



User Guide

AWS End User Messaging Social



AWS End User Messaging Social: User Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is AWS End User Messaging Social?	1
Are you a first-time AWS End User Messaging Social user?	1
Features of AWS End User Messaging Social	2
Related services	2
Accessing AWS End User Messaging Social	2
Regional availability	3
Setting up AWS End User Messaging Social	12
Sign up for an AWS account	12
Next steps	12
Getting started	13
Signing up for WhatsApp	13
Prerequisites	13
Sign up through the console	15
Next steps	19
WhatsApp Business Account (WABA)	20
View a WABA	21
Add a WABA	21
WhatsApp business account types	22
Additional resources	22
Phone numbers	23
Phone number considerations	23
Add a phone number	24
Prerequisites	24
Add a phone number to a WABA	24
View a phone number's status	26
View a phone number's ID	27
Increase messaging conversation limits	27
Increase message throughput	28
Understanding phone number quality rating	29
View a phone number quality rating	29
Message templates	31
Using message templates in the AWS Console	32
Manage templates in the AWS Console	32
Creating a message template	32

Viewing template status	33
Editing a message template	33
Deleting a message template	33
Template quality ratings	34
Template variables	34
Template categories	34
Create templates with <code>CreateWhatsAppMessageTemplate</code>	35
Message template components	35
Create a basic English utility template	35
Create a basic English utility template with button	36
Create a complex English utility message template with a header, body, and button	37
Create a basic marketing message template	38
Create a complex marketing message template	39
Next steps	40
Template pacing	40
Get feedback on a templates lowered status	41
Template status and quality rating	41
Reasons why a template is rejected	43
WhatsApp Flows	44
Key concepts	44
Flow status	45
Creating and editing Flows	46
Creating a Flow (console)	46
Creating a Flow (API)	47
Updating a Flow	48
Listing and viewing Flows	49
Deleting a Flow	50
Publishing and deprecating Flows	50
Publishing a Flow	50
Deprecating a Flow	51
Previewing a Flow	51
Sending Flows	52
Example: Sending a Flow via template message	52
Setting up a template for Flows	54
Receiving Flow responses	54
Flow response messages	54

Flow status change notifications	55
Flow version expiry warnings	56
Limitations	56
Message and event destinations	58
Add an event destination	58
Prerequisites	58
Add a message and event destination	59
Encrypted Amazon SNS topic policies	60
IAM policies for Amazon SNS topics	61
IAM policies for Connect Customer	61
Next steps	63
Message and event format	63
AWS End User Messaging Social event header	63
Example WhatsApp JSON for a message	64
Example WhatsApp JSON for a media message	66
Message status	67
Message statuses	67
Additional resources	68
Uploading media files	69
Supported media file types	71
Media file types	71
Message types	73
Additional resources	73
Sending messages	74
Send a template message	75
Sending a media message	76
Responding to a received message	79
Change a message's status to read	79
Respond with a reaction	80
Download a media file to Amazon S3 from WhatsApp	80
Example of responding to a message	81
Prerequisites	81
Responding	81
Additional resources	84
Understanding your bill	85
Charged per message	85

Example billing SKU names	93
When does the Authentication-International FeeType apply	93
Example 1: Sending a Marketing template message	94
Example 2: Opening a service conversation	94
Volume-tiered pricing	95
Billing ISO codes	96
Charged per conversation (Deprecated)	109
Monitoring	129
Monitoring with CloudWatch	129
CloudTrail logs	131
AWS End User Messaging Social data events in CloudTrail	132
AWS End User Messaging Social management events in CloudTrail	133
AWS End User Messaging Social event examples	134
Monitoring with EventBridge	135
Event for a delivered WhatsApp message	136
Event for a read WhatsApp message	136
Event for a sent WhatsApp message	137
Event for a failed WhatsApp message	137
Best practices	139
Up-to-date business profile	139
Obtain permission	139
Prohibited message content	140
Audit your customer lists	142
Adjust your sending based on engagement	142
Send at appropriate times	142
Security	143
Data protection	144
Data encryption	145
Encryption in transit	145
Key management	145
Inter-network traffic privacy	146
Identity and access management	146
Audience	147
Authenticating with identities	147
Managing access using policies	148
How AWS End User Messaging Social works with IAM	150

Identity-based policy examples	155
AWS managed policies	158
Troubleshooting	159
Compliance validation	161
Resilience	162
Infrastructure Security	162
Cross-service confused deputy prevention	162
Security best practices	163
Using service-linked roles	164
Service-linked role permissions for AWS End User Messaging Social	164
Creating a service-linked role for AWS End User Messaging Social	165
Editing a service-linked role for AWS End User Messaging Social	165
Deleting a service-linked role for AWS End User Messaging Social	165
Supported Regions for AWS End User Messaging Social service-linked roles	166
AWS PrivateLink	167
Considerations	167
Create an interface endpoint	167
Create an endpoint policy	168
Quotas	170
Document history	171

What is AWS End User Messaging Social?

AWS End User Messaging Social, also referred to as Social messaging, is a messaging service that allows developers to integrate WhatsApp into their applications. It provides access to WhatsApp's messaging capabilities, enabling the creation of branded, interactive content with images, videos, and buttons. By using this service, you can add WhatsApp messaging functionality to your applications alongside existing channels like SMS and push notifications. This allows you to engage with customers through their preferred communication channel.

To get started, either create a new WhatsApp Business Account (WABA) using the self-guided onboarding process in the AWS End User Messaging Social console, or link an existing WABA to the service.

Note

AWS End User Messaging Social now supports Meta's Marketing Messages (MM) API to deliver marketing messages using the same AWS API endpoint. To send messages through MM API, you must accept Meta's MM API terms and conditions. You can proceed by clicking the MM API banner on the WhatsApp Business Account (WABA) details page in the AWS End User Messaging Social console, and accepting the terms in Meta's console. According to Meta, MM API has up to 9% higher deliverability for marketing messages, recipient optimization, and conversion tracking. For more information, see [Meta's Marketing Messages API documentation](#).

Topics

- [Are you a first-time AWS End User Messaging Social user?](#)
- [Features of AWS End User Messaging Social](#)
- [Related services](#)
- [Accessing AWS End User Messaging Social](#)
- [Regional availability](#)

Are you a first-time AWS End User Messaging Social user?

If you are a first-time user of AWS End User Messaging Social, we recommend that you begin by reading the following sections:

- [Setting up AWS End User Messaging Social](#)
- [Getting started with AWS End User Messaging Social](#)
- [Best practices for AWS End User Messaging Social](#)

Features of AWS End User Messaging Social

AWS End User Messaging Social provides the following features and capabilities:

- Design consistent messages and reuse content more effectively by [creating and using message templates](#). A message template contains content and settings that you want to reuse in messages that you send.
- Access to rich messaging capabilities for a more engaging experience. Beyond text and media, you can send locations and interactive messages.
- Receive incoming text and media messages from your customers.
- Create interactive, multi-screen experiences using [WhatsApp Flows](#). Collect data through forms, surveys, appointment booking, and other structured interactions without users leaving WhatsApp.
- Build trust with your customers by verifying your business identity through Meta.

Related services

AWS offers other messaging services that can be used together in a multi-channel workflow:

- Use [AWS End User Messaging SMS](#) to send SMS messages
- Use [AWS End User Messaging Push](#) to send push notifications
- Use [Amazon SES](#) to send email

Accessing AWS End User Messaging Social

You can access AWS End User Messaging Social using the following:

AWS End User Messaging Social console

The web interface where you [create](#) and manage resources.

AWS Command Line Interface

Interact with AWS services using commands in your command line shell. The AWS Command Line Interface is supported on Windows, macOS, and Linux. For more information about the AWS CLI, see [AWS Command Line Interface User Guide](#). You can find the AWS End User Messaging Social commands in the [AWS CLI Command Reference](#).

AWS SDKs

If you prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS, use the libraries, sample code, tutorials, and other resources provided by AWS. These libraries provide basic functions that automate tasks, such as cryptographically signing your requests, retrying requests, and handling error responses. These functions make it more efficient for you to get started. For more information, see [Tools to Build on AWS](#).

Regional availability

AWS End User Messaging Social is available in several AWS Regions in North America, Europe, Asia, and Oceania. In each Region, AWS maintains multiple Availability Zones. These Availability Zones are physically isolated from each other, but are united by private, low-latency, high-throughput, and highly redundant network connections. These Availability Zones are used to provide high levels of availability and redundancy, while also minimizing latency.

To learn more about AWS Regions, see [Specify which AWS Regions your account can use](#) in the *Amazon Web Services General Reference*. For a list of all the Regions where AWS End User Messaging Social is currently available and the endpoint for each Region, see [Endpoints and quotas](#) for AWS End User Messaging Social API and [AWS service endpoints](#) in the *Amazon Web Services General Reference*, or the following table. To learn more about the number of Availability Zones that are available in each Region, see [AWS global infrastructure](#).

Region availability

Region name	Region	Endpoint	WhatsApp API version
US East (N. Virginia)	us-east-1	social-messaging.us-east-1.amazonaws.com	Version 20 and later

Region name	Region	Endpoint	WhatsApp API version
		social-messaging.us-east-1.api.aws social-messaging-fips.us-east-1.amazonaws.com social-messaging-fips.us-east-1.api.aws	
US East (Ohio)	us-east-2	social-messaging.us-east-2.amazonaws.com social-messaging.us-east-2.api.aws social-messaging-fips.us-east-2.amazonaws.com social-messaging-fips.us-east-2.api.aws	Version 20 and later
US West (Oregon)	us-west-2	social-messaging.us-west-2.amazonaws.com social-messaging.us-west-2.api.aws social-messaging-fips.us-west-2.amazonaws.com social-messaging-fips.us-west-2.api.aws	Version 20 and later

Region name	Region	Endpoint	WhatsApp API version
Canada (Central)	ca-central-1	social-messaging.ca-central-1.amazonaws.com social-messaging.ca-central-1.api.aws social-messaging-fips.ca-central-1.amazonaws.com social-messaging-fips.ca-central-1.api.aws	Version 20 and later
Canada West (Calgary)	ca-west-1	social-messaging.ca-west-1.amazonaws.com social-messaging.ca-west-1.api.aws social-messaging-fips.ca-west-1.amazonaws.com social-messaging-fips.ca-west-1.api.aws	Version 20 and later
Africa (Cape Town)	af-south-1	social-messaging.af-south-1.amazonaws.com social-messaging.af-south-1.api.aws	Version 20 and later

Region name	Region	Endpoint	WhatsApp API version
Asia Pacific (Tokyo)	ap-northeast-1	social-messaging.ap-northeast-1.amazonaws.com social-messaging.ap-northeast-1.api.aws	Version 20 and later
Asia Pacific (Seoul)	ap-northeast-2	social-messaging.ap-northeast-2.amazonaws.com social-messaging.ap-northeast-2.api.aws	Version 20 and later
Asia Pacific (Mumbai)	ap-south-1	social-messaging.ap-south-1.amazonaws.com social-messaging.ap-south-1.api.aws	Version 20 and later
Asia Pacific (Hyderabad)	ap-south-2	social-messaging.ap-south-2.amazonaws.com social-messaging.ap-south-2.api.aws	Version 20 and later
Asia Pacific (Singapore)	ap-southeast-1	social-messaging.ap-southeast-1.amazonaws.com social-messaging.ap-southeast-1.api.aws	Version 20 and later

Region name	Region	Endpoint	WhatsApp API version
Asia Pacific (Sydney)	ap-southeast-2	social-messaging.ap-southeast-2.amazonaws.com social-messaging.ap-southeast-2.api.aws	Version 20 and later
Asia Pacific (New Zealand)	ap-southeast-6	social-messaging.ap-southeast-6.amazonaws.com social-messaging.ap-southeast-6.api.aws	Version 20 and later
Europe (Frankfurt)	eu-central-1	social-messaging.eu-central-1.amazonaws.com social-messaging.eu-central-1.api.aws	Version 20 and later
Europe (Stockholm)	eu-north-1	social-messaging.eu-north-1.amazonaws.com social-messaging.eu-north-1.api.aws	Version 20 and later
Europe (Spain)	eu-south-2	social-messaging.eu-south-2.amazonaws.com social-messaging.eu-south-2.api.aws	Version 20 and later

Region name	Region	Endpoint	WhatsApp API version
Europe (Ireland)	eu-west-1	social-messaging.eu-west-1.amazonaws.com social-messaging.eu-west-1.api.aws	Version 20 and later
Europe (London)	eu-west-2	social-messaging.eu-west-2.amazonaws.com social-messaging.eu-west-2.api.aws	Version 20 and later
Middle East (Bahrain)	me-south-1	social-messaging.me-south-1.amazonaws.com social-messaging.me-south-1.api.aws	Version 20 and later
Middle East (UAE)	me-central-1	social-messaging.me-central-1.amazonaws.com social-messaging.me-central-1.api.aws	Version 20 and later
Mexico (Central)	mx-central-1	social-messaging.mx-central-1.amazonaws.com social-messaging.mx-central-1.api.aws	Version 20 and later

Region name	Region	Endpoint	WhatsApp API version
South America (São Paulo)	sa-east-1	social-messaging.sa-east-1.amazonaws.com social-messaging.sa-east-1.api.aws	Version 20 and later
Asia Pacific (Hong Kong)	ap-east-1	social-messaging.ap-east-1.amazonaws.com social-messaging.ap-east-1.api.aws	Version 20 and later
Asia Pacific (Taipei)	ap-east-2	social-messaging.ap-east-2.amazonaws.com social-messaging.ap-east-2.api.aws	Version 20 and later
Asia Pacific (Osaka)	ap-northeast-3	social-messaging.ap-northeast-3.amazonaws.com social-messaging.ap-northeast-3.api.aws	Version 20 and later
Asia Pacific (Jakarta)	ap-southeast-3	social-messaging.ap-southeast-3.amazonaws.com social-messaging.ap-southeast-3.api.aws	Version 20 and later

Region name	Region	Endpoint	WhatsApp API version
Asia Pacific (Melbourne)	ap-southeast-4	social-messaging.ap-southeast-4.amazonaws.com social-messaging.ap-southeast-4.api.aws	Version 20 and later
Asia Pacific (Malaysia)	ap-southeast-5	social-messaging.ap-southeast-5.amazonaws.com social-messaging.ap-southeast-5.api.aws	Version 20 and later
Asia Pacific (Thailand)	ap-southeast-7	social-messaging.ap-southeast-7.amazonaws.com social-messaging.ap-southeast-7.api.aws	Version 20 and later
Europe (Zurich)	eu-central-2	social-messaging.eu-central-2.amazonaws.com social-messaging.eu-central-2.api.aws	Version 20 and later
Europe (Milan)	eu-south-1	social-messaging.eu-south-1.amazonaws.com social-messaging.eu-south-1.api.aws	Version 20 and later

Region name	Region	Endpoint	WhatsApp API version
Europe (Paris)	eu-west-3	social-messaging.eu-west-3.amazonaws.com social-messaging.eu-west-3.api.aws	Version 20 and later
Israel (Tel Aviv)	il-central-1	social-messaging.il-central-1.amazonaws.com social-messaging.il-central-1.api.aws	Version 20 and later
US West (N. California)	us-west-1	social-messaging.us-west-1.amazonaws.com social-messaging.us-west-1.api.aws social-messaging-fips.us-west-1.amazonaws.com social-messaging-fips.us-west-1.api.aws	Version 20 and later

Setting up AWS End User Messaging Social

Before you can use AWS End User Messaging Social for the first time, you must complete the following steps.

Topics

- [Sign up for an AWS account](#)
- [Next steps](#)

Sign up for an AWS account

To get started with AWS, you need an AWS account. For information about creating an AWS account, see [Getting started with an AWS account](#) in the *AWS Account Management Reference Guide*.

Next steps

Now that you're prepared to work with AWS End User Messaging Social, see [Getting started with AWS End User Messaging Social](#) for creating your WhatsApp Business Account (WABA) or migrating your existing WABA.

Getting started with AWS End User Messaging Social

These topics guide you through the steps to link or migrate your WhatsApp Business Account (WABA) to AWS End User Messaging Social.

Topics

- [Signing up for WhatsApp](#)

Signing up for WhatsApp

A WhatsApp Business Account (WABA) allows your business to use the WhatsApp Business Platform to send messages directly to your customers. All of your WABAs are part of your Meta business portfolio. A WABA contains your customer facing assets, like phone number, templates, and WhatsApp Business Profile. A WhatsApp Business Profile contains your business's contact information that users see. For more information on WABAs, see [WhatsApp Business Account \(WABA\) in AWS End User Messaging Social](#).

Follow the steps in this section to get started with AWS End User Messaging Social. Use the embedded sign-up process to either create a new WhatsApp Business Account (WABA) or migrate an existing WABA to AWS End User Messaging Social.

Prerequisites

Important

Working with Meta/WhatsApp

- Your use of the WhatsApp Business Solution is subject to the terms and conditions of the [WhatsApp Business Terms of Service](#), the [WhatsApp Business Solution Terms](#), the [WhatsApp Business Messaging Policy](#), the [WhatsApp Messaging Guidelines](#), and all other terms, policies, or guidelines incorporated therein by reference (as each may be updated from time to time).
- Meta or WhatsApp may at any time prohibit your use of the WhatsApp Business Solution.
- You must create a WhatsApp Business Account ("WABA") with Meta and WhatsApp.
- You must create a Business Manager account with Meta and link it to your WABA.

- You must provide control of your WABA to us. At your request, we will transfer control of your WABA back to you in a reasonable and timely manner using the methods Meta makes available to us.
- In connection with your use of the WhatsApp Business Solution, you will not submit any content, information, or data that is subject to safeguarding and/or limitations on distribution pursuant to applicable laws and/or regulation.
- WhatsApp's pricing for use of the WhatsApp Business Solution can be found at [Conversation-Based Pricing](#).

- To create a WhatsApp Business Account (WABA), your business needs a [Meta Business Account](#). Check if your company already has a Meta Business Account. If you don't have a Meta Business Account, you can create one during the sign-up process.
- To use a phone number that's already in use with the WhatsApp Messenger application or WhatsApp Business application, you must delete it first.
- A phone number that can receive either an SMS or a voice One-Time Passcode (OTP). The phone number used for sign-up becomes associated with your WhatsApp account and the phone number is used when you send messages. The phone number can still be used for SMS, MMS, and voice messaging.
- If you are importing an existing WABA, you need the PINs for all the phone numbers associated with the imported WABA. To reset a lost or forgotten PIN, follow the directions in [Updating PIN](#) in the *WhatsApp Business Platform Cloud API Reference*.

The following prerequisites must be met to use either an Amazon SNS topic or Connect Customer instance as a message and event destination.

Amazon SNS topic

- An Amazon SNS topic has been [created](#) and [permissions](#) have been added.

Note

Amazon SNS FIFO topics are not supported.

- **(Optional)** To use an Amazon SNS topic that is encrypted using AWS KMS keys you have to grant AWS End User Messaging Social permissions to the [existing key policy](#).

Connect Customer instance

- An Connect Customer instances has been [created](#) and [permissions](#) have been added.

Sign up through the console

Follow these directions to create a new WhatsApp account, migrate your existing account, or add a phone number to an existing WABA. As part of the sign-up process, you give AWS End User Messaging Social access to your WABA. You also allow AWS End User Messaging Social to bill you for messages. For more information on WABAs, see [Understanding WhatsApp business account types](#).

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Business accounts**.
3. On the **Link business account** page, choose **Launch Facebook portal**. A new login window from Meta will appear.
4. In the Meta login window, enter your Facebook account credentials.

On the **WhatsApp business account** page, choose **Add WhatsApp phone number**. On the **Add WhatsApp phone number** page, choose **Launch Facebook portal**. A new login window from Meta will appear.

5. In the Meta login window, enter your Facebook account credentials.
6. As part of the sign-up process, you give AWS End User Messaging Social access to your WhatsApp Business Account (WABA). You also allow AWS End User Messaging Social to bill you for messages. Choose **Continue**.
7. For **Meta Business account**, choose an existing Meta business account or **Create a Meta Business account**.
 - a. (Optional) If you need to create a Meta Business account, follow these steps:
 - b. For **Business name**, enter the name of your business.
 - c. For **Business website or profile page**, enter either the URL for your company's website, or if your company doesn't have a website, enter the URL to your social media page.
 - d. For **Country**, choose the country your business is located in.
 - e. (Optional) Choose **Add address** and enter your business's address.

8. Choose **Next**.
9. For **Choose a WhatsApp Business Account**, choose an existing WhatsApp Business Account (WABA), or if you need to create an account, choose **Create a WhatsApp Business Account**.

For **Create or Select a WhatsApp Business Profile**, choose an existing WhatsApp business profile, or **Create a new WhatsApp Business Profile**.

10. Choose **Next**.
11. For **Create a Business Profile**, enter the following information:

- For **WhatsApp Business Account Name**, enter a name for your account. This field is not customer facing.
- For **WhatsApp Business Profile display name**, enter the name to display to your customers when they receive a message from you. We recommend that you use your company name as the display name. The name is reviewed by Meta and must comply with [WhatsApp display name rules](#). To use a brand name that is different from your company name, there must be an externally published association between your company and the brand. This association must be displayed on your website and on the brand represented by the display name's website.

Once you complete registration, Meta performs a review of your display name. Meta sends you an email telling you whether the display name has been approved or rejected. If your display name is rejected, your per day messaging limit is lowered and you could be disconnected from WhatsApp.

 **Important**

To change your display name, you have to create a ticket with Meta support.

- For **Timezone**, choose the time zone the business is located in.
- For **Category**, choose a category that best aligns with your business. Customers can view the category you as part of your contact information.
- For **Business Description**, enter a description of your company. Customers can view your business description as part of your contact information.
- For **Website**, enter your company's website. Customers can view your website as part of your contact information.
- Choose **Next**.

12. For **Add a phone number for WhatsApp**, enter a phone number to register. This phone number is displayed to your customers when you send them a message.
13. For **Choose how you would like to verify your number**, choose either **Text message** or **Phone call**.
 - Once you are ready to receive the verification code, choose **Next**.
 - Enter the verification code, and then choose **Next**.
14. Once your number has been verified, you can choose **Next** to close the window from Meta.
15. For **WhatsApp business account** expand **Tags - optional** to add tags to your WhatsApp business account.

Tags are pairs of keys and values that you can optionally apply to your AWS resources to control access or usage. Choose **Add new tag** and enter a key-value pair to attach.

16. A WABA can have one message and event destination to log events for the WABA and all resources associated to the WABA. To enable event logging in Amazon SNS, including logging of receiving a customer message, you must turn on **Message and event publishing**. For more information, see [Message and event destinations in AWS End User Messaging Social](#).

 **Important**

To be able to respond to customer messages, you must enable **Message and event publishing**.

In the **Message and event destination details** section, turn on **Event publishing**. For Amazon SNS, choose either **New Amazon SNS standard topic** and enter a name in **Topic name**, or choose **Existing Amazon SNS standard topic** and choose a topic from the **Topic arn** dropdown list.

17. Under **Phone numbers**:

For each phone number under **WhatsApp Phone numbers**:

- a. For **Phone number verification**, enter the existing PIN or enter a new PIN code. To reset a lost or forgotten PIN, follow the directions in [Updating PIN](#) in the *WhatsApp Business Platform Cloud API Reference*.
- b. For **Additional setting**:

- i. For **Data localization region - optional** choose one of Meta's regions in which to store your data at rest. For more information on Meta's data privacy policies, see [Data Privacy & Security](#) and [Cloud API Local Storage](#) in the *WhatsApp Business Platform Cloud API Reference*.
 - ii. Tags are pairs of keys and values that you can optionally apply to your AWS resources to control access or usage. Choose **Add new tag** and enter a key-value pair to attach.
18. A WABA can have one message and event destination to log events for the WABA and all resources associated to the WABA. To enable event logging, including logging of receiving a customer message, you need to turn on **Message and event publishing**. For more information, see [Message and event destinations in AWS End User Messaging Social](#).

 **Important**

You must enable **Message and event publishing** to be able to respond to customer messages.

In the **Message and event destination details** section, turn on **Event publishing**.

19. For **Destination type** choose either Amazon SNS or Connect Customer
 - a. To send your events to an Amazon SNS destination, enter an existing topic ARN in **Topic ARN**. For example IAM policies, see [IAM policies for Amazon SNS topics](#).
 - b. For Connect Customer
 - i. For **Connect instance** choose an instance from the drop down.
 - ii. For Role ARN, choose either:
 - A. **Choose existing IAM role** – Choose an existing IAM policy from the **Existing IAM roles** drop down. For example IAM policies, see [IAM policies for Connect Customer](#).
 - B. **Enter IAM role ARN** – Enter the ARN of the IAM policy into **Use existing IAM role Arn**. For example IAM policies, see [IAM policies for Connect Customer](#).
20. To complete setup, choose **Add phone number**.

Next steps

Once you've completed sign-up, you can start sending messages. When you're ready to start sending messages at scale, complete [Business Verification](#). Now that your WABA and AWS End User Messaging Social accounts are linked, see the following topics:

- Learn about [event destination](#) to log events and receive incoming messages.
- Learn how to create [message templates](#).
- Learn how to [send a text or media message](#).
- Learn how to [receive a message](#).
- Learn about [Official Business Accounts](#) to have a green check mark beside your display name and increase your message throughput.

WhatsApp Business Account (WABA) in AWS End User Messaging Social

With a WhatsApp Business Account (WABA), you can use the WhatsApp Business Platform to send messages directly to your customers. All of your WABAs are part of your [Meta Business Portfolio](#). A WABA contains your customer facing assets like phone number, templates, and business contact information. A WABA can only exist in one AWS Region. For more information on WABAs, see [WhatsApp Business Accounts](#) in the *WhatsApp Business Platform Cloud API Reference*.

Important

Working with Meta/WhatsApp

- Your use of the WhatsApp Business Solution is subject to the terms and conditions of the [WhatsApp Business Terms of Service](#), the [WhatsApp Business Solution Terms](#), the [WhatsApp Business Messaging Policy](#), the [WhatsApp Messaging Guidelines](#), and all other terms, policies, or guidelines incorporated therein by reference. These might be updated from time to time.
- Meta or WhatsApp may at any time prohibit your use of the WhatsApp Business Solution.
- You must create a WhatsApp Business Account (WABA) with Meta and WhatsApp.
- You must create a Business Manager account with Meta and link it to your WABA.
- You must grant control of your WABA to us. At your request, we will transfer control of your WABA back to you in a reasonable and timely manner using the methods Meta makes available to us.
- In connection with your use of the WhatsApp Business Solution, you will not submit any content, information, or data that is subject to safeguarding or limitations on distribution pursuant to applicable laws or regulations.
- WhatsApp's pricing for use of the WhatsApp Business Solution can be found at <https://developers.facebook.com/docs/whatsapp/pricing>.

Topics

- [View a WhatsApp Business Account \(WABA\) in AWS End User Messaging Social](#)
- [Add a WhatsApp Business Account \(WABA\) in AWS End User Messaging Social](#)

- [Understanding WhatsApp business account types](#)

View a WhatsApp Business Account (WABA) in AWS End User Messaging Social

You can view the WABA associated with your AWS account.

To view the WABA associated with your account

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. In **Business accounts**, choose a WABA.
3. On the **Phone numbers** tab, view your phone number, display name, quality rating, and the number of business initiated conversations that you have left for the day.

On the **Event destinations** tab, view your event destination. To edit your event destination, follow the directions in [Message and event destinations in AWS End User Messaging Social](#).

On the **Templates** tab, choose **Manage message templates** to edit your WhatsApp templates through Meta. Each WABA has a 250 template limit.

On the **Tags** tab, you can manage your WABA resource tags.

Add a WhatsApp Business Account (WABA) in AWS End User Messaging Social

Add a new WABA to your account if you already have a WhatsApp Business Profile. As part of creating a new WABA, you must add a [phone number](#) to the WABA.

- To add a new WABA to your account, follow the steps in [Getting started with AWS End User Messaging Social](#):
 - In step 8, choose your WhatsApp Business Profile, and then choose a **Create a new WhatsApp Business account**.

Understanding WhatsApp business account types

Your WhatsApp business account determines how you appear to your customers. When you create a WhatsApp account, your account will be a *Business Account*. WhatsApp has two types of business accounts:

- *Business Account*: WhatsApp verifies the authenticity of every account on the WhatsApp Business Platform. If a business account has completed the Business Verification process, the business name will be visible to all users. This feature helps users identify verified business accounts on WhatsApp.
- *Official Business Account*: Along with the benefits of a *business account*, an official business account has a green checkmark badge in its profile and chat thread headers.

Approval for a WhatsApp Official Business Account (OBA) requires providing evidence that the business is well known and recognized by consumers, such as articles, blog posts, or independent reviews. Approval for a WhatsApp OBA is not guaranteed, even if the business provides the required documentation. The approval process is subject to review and approval by WhatsApp. WhatsApp does not publicly disclose the specific criteria they use to evaluate and approve applications for Official Business Accounts. The businesses seeking a WhatsApp OBA must demonstrate their reputation and recognition, but final approval is at the discretion of WhatsApp.

When you create a WhatsApp account, your account will be a *Business Account*. You can provide information to your customers about your business, such as website, address, and hours. For businesses that haven't completed WhatsApp Business Verification, the display name is shown in small text next to the phone number in the contacts view, not in the chat list or individual chat. Once the Meta Business Verification is completed, the WhatsApp Sender's display name will be shown in the chat list and individual chat threads.

Additional resources

- For more information on *Business Account* and *Official Business Account*, see [Business Accounts](#) in the *WhatsApp Business Platform Cloud API Reference*.
- For more information on the Business Verification process, see [Business Verification](#) in the *WhatsApp Business Platform Cloud API Reference*.

Phone numbers in AWS End User Messaging Social

All WABAs contain one or more phone numbers used to verify your identity with WhatsApp and are used as part of your sending identity. You can have multiple phone numbers associated with a WhatsApp Business Account (WABA) and use each phone number for a different brand.

Topics

- [Phone number considerations for use with a WABA](#)
- [Add a phone number to a WhatsApp Business Account \(WABA\)](#)
- [View a phone number's status](#)
- [View a phone number's ID in AWS End User Messaging Social](#)
- [Increase messaging conversation limits in WhatsApp](#)
- [Increase message throughput in WhatsApp](#)
- [Understanding phone number quality rating in WhatsApp](#)

Phone number considerations for use with a WABA

When you link a phone number with your WhatsApp Business Account (WABA), you should consider the following:

- Phone numbers can only be linked to one WABA at a time.
- The phone number can still be used for SMS, MMS, and voice calls.
- Each phone number has a quality rating from Meta.

You can obtain an SMS-capable phone number through AWS End User Messaging SMS by doing the following:

1. Make sure that the [country or region](#) for the phone number supports two-way SMS.
2. Request the [phone number](#). Depending on the country or region, you may be required to register the phone number.
3. [Enable two-way SMS messaging](#) for the phone number. Once setup is complete, your incoming SMS messages are sent to an event destination.

Add a phone number to a WhatsApp Business Account (WABA)

You can add phone numbers to an existing WhatsApp Business Account (WABA) or create a new WABA for the phone number.

Prerequisites

Before you begin, the following prerequisites must be met:

- The phone number must be able to receive either an SMS or a voice One-Time Passcode (OTP). This is the phone number that is added to your WABA.
- The phone number must not be associated with any other WABA.

The following prerequisites must be met to use either an Amazon SNS topic or Connect Customer instance as a message and event destination.

Amazon SNS topic

- An Amazon SNS topic has been [created](#) and [permissions](#) have been added.

Note

Amazon SNS FIFO topics are not supported.

- **(Optional)** To use an Amazon SNS topic that is encrypted using AWS KMS keys you have to grant AWS End User Messaging Social permissions to the [existing key policy](#).

Connect Customer instance

- An Connect Customer instances has been [created](#) and [permissions](#) have been added.

Add a phone number to a WABA

To add a new phone number to your existing WABA

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Business Accounts**, and then **Add WhatsApp phone number**.

3. On the **Add WhatsApp phone number** page, choose **Launch Facebook portal**. A new login window from Meta will appear.
4. In the Meta login window, enter your Meta developer account credentials and choose your business portfolio.
5. Choose the WABA and WhatsApp Business Profile that you want to add the phone number to.
6. Choose **Next**.
7. For **Add a phone number for WhatsApp**, enter a phone number to register. This phone number is displayed to your customers when you send them a message.
8. For **Choose how you would like to verify your number**, choose either **Text message** or **Phone call**.
9. Once you are ready to receive the verification code, choose **Next**.
10. Enter the verification code, and then choose **Next**. Once your number has been verified, you can choose **Next** to close the window from Meta.
11. Under **WhatsApp Phone numbers**:
 - a. For **Phone number verification**, enter the existing PIN or enter a new PIN code. To reset a lost or forgotten PIN, follow the directions in [Updating PIN](#) in the *WhatsApp Business Platform Cloud API Reference*.
 - b. For **Additional setting**:
 - i. For **Data localization region - optional**, choose one of Meta's regions in which to store your data at rest. For more information on Meta's data privacy policies, see [Data Privacy & Security](#) and [Cloud API Local Storage](#) in the *WhatsApp Business Platform Cloud API Reference*.
 - ii. Tags are pairs of keys and values that you can optionally apply to your AWS resources to control access or usage. Choose **Add new tag** and enter a key-value pair to attach.
12. A WABA can have one message and event destination to log events for the WABA and all resources associated to the WABA. To enable event logging, including logging of receiving a customer message, turn on **Message and event publishing**. For more information, see [Message and event destinations in AWS End User Messaging Social](#).

 **Important**

You must enable **Message and event publishing** to be able to respond to customer messages.

In the **Message and event destination details** section, turn on **Event publishing**.

13. For **Destination type** choose either Amazon SNS or Connect Customer
 - a. To send your events to an Amazon SNS destination, enter an existing topic ARN in **Topic ARN**. For example IAM policies, see [IAM policies for Amazon SNS topics](#).
 - b. For Connect Customer
 - i. For **Connect instance** choose an instance from the drop down.
 - ii. For **Two-way channel role**, choose either:
 - A. **Choose existing IAM role** – Choose an existing IAM policy from the **Existing IAM roles** drop down. For example IAM policies, see [IAM policies for Connect Customer](#).
 - B. **Enter IAM role ARN** – Enter the ARN of the IAM policy into **Use existing IAM role Arn**. For example IAM policies, see [IAM policies for Connect Customer](#).
14. To complete setup, choose **Add phone number**.

View a phone number's status

To be able to send messages in AWS End User Messaging Social, the phone number's **Status** must be *Active*.

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Phone numbers**.
3. In the **Phone numbers** section, the **Status** column has each phone number's status.

Note

If a phone number's **Status** is **Incomplete setup**, you can choose the phone number and then choose **Complete setup** to finish setting up the phone number.

View a phone number's ID in AWS End User Messaging Social

To be able to send messages with the AWS CLI, you need the **Phone number ID** to identify the phone number to use when sending.

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Phone numbers**.
3. In the **Phone numbers** section, choose a phone number.
4. The **Phone number details** section contains the **Phone number ID** of the phone number.

Increase messaging conversation limits in WhatsApp

Messaging limits refer to the maximum number of business-initiated conversations a business phone number can open in a 24-hour period. Business phone numbers are initially limited to 250 business-initiated conversations in a 24-hour moving period. This limit can be increased by Meta based on the quality rating of your messages and how many messages you send. Business-initiated conversations can only use template messages.

When a customer messages you, this opens a 24-hour service window. During this time, you can send all [message types](#).

You can increase your messaging limit to 1,000 messages on your own by following these guidelines:

- Your business phone number must have an [Active status](#).
- If your business phone number has a [low quality rating](#), it may continue to be limited to 250 business-initiated conversations per day until its quality rating improves.
- Apply for [Business Verification](#). If your business is approved, the messaging quality will be analyzed to determine if your messaging activity warrants an increase to your messaging limit. Based on the analysis, your request for a messaging limit increase will either be approved or denied by Meta.
- Apply for [Identity Verification](#). If you complete identity verification and your identity is confirmed, Meta will approve a messaging limit increase.
- Open 1,000 or more business-initiated conversations in a 30-day moving period using a template with a high quality rating. Once you reach the 1,000 conversation threshold, your messaging

quality will be analyzed to determine if your messaging activity warrants an increase to your messaging limit. The goal is to send high-quality messages consistently to potentially get your messaging limit increased.

If you completed Business Verification or Identity Verification, or opened 1,000 or more business conversations, and you are still limited to 250 business-initiated conversations, submit a request to Meta for a message tier upgrade.

If your business or identity verification is rejected, you can improve your chances of getting approved by sending high-quality messages. By sending high-quality, compliant, and opt-in messages, your messaging activity and quality may be reevaluated, potentially leading to an increase in your approved messaging capabilities.

Your messaging quality score on WhatsApp is calculated based on recent user feedback and interactions, with more weight given to more recent data. This helps assess the overall quality and reliability of your messaging on the platform.

Note

If you opt in to Meta's Marketing Messages (MM) API, marketing template messaging limits are managed dynamically by Meta. According to Meta, templates with high engagement automatically receive expanded reach. Static conversation-based limits continue to apply to utility, authentication, and service messages.

Message limits level increases

- 1K business-initiated conversations
- 10K business-initiated conversations
- 100K business-initiated conversations
- An unlimited number of business-initiated conversations

Increase message throughput in WhatsApp

Message throughput is the number of incoming and outgoing messages per second (MPS) for a phone number. By default, each phone number has an MPS of 80. Meta can increase your MPS to 1,000 if you meet the following requirements:

- The phone number must be able to send an unlimited number of [business-initiated conversations](#)
- The phone number must have a [quality rating](#) of medium or higher.

Understanding phone number quality rating in WhatsApp

The quality of your phone number and messages is determined by Meta. Your messaging quality score is based on how your messages have been received by customers over the past seven days, with more recent messages weighted more heavily. The messaging quality score is calculated based on a combination of quality signals from the conversations between you and your WhatsApp users. These signals include user feedback like blocks, reports, and the reasons users provide when they block a business. Meta evaluates the quality of your messages based on how well they are received by your customers on WhatsApp, with a focus on recent feedback and interactions.

WhatsApp phone number quality ratings

- Green: High quality
- Yellow: Medium quality
- Red: Low quality

WhatsApp phone number status

- *Connected*: You can send messages within your message quota.
- *Flagged*: Your phone number quality is low and needs to be improved. If your quality doesn't improve in seven days, your phone number status is changed to Connected but your business-initiated conversation limit is lowered one tier.
- *Restricted*: You have reached your business-initiated conversation limit for the current 24-hour period. You can still respond to incoming messages. Once the 24-hour period is over, you can send messages again.

View a phone number quality rating

Follow these directions to view a phone numbers quality.

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.

2. In **Business accounts**, choose a WhatsApp Business Account (WABA).
3. On the **Phone numbers** tab, view your phone number, display name, quality rating, and the number of business-initiated conversations that you have left for the day.

Using message templates in AWS End User Messaging Social

Important

Starting on 4/1/2025 Meta will block marketing message templates sent to the US country code of +1. For more information, see [Per-User Marketing Template Message Limits](#) in the *WhatsApp Business Platform Cloud API Reference*.

You can use message templates to create message types that you use frequently, such as weekly newsletters or appointment reminders. Template messages are the only type of message that can be sent to customers who have yet to message you, or who have not sent you a message in the last 24 hours.

Meta assigns each template a quality rating and status. The quality rating impacts a template's status and lowers a template's pacing or sending rate.

Templates are associated with your WhatsApp Business Account (WABA), can be managed through the AWS End User Messaging Social console, and are reviewed by WhatsApp.

You can send the following template types:

- Text-based
- Media-based
- Interactive message
- Location-based
- Authentication templates with one-time password buttons
- Multi-Product Message templates

Meta provides pre-approved sample templates. To learn more, see [Sample message templates](#).

For more information on the types of message templates, see [Message template](#) in the *WhatsApp Business Platform Cloud API Reference*.

Using message templates in the AWS Console

Create and manage your WhatsApp message templates directly in the AWS End User Messaging Social console.

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Business account**, and then choose a WABA.
3. On the **Message templates** tab, you can:
 - **Create new templates** by choosing **Create template** and following the template creation workflow
 - **View template status** to see which templates are approved, pending, or rejected
 - **Edit existing templates** by selecting a template and choosing **Edit**
 - **Delete templates** that are no longer needed

Templates must be approved by Meta before they can be used to send messages to your customers. You can monitor the approval status of your templates in the console.

Managing message templates in the AWS Console

Creating a message template

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Business account**, and then choose a WABA.
3. On the **Message templates** tab, choose **Create template**.
4. Configure your template:
 - **Template name:** Enter a unique name for your template (lowercase letters, numbers, and underscores only)
 - **Category:** Select the template category (Marketing, Utility, or Authentication)
 - **Language:** Choose the language for your template content
 - **Header (optional):** Add text, media, or a variable to the header
 - **Body:** Enter your message text (you can include variables using `{{1}}`, `{{2}}`, etc.)

- **Footer (optional):** Add footer text (up to 60 characters)
- **Buttons (optional):** Add call-to-action or quick reply buttons

5. Choose **Create template**.

Your template is submitted to Meta for review. Approval typically takes a few minutes to 24 hours.

Viewing template status

Templates can have the following statuses:

- **Approved:** The template is ready to use
- **Pending:** The template is under review by Meta
- **Rejected:** The template was not approved (hover over the status to see the reason)
- **Disabled:** The template was paused due to low quality ratings

To view your templates, navigate to the **Message templates** tab in your WABA details page.

Editing a message template

1. Select the template you want to modify.
2. Choose **Edit**.
3. Make your changes to the template content.
4. Choose **Save**.

Important

When you edit and submit an approved template, it enters a pending state and requires Meta approval. The template becomes unavailable for use until Meta approves your changes. Consider the impact on your messaging workflows before editing templates that are actively being used.

Deleting a message template

1. Select the template you want to remove.

2. Choose **Delete**.
3. Confirm the deletion.

Deleted templates cannot be recovered. If you're using the template in active campaigns, those messages will fail after deletion.

Template quality ratings

Meta assigns quality ratings to your templates based on customer feedback and engagement. Templates with low quality ratings may be paused or disabled. To maintain good quality ratings:

- Send relevant, valuable content to your customers
- Avoid sending messages too frequently
- Provide clear opt-out options
- Respond promptly to customer replies

You can view quality ratings in the template details page.

Template variables

Use variables to personalize your messages. Variables are defined using double curly braces with numbers ({{1}}, {{2}}, etc.). When sending a message, you provide the actual values for these variables.

Variables must be provided in sequential order starting from {{1}}.

Example Template with variables

Template text: Hello {{1}}, your order {{2}} will be delivered on {{3}}.

When sending: Replace {{1}} with customer name, {{2}} with order number, and {{3}} with delivery date.

Template categories

Marketing

Promotional content, product announcements, or offers. These templates may have sending limitations.

Utility

Account updates, order notifications, or customer service messages. These templates typically have higher sending limits.

Authentication

One-time passwords or verification codes. These templates have the highest priority and sending limits.

Choose the category that best matches your use case. Misclassifying templates may result in rejection or account restrictions.

Creating message templates with the CreateWhatsAppMessageTemplate API

You can create customized WhatsApp message templates using the API. This topic describes how to use the [CreateWhatsAppMessageTemplate](#) to create a variety of message templates.

Message template components

message templates can include the following components:

- **Header:** Title text that appears at the top
- **Body:** Main message content with variable placeholders
- **Footer:** Additional information at the bottom
- **Buttons:** Clickable elements that link to URLs

In the following examples Replace *ENDPOINT* and *WABA_ID* with your actual endpoint URL and ID.

Create a basic English utility template

This example creates a utility message template in English that uses only the BODY component, and doesn't include HEADER, FOOTER, or BUTTON components. The body text uses variable placeholders.

```
$ aws socialmessaging create-whatsapp-message-template --region us-east-1 --endpoint-  
url ENDPOINT_URL \
```

```

--id WABA_ID \
--template-definition '{
  "name": "order_update_basic",
  "language": "en_US",
  "allow_category_change": true,
  "category": "UTILITY",
  "components": [
    {
      "type": "BODY",
      "text": "Hi {{1}}, your order #{{2}} has been shipped. Track your delivery
below.",
      "example": {
        "body_text": [
          [
            "Jane",
            "12345"
          ]
        ]
      }
    }
  ]
}'

```

Create a basic English utility template with button

This example creates a utility message template in English that includes BODY and BUTTON components.

```

$ aws socialmessaging create-whatsapp-message-template --region us-east-1 --endpoint-
url ENDPOINT_URL \
--id WABA_ID \
--template-definition '{
  "name": "order_update_with_button",
  "language": "en_US",
  "allow_category_change": true,
  "category": "UTILITY",
  "components": [
    {
      "type": "BODY",
      "text": "Hi {{1}}, your order #{{2}} has been shipped. Track your delivery
below.",
      "example": {
        "body_text": [

```

```

        [
            "Jane",
            "12345"
        ]
    ]
},
{
    "type": "BUTTONS",
    "buttons": [
        {
            "type": "URL",
            "text": "Track Order",
            "url": "https://example.com/track"
        }
    ]
}
]
}'

```

Create a complex English utility message template with a header, body, and button

This example creates a utility message template in English that includes HEADER, BODY, and BUTTON components.

```

$ aws socialmessaging create-whatsapp-message-template --region us-east-1 --endpoint-
url ENDPOINT_URL \
--id WABA_ID \
--template-definition '{
    "name": "account_creation_confirmation_3333",
    "category": "UTILITY",
    "language": "en_US",
    "status": "APPROVED",
    "components": [
        {
            "type": "HEADER",
            "format": "TEXT",
            "text": "Finalize account set-up"
        },
        {
            "type": "BODY",

```

```

    "text": "Hi {{1}},\n\nYour new account has been created successfully. \n\nPlease
verify {{2}} to complete your profile.",
    "example": {
      "body_text": [
        [
          "John",
          "your email address"
        ]
      ]
    },
    {
      "type": "BUTTONS",
      "buttons": [
        {
          "type": "URL",
          "text": "Verify account",
          "url": "https://www.example.com/"
        }
      ]
    }
  ]
}'

```

Create a basic marketing message template

This example creates a basic marketing message template that includes only a BODY component.

```

$ aws socialmessaging create-whatsapp-message-template --region us-east-1 --endpoint-
url ENDPOINT_URL \
--id WABA_ID \
--template-definition '{
  "id": "1290345849293233",
  "name": "holiday_special_1395238",
  "category": "MARKETING",
  "language": "en_US",
  "status": "PENDING",
  "components": [
    {
      "type": "BODY",
      "text": "Season's Greetings {{1}}!\n\nCelebrate {{2}} with amazing deals up to
{{3}} off.\n\nPlus, get free gift wrapping on all orders above $50.",
      "example": {

```

```

        "body_text": [
          [
            "Pawan",
            "Christmas",
            "30%"
          ]
        ]
      }
    ],
    "metaTemplateId": "1290345849293233"
  }'

```

Create a complex marketing message template

This example creates a marketing message template in English that includes HEADER, BODY, and BUTTON components.

```

$ aws socialmessaging create-whatsapp-message-template --region us-east-1 \
--endpoint-url ENDPOINT_URL \
--id WABA_ID \
--template-definition '{
  "name": "summer_sale_1",
  "category": "MARKETING",
  "language": "en_US",
  "status": "APPROVED",
  "components": [
    {
      "type": "HEADER",
      "format": "TEXT",
      "text": "Summer Sale!"
    },
    {
      "type": "BODY",
      "text": "Hi {{1}}! Get {{2}} off all summer items. Shop now before stock runs
out!",
      "example": {
        "body_text": [
          [
            "Alex",
            "20%"
          ]
        ]
      }
    }
  ]
}'

```

```
    }
  },
  {
    "type": "FOOTER",
    "text": "Valid until August 31st"
  },
  {
    "type": "BUTTONS",
    "buttons": [
      {
        "type": "URL",
        "text": "Shop Now",
        "url": "https://example.com/sale"
      }
    ]
  }
]
}'
```

Next steps

Once you've created or edited a template, you must submit it for review with WhatsApp. Meta's review can take up to 24 hours. Meta sends an email to your Business Manager admin and updates the template status. You can check the status of your template in the AWS End User Messaging Social console.

Understanding template pacing in WhatsApp

Template pacing is a method, used by Meta, that allows time for early customer feedback on new or modified templates. It identifies and pauses templates that receive poor engagement or feedback, giving you time to adjust the template content before sending it to too many customers. This reduces the risk of negative customer feedback impacting the business. For example, if too many customers "block" your message, or if your template has low read rates, then your template quality rating can be lowered.

Template pacing affects newly created templates, templates that have been unpaused, and templates without a high quality rating. Template pacing is often started by a previous history of low quality or paused templates. When a template is paced, messages using that template are sent normally up to a certain threshold determined by Meta. After that, subsequent messages are held

to allow time for customer feedback. If the feedback is positive, the template pacing is then scaled up. If the feedback is negative, the template pacing is lowered, allowing you to adjust the template content. For more information, see [Template pacing](#) in the *WhatsApp Business Platform Cloud API Reference*.

Get feedback on a template's lowered status with WhatsApp Manager

Meta provides information on the reason a template's status was lowered. Use the feedback from Meta to edit the template and submit it for reapproval, use a different template, or change your application's behavior. If you edit the message template and it is reapproved, its quality rating will gradually improve as long as it doesn't receive frequent negative feedback or low read rates.

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Business account**, and then choose a WABA.
3. On the **Message templates** tab, select the template with lowered status.
4. View the template details to see feedback on why the rating was lowered. You can also hover over the status indicator for additional information.

Understanding a template's status and quality rating in WhatsApp

Each message template is assigned a quality rating based on usage, customer feedback, and customer engagement. A template can be used only if the status is Active, but the quality determines the template pacing. If a message template consistently receives negative feedback or experiences low engagement, it will cause a change in the template's status.

Meta changes a template's status or quality rating automatically based on negative or positive feedback and engagement. If your template status changes, you will receive an email and event notification. You can check the status of your template in the AWS End User Messaging Social console.

If your template is rejected by WhatsApp, you can edit the template and resubmit for approval or file an appeal with WhatsApp. To learn more, see [Appeals](#) in the *WhatsApp Business Platform Cloud API Reference*.

Template status	Quality rating	Meaning
In-Review		The message template is being reviewed. This can take up to 24 hours to complete.
Rejected		The message template was rejected, and you can file an appeal.
Active	Pending	The message template hasn't receive quality feedback or read rate information from customers, but the template can still be used to send messages.
Active	High	The message template has received little to no negative customer feedback and can be used to send messages.
Active	Medium	The message template has received negative feedback from customers, or low read rates, and may be paused or turned off.
Active	Low	<p>The message template has received negative feedback from customers, or low read rates. Message templates with this status can be used, but are at risk of being paused or disabled.</p> <p>When a template moves to the Active-Low status, its</p>

Template status	Quality rating	Meaning
		sending is paused. The first pause is three hours, the second pause is six hours, and the next pause disables the template.
Paused		The message template has been paused due to recurring negative feedback from customers, or low read rates.
Disabled		The message template has been disabled due to recurring negative feedback from customers.
Appeal Requested		An appeal has been requested .

Reasons why a template is rejected in WhatsApp

If your message template is reviewed and rejected by Meta, you will receive an email explaining why the template was rejected. You can appeal the rejection or modify your message template. These are some of the common reasons Meta might reject a message template:

- Variable parameters contain special characters, such as a #, \$, or %.
- Variable parameters are missing, have mismatched curly braces, or are not sequential.
- The message template contains content that violates either [WhatsApp Commerce Policy](#) or [WhatsApps Business Policy](#).

For more information, see [Common Rejection Reasons](#) in the *WhatsApp Business Platform Cloud API Reference*.

Managing WhatsApp Flows in AWS End User Messaging Social

WhatsApp Flows enable businesses to create interactive, multi-screen experiences that users can complete without leaving WhatsApp. Flows support data collection, form submissions, and complex UI components like text inputs, radio buttons, checkboxes, date pickers, and dropdown menus. Use Flows to build experiences such as appointment booking, lead generation forms, customer surveys, and shopping flows.

With AWS End User Messaging Social, you can manage your WhatsApp Flows directly through the AWS End User Messaging Social console or by using the AWS End User Messaging Social API. You can create, update, publish, preview, deprecate, and delete Flows without needing to use the Meta console or Meta's APIs directly.

Flows are associated with your WhatsApp Business Account (WABA). Each Flow is defined by a JSON schema that describes its screens, components, and logic. For more information about the Flow JSON schema, see [Flow JSON](#) in the *Meta WhatsApp Business Platform documentation*.

Note

The current release supports static Flows where all screens, components, and navigation logic are defined in the Flow JSON. Flows with endpoints that exchange data between screens and a business server (the `endpoint_uri` and `data_exchange` capabilities) are planned for an upcoming release. For more information, see [Limitations and considerations](#).

Key concepts

Flow

An interactive, multi-screen experience defined by a JSON schema that users complete within WhatsApp.

Flow JSON

The JSON definition that describes the Flow's screens, components, and logic. Maximum size is 10 MB. We recommend using JSON version 7.3.

Flow category

A classification that describes the business purpose of the Flow. Categories include: Sign Up, Sign In, Appointment Booking, Lead Generation, Contact Us, Customer Support, Survey, Shopping, and Other.

Flow status

The lifecycle state of a Flow. See [Understanding Flow status](#).

Flow token

A unique identifier you provide when sending a Flow to track the session. The token is returned in the Flow response webhook when the user completes the Flow.

Understanding Flow status

Each WhatsApp Flow has a status that represents its lifecycle state. The status determines what actions you can take on the Flow.

Flow status values

Status	Description	Allowed actions
DRAFT	Initial state when a Flow is created. The Flow can be edited and tested but cannot be sent to users.	Update, update assets, publish, delete, preview
PUBLISHED	The Flow is available for use in template messages and can be sent to users. Published Flows cannot be edited directly.	Deprecate, preview. To edit, update assets (which reverts to DRAFT).
DEPRECATED	The Flow is no longer recommended for use. This is an irreversible state.	None (terminal state)
BLOCKED	Meta has blocked the Flow due to integrity or policy violations.	Contact Meta support
THROTTLED	Meta has temporarily throttled the Flow.	Wait for throttling to be lifted

Important

Publishing and deprecating a Flow are irreversible operations. Once a Flow is published, you cannot revert it to DRAFT directly. However, updating a published Flow's assets (JSON definition) will revert it to DRAFT status, requiring you to re-publish.

Meta sends a webhook notification each time a Flow's status changes. You receive these notifications through your configured event destination. For more information, see [Receiving Flow responses and status changes](#).

Creating and editing WhatsApp Flows

You can create a WhatsApp Flow using the AWS End User Messaging Social console or the AWS End User Messaging Social API. When you create a Flow, it is placed in DRAFT status. You can optionally provide a Flow JSON definition at creation time, or add it later using the update assets operation.

Creating a Flow (console)

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Business account**, and then choose a WABA.
3. Choose the **Flows** tab.
4. Choose **Create flow**.
5. For **Flow name**, enter a unique name for your Flow. Flow names must be unique within the WABA (maximum 200 characters).
6. For **Category**, select one or more categories that describe the business purpose of your Flow. At least one category is required. Available categories are:
 - Sign Up
 - Sign In
 - Appointment Booking
 - Lead Generation
 - Contact Us
 - Customer Support

- Survey
 - Shopping
 - Other
7. For **Template**, choose a pre-built template to start from, or choose **Default** to start with a blank Flow. The following templates are available without an endpoint:
- **Default** — A blank starting template
 - **Collect purchase interest** — Collect information from potential buyers
 - **Get feedback** — Gather customer feedback
 - **Send a survey** — Create a multi-question survey
 - **Customer support** — Collect support request details
8. Choose **Create**.

After you create the Flow, you can edit the Flow JSON to customize the screens and components. The Flow preview on the right side of the page shows a live preview of how your Flow will appear to users in WhatsApp.

Creating a Flow (API)

Use the `CreateWhatsAppFlow` API to create a new Flow. The following example uses the AWS CLI to create a Flow with a name and category:

```
aws socialmessaging create-whatsapp-flow \  
  --id {WABA_ID} \  
  --flow-name "my_lead_generation_form" \  
  --categories '["LEAD_GENERATION"]'
```

The response includes the Flow ID assigned by Meta:

```
{  
  "flowId": "{FLOW_ID}",  
  "validationErrors": []  
}
```

You can also provide a Flow JSON definition at creation time and publish immediately:

```
aws socialmessaging create-whatsapp-flow \  
  --flow-definition '{  
    "flowId": "{FLOW_ID}",  
    "validationErrors": []  
  }'  
  --publish
```

```
--id {WABA_ID} \  
--flow-name "appointment_booking" \  
--categories '["APPOINTMENT_BOOKING"]' \  
--flow-json file://flow.json \  
--publish
```

Note

If you include `--publish`, the Flow JSON must be valid and pass Meta's validation. If there are validation errors, the Flow is created in DRAFT status and the errors are returned in the `validationErrors` field.

To clone an existing Flow within the same WABA:

```
aws socialmessaging create-whatsapp-flow \  
  --id {WABA_ID} \  
  --flow-name "lead_form_v2" \  
  --categories '["LEAD_GENERATION"]' \  
  --clone-flow-id {FLOW_ID}
```

Updating a Flow

There are two types of updates you can make to a Flow:

- **Update metadata** — Change the Flow name or categories using `UpdateWhatsAppFlow`.
- **Update assets** — Change the Flow JSON definition using `UpdateWhatsAppFlowAssets`.

Important

Updating a published Flow's assets reverts its status to DRAFT. You must re-publish the Flow for it to be available for use in messages.

The following example updates a Flow's name:

```
aws socialmessaging update-whatsapp-flow \  
  --id {WABA_ID} \  
  --flow-name "lead_form_v2"
```

```
--flow-id {FLOW_ID} \  
--flow-name "updated_lead_form"
```

The following example updates a Flow's JSON definition:

```
aws socialmessaging update-whatsapp-flow-assets \  
--id {WABA_ID} \  
--flow-id {FLOW_ID} \  
--flow-json file://updated_flow.json
```

If the Flow JSON contains validation errors, they are returned in the response. You must fix these errors before you can publish the Flow.

Listing and viewing Flows

Use `ListWhatsAppFlows` to see all Flows for a WABA, and `GetWhatsAppFlow` to retrieve details about a specific Flow.

List all Flows:

```
aws socialmessaging list-whatsapp-flows \  
--id {WABA_ID}
```

Get details for a specific Flow:

```
aws socialmessaging get-whatsapp-flow \  
--id {WABA_ID} \  
--flow-id {FLOW_ID}
```

The `GetWhatsAppFlow` response includes the Flow status, categories, validation errors, JSON version, preview information, and health status.

To retrieve the Flow JSON definition (assets), use `ListWhatsAppFlowAssets`:

```
aws socialmessaging list-whatsapp-flow-assets \  
--id {WABA_ID} \  
--flow-id {FLOW_ID}
```

The response includes a presigned download URL for the Flow JSON file. The URL expires after a short period. No additional authentication is needed to download the file from the URL.

Deleting a Flow

You can only delete Flows that are in DRAFT status. Published or deprecated Flows cannot be deleted.

To delete a Flow using the console, select the Flow on the **Flows** tab and choose **Delete**.

To delete a Flow using the API:

```
aws socialmessaging delete-whatsapp-flow \  
  --id {WABA_ID} \  
  --flow-id {FLOW_ID}
```

Warning

Deleting a Flow is permanent and cannot be undone.

Publishing, deprecating, and previewing WhatsApp Flows

Publishing a Flow

Publishing a Flow makes it available for use in template messages. Before you can publish a Flow, it must meet the following requirements:

- The Flow must be in DRAFT status.
- The Flow must have a valid Flow JSON definition with no validation errors.
- The Flow JSON must pass Meta's validation checks.

Important

Publishing a Flow is irreversible. Once published, a Flow cannot be reverted to DRAFT directly. If you need to edit a published Flow, update its assets using `UpdateWhatsAppFlowAssets`, which reverts the Flow to DRAFT status. You must then re-publish the Flow.

To publish a Flow using the API:

```
aws socialmessaging publish-whatsapp-flow \  
  --id {WABA_ID} \  
  --flow-id {FLOW_ID}
```

If the Flow has validation errors, the publish operation fails. Use `GetWhatsAppFlow` to view validation errors and fix them before retrying.

Deprecating a Flow

Deprecating a Flow marks it as no longer recommended for use. Only Flows in `PUBLISHED` status can be deprecated.

Important

Deprecating a Flow is irreversible. Deprecated Flows cannot be reverted to `PUBLISHED` or `DRAFT` status.

To deprecate a Flow using the API:

```
aws socialmessaging deprecate-whatsapp-flow \  
  --id {WABA_ID} \  
  --flow-id {FLOW_ID}
```

Previewing a Flow

You can generate a web preview URL to test a Flow before publishing. Preview URLs expire after 30 days and can be shared with stakeholders for review. The preview URL opens a simulation of the Flow as it would appear to users in WhatsApp.

To generate a preview URL using the API:

```
aws socialmessaging get-whatsapp-flow-preview \  
  --id {WABA_ID} \  
  --flow-id {FLOW_ID}
```

The response includes the preview URL and its expiration timestamp:

```
{
```

```
"flowId": "{FLOW_ID}",
"preview": {
  "previewUrl": "https://business.facebook.com/wa/manage/flows/{FLOW_ID}/
preview/?token=...",
  "expiresAt": "2026-07-01T23:46:55+0000"
}
}
```

To force generation of a new preview URL (for example, if the previous URL was shared and you want a fresh URL with a new expiration), use the `--invalidate` parameter:

```
aws socialmessaging get-whatsapp-flow-preview \
  --id {WABA_ID} \
  --flow-id {FLOW_ID} \
  --invalidate
```

Sending WhatsApp Flows to users

You send Flows to users through template messages using the existing `SendWhatsAppMessage` API. No changes to the send message API are required. The Flow is embedded in the template message as a button component with a `flow` sub-type.

Before you can send a Flow, the following must be true:

- The Flow must be in `PUBLISHED` status.
- You must have a message template that references the Flow (with a Flow button).
- The user must have opted in to receive messages from you.

Example: Sending a Flow via template message

The following example sends a template message that contains a Flow button. The `flow_token` is a unique identifier you provide to track the session, and `flow_action_data` specifies the starting screen and any initial data.

```
aws socialmessaging send-whatsapp-message \
  --origination-phone-number-id {PHONE_NUMBER_ID} \
  --meta-api-version v20.0 \
  --message '{
```

```
"messaging_product": "whatsapp",
"to": "+14085551234",
"type": "template",
"template": {
  "name": "flow_template_name",
  "language": {"code": "en_US"},
  "components": [
    {
      "type": "button",
      "sub_type": "flow",
      "index": "0",
      "parameters": [
        {
          "type": "action",
          "action": {
            "flow_token": "unique-session-token-123",
            "flow_action_data": {
              "screen": "WELCOME_SCREEN",
              "data": {}
            }
          }
        }
      ]
    }
  ]
}
```

In the preceding command, replace the following:

- *{PHONE_NUMBER_ID}* — Your originating phone number's ID.
- *+14085551234* — The recipient's phone number in international format.
- *flow_template_name* — The name of your approved message template that includes a Flow button.
- *unique-session-token-123* — A unique token you generate to identify this Flow session. This token is returned in the webhook when the user completes the Flow.
- *WELCOME_SCREEN* — The ID of the first screen in your Flow JSON definition.

Note

The `flow_token` should be unique per session to allow you to correlate Flow responses with specific send events. You can use a UUID or any other unique identifier.

Setting up a template for Flows

To send a Flow, you need a message template with a Flow button configured. When creating the template, add a call-to-action button of type FLOW that references your published Flow ID.

For more information about creating message templates, see [Using message templates in AWS End User Messaging Social](#). For information about Flow template configuration, see [Sending a Flow](#) in the *Meta WhatsApp Business Platform documentation*.

Receiving Flow responses and status changes

When a user completes a Flow or when a Flow's status changes, AWS End User Messaging Social delivers webhook notifications through your existing configured event destination. No additional setup or new event types are required — Flow events are delivered automatically to the same event destination you use for messages and delivery receipts. You must have an event destination enabled to receive these notifications. For more information, see [Add a message and event destination to AWS End User Messaging Social](#).

Flow response messages

When a user completes a Flow, you receive an inbound message with the type `interactive` and the subtype `nfm_reply`. The `response_json` field contains the data submitted by the user as a JSON string.

The following example shows a Flow response webhook payload:

```
{
  "messages": [
    {
      "from": "14085551234",
      "id": "wamid.XXX",
      "timestamp": "1234567890",
```

```
        "type": "interactive",
        "interactive": {
            "type": "nfm_reply",
            "nfm_reply": {
                "response_json": "{\"name\": \"John Doe\", \"email\": \"john@example.com\", \"phone\": \"+14085551234\"}",
                "body": "Sent",
                "name": "flow"
            }
        }
    }
]
```

The `response_json` contains key-value pairs corresponding to the form field names defined in your Flow JSON and the values entered by the user. Parse this JSON string to extract the submitted data.

Flow status change notifications

When Meta changes a Flow's status (for example, from PUBLISHED to BLOCKED), you receive a notification through your event destination. The following example shows a status change payload:

```
{
  "entry": [
    {
      "id": "WHATSAPP_BUSINESS_ACCOUNT_ID",
      "changes": [
        {
          "value": {
            "event": "FLOW_STATUS_CHANGE",
            "flow_id": "{FLOW_ID}",
            "old_status": "PUBLISHED",
            "new_status": "BLOCKED",
            "reason": "Policy violation detected"
          },
          "field": "flows"
        }
      ]
    }
  ]
}
```

Monitor these status change events to detect when Meta blocks or throttles your Flows so you can take corrective action.

Flow version expiry warnings

Meta sends a warning webhook when the Flow JSON version used by your Flow is approaching its end-of-life date. When a version expires, you cannot send the Flow until you update to a newer version.

```
{
  "entry": [
    {
      "id": "WHATSAPP_BUSINESS_ACCOUNT_ID",
      "changes": [
        {
          "value": {
            "event": "FLOW_VERSION_EXPIRY_WARNING",
            "flow_id": "{FLOW_ID}",
            "warning": "Your current Flow version will freeze in 21 days.
            You won't be able to send the Flow after it expires. Please migrate to the recommended
            version as soon as possible."
          },
          "field": "flows"
        }
      ]
    }
  ]
}
```

When you receive this warning, update your Flow JSON to use the recommended version (currently 7.3) by calling `UpdateWhatsAppFlowAssets` with the updated JSON, then re-publish the Flow.

Limitations and considerations

Consider the following limitations when working with WhatsApp Flows:

- **Flow name uniqueness** — Flow names must be unique within a WABA.
- **Flow JSON size** — The maximum size of a Flow JSON definition is 10 MB.
- **Flow JSON version** — Supported versions for publishing are 5.1, 6.0–6.3, and 7.0–7.3. We recommend version 7.3.

- **Only DRAFT Flows can be deleted** — Published or deprecated Flows cannot be deleted.
- **Data exchange endpoints** — Flows with endpoints that exchange data between screens and a business server (the `endpoint_uri` and `data_exchange` capabilities) are planned for an upcoming release. Only Flows without an endpoint are currently supported.
- **Flow migration** — Copying or migrating Flows between WABAs is not supported. You can clone a Flow within the same WABA using the `cloneFlowId` parameter.
- **Rate limits** — Standard Meta API rate limits apply to all Flow operations.

Message and event destinations in AWS End User Messaging Social

An event destination is an Amazon SNS topic or Connect Customer instance that WhatsApp events are sent to. When you turn on event publishing, all of your send and receive events are sent to the message and event destination. Use events to monitor, track, and analyze the status of outbound messages and incoming customer communications.

Each WhatsApp Business Account (WABA) can have one event destination. All events from all resources associated to the WABA are logged to that event destination. For example, you could have a WABA with three phone numbers associated to it and all events from those phone numbers are logged to the one event destination.

Topics

- [Add a message and event destination to AWS End User Messaging Social](#)
- [Message and event format in AWS End User Messaging Social](#)
- [WhatsApp message status](#)

Add a message and event destination to AWS End User Messaging Social

When you turn on message and event publishing, all of the events generated by your WhatsApp Business Account (WABA) are sent to the Amazon SNS topic. This includes events for each phone number associated to a WABA. Your WABA can have one Amazon SNS topic associated with it.

Prerequisites

Before you begin, the following prerequisites must be met to use either an Amazon SNS topic or Connect Customer instance as a message and event destination.

Amazon SNS topic

- An Amazon SNS topic has been [created](#) and [permissions](#) have been added.

Note

Amazon SNS FIFO topics are not supported.

- **(Optional)** To use an Amazon SNS topic that is encrypted using AWS KMS keys you have to grant AWS End User Messaging Social permissions to the [existing key policy](#).

Connect Customer instance

- An Connect Customer instances has been [created](#) and [permissions](#) have been added.

Add a message and event destination

1. Open the AWS End User Messaging Social console at <https://console.aws.amazon.com/social-messaging/>.
2. Choose **Business account**, and then choose a WABA.
3. On the **Event destination** tab, choose **Edit destination**.
4. To turn on an event destination, choose **Enable**.
5. For **Destination type** choose either Amazon SNS or Connect Customer
 - a. To send your events to an Amazon SNS destination, enter an existing topic ARN in **Topic ARN**. For example IAM policies, see [IAM policies for Amazon SNS topics](#).
 - b. For Connect Customer
 - i. For **Connect instance** choose an instance from the drop down.
 - ii. For **Two-way channel role**, choose either:
 - A. **Choose existing IAM role** – Choose an existing IAM policy from the **Existing IAM roles** drop down. For example IAM policies, see [IAM policies for Connect Customer](#).
 - B. **Enter IAM role ARN** – Enter the ARN of the IAM policy into **Use existing IAM role Arn**. For example IAM policies, see [IAM policies for Connect Customer](#).
6. Choose **Save changes**.

Encrypted Amazon SNS topic policies

You can use Amazon SNS topics that are encrypted using AWS KMS keys for an additional level of security. This added security can be helpful if your application handles private or sensitive data. For more information about encrypting Amazon SNS topics using AWS KMS keys, see [Enable compatibility between event sources from AWS services and encrypted topics](#) in the *Amazon Simple Notification Service Developer Guide*.

Note

Amazon SNS FIFO topics are not supported.

The example statement uses the, optional but recommended, `SourceAccount` and `SourceArn` conditions to avoid the confused deputy problem and only the AWS End User Messaging Social owner account has access. For more information on the confused deputy problem, see [The confused deputy problem](#) in the *IAM user guide*.

The key that you use must be *symmetric*. Encrypted Amazon SNS topics don't support asymmetric AWS KMS keys.

The key policy must be modified to allow AWS End User Messaging Social to use the key. Follow the directions in [Changing a key policy](#), in the *AWS Key Management Service Developer Guide*, to add the following permissions to the existing key policy:

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "social-messaging.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey*",
    "kms:Decrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "{ACCOUNT_ID}"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:{PARTITION}:social-messaging:{REGION}:{ACCOUNT_ID}:"
    }
  }
}
```

```

    }
  }
}

```

IAM policies for Amazon SNS topics

To use an existing IAM role or to create a new role, attach the following policy to that role so that AWS End User Messaging Social can assume it. For information about how to modify the trust relationship of a role, see [Modifying a Role](#) in the [IAM user guide](#).

The following is the **permission policy** for the IAM role. The permission policy allows for publishing to Amazon SNS topics.

In the following IAM permission policy, make the following changes:

- Replace `{PARTITION}` with the AWS partition that you use AWS End User Messaging Social in.
- Replace `{REGION}` with the AWS Region that you use AWS End User Messaging Social in.
- Replace `{ACCOUNT}` with the unique ID for your AWS account.
- Replace `{TOPIC_NAME}` with the Amazon SNS topics that will receive messages.

```

{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "social-messaging.amazonaws.com"
    ]
  },
  "Action": "sns:Publish",
  "Resource": "arn:{PARTITION}:sns:{REGION}:{ACCOUNT}:{TOPIC_NAME}"
}

```

IAM policies for Connect Customer

If you want AWS End User Messaging Social to use an existing IAM role or if you create a new role, attach the following policies to that role so that AWS End User Messaging Social can assume it. For information about how to modify an existing trust relationship of a role, see [Modifying a Role](#) in the [IAM user guide](#). This role is used for both sending events and importing phone numbers from AWS End User Messaging Social into Connect Customer.

To create new IAM polices, do the following:

1. Create a new **permission policy** by following the directions in [Creating policies using the JSON editor](#) in the IAM User Guide.
 - In step 5 use the **permission policy** for the IAM role to allow for publishing to Connect Customer.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOperationsForEventDelivery",
      "Effect": "Allow",
      "Action": [
        "connect:SendIntegrationEvent"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowOperationsForPhoneNumberImport",
      "Effect": "Allow",
      "Action": [
        "connect:ImportPhoneNumber",
        "social-
messaging:GetLinkedWhatsAppBusinessAccountPhoneNumber",
        "social-messaging:TagResource"
      ],
      "Resource": "*"
    }
  ]
}
```

2. Create a new **trust policy** by following the directions in [Creating a role using custom trust policies](#) in the IAM User Guide.
 - a. In step 4 use the **trust policy** for the IAM role.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "social-messaging.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- b. In step 10 add the **permission policy** that you created in the previous step.

Next steps

Once you have set up your Amazon SNS topic, you must subscribe an endpoint to the topic. The endpoint will start to receive messages published to the associated topic. For more information on subscribing to a topic, see [Subscribing to an Amazon SNS topic](#) in the *Amazon SNS Developer Guide*.

Message and event format in AWS End User Messaging Social

The JSON object for an event contains the AWS event header and WhatsApp JSON payload. For a list of the JSON WhatsApp notification payload and values, see [Webhooks Notification Payload Reference](#) and [Message Status](#) in the *WhatsApp Business Platform Cloud API Reference*.

AWS End User Messaging Social event header

The JSON object for an event contains the AWS event header and WhatsApp JSON. The header contains the AWS identifiers and ARNs of your WhatsApp Business Account (WABA) and phone number.

```
{
```

```

"context": {
  "MetaWabaIds": [
    {
      "wabaId": "1234567890abcde",
      "arn": "arn:aws:social-messaging:us-east-1:123456789012:waba/
fb2594b8a7974770b128a409e2example"
    }
  ],
  "MetaPhoneNumberIds": [
    {
      "metaPhoneNumberId": "abcde1234567890",
      "arn": "arn:aws:social-messaging:us-east-1:123456789012:phone-number-
id/976c72a700aac43eaf573ae050example"
    }
  ]
},
"whatsAppWebhookEntry": "{\\"...JSON STRING....",
"aws_account_id": "123456789012",
"message_timestamp": "2025-01-08T23:30:43.271279391Z",
"messageId": "6d69f07a-c317-4278-9d5c-6a84078419ec"
}
//Decoding the contents of whatsAppWebhookEntry
{
//WhatsApp notification payload
}

```

In the preceding example event:

- *1234567890abcde* is the WABA id from Meta.
- *abcde1234567890* is the phone number id from Meta.
- *fb2594b8a7974770b128a409e2example* is the ID of the WhatsApp Business Account (WABA).
- *976c72a700aac43eaf573ae050example* is the ID of the phone number.

Example WhatsApp JSON for receiving a message

The following shows the event record for an incoming message from WhatsApp. The JSON received from WhatsApp in the `whatsAppWebhookEntry` is received as a JSON string and can be converted to JSON. For a list of fields and their meaning, see [Webhooks Notification Payload Reference](#) in the *WhatsApp Business Platform Cloud API Reference*.

```
{
  "context": {
    "MetaWabaIds": [
      {
        "wabaId": "1234567890abcde",
        "arn": "arn:aws:social-messaging:us-east-1:123456789012:waba/fb2594b8a7974770b128a409e2example"
      }
    ],
    "MetaPhoneNumberIds": [
      {
        "metaPhoneNumberId": "abcde1234567890",
        "arn": "arn:aws:social-messaging:us-east-1:123456789012:phone-number-id/976c72a700aac43eaf573ae050example"
      }
    ]
  },
  "whatsAppWebhookEntry": "{\\\"...JSON STRING....\"",
  "aws_account_id": "123456789012",
  "message_timestamp": "2025-01-08T23:30:43.271279391Z",
  "messageId": "6d69f07a-c317-4278-9d5c-6a84078419ec"
}
```

You can use a tool, such as [jq](#), to convert the JSON string to JSON. The following is the `whatsAppWebhookEntry` in JSON form:

```
{
  "id": "503131219501234",
  "changes": [
    {
      "value": {
        "messaging_product": "whatsapp",
        "metadata": {
          "display_phone_number": "14255550123",
          "phone_number_id": "46271669example"
        }
      },
      "statuses": [
        {
          "id": "wamid.HBgLMTkxNzM5OTI3MzkVAgARGBJBMTM4NDdGRENEREI5Rexample",
          "status": "sent",
          "timestamp": "1736379042",
          "recipient_id": "01234567890",
          "conversation": {
```

```
        "id": "62374592e84cb58e52bdaed31example",
        "expiration_timestamp": "1736461020",
        "origin": {
            "type": "utility"
        }
    },
    "pricing": {
        "billable": true,
        "pricing_model": "CBP",
        "category": "utility"
    }
}
],
"field": "messages"
}
]
```

Example WhatsApp JSON for receiving a media message

The following shows the event record for an incoming media message. To retrieve the media file, use the `GetWhatsAppMessageMedia` API command. For a list of fields and their meaning, see [Webhooks Notification Payload Reference](#)

```
{
//AWS End User Messaging Social header
}
//Decoding the contents of whatsAppWebhookEntry
{
  "id": "365731266123456",
  "changes": [
    {
      "value": {
        "messaging_product": "whatsapp",
        "metadata": {
          "display_phone_number": "12065550100",
          "phone_number_id": "321010217760100"
        },
        "contacts": [
          {
            "profile": {
              "name": "Diego"
            }
          }
        ]
      }
    }
  ]
}
```

```

    },
    "wa_id": "12065550102"
  }
],
"messages": [
  {
    "from": "14255550150",
    "id":
"wamid.HBgLMTQyNTY5ODgzMDIVAgASGCBDNzBDRjM5MDU2ODEwMDkwREY4ODBDRE0RjVGRkexample",
    "timestamp": "1723506230",
    "type": "image",
    "image": {
      "mime_type": "image/jpeg",
      "sha256": "BTD0xlqSZ7l02o+/upusiNStlEZhA/urkvKf143Uqjk=",
      "id": "530339869524171"
    }
  }
]
},
"field": "messages"
}
]
}

```

WhatsApp message status

When you send a message, you receive status updates about the message. You have to enable event logging to receive these notifications, see [Message and event destinations in AWS End User Messaging Social](#).

Message statuses

The following table contains possible message statuses.

Status name	Description
accepted	The message has been accepted by WhatsApp for processing.

Status name	Description
deleted	The customer deleted the message, and you should also delete the message if it was downloaded to your server.
delivered	The message was successfully delivered to the customer.
failed	The message failed to send.
Message retries exhausted, dropping message.	The message could not be handed off to WhatsApp within the 180-minute retry period and was dropped.
read	The customer read the message. This status is only sent if the customer has read receipts turned on.
sent	The message has been sent but is still in transit.
warning	The message contains an item that is unavailable or doesn't exist.

Additional resources

For more information, see [Message Status](#) in the *WhatsApp Business Platform Cloud API Reference*.

Uploading media files to send with WhatsApp

When you send or receive a media file, it has to be stored in an Amazon S3 bucket and uploaded or retrieved from WhatsApp. The Amazon S3 bucket must be in the same AWS account and AWS Region as your WhatsApp Business Account (WABA). These directions show how to create an Amazon S3 bucket, upload a file, and build the URL to the file. For more information on Amazon S3 commands, see [Use high-level \(s3\) commands with the AWS CLI](#). For more information on configuring the AWS CLI, see [Configure the AWS CLI](#) in the [AWS Command Line Interface User Guide](#), and [Creating a bucket](#), and [Uploading objects](#) in the [Amazon S3 User Guide](#).

Note

WhatsApp stores media files for 30 days before deleting them, see [Upload Media](#) in the *WhatsApp Business Platform Cloud API Reference*.

You can also create a [presigned URL](#) to the media file. With a presigned URL, you can grant time-limited access to objects and upload them without requiring another party to have AWS security credentials or permissions.

1. To create an Amazon S3 bucket, use the [create-bucket](#) AWS CLI command. At the command line, enter the following command:

```
aws s3api create-bucket --region 'us-east-1' --bucket BucketName
```

In the preceding command:

- Replace *us-east-1* with the AWS Region that your WABA is in.
 - Replace *BucketName* with the name of the new bucket.
2. To copy a file to the Amazon S3 bucket, use the [cp](#) AWS CLI command. At the command line, enter the following command:

```
aws s3 cp SourceFilePathAndName s3://BucketName/FileName
```

In the preceding command:

- Replace *SourceFilePathAndName* with the file path and name of the file to copy.

- Replace *BucketName* with the name of the bucket.
- Replace *FileName* with the name to use for the file.

The url to use when sending is:

```
s3://BucketName/FileName
```

To create a [presigned URL](#), replace the *user input placeholders* with your own information.

```
aws s3 presign s3://amzn-s3-demo-bucket1/mydoc.txt --expires-in 604800 --region af-south-1 --endpoint-url https://s3.af-south-1.amazonaws.com
```

The returned URL will be: `https://amzn-s3-demo-bucket1.s3.af-south-1.amazonaws.com/mydoc.txt?{Headers}`

3. Upload the media file to WhatsApp using the [post-whatsapp-message-media](#) command. On successful completion, the command will return the *{MEDIA_ID}*, which is required for sending the media message.

```
aws socialmessaging post-whatsapp-message-media --origination-phone-number-id {ORIGINATION_PHONE_NUMBER_ID} --source-s3-file bucketName={BUCKET},key={MEDIA_FILE}
```

In the preceding command, do the following:

- Replace *{ORIGINATION_PHONE_NUMBER_ID}* with your phone number's ID.
- Replace *{BUCKET}* with the name of the Amazon S3 bucket.
- Replace *{MEDIA_FILE}* with the name of the media file.

You can also upload using a [presign url](#) by using `--source-s3-presigned-url` instead of `--source-s3-file`. You must add Content-Type in the headers field. If you use both then an `InvalidParameterException` is returned.

```
--source-s3-presigned-url headers={"Name":"Value"},url=https://BUCKET.s3.REGION/MEDIA_FILE
```

- On successful completion the *MEDIA_ID* is returned. The *MEDIA_ID* is used to reference the media file when [sending a media message](#).

Supported media file types and sizes in WhatsApp

When sending or receiving a media message, the file type must be supported and under the maximum file size. For more information, see [Supported Media Types](#) in the *WhatsApp Business Platform Cloud API Reference*.

Media file types

Audio formats

Audio Type	Extension	MIME Type	Max Size
AAC	.aac	audio/aac	16 MB
AMR	.amr	audio/amr	16 MB
MP3	.mp3	audio/mpeg	16 MB
MP4 Audio	.m4a	audio/mp4	16 MB
OGG Audio	.ogg	audio/ogg	16 MB

Document formats

Document Type	Extension	MIME Type	Max Size
Text	.text	text/plain	100 MB
Microsoft Excel	.xls, .xlsx	application/vnd.ms-excel, application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	100 MB
Microsoft Word	.doc, .docx	application/msword, application/vnd.op	100 MB

Document Type	Extension	MIME Type	Max Size
		enxmlformats-officedocument.wordprocessingml.document	
Microsoft PowerPoint	.ppt, .pptx	application/vnd.ms-powerpoint, application/vnd.openxmlformats-officedocument.presentationml.presentation	100 MB
PDF	.pdf	application/pdf	100 MB

Image formats

Image Type	Extension	MIME Type	Max Size
JPEG	.jpeg	image/jpeg	5 MB
PNG	.png	image/png	5 MB

Sticker formats

Sticker Type	Extension	MIME Type	Max Size
Animated sticker	.webp	image/webp	500 KB
Static sticker	.webp	image/webp	100 KB

Video formats

Video Type	Extension	MIME Type	Max Size
3GPP	.3gp	video/3gp	16 MB
MP4 Video	.mp4	video/mp4	16 MB

WhatsApp message types

This topic lists the supported message types and a description of their use. For a list of message types, see [Messages](#) in the *WhatsApp Business Platform Cloud API Reference*.

Message Type	Description
Text	Send a text message or URL to your customer.
Media	Send an audio, document, image, sticker, or video file. You can also send links of the media file.
Reaction	Send an emoji as a reaction to a message, like a thumbs up.
Template	Send a template message.
Location	Send a location.
Contacts	Send a contact card.
Interactive	Send an interactive message.

Additional resources

For a list of WhatsApp message objects, see [Messages](#) in the *WhatsApp Business Platform Cloud API Reference*.

Sending messages through WhatsApp with AWS End User Messaging Social

Before sending a message, you must set up your WhatsApp Business Account (WABA), and your user must opt in to receive messages from you. For more information, see [Obtain permission](#).

When a user messages you, a 24-hour timer called a customer service window starts or refreshes. All message types, except for template messages, can only be sent when a customer service window is open between you and the user. Template messages can be sent at any time, as long as the user has opted in to receive messages from you.

For each message that you send or receive, a message status is generated and sent to the event destination. If your customer has not signed up for WhatsApp, an event is generated with a message status of `fail`. You must turn on a [message and event destination](#) to receive the [message status](#).

For a list of message types, see [Messages](#) in the *WhatsApp Business Platform Cloud API Reference*.

Important

Working with Meta/WhatsApp

- Your use of the WhatsApp Business Solution is subject to the terms and conditions of the [WhatsApp Business Terms of Service](#), the [WhatsApp Business Solution Terms](#), the [WhatsApp Business Messaging Policy](#), the [WhatsApp Messaging Guidelines](#), and all other terms, policies, or guidelines incorporated therein by reference. These might be updated from time to time.
- Meta or WhatsApp may at any time prohibit your use of the WhatsApp Business Solution.
- In connection with your use of the WhatsApp Business Solution, you will not submit any content, information, or data that is subject to safeguarding or limitations on distribution pursuant to applicable laws or regulations.

Note

AWS End User Messaging Social now supports Meta's Marketing Messages (MM) API for marketing template messages. You use the same AWS API `SendWhatsAppMessage` — no

code changes required. To use MM API, accept Meta's terms and conditions through the MM API banner on the WABA details page in the AWS End User Messaging Social console. According to Meta, MM API provides up to 9% higher deliverability, recipient optimization, and conversion tracking.

Topics

- [Example of sending a template message in AWS End User Messaging Social](#)
- [Example of sending a media message in AWS End User Messaging Social](#)

Example of sending a template message in AWS End User Messaging Social

For more information on the types of message templates that can be sent, see [Message template](#) in the *WhatsApp Business Platform Cloud API Reference*. For a list of message types that can be sent, see [Messages](#) in the *WhatsApp Business Platform Cloud API Reference*.

The following example shows how to use a template to [send a message](#) to your customer using the AWS CLI. For more information on configuring the AWS CLI, see [Configure the AWS CLI](#) in the [AWS Command Line Interface User Guide](#).

Note

You must specify base64 encoding when you use the AWS CLI version 2. This can be done by adding the AWS CLI parameter `--cli-binary-format raw-in-base64-out` or changing the AWS CLI global configuration file. For more information, see [cli_binary_format](#) in the *AWS Command Line Interface User Guide for Version 2*.

```
aws socialmessaging send-whatsapp-message --message
 '{"messaging_product":"whatsapp","to":"' {PHONE_NUMBER} ','type":"template","template":
 {"name":"statement","language":{"code":"en_US"},"components":
 [{"type":"body","parameters":[{"type":"text","text":"1000"}]}]}' --origination-phone-
 number-id {ORIGINATION_PHONE_NUMBER_ID} --meta-api-version v20.0
```

In the preceding command, do the following:

- Replace `{PHONE_NUMBER}` with your customer's phone number.
- Replace `{ORIGINATION_PHONE_NUMBER_ID}` with your phone number's ID.

The following example shows how to send a template message that doesn't contain any components.

```
aws socialmessaging send-whatsapp-message --message '{"messaging_product":
  "whatsapp","to": "'{PHONE_NUMBER}'","type": "template","template":
  {"name":"simple_template","language": {"code": "en_US"}}}' --origination-phone-number-
id {ORIGINATION_PHONE_NUMBER_ID} --meta-api-version v20.0
```

- Replace `{PHONE_NUMBER}` with your customer's phone number.
- Replace `{ORIGINATION_PHONE_NUMBER_ID}` with your phone number's ID.

Example of sending a media message in AWS End User Messaging Social

The following example shows how to send a media message to your customer using the AWS CLI. For more information on configuring the AWS CLI, see [Configure the AWS CLI](#) in the [AWS Command Line Interface User Guide](#). For a list of supported media file types, see [Supported media file types and sizes in WhatsApp](#).

Note

WhatsApp stores media files for 30 days before deleting them, see [Upload Media](#) in the *WhatsApp Business Platform Cloud API Reference*.

1. Upload the media file to an Amazon S3 bucket. For more information, see [Uploading media files to send with WhatsApp](#).
2. Upload the media file to WhatsApp using the [post-whatsapp-message-media](#) command. On successful completion, the command will return the `{MEDIA_ID}`, which is required for sending the media message.

```
aws socialmessaging post-whatsapp-message-media --origination-
phone-number-id {ORIGINATION_PHONE_NUMBER_ID} --source-s3-file
bucketName={BUCKET},key={MEDIA_FILE}
```

In the preceding command, do the following:

- Replace `{ORIGINATION_PHONE_NUMBER_ID}` with your phone number's ID.
- Replace `{BUCKET}` with the name of the Amazon S3 bucket.
- Replace `{MEDIA_FILE}` with the name of the media file.

You can also upload using a [presign url](#) by using `--source-s3-presigned-url` instead of `--source-s3-file`. You must add Content-Type in the headers field. If you use both then an `InvalidParameterException` is returned.

```
--source-s3-presigned-url headers={"Name":"Value"},url=https://BUCKET.s3.REGION/
MEDIA_FILE
```

3. Use the [send-whatsapp-message](#) command to send the media message.

```
aws socialmessaging send-whatsapp-message --message
'{"messaging_product":"whatsapp","to":"' {PHONE_NUMBER} "',"type":"image","image":
{"id":"' {MEDIA_ID} '"}' --origination-phone-number-id {ORIGINATION_PHONE_NUMBER_ID}
--meta-api-version v20.0
```

Note

You must specify base64 encoding when you use the AWS CLI version 2. This can be done by adding the AWS CLI parameter `--cli-binary-format raw-in-base64-out` or changing the AWS CLI global configuration file. For more information, see [cli_binary_format](#) in the *AWS Command Line Interface User Guide for Version 2*.

```
aws socialmessaging send-whatsapp-message --message
'{"messaging_product":"whatsapp","to":"' {PHONE_NUMBER} "',"type":"image","image":
{"id":"' {MEDIA_ID} '"}' --origination-phone-number-
id {ORIGINATION_PHONE_NUMBER_ID} --meta-api-version v20.0 --cli-binary-
format raw-in-base64-out
```

In the preceding command, do the following:

- Replace *{PHONE_NUMBER}* with your customer's phone number.
 - Replace *{ORIGINATION_PHONE_NUMBER_ID}* with your phone number's ID.
 - Replace *{MEDIA_ID}* with the media ID returned from the previous step.
4. When you no longer need the media file, you can delete it from WhatsApp using the [delete-whatsapp-message-media](#) command. This only removes the media file from WhatsApp and not your Amazon S3 bucket.

```
aws socialmessaging delete-whatsapp-message-media --media-id {MEDIA_ID} --  
origination-phone-number-id {ORIGINATION_PHONE_NUMBER_ID}
```

In the preceding command, do the following:

- Replace *{ORIGINATION_PHONE_NUMBER_ID}* with your phone number's ID.
- Replace *{MEDIA_ID}* with the media ID.

Responding to a message in AWS End User Messaging Social

Before you can receive a text or media message, you must have set up your WhatsApp Business Account (WABA) and an event destination. When you receive an incoming message, an event is saved in the event destination Amazon SNS topic. To receive a notification, you must subscribe to the Amazon SNS topics endpoint.

For an example event of a received media message, see [Example WhatsApp JSON for receiving a media message](#). For more information on configuring the AWS CLI, see [Configure the AWS CLI](#) in the [AWS Command Line Interface User Guide](#). For a list of supported media file types, see [Supported media file types and sizes in WhatsApp](#).

Important

To receive incoming messages, you must have [event destinations](#) enabled for the WABA. For more information, see [Add a message and event destination to AWS End User Messaging Social](#).

Example of changing a message's status to read in AWS End User Messaging Social

You can set the [status of the message](#) to read to show the end user two blue check marks on their screen.

```
aws socialmessaging send-whatapp-message --message
 '{"messaging_product":"whatsapp","message_id":"' {MESSAGE_ID} ','status":"read"}' --
 origination-phone-number-id {ORIGINATION_PHONE_NUMBER_ID} --meta-api-version v20.0
```

In the preceding command, do the following:

- Replace `{ORIGINATION_PHONE_NUMBER_ID}` with your phone number's ID.
- Replace `{MESSAGE_ID}` with the unique identifier of the message. Use the value of the `id` field in the message object of the Amazon SNS topic.

Example of responding to a message with a reaction in AWS End User Messaging Social

You can add a reaction to the message, like a thumbs up.

```
aws socialmessaging send-whatsapp-message --message
 '{"messaging_product":"whatsapp","recipient_type":"individual","to":"' {PHONE_NUMBER} ','type":
 "reaction","reaction": {"message_id": "' {MESSAGE_ID} ','emoji":"\uD83D\uDC4D"}' --
 origination-phone-number-id {ORIGINATION_PHONE_NUMBER_ID} --meta-api-version v20.0
```

In the preceding command, do the following:

- Replace `{PHONE_NUMBER}` with your customer's phone number.
- Replace `{MESSAGE_ID}` with the unique identifier of the message. Use the value of the `id` field in the message object of the Amazon SNS topic.
- Replace `{ORIGINATION_PHONE_NUMBER_ID}` with your phone number's ID.

Download a media file from WhatsApp to Amazon S3

To retrieve a media file and save it to an Amazon S3 bucket, use the [get-whatsapp-message-media](#) command.

```
aws socialmessaging get-whatsapp-message-media --media-id {MEDIA_ID} --
 origination-phone-number-id {ORIGINATION_PHONE_NUMBER_ID} --destination-s3-file
 bucketName={BUCKET},key=inbound_
 {
   "mimeType": "image/jpeg",
   "fileSize": 78144
 }
```

In the preceding command, do the following:

- Replace `{BUCKET}` with the name of the Amazon S3 bucket.
- Replace `{MEDIA_ID}` with the value of the `id` field from the received event. For an example incoming media event, see [Example WhatsApp JSON for receiving a media message](#).
- Replace `{ORIGINATION_PHONE_NUMBER_ID}` with your phone number's ID.

To retrieve the media from the Amazon S3 bucket, use the following command:

```
aws s3 cp s3://{BUCKET}/inbound_{MEDIA_ID}.jpeg
```

In the preceding command, do the following:

- Replace `{BUCKET}` with the name of the Amazon S3 bucket.
- Replace `{MEDIA_ID}` with the MEDIA_ID returned from the previous step.

Example of responding to a message with a read receipt and reaction

In this example, your customer, Diego, sent you a message saying "Hi" and you respond to him with a read receipt and hand wave emoji.

Prerequisites

To receive a notification that Diego sent a message, you must have set up an event destination Amazon SNS topic and subscribed to a topic endpoint.

Responding

1. When the message from Diego is received, an event is published to the endpoints of the topic. The following is a snippet of what the topic publishes.

Note

Because Diego initiated the conversation, it doesn't count against the quota for your business initiated conversations.

The `whatsAppWebhookEntry` in this example is shown in JSON notation. For an example of converting the `whatsAppWebhookEntry` from JSON sting to JSON, see [Example WhatsApp JSON for receiving a message](#).

```
{
  "context": {
    "MetaWabaIds": [
      {
```

```

    "wabaId": "1234567890abcde",
    "arn": "arn:aws:social-messaging:us-east-1:123456789012:waba/
fb2594b8a7974770b128a409e2example"
  }
],
"MetaPhoneNumberIds": [
  {
    "metaPhoneNumberId": "abcde1234567890",
    "arn": "arn:aws:social-messaging:us-east-1:123456789012:phone-number-
id/976c72a700aac43eaf573ae050example"
  }
]
},
"whatsAppWebhookEntry": "{\\\"...JSON STRING....\",
"aws_account_id": "123456789012",
"message_timestamp": "2025-01-08T23:30:43.271279391Z"
}
//Decoding the contents of whatsAppWebhookEntry
{
  "id": "365731266123456",
  "changes": [
    {
      "value": {
        "messaging_product": "whatsapp",
        "metadata": {
          "display_phone_number": "12065550100",
          "phone_number_id": "321010217712345"
        },
        "contacts": [
          {
            "profile": {
              "name": "Diego"
            },
            "wa_id": "12065550102"
          }
        ],
        "messages": [
          {
            "from": "14255550150",
            "id":
"wamid.HBgLMTQyNTY5ODgzMDIVAgASGCBDNzBDRjM5MDU2ODEwMDkwREY4ODBDRE0RjVGRkexample",
            "timestamp": "1723506035",
            "text": {
              "body": "Hi"
            }
          }
        ]
      }
    }
  ]
}

```

```

        },
        "type": "text"
      }
    ]
  },
  "field": "messages"
}
]
}

```

- To show Diego you received the message, set the status to `read`. Diego will see two blue check marks next to the message on his device.

Note

You must specify base64 encoding when you use the AWS CLI version 2. This can be done by adding the AWS CLI parameter `--cli-binary-format raw-in-base64-out` or changing the AWS CLI global configuration file. For more information, see [cli_binary_format](#) in the *AWS Command Line Interface User Guide for Version 2*.

```

aws socialmessaging send-whatsapp-message --message
 '{"messaging_product":"whatsapp","message_id":"' {MESSAGE_ID} "',"status":"read"}'
 --origination-phone-number-id {ORIGINATION_PHONE_NUMBER_ID} --meta-api-version
 v20.0

```

In the preceding command, do the following:

- Replace `{ORIGINATION_PHONE_NUMBER_ID}` with the phone number ID that Diego sent his message to `phone-number-id-976c72a700aac43eaf573ae050example`.
- Replace `{MESSAGE_ID}` with the unique identifier of the message.

This is the same value of the `id` field in the received message

```
wamid.HBgLMTQyNTY5ODgzMDIVAgASGCBDNzBDRjM5MDU2ODEwMDkwREY4ODBDRE0RjVGRkexa
```

- You can send Diego a hand wave reaction.

```

aws socialmessaging send-whatsapp-message --message
 '{"messaging_product":"whatsapp","recipient_type":"individual","to":"' {PHONE_NUMBER} "',"ty
 "reaction","reaction": {"message_id": "' {MESSAGE_ID} "',"emoji": "\uD83D\uDC4B"}'

```

```
--origination-phone-number-id {ORIGINATION_PHONE_NUMBER_ID} --meta-api-version  
v20.0
```

In the preceding command, do the following:

- Replace *{PHONE_NUMBER}* with Diego's phone number, 14255550150.
- Replace *{MESSAGE_ID}* with the unique identifier of the message.
This is the same value of the `id` field in the received message
`wamid.HBgLMTQyNTY5ODgzMDIVAgASGCBDNzBDRjM5MDU2ODEwMDkwREY4ODBDRDE0RjVGRkexa`
- Replace *{ORIGINATION_PHONE_NUMBER_ID}* with the phone number ID that Diego sent his message to: `phone-number-id-976c72a700aac43eaf573ae050example`.

Additional resources

- Enable [event destinations](#) to log events and receive incoming messages.
- For a list of WhatsApp message objects, see [Messages](#) in the *WhatsApp Business Platform Cloud API Reference*.

Understanding billing and usage reports for AWS End User Messaging Social

You are billed on a per message basis when you use AWS End User Messaging Social. This topic explains the different types of `MetaTemplateMessageFee` and `AWS MessageFee` charges that you can incur per message while using End User Messaging Social.

The `MetaTemplateMessageFees` are set by Meta and subject to change by Meta. AWS will make reasonable efforts to notify you whenever Meta increases the `MetaTemplateMessageFees`.

Effective February 1, 2026, AWS will charge the `MetaTemplateMessageFees` using the rates in Meta's INR rate card, including for volume-tiered pricing. AWS will charge the `MetaTemplateMessageFees` in USD by applying the INR to USD exchange rate as of February 1, 2026. On the first day of each quarter, AWS may update the USD price it charges for the `MetaTemplateMessageFees` to reflect changes to the INR to USD exchange rate.

Charged per message

The AWS End User Messaging Social channel generates a usage type that contains five fields in the following format:

```
Region code-MessagingType-ISO-FeeDescription-FeeType
```

There are two possible billing items for each WhatsApp message: the `MetaTemplateMessageFee` charged by Meta, and the `MessageFee` charged by AWS.

When you send a template message, you are billed for one WhatsApp `MetaTemplateMessageFee` and one AWS `MessageFee`. The following table provides the descriptions and possible values for the fields in the usage type. For more information about AWS End User Messaging Social pricing, see [WhatsApp](#) in AWS End User Messaging Pricing.

Field	Description	Possible values
<i>Region code</i>	The AWS Region prefix that indicates where the WhatsApp message was sent or received from.	<ul style="list-style-type: none"> USE1 – US East (N. Virginia) Region USE2 – US East (Ohio) Region

Field	Description	Possible values
		<ul style="list-style-type: none"> • USW2 – US West (Oregon) Region • APS3 – Asia Pacific (Mumbai) Region • APS5 – Asia Pacific (Hyderabad) Region • APS1 – Asia Pacific (Singapore) Region • APS2 – Asia Pacific (Sydney) Region • APS6 – Asia Pacific (New Zealand) Region • EU – Europe (Ireland) Region • EUW2 – Europe (London) Region • APN1 – Asia Pacific (Tokyo) Region • APN2 – Asia Pacific (Seoul) Region • EUC1 – Europe (Frankfurt) Region • EUN1 – Europe (Stockholm) Region • EUS2 – Europe (Spain) Region • MEC1 – Middle East (UAE) Region • MES1 – Middle East (Bahrain) Region • MXC1 – Mexico (Central) Region

Field	Description	Possible values
		<ul style="list-style-type: none"> • SAE1 – South America (São Paulo) Region • AFS1 – Africa (Cape Town) Region • CAN1 – Canada (Central) Region • CAW1 – Canada West (Calgary) Region
<i>MessagingType</i>	Identifies the message type that the charges are for.	WhatsApp
<i>ISO</i>	The two-digit ISO country code that the message was sent to.	See supported countries for possible values.
<i>FeeDescription</i>	Describes the fee.	<ul style="list-style-type: none"> • MetaTemplateMessageFee charged by Meta¹ • MessageFee charged by End User Messaging Social

Field	Description	Possible values
Meta <i>FeeType</i>	This field displays the MetaTemplateMessageFee fee type for the message.	<p>MetaTemplateMessageFee types for business-initiated messages</p> <ul style="list-style-type: none"> • Authentication – Used to authenticate users with one-time passcodes, which could occur at multiple steps in the login process. This can include account verification, account recovery, and account integrity challenges. • Authentication-International – Used the same as Authentication but your business is eligible for Authentication-International rates, based in another country, and the message was sent on or after the start time for the country. • Marketing – Used to achieve a wide range of goals, from generating awareness to driving sales and retargeting customers. Examples include new product, service, or feature announcements, targeted

Field	Description	Possible values
		<p>promotions/offers, and cart abandonment reminders.</p> <ul style="list-style-type: none"> • Service – Used to resolve customer inquiries. • Utility – Used to follow up on user actions or requests. Examples include opt-in confirmation, order/delivery management (for example a delivery update); account updates or alerts (for example a payment reminder); or feedback surveys. <p>MetaTemplateMessageFee types for user-initiated messages</p> <ul style="list-style-type: none"> • Service – Used to resolve customer inquiries. <p>Volume-tiered pricing is applicable on Authentication, Authentication-International, and Utility message types. For information about volume-tiered</p>

Field	Description	Possible values
		pricing, see Volume-tiered pricing .

Field	Description	Possible values
AWS FeeType	This field displays the AWS MessageFee type for the message.	<p>AWS MessageFee types for business-initiated messages</p> <ul style="list-style-type: none"> • Authentication – Used to authenticate users with one-time passcodes, which could occur at multiple steps in the login process. This can include account verification, account recovery, and account integrity challenges. • Authentication-International – Used the same as Authentication but your business is eligible for Authentication-International rates, based in another country, and the message was sent on or after the start time for the country. • Marketing – for messages used for marketing purposes, from generating awareness to driving sales and retargeting customers. Examples include new product, service, or feature

Field	Description	Possible values
		<p>announcements, targeted promotions/offers, and cart abandonment reminders.</p> <ul style="list-style-type: none"> • Service – for messages related to resolving customer inquiries. • Standard – for free-form messages • Utility – Used to follow up on user actions or requests. Examples include opt-in confirmation, order/delivery management (for example a delivery update); account updates or alerts (for example a payment reminder); or feedback surveys. <p>MessageFee types for user-initiated messages</p> <ul style="list-style-type: none"> • Inbound – for user-initiated messages.

 **Note**

¹The `MetaTemplateMessageFees` are set by Meta and subject to change by Meta. AWS will make reasonable efforts to notify you whenever Meta increases the `MetaTemplateMessageFees`.

When you send a template message, you are billed for one `MetaTemplateMessageFee` and one `AWS MessageFee`. When you send free form text messages, or receive inbound messages, you are billed for one `MessageFee`.

For example, if you send a marketing template message to a customer, you are billed for one `MetaTemplateMessageFee` and one `MessageFee`.

Example billing SKU names

The following are examples of actual billing SKU names. These examples show `AWS MessageFee FeeTypes` for messages sent or received from the `APS3` region code (`ap-south-1`, Asia Pacific (Mumbai)), and sent to the ISO code `IN` (India).

```
APS3-WhatsApp-IN-MessageFee-Authentication
APS3-WhatsApp-IN-MessageFee-Service
APS3-WhatsApp-IN-MessageFee-Authentication_International
APS3-WhatsApp-IN-MessageFee-Marketing
APS3-WhatsApp-IN-MessageFee-Inbound
APS3-WhatsApp-IN-MessageFee-Utility
APS3-WhatsApp-IN-MessageFee-Standard
```

When does the Authentication-International FeeType apply

For a list of countries with an `Authentication-International FeeType`, see [WhatsApp](#) in *AWS End User Messaging Pricing*.

If you send an `Authentication` message to a WhatsApp user whose country calling code has an `Authentication-International FeeType`, you will be billed that country's `Authentication-International` rate if:

1. Your business opens more than 750K messages in a moving 30-day period across all of your WhatsApp Business Accounts with WhatsApp users whose country calling codes are for a country that has an `Authentication-International` rate. For more information, see [Eligibility](#) in the *WhatsApp Business Platform Developer Guide*.

⚠ Important

If Meta determines that your business is eligible for Authentication-International then they will attempt to send you an email notification with applicable countries and moving 30-day period start times.

2. Your business is based in another country. For more information on managing your business's location, see [Primary business location](#) in the *WhatsApp Business Platform Developer Guide*.
3. The message was sent on or after your start time of the 30-day period for that country.

Example 1: Sending a Marketing template message

📘 Note

If you opt in to Meta's Marketing Messages (MM) API, marketing messages sent through MM API continue to appear under the existing marketing message billing category with no changes to your billing configuration.

For example, if you send a marketing template message to a customer, you are billed for one WhatsApp MetaTemplateMessageFee and one AWS MessageFee.

```
Marketing Template Message 1: APS1-WhatsApp-CA-MetaTemplateMessageFee-Marketing
Marketing Template Message 1: APS1-WhatsApp-CA-MessageFee-Marketing
```

Example 2: Opening a service conversation

A service conversation is created when a business responds to a user's inbound message within 24 hours of receiving the user-initiated message. In this scenario, you are billed one MetaTemplateMessageFee-Service_Regular fee and an AWS MessageFee-Service for each outbound message, and billed one MessageFee-Inbound for each user-initiated message. For example, if a user initiated the first message, the business responded, and the user responded back, the charges for the three messages would be the following:

```
User Initiated Message 1: APS1-WhatsApp-CA-MessageFee-Inbound
Business Initiated Message 2: APS1-WhatsApp-CA-MetaTemplateMessageFee-Service_Regular
Business Initiated Message 2: APS1-WhatsApp-CA-MessageFee-Service
```

User Initiated Message 3: APS1-WhatsApp-CA-MessageFee-Inbound

Volume-tiered pricing

AWS End User Messaging Social uses Meta's volume tiered pricing for WhatsApp template messages. When your monthly volume crosses tier thresholds, you benefit from lower per-message rates for Meta message fees for the remainder of that billing period. Currently, volume tiers apply to authentication, authentication-international and utility template messages by country, and do not apply to marketing template messages. The volume-tiered prices, tier thresholds, and discounts are set by Meta, and subject to change by Meta. AWS will make reasonable efforts to notify you of changes to volume tiers.

Key pricing considerations include the following:

- Only template messages count toward volume calculations—non-template messages sent within customer service windows are always free.
- Tiers vary by country and message type (authentication versus utility), and reset monthly.
- Multiple WABAs in your account are aggregated together for tier determination.
- Mid-month tier changes apply discounted rates only to subsequent messages.

Volume tiers on ongoing monthly usage are determined solely by Meta. AWS applies the volume tiers to ongoing usage as and when Meta informs us of tier changes. For more information about tier thresholds and current rates, see [Meta's WhatsApp Business Platform pricing documentation](#).

The following SKU examples show the volume-tiered pricing for Authentication, Authentication-International, and Utility message types.

The following are examples of actual billing SKU names. These examples show Meta FeeType for messages sent or received from the APS3 Region code (ap-south-1, Asia Pacific (Mumbai) Region), and sent to the ISO code IN (India).

```
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_Regular
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_Regular_TIER_1
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_Regular_TIER_2
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_Regular_TIER_3
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_Regular_TIER_4
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_Regular_TIER_5
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_International_Regular
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_International_Regular_TIER_1
```

```

APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_International_Regular_TIER_2
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_International_Regular_TIER_3
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_International_Regular_TIER_4
APS3-WhatsApp-IN-MetaTemplateMessageFee-Authentication_International_Regular_TIER_5
APS3-WhatsApp-IN-MetaTemplateMessageFee-Utility_Regular
APS3-WhatsApp-IN-MetaTemplateMessageFee-Utility_Regular_TIER_1
APS3-WhatsApp-IN-MetaTemplateMessageFee-Utility_Regular_TIER_2
APS3-WhatsApp-IN-MetaTemplateMessageFee-Utility_Regular_TIER_3
APS3-WhatsApp-IN-MetaTemplateMessageFee-Utility_Regular_TIER_4
APS3-WhatsApp-IN-MetaTemplateMessageFee-Utility_Regular_TIER_5

```

AWS End User Messaging Social billing ISO codes and MetaTemplateMessageFee mapping

Supported countries

Two-digit ISO country code	Country name	WhatsApp conversation billing region
AF	Afghanistan	Rest of Asia Pacific
AX	Aland Islands	Other
AL	Albania	Rest of Central & Eastern Europe
DZ	Algeria	Rest of Africa
AS	American Samoa	Other
AD	Andorra	Other
AO	Angola	Rest of Africa
AI	Anguilla	Other
AQ	Antarctica	Other
AG	Antigua and Barbuda	Other
AR	Argentina	Argentina

Two-digit ISO country code	Country name	WhatsApp conversation billing region
AM	Armenia	Rest of Central & Eastern Europe
AW	Aruba	Other
AC	Ascension Island	Other
AU	Australia	Rest of Asia Pacific
AT	Austria	Rest of Western Europe
AZ	Azerbaijan	Rest of Central & Eastern Europe
BS	Bahamas	Other
BH	Bahrain	Rest of Middle East
BD	Bangladesh	Rest of Asia Pacific
BB	Barbados	Other
BY	Belarus	Rest of Central & Eastern Europe
BE	Belgium	Rest of Western Europe
BZ	Belize	Other
BJ	Benin	Rest of Africa
BM	Bermuda	Other
BT	Bhutan	Other
BO	Bolivia	Rest of Latin America
BQ	Bonaire	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
BA	Bosnia and Herzegovina	Other
BW	Botswana	Rest of Africa
BV	Bouvet Island	Other
BR	Brazil	Brazil
IO	British Indian Ocean Territory	Other
VG	British Virgin Islands	Other
BN	Brunei Darussalam	Other
BG	Bulgaria	Rest of Central & Eastern Europe
BF	BurkinaFaso	Rest of Africa
BI	Burundi	Rest of Africa
KH	Cambodia	Rest of Asia Pacific
CM	Cameroon	Rest of Africa
CA	Canada	North America
CV	Cape Verde	Other
KY	Cayman Islands	Other
CF	Central African Republic	Other
TD	Chad	Rest of Africa
CL	Chile	Chile
CN	China	Rest of Asia Pacific

Two-digit ISO country code	Country name	WhatsApp conversation billing region
CX	Christmas Island	Other
CC	Cocos(Keeling) Islands	Other
CO	Colombia	Colombia
KM	Comoros	Other
CK	Cook Islands	Other
CR	Costa Rica	Rest of Latin America
CI	Cote d'Ivoire	Rest of Africa
HR	Croatia	Rest of Central & Eastern Europe
CW	Curacao	Other
CY	Cyprus	Other
CZ	Czech Republic	Rest of Central & Eastern Europe
CD	Democratic Republic of the Congo	Rest of Africa
DK	Denmark	Rest of Western Europe
DJ	Djibouti	Other
DM	Dominica	Other
DO	Dominican Republic	Rest of Latin America
EC	Ecuador	Rest of Latin America
EG	Egypt	Egypt

Two-digit ISO country code	Country name	WhatsApp conversation billing region
SV	El Salvador	Rest of Latin America
GQ	Equatorial Guinea	Other
ER	Eritrea	Rest of Africa
EE	Estonia	Other
ET	Ethiopia	Rest of Africa
SZ	Eswatini	Rest of Africa
FK	Falkland Islands	Other
FO	Faroe Islands	Other
FJ	Fiji	Other
FI	Finland	Rest of Western Europe
FR	France	France
GF	French Guiana	Other
PF	French Polynesia	Other
TF	French Southern Territories	Other
GA	Gabon	Rest of Africa
GM	Gambia	Rest of Africa
GE	Georgia	Rest of Central & Eastern Europe
DE	Germany	Germany
GH	Ghana	Rest of Africa

Two-digit ISO country code	Country name	WhatsApp conversation billing region
GI	Gibraltar	Other
GR	Greece	Rest of Central & Eastern Europe
GL	Greenland	Other
GD	Grenada	Other
GP	Guadeloupe	Other
GU	Guam	Other
GT	Guatemala	Rest of Latin America
GG	Guernsey	Other
GN	Guinea	Other
GW	Guinea-Bissau	Rest of Africa
GY	Guyana	Other
HT	Haiti	Rest of Latin America
HM	Heard and McDonald Islands	Other
HN	Honduras	Rest of Latin America
HK	Hong Kong	Rest of Asia Pacific
HU	Hungary	Rest of Central & Eastern Europe
IS	Iceland	Other
IN	India	India
ID	Indonesia	Indonesia

Two-digit ISO country code	Country name	WhatsApp conversation billing region
IQ	Iraq	Rest of Middle East
IE	Ireland	Rest of Western Europe
IM	Isle of Man	Other
IL	Israel	Israel
IT	Italy	Italy
JM	Jamaica	Rest of Latin America
JP	Japan	Rest of Asia Pacific
JE	Jersey	Other
JO	Jordan	Rest of Middle East
KZ	Kazakhstan	Other
KE	Kenya	Rest of Africa
KI	Kiribati	Other
XK	Kosovo	Other
KW	Kuwait	Rest of Middle East
KG	Kyrgyzstan	Other
LA	Lao PDR	Rest of Asia Pacific
LV	Latvia	Rest of Central & Eastern Europe
LB	Lebanon	Rest of Middle East
LS	Lesotho	Rest of Africa

Two-digit ISO country code	Country name	WhatsApp conversation billing region
LR	Liberia	Rest of Africa
LY	Libya	Rest of Africa
LI	Liechtenstein	Other
LT	Lithuania	Rest of Central & Eastern Europe
LU	Luxembourg	Other
MO	Macao	Other
MK	Macedonia	Rest of Central & Eastern Europe
MG	Madagascar	Rest of Africa
MW	Malawi	Rest of Africa
MY	Malaysia	Malaysia
MV	Maldives	Other
ML	Mali	Rest of Africa
MT	Malta	Other
MH	Marshall Islands	Other
MQ	Martinique	Other
MR	Mauritania	Rest of Africa
MU	Mauritius	Other
YT	Mayotte	Other
MX	Mexico	Mexico

Two-digit ISO country code	Country name	WhatsApp conversation billing region
FM	Micronesia	Other
MD	Moldova	Rest of Central & Eastern Europe
MC	Monaco	Other
MN	Mongolia	Rest of Asia Pacific
ME	Montenegro	Other
MS	Montserrat	Other
MA	Morocco	Rest of Africa
MZ	Mozambique	Rest of Africa
MM	Myanmar	Other
NA	Namibia	Rest of Africa
NR	Nauru	Other
NP	Nepal	Rest of Asia Pacific
NL	Netherlands	Netherlands
NC	New Caledonia	Other
NZ	New Zealand	Rest of Asia Pacific
NI	Nicaragua	Rest of Latin America
NE	Niger	Rest of Africa
NG	Nigeria	Nigeria
NU	Niue	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
NF	Norfolk Island	Other
MP	Northern Mariana Islands	Other
NO	Norway	Rest of Western Europe
OM	Oman	Rest of Middle East
PK	Pakistan	Pakistan
PW	Palau	Other
PS	Palestinian Territory	Other
PA	Panama	Rest of Latin America
PG	Papua New Guinea	Rest of Asia Pacific
PY	Paraguay	Rest of Latin America
PE	Peru	Peru
PH	Philippines	Rest of Asia Pacific
PN	Pitcairn	Other
PL	Poland	Rest of Central & Eastern Europe
PT	Portugal	Rest of Western Europe
PR	Puerto Rico	Rest of Latin America
QA	Qatar	Rest of Middle East
CG	Republic of Congo	Other
RE	Reunion	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
RO	Romania	Rest of Central & Eastern Europe
RU	Russian Federation	Russia
RW	Rwanda	Rest of Africa
SH	Saint Helena	Other
KN	Saint Kitts and Nevis	Other
LC	Saint Lucia	Other
PM	Saint Pierre and Miquelon	Other
VC	Saint Vincent and Grenadines	Other
BL	Saint-Barthelemy	Other
MF	Saint-Martin	Other
WS	Samoa	Other
SM	San Marino	Other
ST	Sao Tome and Principe	Other
SA	Saudi Arabia	Saudi Arabia
SN	Senegal	Rest of Africa
RS	Serbia	Rest of Central & Eastern Europe
SC	Seychelles	Other
SL	Sierra Leone	Rest of Africa
SG	Singapore	Rest of Asia Pacific

Two-digit ISO country code	Country name	WhatsApp conversation billing region
SX	Sint Maarten	Other
SK	Slovakia	Rest of Central & Eastern Europe
SI	Slovenia	Rest of Central & Eastern Europe
SB	Solomon Islands	Other
SO	Somalia	Rest of Africa
ZA	South Africa	South Africa
GS	South Georgia and the South Sandwich Islands	Other
KR	South Korea	Other
SS	South Sudan	Rest of Africa
ES	Spain	Spain
LK	Sri Lanka	Rest of Asia Pacific
SR	Suriname	Other
SJ	Svalbard and Jan Mayen Islands	Other
SE	Sweden	Rest of Western Europe
CH	Switzerland	Rest of Western Europe
TW	Taiwan	Rest of Asia Pacific
TJ	Tajikistan	Rest of Asia Pacific

Two-digit ISO country code	Country name	WhatsApp conversation billing region
TZ	Tanzania	Rest of Africa
TH	Thailand	Rest of Asia Pacific
TL	Timor-Leste	Other
TG	Togo	Rest of Africa
TK	Tokelau	Other
TO	Tonga	Other
TT	Trinidad and Tobago	Other
TA	Tristan da Cunha	Other
TN	Tunisia	Rest of Africa
TR	Turkey	Turkey
TM	Turkmenistan	Rest of Asia Pacific
TC	Turks and Caicos Islands	Other
TV	Tuvalu	Other
UG	Uganda	Rest of Africa
UA	Ukraine	Rest of Central & Eastern Europe
AE	United Arab Emirates	United Arab Emirates
GB	United Kingdom	United Kingdom
US	United States	North America
UY	Uruguay	Rest of Latin America

Two-digit ISO country code	Country name	WhatsApp conversation billing region
UM	US Minor Outlying Islands	Other
UZ	Uzbekistan	Rest of Asia Pacific
VU	Vanuatu	Other
VA	Vatican City State	Other
VE	Venezuela	Rest of Latin America
VN	Vietnam	Rest of Asia Pacific
VI	Virgin Islands	Other
WF	Wallis and Futuna Islands	Other
EH	Western Sahara	Other
YE	Yemen	Rest of Middle East
ZM	Zambia	Rest of Africa
ZW	Zimbabwe	Other

Charged per conversation (Deprecated)

Note

The Charged per conversation billing categories are deprecated as of July 1, 2025 and are no longer in use. This documentation

The AWS End User Messaging Social channel generates a usage type that contains five fields in the following format: *Region code-MessagingType-ISO-FeeDescription-FeeType*. There are two possible billing items for each WhatsApp conversation the WhatsApp ConversationFee, and the AWS per MessageFee.

Note

Starting on July 1 2025, AWS End User Messaging Social, charges per message instead of per conversation for WhatsApp messages, inline with Meta's change. This means that you are charged a `MetaTemplateMessageFee` instead of `ConversationFee` for the Meta fee, in addition to the `AWS MessageFee`. These changes are included in your monthly bills, starting from the bill for July 2025 that you receive in August 2025. For information about billing and usage reports when you are charged per message, see [the section called "Charged per message"](#).

When you initiate a conversation by sending a template message, you are billed for one `WhatsApp ConversationFee` and one `AWS per MessageFee`. This opens a 24-hour window where each message that you send or receive from the same customer is billed as an `AWS per MessageFee`.

The WhatsApp Conversation type and pricing detail can be found at [Conversation-Based Pricing](#) in the *WhatsApp Business Platform Developer Guide*.

The following table displays the possible values and descriptions for the fields in the usage type. For more information about AWS End User Messaging Social pricing, see [WhatsApp](#) in AWS End User Messaging Pricing.

Field	Options	Description
<i>Region code</i>	<ul style="list-style-type: none"> USE1 – US East (N. Virginia) Region USE2 – US East (Ohio) Region USW2 – US West (Oregon) Region APS3 – Asia Pacific (Mumbai) Region APS5 – Asia Pacific (Hyderabad) Region APS1 – Asia Pacific (Singapore) Region 	The AWS Region prefix that indicates where the WhatsApp message was sent or received from.

Field	Options	Description
	<ul style="list-style-type: none"> • APS2 – Asia Pacific (Sydney) Region • EU – Europe (Ireland) Region • EUW2 – Europe (London) Region • APN1 – Asia Pacific (Tokyo) Region • APN2 – Asia Pacific (Seoul) Region • EUC1 – Europe (Frankfurt) Region • EUS2 – Europe (Spain) Region • SAE1 – South America (São Paulo) Region • AFS1 – Africa (Cape Town) Region • CAN1 – Canada (Central) Region 	
<i>MessagingType</i>	WhatsApp	This field identifies the message type being sent.
<i>ISO</i>	See supported countries	The two-digit ISO country code that the message was sent to.
<i>FeeDescription</i>	ConversationFee , MessageFee	This field specifies either the WhatsApp ConversationFee or the AWS per MessageFee .

Field	Options	Description
<i>FeeType</i>	Authentication , Authentication-International ,Marketing , Service, Utility, Standard	<p>This field displays the type of conversation type was used, or specifies standard for the per message fee</p> <p>Business initiated ConversationFee categories</p> <ul style="list-style-type: none"> • Marketing – Used to achieve a wide range of goals, from generating awareness to driving sales and retargeting customers . Examples include new product, service, or feature announcements, targeted promotions/offers, and cart abandonment reminders. • Utility – Used to follow up on user actions or requests. Examples include opt-in confirmation, order/delivery management (for example a delivery update); account updates or alerts (for example a payment reminder); or feedback surveys. • Authentication – Used to authenticate users with one-time passcodes , potentially at multiple steps in the login process (for example account

Field	Options	Description
		<p>verification, account recovery, and integrity challenges).</p> <ul style="list-style-type: none"> • Authentication-International – Used the same as Authentication but your business is eligible for Authentication-International rates, based in another country, and the conversation was opened on or after the start time for the country. • Service – Used to resolve customer inquiries. <p>User initiated ConversationFee categories</p> <ul style="list-style-type: none"> • Service – Used to resolve customer inquiries. <p>MessageFee categories</p> <ul style="list-style-type: none"> • Standard – Per message sent or received fee.

When you initiate a conversation by sending a template message, you are billed for one **ConversationFee** and one **MessageFee**. This opens a 24-hour window where each template message that you send to the same customer is billed as an individual **MessageFee**. During the 24-hour window, the template messages must be the same type or a new conversation is started.

For example, if you send a marketing template message to a customer you are billed for the **ConversationFee** and **MessageFee**.

```
Marketing Template Message 1: APS1-WhatsApp-CA-ConversationFee-Marketing
Marketing Template Message 1: APS1-WhatsApp-CA-MessageFee-Standard
Marketing Template Message 2: APS1-WhatsApp-CA-MessageFee-Standard
```

If the customer sends you a message and you respond, then you are billed for opening a new Service conversation and message.

```
Service Message 1: APS1-WhatsApp-CA-ConversationFee-Service
Service Message 1: APS1-WhatsApp-CA-MessageFee-Standard
Service Message 2: APS1-WhatsApp-CA-MessageFee-Standard
Service Message 3: APS1-WhatsApp-CA-MessageFee-Standard
```

When does the Authentication-International FeeType apply

For a list of countries with an Authentication-International FeeType, see [WhatsApp](#) in AWS End User Messaging Pricing.

If you open an Authentication conversation with a WhatsApp user whose country calling code has an Authentication-International FeeType, you will be billed that country's Authentication-International rate if:

1. Your business opens more than 750K conversations in a moving 30-day period across all of your WhatsApp Business Accounts with WhatsApp users whose country calling codes are for a country that has an Authentication-International rate. For more information, see [Eligibility](#) in the *WhatsApp Business Platform Developer Guide*.

Important

If Meta determines that your business is eligible for Authentication-International then they will attempt to send you an email notification with applicable countries and moving 30-day period start times.

2. Your business is based in another country. For more information on managing your business's location, see [Primary business location](#) in the *WhatsApp Business Platform Developer Guide*.
3. The conversation was opened on or after your start time for that country

Example 1: Sending a Marketing template message

For example, if you send a marketing template message to a customer, you are billed for one WhatsApp ConversationFee and one AWS per MessageFee.

```
Marketing Template Message 1: APS1-WhatsApp-CA-ConversationFee-Marketing
Marketing Template Message 1: APS1-WhatsApp-CA-MessageFee-Standard
```

Example 2: Opening a Service conversation

A service conversation fee applies when a business responds to a user's inbound message that falls outside of any active 24-hour conversation window initiated by the business. In this scenario, you are billed one WhatsApp ConversationFee and an AWS MessageFee for each inbound and outbound message.

```
Service Message 1: APS1-WhatsApp-CA-ConversationFee-Service
Service Message 1: APS1-WhatsApp-CA-MessageFee-Standard
Service Message 2: APS1-WhatsApp-CA-MessageFee-Standard
Service Message 3: APS1-WhatsApp-CA-MessageFee-Standard
```

AWS End User Messaging Social billing ISO codes and WhatsApp Conversation Fee mapping

Supported countries

Two-digit ISO country code	Country name	WhatsApp conversation billing region
AF	Afghanistan	Rest of Asia Pacific
AX	Aland Islands	Other
AL	Albania	Rest of Central & Eastern Europe
DZ	Algeria	Rest of Africa
AS	American Samoa	Other
AD	Andorra	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
AO	Angola	Rest of Africa
AI	Anguilla	Other
AQ	Antarctica	Other
AG	Antigua and Barbuda	Other
AR	Argentina	Argentina
AM	Armenia	Rest of Central & Eastern Europe
AW	Aruba	Other
AC	Ascension Island	Other
AU	Australia	Rest of Asia Pacific
AT	Austria	Rest of Western Europe
AZ	Azerbaijan	Rest of Central & Eastern Europe
BS	Bahamas	Other
BH	Bahrain	Rest of Middle East
BD	Bangladesh	Rest of Asia Pacific
BB	Barbados	Other
BY	Belarus	Rest of Central & Eastern Europe
BE	Belgium	Rest of Western Europe
BZ	Belize	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
BJ	Benin	Rest of Africa
BM	Bermuda	Other
BT	Bhutan	Other
BO	Bolivia	Rest of Latin America
BQ	Bonaire	Other
BA	Bosnia and Herzegovina	Other
BW	Botswana	Rest of Africa
BV	Bouvet Island	Other
BR	Brazil	Brazil
IO	British Indian Ocean Territory	Other
VG	British Virgin Islands	Other
BN	Brunei Darussalam	Other
BG	Bulgaria	Rest of Central & Eastern Europe
BF	BurkinaFaso	Rest of Africa
BI	Burundi	Rest of Africa
KH	Cambodia	Rest of Asia Pacific
CM	Cameroon	Rest of Africa
CA	Canada	North America
CV	Cape Verde	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
KY	Cayman Islands	Other
CF	Central African Republic	Other
TD	Chad	Rest of Africa
CL	Chile	Chile
CN	China	Rest of Asia Pacific
CX	Christmas Island	Other
CC	Cocos(Keeling) Islands	Other
CO	Colombia	Colombia
KM	Comoros	Other
CK	Cook Islands	Other
CR	Costa Rica	Rest of Latin America
CI	Cote d'Ivoire	Rest of Africa
HR	Croatia	Rest of Central & Eastern Europe
CW	Curacao	Other
CY	Cyprus	Other
CZ	Czech Republic	Rest of Central & Eastern Europe
CD	Democratic Republic of the Congo	Rest of Africa
DK	Denmark	Rest of Western Europe

Two-digit ISO country code	Country name	WhatsApp conversation billing region
DJ	Djibouti	Other
DM	Dominica	Other
DO	Dominican Republic	Rest of Latin America
EC	Ecuador	Rest of Latin America
EG	Egypt	Egypt
SV	El Salvador	Rest of Latin America
GQ	Equatorial Guinea	Other
ER	Eritrea	Rest of Africa
EE	Estonia	Other
ET	Ethiopia	Rest of Africa
SZ	Eswatini	Rest of Africa
FK	Falkland Islands	Other
FO	Faroe Islands	Other
FJ	Fiji	Other
FI	Finland	Rest of Western Europe
FR	France	France
GF	French Guiana	Other
PF	French Polynesia	Other
TF	French Southern Territories	Other
GA	Gabon	Rest of Africa

Two-digit ISO country code	Country name	WhatsApp conversation billing region
GM	Gambia	Rest of Africa
GE	Georgia	Rest of Central & Eastern Europe
DE	Germany	Germany
GH	Ghana	Rest of Africa
GI	Gibraltar	Other
GR	Greece	Rest of Central & Eastern Europe
GL	Greenland	Other
GD	Grenada	Other
GP	Guadeloupe	Other
GU	Guam	Other
GT	Guatemala	Rest of Latin America
GG	Guernsey	Other
GN	Guinea	Other
GW	Guinea-Bissau	Rest of Africa
GY	Guyana	Other
HT	Haiti	Rest of Latin America
HM	Heard and McDonald Islands	Other
HN	Honduras	Rest of Latin America
HK	Hong Kong	Rest of Asia Pacific

Two-digit ISO country code	Country name	WhatsApp conversation billing region
HU	Hungary	Rest of Central & Eastern Europe
IS	Iceland	Other
IN	India	India
ID	Indonesia	Indonesia
IQ	Iraq	Rest of Middle East
IE	Ireland	Rest of Western Europe
IM	Isle of Man	Other
IL	Israel	Israel
IT	Italy	Italy
JM	Jamaica	Rest of Latin America
JP	Japan	Rest of Asia Pacific
JE	Jersey	Other
JO	Jordan	Rest of Middle East
KZ	Kazakhstan	Other
KE	Kenya	Rest of Africa
KI	Kiribati	Other
XK	Kosovo	Other
KW	Kuwait	Rest of Middle East
KG	Kyrgyzstan	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
LA	Lao PDR	Rest of Asia Pacific
LV	Latvia	Rest of Central & Eastern Europe
LB	Lebanon	Rest of Middle East
LS	Lesotho	Rest of Africa
LR	Liberia	Rest of Africa
LY	Libya	Rest of Africa
LI	Liechtenstein	Other
LT	Lithuania	Rest of Central & Eastern Europe
LU	Luxembourg	Other
MO	Macao	Other
MK	Macedonia	Rest of Central & Eastern Europe
MG	Madagascar	Rest of Africa
MW	Malawi	Rest of Africa
MY	Malaysia	Malaysia
MV	Maldives	Other
ML	Mali	Rest of Africa
MT	Malta	Other
MH	Marshall Islands	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
MQ	Martinique	Other
MR	Mauritania	Rest of Africa
MU	Mauritius	Other
YT	Mayotte	Other
MX	Mexico	Mexico
FM	Micronesia	Other
MD	Moldova	Rest of Central & Eastern Europe
MC	Monaco	Other
MN	Mongolia	Rest of Asia Pacific
ME	Montenegro	Other
MS	Montserrat	Other
MA	Morocco	Rest of Africa
MZ	Mozambique	Rest of Africa
MM	Myanmar	Other
NA	Namibia	Rest of Africa
NR	Nauru	Other
NP	Nepal	Rest of Asia Pacific
NL	Netherlands	Netherlands
NC	New Caledonia	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
NZ	New Zealand	Rest of Asia Pacific
NI	Nicaragua	Rest of Latin America
NE	Niger	Rest of Africa
NG	Nigeria	Nigeria
NU	Niue	Other
NF	Norfolk Island	Other
MP	Northern Mariana Islands	Other
NO	Norway	Rest of Western Europe
OM	Oman	Rest of Middle East
PK	Pakistan	Pakistan
PW	Palau	Other
PS	Palestinian Territory	Other
PA	Panama	Rest of Latin America
PG	Papua New Guinea	Rest of Asia Pacific
PY	Paraguay	Rest of Latin America
PE	Peru	Peru
PH	Philippines	Rest of Asia Pacific
PN	Pitcairn	Other
PL	Poland	Rest of Central & Eastern Europe

Two-digit ISO country code	Country name	WhatsApp conversation billing region
PT	Portugal	Rest of Western Europe
PR	Puerto Rico	Rest of Latin America
QA	Qatar	Rest of Middle East
CG	Republic of Congo	Other
RE	Reunion	Other
RO	Romania	Rest of Central & Eastern Europe
RU	Russian Federation	Russia
RW	Rwanda	Rest of Africa
SH	Saint Helena	Other
KN	Saint Kitts and Nevis	Other
LC	Saint Lucia	Other
PM	Saint Pierre and Miquelon	Other
VC	Saint Vincent and Grenadines	Other
BL	Saint-Barthelemy	Other
MF	Saint-Martin	Other
WS	Samoa	Other
SM	San Marino	Other
ST	Sao Tome and Principe	Other
SA	Saudi Arabia	Saudi Arabia

Two-digit ISO country code	Country name	WhatsApp conversation billing region
SN	Senegal	Rest of Africa
RS	Serbia	Rest of Central & Eastern Europe
SC	Seychelles	Other
SL	Sierra Leone	Rest of Africa
SG	Singapore	Rest of Asia Pacific
SX	Sint Maarten	Other
SK	Slovakia	Rest of Central & Eastern Europe
SI	Slovenia	Rest of Central & Eastern Europe
SB	Solomon Islands	Other
SO	Somalia	Rest of Africa
ZA	South Africa	South Africa
GS	South Georgia and the South Sandwich Islands	Other
KR	South Korea	Other
SS	South Sudan	Rest of Africa
ES	Spain	Spain
LK	Sri Lanka	Rest of Asia Pacific
SR	Suriname	Other

Two-digit ISO country code	Country name	WhatsApp conversation billing region
SJ	Svalbard and Jan Mayen Islands	Other
SE	Sweden	Rest of Western Europe
CH	Switzerland	Rest of Western Europe
TW	Taiwan	Rest of Asia Pacific
TJ	Tajikistan	Rest of Asia Pacific
TZ	Tanzania	Rest of Africa
TH	Thailand	Rest of Asia Pacific
TL	Timor-Leste	Other
TG	Togo	Rest of Africa
TK	Tokelau	Other
TO	Tonga	Other
TT	Trinidad and Tobago	Other
TA	Tristan da Cunha	Other
TN	Tunisia	Rest of Africa
TR	Turkey	Turkey
TM	Turkmenistan	Rest of Asia Pacific
TC	Turks and Caicos Islands	Other
TV	Tuvalu	Other
UG	Uganda	Rest of Africa

Two-digit ISO country code	Country name	WhatsApp conversation billing region
UA	Ukraine	Rest of Central & Eastern Europe
AE	United Arab Emirates	United Arab Emirates
GB	United Kingdom	United Kingdom
US	United States	North America
UY	Uruguay	Rest of Latin America
UM	US Minor Outlying Islands	Other
UZ	Uzbekistan	Rest of Asia Pacific
VU	Vanuatu	Other
VA	Vatican City State	Other
VE	Venezuela	Rest of Latin America
VN	Vietnam	Rest of Asia Pacific
VI	Virgin Islands	Other
WF	Wallis and Futuna Islands	Other
EH	Western Sahara	Other
YE	Yemen	Rest of Middle East
ZM	Zambia	Rest of Africa
ZW	Zimbabwe	Other

Monitoring AWS End User Messaging Social

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS End User Messaging Social and your other AWS solutions. AWS provides the following monitoring tools to watch AWS End User Messaging Social, report when something is wrong, and take automatic actions when appropriate:

- *Amazon CloudWatch* monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the [Amazon CloudWatch User Guide](#).
- *Amazon CloudWatch Logs* enables you to monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the [Amazon CloudWatch Logs User Guide](#).
- *Amazon EventBridge* can be used to automate your AWS services and respond automatically to system events, such as application availability issues or resource changes. Events from AWS services are delivered to EventBridge in near real time. You can write simple rules to indicate which events are of interest to you and which automated actions to take when an event matches a rule. For more information, see [Amazon EventBridge User Guide](#).
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the [AWS CloudTrail User Guide](#).

Monitoring AWS End User Messaging Social with Amazon CloudWatch

You can monitor AWS End User Messaging Social using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. You can also set alarms that watch for certain thresholds, and send

notifications or take actions when those thresholds are met. For more information, see the [Amazon CloudWatch User Guide](#).

For AWS End User Messaging Social, consider monitoring the `MetaTemplateMessageFeeCount` and `WhatsAppMessageFeeCount` metrics, and trigger an alarm when a spend threshold has been reached.

Note

Before you can use the CloudWatch metrics you must [create a service-link role](#).

The following tables list the metrics and dimensions that AWS End User Messaging Social exports to the `AWS/SocialMessaging` namespace.

Metric	Unit	Description
<code>WhatsAppMessageFeeCount</code>	Count	The count of WhatsApp message fees
<code>MetaTemplateMessageFeeCount</code>	Count	The count of Meta template message fees

Dimension	Description
<code>MessageFeeType</code>	Valid fee types are Authentication, Authentication_International, Inbound, Marketing, Service,Standard, Utility
<code>DestinationCountryCode</code>	The two letter ISO code for the country
<code>WhatsAppPhoneNumberArn</code>	The arn of the phone number
<code>MetaTemplateMessageFeeType</code>	Valid fee types are regular, free_customer_service, free_entry_point

Dimension	Description
MetaTemplateMessageFeeCategory	Valid fee categories are service, marketing , utility, authentication, authentication_international

Logging AWS End User Messaging Social API calls using AWS CloudTrail

AWS End User Messaging Social is integrated with [AWS CloudTrail](#), a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls for AWS End User Messaging Social as events. The calls captured include calls from the AWS End User Messaging Social console and code calls to the AWS End User Messaging Social API operations. Using the information collected by CloudTrail, you can determine the request that was made to AWS End User Messaging Social, the IP address from which the request was made, when it was made, and additional details.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user or user credentials.
- Whether the request was made on behalf of an IAM Identity Center user.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

CloudTrail is active in your AWS account when you create the account and you automatically have access to the CloudTrail **Event history**. The CloudTrail **Event history** provides a viewable, searchable, downloadable, and immutable record of the past 90 days of recorded management events in an AWS Region. For more information, see [Working with CloudTrail Event history](#) in the *AWS CloudTrail User Guide*. There are no CloudTrail charges for viewing the **Event history**.

For an ongoing record of events in your AWS account past 90 days, create a trail or a [CloudTrail Lake](#) event data store.

CloudTrail trails

A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. All trails created using the AWS Management Console are multi-Region. You can create a single-Region or a multi-Region trail by using the AWS CLI. Creating a multi-Region trail is recommended because you capture activity in all AWS Regions in your account. If you create a single-Region trail, you can view only the events logged in the trail's AWS Region. For more information about trails, see [Creating a trail for your AWS account](#) and [Creating a trail for an organization](#) in the *AWS CloudTrail User Guide*.

You can deliver one copy of your ongoing management events to your Amazon S3 bucket at no charge from CloudTrail by creating a trail, however, there are Amazon S3 storage charges. For more information about CloudTrail pricing, see [AWS CloudTrail Pricing](#). For information about Amazon S3 pricing, see [Amazon S3 Pricing](#).

CloudTrail Lake event data stores

CloudTrail Lake lets you run SQL-based queries on your events. CloudTrail Lake converts existing events in row-based JSON format to [Apache ORC](#) format. ORC is a columnar storage format that is optimized for fast retrieval of data. Events are aggregated into *event data stores*, which are immutable collections of events based on criteria that you select by applying [advanced event selectors](#). The selectors that you apply to an event data store control which events persist and are available for you to query. For more information about CloudTrail Lake, see [Working with AWS CloudTrail Lake](#) in the *AWS CloudTrail User Guide*.

CloudTrail Lake event data stores and queries incur costs. When you create an event data store, you choose the [pricing option](#) you want to use for the event data store. The pricing option determines the cost for ingesting and storing events, and the default and maximum retention period for the event data store. For more information about CloudTrail pricing, see [AWS CloudTrail Pricing](#).

AWS End User Messaging Social data events in CloudTrail

[Data events](#) provide information about the resource operations performed on or in a resource (for example, reading or writing to an Amazon S3 object). These are also known as data plane operations. Data events are often high-volume activities. By default, CloudTrail doesn't log data events. The CloudTrail **Event history** doesn't record data events.

Additional charges apply for data events. For more information about CloudTrail pricing, see [AWS CloudTrail Pricing](#).

You can log data events for the AWS End User Messaging Social resource types by using the CloudTrail console, AWS CLI, or CloudTrail API operations. For more information about how to log data events, see [Logging data events with the AWS Management Console](#) and [Logging data events with the AWS Command Line Interface](#) in the *AWS CloudTrail User Guide*.

The following table lists the AWS End User Messaging Social resource types for which you can log data events. The **Data event type (console)** column shows the value to choose from the **Data event type** list on the CloudTrail console. The **resources.type value** column shows the `resources.type` value, which you would specify when configuring advanced event selectors using the AWS CLI or CloudTrail APIs. The **Data APIs logged to CloudTrail** column shows the API calls logged to CloudTrail for the resource type.

Data event type (console)	resources.type value	Data APIs logged to CloudTrail
Social-Messaging Phone Number ID	<code>AWS::SocialMessaging::PhoneNumberId</code>	<ul style="list-style-type: none"> • DeleteWhatsAppMessageMedia • GetWhatsAppMessageMedia • PostWhatsAppMessageMedia • SendWhatsAppMessage

You can configure advanced event selectors to filter on the `eventName`, `readOnly`, and `resources.ARN` fields to log only those events that are important to you. For more information about these fields, see [AdvancedFieldSelector](#) in the *AWS CloudTrail API Reference*.

AWS End User Messaging Social management events in CloudTrail

[Management events](#) provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations. By default, CloudTrail logs management events.

AWS End User Messaging Social logs all AWS End User Messaging Social control plane operations as management events. For a list of the AWS End User Messaging Social control plane operations that AWS End User Messaging Social logs to CloudTrail, see the [AWS End User Messaging Social API Reference](#).

AWS End User Messaging Social event examples

An event represents a single request from any source and includes information about the requested API operation, the date and time of the operation, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so events don't appear in any specific order.

The following example shows a CloudTrail event that demonstrates the operation.

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "GR632462JDSBDSHHGS39:session",
    "arn": "arn:aws:sts::123456789101:assumed-role/Role_name/Session_name",
    "accountId": "123456789101",
    "accessKeyId": "12345678901234567890",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "GR632462JDSBDEXAMPLE",
        "arn": "arn:aws:sts::123456789101:assumed-role/Role_name/
Session_name",
        "accountId": "123456789101",
        "userName": "user"
      },
      "attributes": {
        "creationDate": "2024-10-03T17:25:08Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2024-10-03T17:25:23Z",
  "eventSource": "social-messaging.amazonaws.com",
  "eventName": "SendWhatsAppMessage",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "1.x.x.x",
```

```
    "userAgent": "agent",
    "requestParameters": {
      "originationPhoneNumberId": "phone-number-id-
aa012345678901234567890123456789",
      "metaApiVersion": "v20.0",
      "message": "Hi"
    },
    "responseElements": {
      "messageId": "message_id"
    },
    "requestID": "request_id",
    "eventID": "event_id",
    "readOnly": false,
    "resources": [{
      "accountId": "123456789101",
      "type": "AWS::SocialMessaging::PhoneNumberId",
      "ARN": "arn:aws:social-messaging:us-east-1:123456789101:phone-number-id/
phone-number-id-aa012345678901234567890123456789"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789101",
    "eventCategory": "Data",
    "tlsDetails": {
      "clientProvidedHostHeader": "social-messaging.us-east-1.amazonaws.com"
    }
  }
}
```

For information about CloudTrail record contents, see [CloudTrail record contents](#) in the *AWS CloudTrail User Guide*.

Monitoring with Amazon EventBridge

AWS End User Messaging Social sends events to EventBridge for WhatsApp events. You can use EventBridge to write rules that take actions, such as notifying you, when specific event types are received. For more information, see the following topics in the Amazon EventBridge User Guide:

- [What is Amazon EventBridge?](#)
- [Amazon EventBridge events](#)
- [Amazon EventBridge event patterns](#)
- [Amazon EventBridge rules](#)

AWS End User Messaging Social sends the following events to Amazon EventBridge:

- WhatsApp Message Delivered
- WhatsApp Message Failed
- WhatsApp Message Handover Failed
- WhatsApp Message Handover Succeeded
- WhatsApp Message Read
- WhatsApp Message Received
- WhatsApp Message Sent
- WhatsApp Message Undeliverable
- WhatsApp Unknown Notification Received

Event for a delivered WhatsApp message

```
{
  "version": "0",
  "id": "76a454c4-26df-6e2c-bc50-2137de2a0fe7",
  "detail-type": "WhatsApp Message Delivered",
  "source": "aws.social-messaging",
  "account": "111122223333",
  "time": "2025-10-03T04:34:24Z",
  "region": "ca-central-1",
  "resources": [],
  "detail": {
    "wabaId": "1163793972139053",
    "originationPhoneNumber": "15815555555",
    "originationPhoneNumberId": "574086085794038",
    "messageId": "4bee7d29-70f5-4a49-b84e-facf4424fe58",
    "status": "delivered",
    "eventTimestamp": "1759466063"
  }
}
```

Event for a read WhatsApp message

```
{
  "version": "0",
  "id": "c730142c-96d6-28f6-00d4-1a8a12e509da",
```

```

"detail-type": "WhatsApp Message Read",
"source": "aws.social-messaging",
"account": "111122223333",
"time": "2025-10-03T04:53:07Z",
"region": "ca-central-1",
"resources": [],
"detail": {
  "wabaId": "1163793972139053",
  "originationPhoneNumber": "15815555555",
  "originationPhoneNumberId": "574086085794038",
  "messageId": "e16d4108-fe93-471f-a057-4c6473e68430",
  "status": "read",
  "eventTimestamp": "1759467185"
}
}

```

Event for a sent WhatsApp message

```

{
  "version": "0",
  "id": "09b261c5-5c4b-b912-afa8-845214926b2f",
  "detail-type": "WhatsApp Message Sent",
  "source": "aws.social-messaging",
  "account": "111122223333",
  "time": "2025-10-03T04:26:16Z",
  "region": "ca-central-1",
  "resources": [],
  "detail": {
    "wabaId": "1163793972139053",
    "originationPhoneNumber": "15815555555",
    "originationPhoneNumberId": "574086085794038",
    "messageId": "a8dec257-b52e-4acc-8654-3dfb96468a43",
    "status": "sent",
    "eventTimestamp": "1759465574"
  }
}

```

Event for a failed WhatsApp message

```

{
  "version": "0",
  "id": "281b8c6b-f39f-f768-4229-d66e893592c1",

```

```
"detail-type": "WhatsApp Message Failed",
"source": "aws.social-messaging",
"account": "111122223333",
"time": "2025-10-03T04:49:25Z",
"region": "ca-central-1",
"resources": [],
"detail": {
  "wabaId": "1163793972139053",
  "originationPhoneNumber": "15815555555",
  "originationPhoneNumberId": "574086085794038",
  "messageId": "7bd36188-6123-4ded-a1dd-f017f0814405",
  "status": "failed",
  "eventTimestamp": "1759466965",
  "errors": [
    {
      "code": 131026,
      "title": "Message undeliverable",
      "message": "Message undeliverable"
    }
  ]
}
```

Best practices for AWS End User Messaging Social

This section describes several best practices that might help you improve your customer engagement and avoid account suspension. However, note that this section doesn't contain legal advice. Always consult an attorney to obtain legal advice.

For the most recent list of WhatsApp best practices, see the [WhatsApp Business Messaging Policy](#).

Topics

- [Up-to-date business profile](#)
- [Obtain permission](#)
- [Prohibited message content](#)
- [Audit your customer lists](#)
- [Adjust your sending based on engagement](#)
- [Send at appropriate times](#)

Up-to-date business profile

Maintain an accurate and up-to-date WhatsApp Business profile that includes customer support contact information, such as an email address, website address, or telephone number. Ensure that the information provided is truthful and does not misrepresent or impersonate another business.

Obtain permission

Never send messages to recipients who haven't explicitly asked to receive the specific types of messages that you plan to send. Maintain the following opt-in information:

- The opt-in process must clearly inform the person that they are consenting to receive messages or calls from your business over WhatsApp. You must explicitly state the name of your business.
- You are solely responsible for determining the method of obtaining opt-in consent. Ensure that the opt-in process complies with all applicable laws governing your communications. Provide all required notices and obtain all necessary permissions under relevant laws.

For more information on WhatsApp Opt-in requirements, see [Get Opt-in for WhatsApp](#)

If recipients can sign up to receive your messages by using an online form, prevent automated scripts from subscribing people without their knowledge. Also limit the number of times a user can submit a phone number in a single session.

Respect all requests made by a person, whether on or off WhatsApp, to block, discontinue, or otherwise opt out of communications, including removing that person from your contacts list.

Maintain records that include the date, time, and source of each opt-in request and confirmation. This can also help you perform routine audits of your customer list.

Prohibited message content

Important

Working with Meta/WhatsApp

- Your use of the WhatsApp Business Solution is subject to the terms and conditions of the [WhatsApp Business Terms of Service](#), the [WhatsApp Business Solution Terms](#), the [WhatsApp Business Messaging Policy](#), the [WhatsApp Messaging Guidelines](#), and all other terms, policies, or guidelines incorporated therein by reference (as each may be updated from time to time).
- Meta or WhatsApp may at any time prohibit your use of the WhatsApp Business Solution.
- In connection with your use of the WhatsApp Business Solution, you will not submit any content, information, or data that is subject to safeguarding or limitations on distribution according to applicable laws or regulation.

If you violate the WhatsApp policy your account could be blocked from sending messages for a period of time, locked until you file an appeal, or permanently blocked. Meta will inform you if any of your accounts or assets have violated the policy, through email and the WhatsApp Business Manager. All appeals must be made to Meta. To view a policy violation or file an appeal with Meta, see [View policy violation details for your WhatsApp Business account](#) in the *Meta Business Help Center*. For the most recent list of prohibited message content, see the [WhatsApp Business Messaging Policy](#).

The following are prohibited content categories for all message types globally. When sending a message with WhatsApp, follow these guidelines:

Category	Examples
Gambling	<ul style="list-style-type: none"> • Casinos • Sweepstakes • App/Websites
High-risk financial services	<ul style="list-style-type: none"> • Payday loans • Short-term high-interest loans • Auto loans • Mortgage loans • Student loans • Debt collection • Stock alerts • Cryptocurrency
Debt forgiveness	<ul style="list-style-type: none"> • Debt consolidation • Debt reduction • Credit repair programs
Get-rich-quick schemes	<ul style="list-style-type: none"> • Work-from-home programs • Risk-investment opportunities • Pyramid or multi-level marketing schemes
Illegal substances	<ul style="list-style-type: none"> • Cannabis/CBD
Phishing/smishing	<ul style="list-style-type: none"> • Attempts to get users to reveal personal information or website login information.
S.H.A.F.T.	<ul style="list-style-type: none"> • Sex • Hate • Alcohol • Firearms • Tobacco/Vape

Category	Examples
Third-Party Lead Generation	<ul style="list-style-type: none">• Companies that buy, sell, or share consumer information

Audit your customer lists

If you send recurring WhatsApp messages, audit your customer lists on a regular basis. Auditing your customer lists helps to make sure that the only customers who receive your messages are those who want to receive them.

When you audit your list, send each opted-in customer a message that reminds them that they're subscribed, and provides them with information about unsubscribing.

Adjust your sending based on engagement

Your customers' priorities can change over time. If customers no longer find your messages to be useful, they might opt out of your messages entirely, or even report your messages as unsolicited. For these reasons, it's important that you adjust your sending practices based on customer engagement.

For customers who rarely engage with your messages, you should adjust the frequency of your messages. For example, if you send weekly messages to engaged customers, you could create a separate monthly digest for customers who are less engaged.

Finally, remove customers who are completely unengaged from your customer lists. This step prevents customers from becoming frustrated with your messages. It also saves you money and helps protect your reputation as a sender.

Send at appropriate times

Send messages during normal daytime business hours. If you send messages at dinner time or in the middle of the night, there's a good chance that your customers will unsubscribe from your lists to avoid being disturbed. You might want to avoid sending WhatsApp messages when your customers can't respond to them immediately.

Security in AWS End User Messaging Social

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS Compliance Programs](#). To learn about the compliance programs that apply to AWS End User Messaging Social, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS End User Messaging Social. The following topics show you how to configure AWS End User Messaging Social to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS End User Messaging Social resources.

Topics

- [Data protection in AWS End User Messaging Social](#)
- [Identity and access management for AWS End User Messaging Social](#)
- [Compliance validation for AWS End User Messaging Social](#)
- [Resilience in AWS End User Messaging Social](#)
- [Infrastructure Security in AWS End User Messaging Social](#)
- [Cross-service confused deputy prevention](#)
- [Security best practices](#)
- [Using service-linked roles for AWS End User Messaging Social](#)

Data protection in AWS End User Messaging Social

The AWS [shared responsibility model](#) applies to data protection in AWS End User Messaging Social. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see [Data Privacy FAQ](#). For information about data protection in Europe, see the [General Data Protection Regulation \(GDPR\) Center](#).

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see [Working with CloudTrail trails](#) in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-3](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with AWS End User Messaging Social or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

⚠ Important

WhatsApp uses the Signal protocol for secure communications. However, because AWS End User Messaging Social is a third party, WhatsApp does not consider these messages end-to-end encrypted. For more information on WhatsApp data protection, see [Data Privacy & Security](#) and [WhatsApp Encryption Overview](#) whitepaper.

Data encryption

AWS End User Messaging Social data is encrypted in transit and at rest within the AWS boundary. When you submit data to AWS End User Messaging Social, it encrypts the data as it's received and stores it. When you retrieve data from AWS End User Messaging Social, it transmits the data to you by using current security protocols.

Encryption at rest

AWS End User Messaging Social encrypts all the data that it stores for you within the AWS boundary. This includes configuration data, registration data, and any data that you add into AWS End User Messaging Social. To encrypt your data, AWS End User Messaging Social uses internal AWS Key Management Service (AWS KMS) keys that the service owns and maintains on your behalf. For information about AWS KMS, see the [AWS Key Management Service Developer Guide](#).

Encryption in transit

AWS End User Messaging Social uses HTTPS and Transport Layer Security (TLS) 1.2 to communicate with your clients, applications, and Meta. To communicate with other AWS services, AWS End User Messaging Social uses HTTPS and TLS 1.2. In addition, when you create and manage AWS End User Messaging Social resources by using the console, an AWS SDK, or the AWS Command Line Interface, all communications are secured using HTTPS and TLS 1.2.

Key management

To encrypt your data, AWS End User Messaging Social uses internal AWS KMS keys that the service owns and maintains on your behalf. We rotate these keys on a regular basis. You can't provision and use your own AWS KMS or other keys to encrypt data that you store in AWS End User Messaging Social.

Inter-network traffic privacy

Internetwork traffic privacy refers to securing connections and traffic between AWS End User Messaging Social and your on-premises clients and applications, and between AWS End User Messaging Social and other AWS resources in the same AWS Region. The following features and practices can help you secure internetwork traffic privacy for AWS End User Messaging Social.

Traffic between AWS End User Messaging Social and on-premises clients and applications

To establish a private connection between AWS End User Messaging Social and clients and applications on your on-premises network, you can use Direct Connect. This enables you to link your network to an AWS Direct Connect location by using a standard, fiber-optic Ethernet cable. One end of the cable is connected to your router. The other end is connected to a Direct Connect router. For more information, see [What is Direct Connect?](#) in the *Direct Connect User Guide*.

To help secure access to AWS End User Messaging Social through published APIs, we recommend that you comply with AWS End User Messaging Social requirements for API calls. AWS End User Messaging Social requires clients to use Transport Layer Security (TLS) 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS), such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Diffie-Hellman Ephemeral (ECDHE). Most modern systems such as Java 7 and later support these modes.

In addition, requests must be signed using an access key ID and a secret access key that's associated with an AWS Identity and Access Management (IAM) principal for your AWS account. Alternatively, you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

Identity and access management for AWS End User Messaging Social

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS End User Messaging Social resources. IAM is an AWS service that you can use with no additional charge.

Topics

- [Audience](#)
- [Authenticating with identities](#)
- [Managing access using policies](#)
- [How AWS End User Messaging Social works with IAM](#)
- [Identity-based policy examples for AWS End User Messaging Social](#)
- [AWS managed policies for AWS End User Messaging Social](#)
- [Troubleshooting AWS End User Messaging Social identity and access](#)

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see [Troubleshooting AWS End User Messaging Social identity and access](#))
- **Service administrator** - determine user access and submit permission requests (see [How AWS End User Messaging Social works with IAM](#))
- **IAM administrator** - write policies to manage access (see [Identity-based policy examples for AWS End User Messaging Social](#))

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see [How to sign in to your AWS account](#) in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see [AWS Signature Version 4 for API requests](#) in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you

don't use the root user for everyday tasks. For tasks that require root user credentials, see [Tasks that require root user credentials](#) in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see [What is IAM Identity Center?](#) in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see [Require human users to use federation with an identity provider to access AWS using temporary credentials](#) in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see [Use cases for IAM users](#) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by [switching from a user to an IAM role \(console\)](#) or by calling an AWS CLI or AWS API operation. For more information, see [Methods to assume a role](#) in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see [Cross account resource access in IAM](#) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For

more information about JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Define custom IAM permissions with customer managed policies](#) in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see [Choose between managed policies and inline policies](#) in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must [specify a principal](#) in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.

- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see [Service control policies](#) in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see [Resource control policies \(RCPs\)](#) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see [Session policies](#) in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see [Policy evaluation logic](#) in the *IAM User Guide*.

How AWS End User Messaging Social works with IAM

Before you use IAM to manage access to AWS End User Messaging Social, learn what IAM features are available to use with AWS End User Messaging Social.

IAM features you can use with AWS End User Messaging Social

IAM feature	AWS End User Messaging Social support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys	Yes
ACLs	No
ABAC (tags in policies)	Partial

IAM feature	AWS End User Messaging Social support
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes
Service-linked roles	Yes

To get a high-level view of how AWS End User Messaging Social and other AWS services work with most IAM features, see [AWS services that work with IAM](#) in the *IAM User Guide*.

Identity-based policies for AWS End User Messaging Social

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Define custom IAM permissions with customer managed policies](#) in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see [IAM JSON policy elements reference](#) in the *IAM User Guide*.

Identity-based policy examples for AWS End User Messaging Social

To view examples of AWS End User Messaging Social identity-based policies, see [Identity-based policy examples for AWS End User Messaging Social](#).

Resource-based policies within AWS End User Messaging Social

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific

resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must [specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see [Cross account resource access in IAM](#) in the *IAM User Guide*.

Policy actions for AWS End User Messaging Social

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The **Action** element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS End User Messaging Social actions, see [Actions Defined by AWS End User Messaging Social](#) in the *Service Authorization Reference*.

Policy actions in AWS End User Messaging Social use the following prefix before the action:

```
social-messaging
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [  
  "social-messaging:action1",  
  "social-messaging:action2"  
]
```

To view examples of AWS End User Messaging Social identity-based policies, see [Identity-based policy examples for AWS End User Messaging Social](#).

Policy resources for AWS End User Messaging Social

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Resource JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its [Amazon Resource Name \(ARN\)](#). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of AWS End User Messaging Social resource types and their ARNs, see [Resources Defined by AWS End User Messaging Social](#) in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see [Actions Defined by AWS End User Messaging Social](#).

To view examples of AWS End User Messaging Social identity-based policies, see [Identity-based policy examples for AWS End User Messaging Social](#).

Policy condition keys for AWS End User Messaging Social

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The Condition element specifies when statements execute based on defined criteria. You can create conditional expressions that use [condition operators](#), such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

To see a list of AWS End User Messaging Social condition keys, see [Condition Keys for AWS End User Messaging Social](#) in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see [Actions Defined by AWS End User Messaging Social](#).

To view examples of AWS End User Messaging Social identity-based policies, see [Identity-based policy examples for AWS End User Messaging Social](#).

ACLs in AWS End User Messaging Social

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with AWS End User Messaging Social

Supports ABAC (tags in policies): Partial

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the [condition element](#) of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see [Define permissions with ABAC authorization](#) in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see [Use attribute-based access control \(ABAC\)](#) in the *IAM User Guide*.

Using temporary credentials with AWS End User Messaging Social

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see [Temporary security credentials in IAM](#) and [AWS services that work with IAM](#) in the *IAM User Guide*.

Cross-service principal permissions for AWS End User Messaging Social

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see [Forward access sessions](#).

Service roles for AWS End User Messaging Social

Supports service roles: Yes

A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Create a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.

Warning

Changing the permissions for a service role might break AWS End User Messaging Social functionality. Edit service roles only when AWS End User Messaging Social provides guidance to do so.

Service-linked roles for AWS End User Messaging Social

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see [AWS services that work with IAM](#). Find a service in the table that includes a Yes in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS End User Messaging Social

By default, users and roles don't have permission to create or modify AWS End User Messaging Social resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see [Create IAM policies \(console\)](#) in the *IAM User Guide*.

For details about actions and resource types defined by AWS End User Messaging Social, including the format of the ARNs for each of the resource types, see [Actions, Resources, and Condition Keys for AWS End User Messaging Social](#) in the *Service Authorization Reference*.

Topics

- [Policy best practices](#)
- [Using the AWS End User Messaging Social console](#)
- [Allow users to view their own permissions](#)

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS End User Messaging Social resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see [AWS managed policies](#) or [AWS managed policies for job functions](#) in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see [Policies and permissions in IAM](#) in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies

adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see [Validate policies with IAM Access Analyzer](#) in the *IAM User Guide*.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see [Secure API access with MFA](#) in the *IAM User Guide*.

For more information about best practices in IAM, see [Security best practices in IAM](#) in the *IAM User Guide*.

Using the AWS End User Messaging Social console

To access the AWS End User Messaging Social console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the AWS End User Messaging Social resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the AWS End User Messaging Social console, also attach the AWS End User Messaging Social *ConsoleAccess* or *ReadOnly* AWS managed policy to the entities. For more information, see [Adding permissions to a user](#) in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Sid": "ViewOwnUserInfo",
    "Effect": "Allow",
    "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

AWS managed policies for AWS End User Messaging Social

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to [create IAM customer managed policies](#) that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the *IAM User Guide*.

AWS End User Messaging Social updates to AWS managed policies

View details about updates to AWS managed policies for AWS End User Messaging Social since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AWS End User Messaging Social Document history page.

Change	Description	Date
AWS End User Messaging Social started tracking changes	AWS End User Messaging Social started tracking changes for its AWS managed policies.	October 10, 2024

Troubleshooting AWS End User Messaging Social identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS End User Messaging Social and IAM.

Topics

- [I am not authorized to perform an action in AWS End User Messaging Social](#)
- [I am not authorized to perform iam:PassRole](#)
- [I want to allow people outside of my AWS account to access my AWS End User Messaging Social resources](#)

I am not authorized to perform an action in AWS End User Messaging Social

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional `my-example-widget` resource but doesn't have the fictional `social-messaging:GetWidget` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform: social-messaging:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the `my-example-widget` resource by using the `social-messaging:GetWidget` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to AWS End User Messaging Social.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in AWS End User Messaging Social. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my AWS End User Messaging Social resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS End User Messaging Social supports these features, see [How AWS End User Messaging Social works with IAM](#).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [Cross account resource access in IAM](#) in the *IAM User Guide*.

Compliance validation for AWS End User Messaging Social

To learn whether an AWS service is within the scope of specific compliance programs, see [AWS services in Scope by Compliance Program](#) and choose the compliance program that you are interested in. For general information, see [AWS Compliance Programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading Reports in AWS Artifact](#).

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more

information about your compliance responsibility when using AWS services, see [AWS Security Documentation](#).

Resilience in AWS End User Messaging Social

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see [AWS Global Infrastructure](#).

In addition to the AWS global infrastructure, AWS End User Messaging Social offers several features to help support your data resiliency and backup needs.

Infrastructure Security in AWS End User Messaging Social

As a managed service, AWS End User Messaging Social is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](#) whitepaper.

You use AWS published API calls to access AWS End User Messaging Social through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur

when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the [aws:SourceArn](#) and [aws:SourceAccount](#) global condition context keys in resource policies to limit the permissions that Social Messaging gives another service to the resource. Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access. Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with the cross-service use.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcard characters (*) for the unknown portions of the ARN. For example, `arn:aws:social-messaging:*:123456789012:*`.

If the `aws:SourceArn` value does not contain the account ID, such as an Amazon S3 bucket ARN, you must use both global condition context keys to limit permissions.

The value of `aws:SourceArn` must be `ResourceDescription`.

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in Social Messaging to prevent the confused deputy problem.

Security best practices

AWS End User Messaging Social provides a number of security features to consider as you develop and implement your own security policies. The following best practices are general guidelines and don't represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

- Create an individual user for each person who manages AWS End User Messaging Social resources, including yourself. Don't use AWS root credentials to manage AWS End User Messaging Social resources.
- Grant each user the minimum set of permissions required to perform his or her duties.
- Use IAM groups to effectively manage permissions for multiple users.

- Rotate your IAM credentials regularly.

Using service-linked roles for AWS End User Messaging Social

AWS End User Messaging Social uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to AWS End User Messaging Social. Service-linked roles are predefined by AWS End User Messaging Social and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up AWS End User Messaging Social easier because you don't have to manually add the necessary permissions. AWS End User Messaging Social defines the permissions of its service-linked roles, and unless defined otherwise, only AWS End User Messaging Social can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your AWS End User Messaging Social resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see [AWS services that work with IAM](#) and look for the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for AWS End User Messaging Social

AWS End User Messaging Social uses the service-linked role named **AWSServiceRoleForSocialMessaging** – To publish metrics and provide insights for your social message sending.

The AWSServiceRoleForSocialMessaging service-linked role trusts the following services to assume the role:

- `social-messaging.amazonaws.com`

The role permissions policy named AWSSocialMessagingServiceRolePolicy allows AWS End User Messaging Social to complete the following actions on the specified resources:

- Action: "cloudwatch:PutMetricData" on all AWS resources in the AWS/SocialMessaging namespace.

You must configure permissions to allow your users, groups, or roles to create, edit, or delete a service-linked role. For more information, see [Service-linked role permissions](#) in the *IAM User Guide*.

For updates to the policy, see [AWS End User Messaging Social updates to AWS managed policies](#).

Creating a service-linked role for AWS End User Messaging Social

You can use the IAM console to create a service-linked role with the **AWSEndUserMessagingSocial - Metrics** use case. In the AWS CLI or the AWS API, create a service-linked role with the `social-messaging.amazonaws.com` service name. For more information, see [Creating a service-linked role](#) in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

You can create the service-linked role for AWS End User Messaging Social with the following AWS CLI command:

```
aws iam create-service-linked-role --aws-service-name social-messaging.amazonaws.com
```

Editing a service-linked role for AWS End User Messaging Social

AWS End User Messaging Social does not allow you to edit the `AWSServiceRoleForSocialMessaging` service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see [Editing a service-linked role](#) in the *IAM User Guide*.

Deleting a service-linked role for AWS End User Messaging Social

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

Note

If the AWS End User Messaging Social service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To remove AWS End User Messaging Social resources used by the `AWSServiceRoleForSocialMessaging`

1. Call `list-linked-whatsapp-business-accounts` API to see the resources you have.
2. For each linked Whats App Business Account, call the `disassociate-whatsapp-business-account` API to remove the resource from SocialMessaging service.
3. Verify no resources are returned by calling the `list-linked-whatsapp-business-accounts` API again.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSServiceRoleForSocialMessaging` service-linked role. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

Supported Regions for AWS End User Messaging Social service-linked roles

AWS End User Messaging Social supports using service-linked roles in all of the Regions where the service is available. For more information, see [AWS Regions and endpoints](#).

Access AWS End User Messaging Social using an interface endpoint (AWS PrivateLink)

You can use AWS PrivateLink to create a private connection between your VPC and AWS End User Messaging Social. You can access AWS End User Messaging Social as if it were in your VPC, without the use of an internet gateway, NAT device, VPN connection, or Direct Connect connection. Instances in your VPC don't need public IP addresses to access AWS End User Messaging Social.

You establish this private connection by creating an *interface endpoint*, powered by AWS PrivateLink. We create an endpoint network interface in each subnet that you enable for the interface endpoint. These are requester-managed network interfaces that serve as the entry point for traffic destined for AWS End User Messaging Social.

For more information, see [Access AWS services through AWS PrivateLink](#) in the *AWS PrivateLink Guide*.

Considerations for AWS End User Messaging Social

Before you set up an interface endpoint for AWS End User Messaging Social, review [Considerations](#) in the *AWS PrivateLink Guide*.

AWS End User Messaging Social supports making calls to all of its API actions through the interface endpoint.

VPC endpoint policies are not supported for AWS End User Messaging Social. By default, full access to AWS End User Messaging Social is allowed through the interface endpoint. Alternatively, you can associate a security group with the endpoint network interfaces to control traffic to AWS End User Messaging Social through the interface endpoint.

Create an interface endpoint for AWS End User Messaging Social

You can create an interface endpoint for AWS End User Messaging Social using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see [Create an interface endpoint](#) in the *AWS PrivateLink Guide*.

Create an interface endpoint for AWS End User Messaging Social using the following service name:

- `com.amazonaws.region.social-messaging`

If you enable private DNS for the interface endpoint, you can make API requests to AWS End User Messaging Social using its default Regional DNS name. For example, `service-name.us-east-1.amazonaws.com`.

Create an endpoint policy for your interface endpoint

An endpoint policy is an IAM resource that you can attach to an interface endpoint. The default endpoint policy allows full access to AWS End User Messaging Social through the interface endpoint. To control the access allowed to AWS End User Messaging Social from your VPC, attach a custom endpoint policy to the interface endpoint.

An endpoint policy specifies the following information:

- The principals that can perform actions (AWS accounts, IAM users, and IAM roles).
- The actions that can be performed.
- The resources on which the actions can be performed.

For more information, see [Control access to services using endpoint policies](#) in the *AWS PrivateLink Guide*.

Example: VPC endpoint policy for AWS End User Messaging Social actions

The following is an example of a custom endpoint policy. When you attach this policy to your interface endpoint, it grants access to the listed AWS End User Messaging Social actions for all principals on all resources.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "social-messaging:DeleteWhatsAppMessageMedia",
        "social-messaging:PostWhatsAppMessageMedia",
        "social-messaging:SendWhatsAppMessage"
      ]
    }
  ],
}
```

```
    "Resource": "*"
  }
]
}
```

Quotas for AWS End User Messaging Social

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

Your AWS account has the following quotas related to AWS End User Messaging Social.

Resource	Default
WhatsApp Business Account (WABA)	25 per Region

AWS End User Messaging Social implements quotas that restrict the number of requests that you can make to the AWS End User Messaging Social API from your AWS account.

Operation	Default quota rate (requests per second)
SendWhatsAppMessage	1,000
PostWhatsAppMessageMedia	100
GetWhatsAppMessageMedia	100
DeleteWhatsAppMessageMedia	100
DisassociateWhatsAppBusinessAccount	10
ListWhatsAppBusinessAccount	10
TagResource	10
UntagResourceRate	10
ListTagsForResourceRate	10

Document history for the AWS End User Messaging Social User Guide

The following table describes the documentation releases for AWS End User Messaging Social.

Change	Description	Date
WhatsApp Flows support	AWS End User Messaging Social now supports creating and managing WhatsApp Flows for interactive, multi-screen experiences. For more information, see Managing WhatsApp Flows .	June 8, 2026
Meta's Marketing Messages (MM) API support	AWS End User Messaging Social now supports Meta's Marketing Messages API for marketing template messages. For more information, see What is AWS End User Messaging Social? .	April 23, 2026
Regional availability	Added support for Asia Pacific (Hong Kong), Asia Pacific (Taipei), Asia Pacific (Osaka), Asia Pacific (Jakarta), Asia Pacific (Melbourne), Asia Pacific (Malaysia), Asia Pacific (Thailand), Europe (Zurich), Europe (Milan), Europe (Paris), Israel (Tel Aviv), and US West (N. California) regions. For more information, see Regional availability .	March 30, 2026

Regional availability	Added support for Asia Pacific (New Zealand), Canada West (Calgary), Europe (Stockholm), Middle East (Bahrain), Middle East (UAE), and Mexico (Central) regions. For more information, see Regional availability .	February 3, 2026
Support added for template management in the AWS Console	Added support for WhatsApp template management in the AWS Console. For more information, see Managing message templates in the AWS Console .	December 18, 2025
Support added for Amazon EventBridge events	Added support for sending WhatsApp message events to EventBridge. For more information see Monitoring with EventBridge	October 3, 2025
Support added for new MessageFee types	Six new MessageFee types that provide additional billing information are now available . For more information see Understanding your bill .	September 30, 2025
Deprecated the Charged per conversation topic	The Charged per conversation topic in the Understanding your bill section has been deprecated.	September 30, 2025

New API for creating WhatsApp message templates	Added support for creating WhatsApp message templates using the AWS End User Messaging Social API. For more information, see Creating message templates with the CreateWhatsAppMessageTemplate API .	July 25, 2025
Charged per message	Starting on July 1 2025, AWS End User Messaging Social, charges per message instead of per conversation for WhatsApp messages. For more information, see Charged per message .	June 30, 2025
Regional availability	Added support for Europe (Frankfurt) region. For more information, see Regional availability .	December 5, 2024
Add a message and event destination	Added support for Connect Customer as an event destination. For more information, see Add a message and event destination .	December 1, 2024
AWS PrivateLink	Added support for AWS PrivateLink. For more information, see AWS PrivateLink .	October 22, 2024
Initial release	Initial release of the AWS End User Messaging Social User Guide	October 10, 2024