SNS.

Q: Can a publisher/subscriber use SSL to secure messages?

Yes, both publishers and subscribers can use SSL to help secure the channel to send and receive messages. Publishers can connect to Amazon SNS over HTTPS and publish messages over the SSL channel. Subscribers should register an SSL-enabled end-point as part of the subscription registration, and notifications will be delivered over a SSL channel to that end-point.

Q: What permissions does a subscriber need to allow Amazon SNS to send notifications to a registered endpoint?

The owner of the end-point receiving the notifications has to grant permissions for Amazon SNS to send messages to that end-point.

Q: How can subscriptions be unsubscribed?

Subscribers can be unsubscribed either by the topic owner, the subscription owner or others – depending on the mechanism used for confirming the subscription request.

* A subscription that was confirmed with the AuthenticateOnUnsubscribe flag set to True in the call to the ConfirmSubscription API call can only be unsubscribed by a topic owner or the subscription owner.
* If the subscription was confirmed anonymously without the AuthenticateOnUnsubscribe flag set to True, then it can be anonymously unsubscribed.

In all cases except when unsubscribed by the subscription owner, a final cancellation message will be sent to the end-point, allowing the endpoint owner to easily re-subscribe to the topic (if the Unsubscribe request was unintended or in error). For further details on the ConfirmSubscription API, please refer to the Amazon SNS documentation.

## **Compliance**

Q: Is Amazon SNS HIPAA eligible?

Yes, the AWS HIPAA compliance program includes Amazon SNS as a HIPAA eligible Service. If you have an executed Business Associate Agreement (BAA) with AWS, you can now use Amazon SNS to build HIPAA-compliant applications. If you don't have a BAA or have other questions about using AWS for your HIPAA-compliant applications, [contact us](https://aws.amazon.com/health/providers-and-insurers/hipaa/inbound/) for more information.

To learn more, see the following resources:

* AWS [HIPAA Compliance](https://aws.amazon.com/compliance/hipaa-compliance/) page
* AWS [Cloud Computing in Healthcare](https://aws.amazon.com/health/) page

To see the current list of compliance programs that Amazon SNS is in scope for, see [AWS Services in Scope by Compliance Program](https://aws.amazon.com/compliance/services-in-scope/).

Q: What else is Amazon SNS compliant with?

Please see [AWS Services in Scope by Compliance Program](https://aws.amazon.com/compliance/services-in-scope/) for the latest information about SNS and other AWS services.

## **Reliability**

Q: How reliable is my data once published to Amazon SNS?

Amazon SNS stores all topic and message information within Amazon’s proven network infrastructure and datacenters. At least three copies of the data are stored across multiple availability zones, which means that no single computer or network failure renders Amazon SNS inaccessible.

Q: Will a notification contain more than one message?

No, all notification messages will contain a single published message.

Q: How many times will a subscriber receive each message?

Although most of the time each message will be delivered to your application exactly once, the distributed nature of Amazon SNS and transient network conditions could result in occasional, duplicate messages at the subscriber end. Developers should design their applications such that processing a message more than once does not create any errors or inconsistencies.

Q: Will messages be delivered to me in the exact order they were published?

The Amazon SNS service will attempt to deliver messages from the publisher in the order they were published into the topic. However, network issues could potentially result in out-of-order messages at the subscriber end.

Q: Can a message be deleted after being published?

No, once a message has been successfully published to a topic, it cannot be recalled.

Q: Does Amazon SNS guarantee that messages are delivered to the subscribed endpoint?

When a message is published to a topic, Amazon SNS will attempt to deliver notifications to all subscribers registered for that topic. Due to potential Internet issues or Email delivery restrictions, sometimes the notification may not successfully reach an HTTP or Email end-point. In the case of HTTP, an SNS Delivery Policy can be used to control the retry pattern (linear, geometric, exponential backoff), maximum and minimum retry delays, and other parameters. If it is critical that all published messages be successfully processed, developers should have notifications delivered to an SQS queue (in addition to notifications over other transports).

Q: What happens to Amazon SNS messages if the subscribing endpoint is not available?

All messages sent to SNS are processed and delivered immediately. If a message cannot be successfully delivered on the first attempt, SNS implements a 4-phase retry policy: 1) retries with no delay in between attempts, 2) retries with some minimum delay between attempts, 3) retries with some back-off model (linear or exponential), and 4) retries with some maximum delay between attempts.

The policies for each endpoint vary and are listed below.

* SQS: If a SQS queue is not available, SNS will retry 10 times immediately, then 100,000 times every 20 seconds for a total of 100,010 attempts over more than 23 days before the message is discarded from SNS.
* Lambda: If Lambda is not available, SNS will retry 2 times at 1 seconds apart, then 10 times exponentially backing off from 1 seconds to 20 minutes and finally 38 times every 20 minutes for a total 50 attempts over more than 13 hours before the message is discarded from SNS.
* HTTP/S: Delivery policies can be set for HTTP/S endpoints associated with a particular topic. Each delivery policy contains a retry policy and a throttle policy. For example, a retry policy can specify minimum/maximum delay for a retry, number of retries to be done before delay, and the type of backoff (linear, exponential, or arithmetic) to be used for retries.  After the number of retries has been exhausted, the message is deleted from SNS. Throttle policy is defined by the maximum number of delivery attempts per second per subscription.  More information can be found at [Setting Amazon SNS Delivery Retry Policies for HTTP/HTTPS Endpoints](http://docs.aws.amazon.com/sns/latest/dg/DeliveryPolicies.html).
* Email: If an email endpoint is not available, SNS will retry 1 time immediately, 1 time at 10 seconds apart, then 10 times with linear back-off from 10 seconds to 5 minutes and finally 90 times every 5 minutes for a total 102 attempts over more than 7 hours before the message is discarded from SNS.
* Mobile push: If the mobile endpoint is not available, SNS will retry 2 times immediately, 5 times at 20 seconds apart, then 31 times exponentially backing off from 20 seconds to 20 minutes and finally 12 times every 20 minutes for a total 50 attempts over more than 4 hours before the message is discarded from SNS.
* SMS: If the SMS endpoint is not available, SNS will retry 2 times at 1 seconds apart, then 10 times exponentially backing off from 1 seconds to 10 minutes and finally 38 times every 10 minutes for a total 50 attempts over more than 6 hours before the message is discarded from SNS.

## **Worldwide SMS**

Q: What features are part of the new Worldwide SMS capability?

You can use Amazon SNS to deliver SMS (text) messages to 200+ countries, and you do not require recipients to explicitly opt in as before. You must obtain prior permission from recipients to send SMS messages to their phone numbers, where required by local law and regulations. Additionally, you can now mark your SMS messages as Transactional to optimize for reliable delivery, or you can mark it as Promotional to optimize for cost savings. Furthermore, you can set account and message-level spend limits to avoid inadvertent overruns.

Q: When should I mark an SMS message as Transactional?

SMS messages that are of high priority to your business should be marked as Transactional. This ensures that messages such as those that contain one-time passwords (OTP) or PINs get delivered over routes with the highest delivery reliability. These routes tend to be more expensive than Promotional messaging routes in countries other than the US. You should never mark marketing messages as Transactional, because this violates the local regulatory policies in certain countries, and your account may be marked for abuse and suspended.

Q: When should I mark an SMS message as Promotional?

SMS messages that carry marketing messaging should be marked Promotional. Amazon SNS ensures that such messages are sent over routes that have a reasonable delivery reliability but are substantially cheaper than the most reliable routes. This also allows Amazon SNS to handle and deliver your messages in compliance with on local laws and regulation

Q: What are account-level and message-level spend limits and how do they work?

Spend limits can be specified for an AWS account and for individual messages, and the limits apply only to the cost of sending SMS messages.

The default spend limit per account (if not specified) is 1.00 USD per month. If you want to raise the limit, submit an [SNS Limit Increase case](https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-sns). For New limit value, enter your desired monthly spend limit. In the Use Case Description field, explain that you are requesting an SMS monthly spend limit increase.

Amazon SNS sends SMS messages that you publish while the total cost incurred for your SMS traffic is below your spend limit for that calendar month. Once the spend limit is exceeded, Amazon SNS stops delivering messages until you either increase the spend limit or a new calendar month begins. Similarly, you can also specify a spend limit for an individual message, and Amazon SNS will send the message only if the cost is below the limit. Amazon SNS will not send your SMS messages if the account-level spend limit is exceeded, regardless of whether the message-level spend limit is exceeded.

Q: Is two-way SMS supported?

Amazon SNS does not currently support two-way SMS capabilities, except for opt out where required by local regulations.

Q: Do I need to subscribe phone numbers to an SNS Topic before sending an SMS message to it?

You no longer need to subscribe a phone number to an Amazon SNS topic before you publish messages to it. Now, you can directly publish messages to a phone number using the Amazon SNS console or the Publish request in the Amazon SNS API.

Q: Does AWS offer short codes for purchase?

Yes. You can reserve a dedicated short code that is assigned to your account and available exclusively to you.

To reserve a short code, create a case using the [AWS Support Center.](https://console.aws.amazon.com/support/home#/) For more information, see [Reserving a Dedicated Short Code for SMS Messaging](http://docs.aws.amazon.com/sns/latest/dg/sms_shortcodes.html) in the *Amazon SNS Developer Guide*.

For pricing information, see [Worldwide SMS Pricing](https://aws.amazon.com/sns/sms-pricing/).

Q: Does AWS offer long codes for purchase?

No. AWS does not currently offer long codes for purchase.

Q: Will SMS notifications come from a specific number of short codes or long codes?

Amazon SNS uses a pool of long codes or short codes to send SMS notifications. While there is a possibility that SMS notifications come from multiple numbers, Amazon SNS ensures that the messages sent from an AWS account to a specific phone number always come from the same long code or short code. This is called "Sticky Sender ID".

You can reserve a dedicated short code to ensure that all SMS messages that you send through Amazon SNS have a persistent short code. By reserving a short code, you make it easier for your audience to recognize that your organization is the source of your messages. For more information, see [Reserving a Dedicated Short Code for SMS Messaging](http://docs.aws.amazon.com/sns/latest/dg/sms_shortcodes.html) in the *Amazon SNS Developer Guide*.

Q: Which countries does Amazon SNS support for Worldwide SMS?

Amazon SNS supports more than 200 countries, and we keep growing our reach. Please refer to the SMS Supported [Country List](http://docs.aws.amazon.com/sns/latest/dg/sms_supported-countries.html) for a comprehensive list of supported calling countries.

Q: Which AWS regions support Worldwide SMS?

1) US-East-1 (Virginia), 2) US-West-2 (Oregon), 3) EU-West-1 (Dublin), 4) Asia Pacific (Tokyo), 5) Asia Pacific (Singapore), and 6) Asia Pacific (Sydney).

Q: Do the AWS phone numbers change?

Yes. Amazon SNS uses a pool of long codes or short codes to send SMS notifications. So while there is a possibility that SMS notifications come from multiple numbers, Amazon SNS ensures that the messages sent from an AWS account to a specific phone number, always come from the same long code or short code. This is called "Sticky Sender ID".

Q: Why do some devices on the same carrier receive messages from different phone numbers?

Amazon SNS uses a pool of long codes or short codes to send SMS notifications. So while there is a possibility that SMS notifications come from multiple numbers, Amazon SNS ensures that the messages sent from an AWS account to a specific phone number always come from the same long code or short code. This is called "Sticky Sender ID".

Q: What is the phone number format for sending messages to other countries?

AWS strongly encourages [E.164 number formatting](http://en.wikipedia.org/wiki/E.164) for all phone numbers both in the ‘to’ and ‘from’ (when applicable) fields. Please refer to the [SMS Supported Country List](http://docs.aws.amazon.com/sns/latest/dg/sms_supported-countries.html) for a comprehensive list of supported countries.

Q: Does Amazon SNS determine if a phone number is a mobile, landline, or VoIP number?

No. Currently, Amazon SNS does not detect whether a phone number is mobile, landline, or VoIP.

Q: Is time-based or scheduled delivery supported for SMS messages?

No. Amazon SNS does not currently support time-based or scheduled delivery.

Q: How do I track the delivery status of my SMS messages?

By enabling the Delivery Status feature in Amazon SNS, you can get information on the following for each message: MessageID, Time Sent, Destination Phone Number, Disposition, Disposition Reason (if applicable), Price, and Dwell Time.

Q: Do you support MMS ?

No. Currently Amazon SNS does not support MMS messages.

Q: What is the cost of receiving SMS messages from Amazon SNS?

Costs for receiving SMS messages depend on the Data and Messaging of the recipient's wireless / mobile carrier plans.

Q: How do recipients opt out from receiving SMS messages from AWS?

Recipients can use their devices to opt out by replying to the message with any of the following:

* ARRET (French)
* CANCEL
* END
* OPT-OUT
* OPTOUT
* QUIT
* REMOVE
* STOP
* TD
* UNSUBSCRIBE

To opt out, the recipient must reply to the same long code or short code that Amazon SNS used to deliver the message. After opting out, the recipient will no longer receive SMS messages delivered from your AWS account unless you opt in the phone number.

Q: How do I know if a recipient device has ‘opted out’ of Global SMS?

The SNS console displays the list of opted out numbers for your account. Additionally, the Amazon SNS API provides the ListPhoneNumbersOptedOut request for listing opted out phone numbers.

Q: If a user opts out, will that number be unsubscribed automatically from the SNS Topic?

No. Opt-outs do not unsubscribe a number from an Amazon SNS topic, but rather disable the subscription. This means if you opt-in a phone number you do not need to re-subscribe the phone number to the topic.

Q: How do I confirm the end user received the SMS message?

You can use our Delivery Status feature to get information on the final disposition of your SMS message. For more information on the feature and how to use it, please refer to our [documentation](http://docs.aws.amazon.com/sns/latest/dg/welcome.html).

Q: Does Amazon SNS provide delivery receipts for SMS messages?

Our Delivery Status feature provides information based on delivery receipts received from the destination carrier. For more information on the Delivery Status feature and how to use it, please refer to our [documentation](http://docs.aws.amazon.com/sns/latest/dg/welcome.html).

Q: Does SMS support delivery to VoIP services like Google Voice or Hangouts?

Yes. Amazon SNS does support delivery to VoIP services that can receive SMS messages.

### **SMS Pricing**

Q: Where can I find the current SMS pricing per country?

Our pricing is based on destination country and carrier, and it is provided [here.](http://aws.amazon.com/sns/sms-pricing)

Q: Why does pricing for SMS keeps changing for the same destination country and carrier?

Pricing in the SMS industry is not static and the cost of sending to different countries and carriers in those countries tend to vary over time. Amazon SNS has adopted a transparent approach and has exposed those price variations to customers, so that you get the maximum cost benefits.

Q: Will I be charged for failed deliveries or messages rejected by carriers?

You may be charged for failed deliveries if the destination carrier reports back that you attempted to send messages to an invalid phone number. Phone numbers can be invalid for a number of reasons, such as the phone number doesn't exist, the phone holder's account does not have sufficient credits, or if the destination number is a landline number.

Q: Is there a 'free tier' allowance for SMS messages?

There is a monthly free tier allowance for SMS messages. The first 100 SMS messages sent to US phone numbers each month are free. Additional SMS messages to the US or any messages sent to non-US phone numbers are charged based on current pricing provided [here](https://aws.amazon.com/sns/sms-pricing).

## **Limits and Restrictions**

Q: Are there limits to the number of topics or number of subscribers per topic?

By default, SNS offers 10 million subscriptions per topic, and 100,000 topics per account.  To request a higher limit, please [contact Support](http://aws.amazon.com/support).

Q: How much and what kind of data can go in a message?

With the exception of SMS messages, Amazon SNS messages can contain up to 256 KB of text data, including XML, JSON and unformatted text.

The following Unicode characters are accepted:

#x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFD] | [#x10000 to #x10FFFF]

(according to <http://www.w3.org/TR/REC-xml/#charsets>).

Each 64KB chunk of published data is billed as 1 request. For example, a single API call with a 256KB payload will be billed as four requests.

SMS messages

Each SMS message can contain up to 140 bytes, and the character limit depends on the encoding scheme. For example, an SMS message can contain:

* 160 GSM characters
* 140 ASCII characters
* 70 UCS-2 characters

If you publish a message that exceeds the size limit, Amazon SNS sends it as multiple messages, each fitting within the size limit. Messages are not cut off in the middle of a word but on whole-word boundaries. The total size limit for a single SMS publish action is 1600 bytes.

Q: How many message filters can be applied to a topic?

By default, 100 filter policies per account per region can be applied to a topic. Please contact us if more is required.

Q: Are there TCP ports that should be used for cross-region communication between SNS and EC2?

Yes, cross-region communication between SNS and EC2 on ports other than 80/443/4080/8443 is not guaranteed to work and should be avoided.

## **Raw Message Delivery**

Q: What is raw message delivery?

You can opt-in to get your messages delivered in raw form, i.e. exactly as you published them. By default, messages are delivered encoded in JSON that provides metadata about the message and topic. Raw message delivery can be enabled by setting the “RawMessageDelivery” property on the subscriptions. This property can be set by using the AWS Management Console, or by using the API SetSubscriptionAttributes.

Q: What is the default behavior if the raw message delivery property on the subscription is not set?

By default, if this property is not set, messages will be delivered in JSON format, which is the current behavior. This ensures existing applications will continue to operate as expected.

Q: Which types of endpoints support raw message delivery?

Raw message delivery support is supported with SQS and HTTP(S) endpoints. Deliveries to Lambda, email, and SMS endpoints will behave the same independent of the “RawMessageDelivery” property.

Q: How will raw messages be delivered to HTTP endpoints?

When raw-formatted messages are delivered to HTTP/s endpoints, the message body will be included in the body of the HTTP POST.

## **Mobile Push Notifications**

Q: What is SNS Mobile Push?

SNS Mobile Push lets you use Simple Notification Service (SNS) to deliver push notifications to Apple, Google, Fire OS, and Windows devices, as well as Android devices in China with [Baidu Cloud Push](http://developer.baidu.com/cloud/push). With push notifications, an installed mobile application can notify its users immediately by popping a notification about an event, without opening the application. For example, if you install a sports app and enable push notifications, the app can send you the latest score of your favorite team even if the app isn’t running. The notification appears on your device, and when you acknowledge it, the app launches to display more information. Users’ experiences are similar to receiving an SMS, but with enhanced functionality and at a fraction of the cost.

Q: How do I get started sending push notifications?

Push notifications can only be sent to devices that have your app installed, and whose users have opted in to receive them. SNS Mobile Push does not require explicit opt-in for sending push notifications, but iOS, Android and Kindle Fire operating systems do require it. In order to send push notifications with SNS, you must also register your app and each installed device with SNS. For more information, see [Using Amazon SNS Mobile Push Notifications](http://docs.aws.amazon.com/sns/latest/dg/SNSMobilePush.html).

Q: Which push notifications platforms are supported?

Currently, the following push notifications platforms are supported:

* Amazon Device Messaging (ADM)
* Apple Push Notification Service (APNS)
* Google Cloud Messaging (GCM)
* Windows Push Notification Service (WNS) for Windows 8+ and Windows Phone 8.1+
* Microsoft Push Notification Service (MPNS) for Windows Phone 7+
* Baidu Cloud Push for Android devices in China

Q: How many push notifications can I send with the SNS Free Tier?

The SNS free tier includes 1 million publishes, plus 1 million mobile push deliveries. So you can send 1 million free push notifications every month. Notifications to all mobile push endpoints are all counted together toward your 1 million free mobile push deliveries.

Q: Does enabling push notifications require any special confirmations with SNS Mobile Push?

No, they do not. End-users opt-in to receive push notifications when they first run an app, whether or not SNS delivers the push notifications.

Q: Do I have to modify my client app to use SNS Mobile Push?

SNS does not require you to modify your client app.  Baidu Cloud Push requires Baidu-specific components to be added to your client code in order to work properly, whether or not you choose to use SNS.

Q: How do SNS topics work with Mobile Push?

SNS topics can have subscribers from any supported push notifications platform, as well as any other endpoint type such as SMS or email. When you publish a notification to a topic, SNS will send identical copies of that message to each endpoint subscribed to the topic. If you use platform-specific payloads to define the exact payload sent to each push platform, the publish will fail if it exceeds the maximum payload size imposed by the relevant push notifications platform.

Q: What payload size is supported for various target platforms?

SNS will support maximum payload size that is supported by the underlying native platform. Customers can use a JSON object to send platform specific messages. See [Using SNS Mobile Push API](http://docs.aws.amazon.com/sns/latest/dg/mobile-push-api.html) for additional details.

Q: How do platform-specific payloads work?

When you publish to a topic and want to have customized messages sent to endpoints for the different push notification platforms then you need to select “Use different message body for different protocols” option on the Publish dialog box and then update the messages. You can use platform-specific payloads to specify the exact API string that is relayed to each push notifications service. For example, you can use platform-specific payloads to manipulate the badge count of your iOS application via APNS. For more information, see [Using Amazon SNS Mobile Push Notifications](http://docs.aws.amazon.com/sns/latest/dg/SNSMobilePush.html).

Q: Can one token subscribe to multiple topics?

Yes. Each token can be subscribed to an unlimited number of SNS topics.

Q: What is direct addressing? How does it work?

Direct addressing allows you to deliver notifications directly to a single endpoint, rather than sending identical messages to all subscribers of a topic. This is useful if you want to deliver precisely targeted messages to each recipient. When you register device tokens with SNS, SNS creates an endpoint that corresponds to the token. You can publish to the token endpoint just as you would publish to a topic. You can direct publish either the text of your notification, or a platform-specific payload that takes advantage of platform-specific features such as updating the badge count of your app. Direct addressing is currently only available for push notifications endpoints.

Q: Does SNS support direct addressing for SMS or Email?

At this time, direct addressing is only supported for mobile push endpoints (APNS, GCM, ADM, WNS, MPNS, Baidu) and SMS. Email messaging requires the use of topics.

Q: How does SNS Mobile Push handle token feedback from notification services?

Push notification services such as APNS and GCM provide feedback on tokens which may have expired or may have been replaced by new tokens. If either APNS or GCM reports that a particular token has either expired or is invalid, SNS automatically "disables" the application endpoint associated with the token, and notifies you of this change via an event.GCM specifically, at times not only indicates that a token is invalid, but also provides the new token associated with the application endpoint in its response to SNS. When this happens, SNS automatically updates the associated endpoint with the new token value, leaving the endpoint enabled, and then notifies you of this change via an event.

Q: Can I migrate existing apps to SNS Mobile Push?

Yes. You can perform a bulk upload of existing device tokens to Amazon SNS, either via the console interface or API. You would also register your app with SNS by uploading your credentials for the relevant push notifications services, and configure your proxy or app to register future new tokens with SNS.

Q: Can I monitor my push notifications through Amazon CloudWatch?

Yes. SNS publishes Cloudwatch metrics for number of messages published, number of successful notifications, number of failed notifications and size of data published. Metrics are available on per application basis. You can access Cloudwatch metrics via AWS Management Console or CloudWatch APIs.

Q: What types of Windows Push Notifications does Amazon SNS support?

SNS supports all types of push notifications types offered by Microsoft WNS and MPNS, including toast, tile, badge and raw notifications.  Use the TYPE message attribute to specify which notification type you wish to use.  When you use default payloads to send the same message to all mobile platforms, SNS will select toast notifications by default for Windows platforms.  It is required to specify a notification type for Windows platforms when you use platform-specific payloads.

Q: Does SNS support Windows raw push notifications?

Yes.  You must encode the notification payload as text to send raw notifications via SNS.

Q: What is Baidu Cloud Push?

Baidu Cloud Push is a third-party alternative push notifications relay service for Android devices.  You can use Baidu Cloud Push to reach Android customers in China, no matter what Android app store those customers choose to use for downloading your app.  For more information about Baidu Cloud Push, visit: <http://developer.baidu.com/cloud/push>.

Q: Can I publish Baidu notifications from all public AWS regions?

Yes, SNS supports Baidu push notifications from all public AWS regions.

Q: Can I use Baidu notifications to any Android app store?

Yes, Baidu push notifications work for apps installed via any Android app store.

Q: What are message attributes?

Message attributes allow you to provide structured metadata items (such as timestamps, geospatial data, signatures, and identifiers) about the message. Message attributes are optional and separate from, but sent along with, the message body. This information can be used by the receiver of the message to help decide how to handle the message without having to first process the message body.

You can use SNS message attributes in conjunction with SQS and mobile push endpoints.  To learn more about message attributes, please see the [SNS Getting Started Guide](http://docs.aws.amazon.com/sns/latest/dg/GettingStarted.html).

Q: What message attributes are supported in SNS?

SNS supports different message attributes for each endpoint type, depending on what the endpoint types each support themselves.

* For SQS endpoints, you can specify up to 10 name-type-value triples per message. Types supported include: String, Binary and Number (including integers, floating point, and doubles).
* For mobile push endpoints, you can take advantage of specific message attributes that each mobile platform supports (such as notification type).

Q: What is Time to Live (TTL)?

Some messages that you can send with SNS are relevant or valuable only for a limited period of time. Amazon SNS now allows you to set a TTL (Time to Live) value for each message. When the TTL expires for a given message that was not delivered and read by an end user, the message is deleted. TTL is specified in seconds and is relative to the time Publish call is made.

Q: How do I specify a TTL for my messages?

You can specify a TTL using the console or via API.  TTL can be specified at publish time for a message, using the message attribute below. There is a different attribute for each platform. An attribute specified for a platform is applicable only for notification deliveries to that platform.

Q: What is the default TTL?

SNS uses a default Time to Live (TTL) of 4 weeks for all mobile platforms.

Q: Do TTL message attributes override TTLs specified in a message payload?

Yes.  Google GCM and Amazon ADM allow you to specify a TTL within the message payload.  If you specify TTL within the message payload and also within a message attribute, SNS will follow the message attribute.

Q: What happens if I specify TTL=0?

Some platforms treat TTL = 0 as a special case and attempt to deliver the message immediately, else let it expire.  If you specify TTL = 0, SNS will relay your message to the appropriate service with TTL = 0 in order to take advantage of this special case.

Q: What SNS endpoints support TTL?

You can use TTL with the following mobile push endpoints: APNS, APNS\_Sandbox, GCM, ADM, Baidu, and WNS.  Microsoft MPNS does not currently support TTL.  TTL is also not supported for SQS, HTTP, email or SMS endpoints.

Q: What does the Delivery Status feature of Amazon SNS do?

The Delivery Status feature lets you collect information on success rates, failure rates and dwell times of your push notifications for the supported mobile notification platforms. The currently supported platforms include Apple (APNS), Google (GCM), Windows (WNS and MPNS), Amazon (ADM), and Baidu. The status information is captured in the Amazon CloudWatch log groups created by Amazon SNS on your behalf. Additionally, you can create actionable metrics in Amazon CloudWatch and trigger alarms based on the patterns you are interested in.

Q: Is the Delivery Status feature in Amazon SNS available only for mobile push notifications? Do you plan to support this feature for other endpoint types?

Currently the Delivery Status feature is available for mobile push notifications and SMS. We will evaluate extending this to other endpoint types based on feedback from customers.

Q: How do I activate the Delivery Status feature?

You can activate the Delivery Status feature from the [Amazon SNS console](https://us-west-2.console.aws.amazon.com/sns/v2/home). From your Application, choose the Delivery Status option in the Application Actions drop-down menu. For details, please read our [documentation](http://docs.aws.amazon.com/sns/latest/dg/sns-msg-status.html).

Q: Can I activate the Delivery Status feature from the Amazon SNS APIs?

Yes, you can activate this feature from Amazon SNS APIs by adding the relevant application-level attributes. Our [documentation](http://docs.aws.amazon.com/sns/latest/dg/sns-msg-status.html) goes over the application-level attributes that you need to add and the specific API calls that need to be made to enable this feature.

Q: How much does the Delivery Status feature cost?

There is currently no additional Amazon SNS charge for using the Delivery Status feature. However, depending upon your usage, you may incur charges for using CloudWatch since this feature creates Amazon CloudWatch log groups. Read our [pricing page](https://aws.amazon.com/cloudwatch/pricing/) for more information about CloudWatch pricing and free tier.

Q: Why can you only choose a sampling percentage for successful delivery attempts and not sample failed delivery attempts?

Based on feedback we received from customers, we found that most developers are interested in knowing all the delivery attempt failures for their applications – and prefer to only store sample successful deliveries rather than logging all of them.

Q: How can I set alarms based on failure metrics or dwell time metrics?

After activating the Delivery Status feature, you need to define a Log Metrics Filter in [Amazon CloudWatch](https://aws.amazon.com/cloudwatch/) Logs for the log group that gets created by [Amazon SNS](https://aws.amazon.com/sns/) on your behalf. This metrics filter can be defined to extract information that you are interested in, such as failure rate and dwell time. Once a Metric Filter is defined, you can create it and assign it to a Metric. This metric can then be used to set alarms or send notifications based on thresholds you define. For more information, take a look at our [documentation](http://docs.aws.amazon.com/sns/latest/dg/sns-msg-status.html) or [blog](http://mobile.awsblog.com/post/TxHTXGC8711JNF/Using-the-Delivery-Status-feature-of-Amazon-SNS).

## **SNS Support for AWS Lambda**

Q: What does support for AWS Lambda endpoints in Amazon SNS mean?

You can invoke your AWS Lambda functions by publishing messages to Amazon SNS topics that have AWS Lambda functions subscribed to them. Because Amazon SNS supports message fan-out, publishing a single message can invoke different AWS Lambda functions or invoke Lambda functions in addition to delivering notifications to supported Amazon SNS destinations such as mobile push, HTTP endpoints, SQS, email and SMS.

Q: What is AWS Lambda?

AWS Lambda is a compute service that runs your code in response to events and automatically manages the compute resources for you, making it easy to build applications that respond quickly to new information. More information on AWS Lambda and how to create AWS Lambda functions can be found [here](http://aws.amazon.com/lambda).

Q: What can I do with AWS Lambda functions and Amazon SNS?

By subscribing AWS Lambda functions to Amazon SNS topics, you can perform custom message handling. You can invoke an AWS Lambda function to provide custom message delivery handling by first publishing a message to an AWS Lambda function, have your Lambda function modify a message (e.g. localize language) and then filter and route those messages to other topics and endpoints. Apps and services that already send Amazon SNS notifications, such as Amazon CloudWatch, can now immediately take advantage of AWS Lambda without having to provision or manage infrastructure for custom message handling. You can also use delivery to an AWS Lambda function as a way to publish to other AWS services such as Amazon Kinesis or Amazon S3. You can subscribe an AWS Lambda function to the Amazon SNS topic, and then have the Lambda function in turn write to another service.

Q: How do I activate AWS Lambda endpoint support in Amazon SNS?

You need to first create an AWS Lambda function via your AWS account and the [AWS Lambda console](https://console.aws.amazon.com/lambda), and then subscribe that AWS Lambda function to a topic using the [Amazon SNS console](https://us-west-2.console.aws.amazon.com/sns/v2/home) or the [Amazon SNS APIs](http://docs.aws.amazon.com/sns/latest/dg/mobile-push-api.html). Once that is complete, any messages that you publish to the Amazon SNS topics which have Lambda functions subscribed to them will be delivered to the appropriate Lambda functions in addition to any other destinations subscribed to that topic.

Q: What does delivery of a message from Amazon SNS to an AWS Lambda function do?

A message delivery from Amazon SNS to an AWS Lambda function creates an instance of the AWS Lambda function and invokes it with your message as an input. For more information on message formats, please refer to the [Amazon SNS documentation](http://aws.amazon.com/documentation/sns/) and the AWS Lambda documentation.

Q: How much does this feature cost?

Publishing a message with Amazon SNS costs $0.50 per million requests. Aside from charges incurred in using AWS services, there are no additional fees for delivering a message to an AWS Lambda function. Amazon SNS has a [Free Tier](http://aws.amazon.com/free) of 1 million requests per month. For more information, please refer to [Amazon SNS pricing](http://aws.amazon.com/sns/pricing/). AWS Lambda function costs are based on the number of requests for your functions and the time your code executes. The AWS Lambda Free-Tier includes 1M requests per month and 400,000 GB-seconds of compute time per month. For more information, please refer to [AWS Lambda pricing](http://aws.amazon.com/lambda/pricing/).

Q: Can I subscribe AWS Lambda functions created by someone else to Amazon SNS topics that I own?

We currently do not allow an AWS account owner to subscribe an AWS Lambda function that belongs to another account. You can subscribe your own AWS Lambda functions to your own Amazon SNS topics or subscribe your AWS Lambda functions to an Amazon SNS topic that was created by another account so long as the topic policy for that SNS topic allows it.

Q: Is there a limit to the number of AWS Lambda functions that I can subscribe to an Amazon SNS topic?

Amazon SNS treats AWS Lambda functions like any other destination. By default, SNS offers 10 million subscriptions per topic. To request a higher limit, please [contact us.](https://aws.amazon.com/support)

Q: What data can I pass to my AWS Lambda function?

When an AWS Lambda function is invoked as a result of an Amazon SNS message delivery, the AWS Lambda function receives data such as the Message ID, the topic ARN, the message payload and message attributes via an SNS Event. For more information on the event structure passed to the AWS Lambda function please read our [blog](http://mobile.awsblog.com/post/Tx1VE917Z8J4UDY/Invoking-AWS-Lambda-functions-via-Amazon-SNS).

Q: Can I track delivery status for message delivery attempts to AWS Lambda functions?

To track the success or failure status of message deliveries, you need to activate the Delivery Status feature of Amazon SNS. For more information about how to activate this feature please read our blog.

Q: What regions is AWS Lambda available in?

See [AWS Regions and Endpoints](http://docs.aws.amazon.com/general/latest/gr/rande.html#lambda_region) for a complete list.

Q: Do my AWS Lambda functions need to be in the same region as my Amazon SNS usage?

You can subscribe your AWS Lambda functions to an Amazon SNS topic in any region.

Q: Are there any data transfer costs for invoking AWS Lambda functions?

Data transfer costs are applicable to message deliveries to AWS Lambda functions. Please refer to our [pricing](http://aws.amazon.com/sns/pricing/) for more information.

Q: Are there any limits to the concurrency of AWS Lambda functions?

AWS Lambda currently supports 100 concurrent requests per AWS account. If your Amazon SNS message deliveries to AWS Lambda contribute to crossing these concurrency limits, your Amazon SNS message deliveries will be throttled. If AWS Lambda throttles an Amazon SNS message, Amazon SNS will retry the delivery attempts. For more information about AWS Lambda concurrency limits, please refer to AWS [Lambda documentation](http://docs.aws.amazon.com/lambda/latest/dg/welcome.html).

Q: Can Amazon SNS use the same AWS Lambda functions that I use with other services (e.g. Amazon S3)?

You can use the same AWS Lambda functions that you use with other services as long as the same function can parse the event formats from Amazon SNS in addition to the event format of the other services. For the SNS event format please read our [blog](http://mobile.awsblog.com/post/Tx1VE917Z8J4UDY/Invoking-AWS-Lambda-functions-via-Amazon-SNS).

## **[VoIP iOS and Mac OS Notifications](https://aws.amazon.com/sns/faqs/" \l "VoIP-iOS-and-Mac-OS-Notifications)**

Q: What are VoIP Push Notifications for iOS?  
In iOS 8 and later, voice-over-IP (VoIP) apps can register for VoIP remote notifications such that iOS can launch or wake the app, as appropriate, when an incoming VoIP call arrives for the user. The procedure to register for VoIP notifications is similar to registering for regular push notifications on iOS. For more information, please refer to our [documentation](http://docs.aws.amazon.com/sns/latest/dg/mobile-push-apns.html).

Q: Can I use VoIP Push Notifications and other Push Notifications in the same iOS app?  
Yes, you can have an iOS application that is registered to receive both types of push notifications. However, you will need to obtain the VoIP push notification certificate from Apple in addition to the regular push notification certificate and create a new Platform Application in Amazon SNS and choose Apple VoIP Push as the platform type. For more information, please refer to our [documentation](http://docs.aws.amazon.com/sns/latest/dg/mobile-push-apns.html).

Q: What are Mac OS push notifications?  
You can now send push notifications to Mac OS desktops that run Mac OS X Lion (10.7) or later using Amazon SNS. For more information, please refer to our [documentation](http://docs.aws.amazon.com/sns/latest/dg/mobile-push-apns.html).