Administration Guide

AWS Wickr

AWS Wickr: Administration Guide

Table of Contents

What is AWS Wickr?

AWS Wickr is an end-to-end encrypted service that helps organizations and government agencies to communicate securely through one-to-one and group messaging, voice and video calling, file sharing, screen sharing, and more. Wickr can help customers overcome data retention obligations associated with consumer-grade messaging apps, and safely facilitate collaboration. Advanced security and administrative controls help organizations meet legal and regulatory requirements, and build custom solutions for data security challenges.

Information can be logged to a private, customer-controlled data store for retention and auditing purposes. Users have comprehensive administrative control over data, which includes setting permissions, configuring ephemeral messaging options, and defining security groups. Wickr integrates with additional services such as Active Directory (AD), single sign-on (SSO) with OpenID Connect (OIDC), and more. You can quickly create and manage a Wickr network through the AWS Management Console, and securely automate workflows using Wickr bots. To get started, see Setting up for AWS Wickr.

Topics

Features of Wickr

Enhanced security and privacy

Wickr uses 256-bit Advanced Encryption Standard (AES) end-to-end encryption for every feature. Communications are encrypted locally on user devices, and remain undecipherable in transit to anyone other than sender and receiver. Every message, call, and file is encrypted with a new random key, and no one but intended recipients (not even AWS) can decrypt them. Whether they are sharing sensitive and regulated data, discussing legal or HR matters, or even conducting tactical military operations, customers use Wickr to communicate when security and privacy are paramount.

Data retention

Flexible administrative features are designed not only to safeguard sensitive information, but to retain data as required for compliance obligations, legal hold, and auditing purposes. Messages and files can be archived in a secure, customer-controlled data store.

Flexible access

Users have multi-device (mobile, desktop) access and the ability to function in low-bandwidth environments, including disconnected and out-of-band communications.

Administrative controls

Users have comprehensive administrative control over data, which includes setting permissions, configuring responsible ephemeral messaging options, and defining security groups.

Powerful integrations and bots

Wickr integrates with additional services such as Active Directory, single sign-on (SSO) with OpenID Connect (OIDC), and more. Customers can quickly create and manage a Wickr network through the AWS Management Console, and securely automate workflows with Wickr Bots.

Following is a breakdown of Wickr collaboration offerings:

- 1:1 and group messaging: Securely chat with your team in rooms with up to 500 members
- Audio and video calling: Hold conference calls with up to 70 people
- Screen sharing and broadcasting: Present with up to 500 participants
- File sharing and saving: Transfer files up to 5GBs with unlimited storage
- Ephemeral: Control expiration and burn-on-read timers
- Global federation: Connect with Wickr users outside of your network

> **Note**
>
> Wickr networks in AWS GovCloud (US-West) can be federated only with other Wickr networks in AWS GovCloud (US-West).

Regional availability

Wickr is available in US East (N. Virginia), Asia Pacific (Malaysia), Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Tokyo), Canada (Central), Europe (Frankfurt), Europe (London), Europe (Stockholm), and Europe (Zurich) AWS Regions. Wickr is also available in AWS GovCloud (US-West) Region. Each Region contains multiple Availability Zones, which are physically separate but connected by private, low-latency, high-bandwidth, and redundant network connections. These Availability Zones are used to provide enhanced availability, fault-tolerance, and minimized latency.

To learn more about AWS Regions, see Specify which AWS Regions your account can use in the *AWS General Reference*. For more information about the number of Availability Zones available in each Region, see AWS Global Infrastructure.

Accessing Wickr

Administrators access the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/. Before you get started using Wickr you should complete the Setting up for AWS Wickr and Getting started with AWS Wickr guides.

> **Note**
>
> The Wickr service does not have an application programming interface (API).

End users access Wickr through the Wickr client. For more information, see the *AWS Wickr User Guide*.

Pricing

Wickr is available in different plans for individuals, small teams, and large businesses. For more information, see AWS Wickr Pricing.

Wickr end user documentation

If you are an end user of the Wickr client and need to access its documentation, see the AWS Wickr User Guide.

Setting up for AWS Wickr

If you're a new AWS customer, complete the setup prerequisites that are listed on this page before you start using AWS Wickr. For these setup procedures, you use the AWS Identity and Access Management (IAM) service. For complete information about IAM, see the IAM User Guide.

Topics

Sign up for AWS

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call or text message and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

Create an IAM user

To create an administrator user, choose one of the following options.

Choose one way to manage your administrator	To	By	You can also
In IAM Identity Center (Recommended)	Use short-term credentials to access AWS. This aligns with the security best practices. For information about best practices, see Security best practices in IAM in the *IAM User Guide*.	Following the instructions in Getting started in the *AWS IAM Identity Center User Guide*.	Configure programmatic access by Configuring the AWS CLI to use AWS IAM Identity Center in the *AWS Command Line Interface User Guide*.
In IAM (Not recommended)	Use long-term credentials to access AWS.	Following the instructions in Creating your first IAM admin user and user group in the *IAM User Guide*.	Configure programmatic access by Managing access keys for IAM users in the *IAM User Guide*.

Note

You can also assign the `AWSWickrFullAccess` managed policy to grant full administrative permission to the Wickr service. For more information, see AWS managed policy: AWSWickrFullAccess.

What's next

You completed the prerequisite set up steps. To begin configuring Wickr, see *Getting started*.

Getting started with AWS Wickr

In this guide, we show you how to get started with Wickr by creating a network, configuring your network, and creating users.

Topics

Prerequisites

Before you start, be sure to complete the following prerequisites if you haven't already:

- Sign up for Amazon Web Services (AWS). For more information, see Setting up for AWS Wickr.
- Ensure that you have the permissions required to administer Wickr. For more information, see AWS managed policy: AWSWickrFullAccess.
- Make sure you allow list the appropriate ports and domains for Wickr. For more information, see Ports and domains to allow list for your Wickr network.

Step 1: Create a network

You can create a Wickr network.

Complete the following procedure to create a Wickr network for your account.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.

> **Note**
>
> If you haven't created a Wickr networking before, you will see the informational page for the Wickr service. After you create one or more Wickr networks, you will see the **Networks** page, which contains a list view of all the Wickr networks you have created.

2. Choose **Create a network**.
3. Enter a name for your network in the **Network name** text box. Choose a name that members of your organization will recognize, such as the name of your company or the name of your team.
4. Choose a plan. You can choose one of the following Wickr network plans:

   - **Standard —** For small and large business teams that need administrative controls and flexibility.
   - **Premium** or **Premium Free Trial —** For businesses that require the highest feature limits, granular administrative controls, and data retention.

   Administrators have the option to select a premium free trial, which is available for up to 30 users and lasts for three months. For AWS WickrGov, the premium free trial option allows up to 50 users and also last for three months. During the premium free trial period, administrators can upgrade or downgrade to Premium or Standard plans.

   For more information about available Wickr plans and pricing, see the Wickr pricing page.
5. (Optional) Choose **Add new tag** to add a tag to your network. Tags consist of a key value pair. Tags can be used to search and filter resources or track your AWS costs. For more information, see Network tags.
6. Choose **Create Network**.

   You are redirected to the **Networks** page of the AWS Management Console for Wickr, and the new network is listed on the page.

Step 2: Configure your network

Complete the following procedure to access the AWS Management Console for Wickr, where you can add users, add security groups, configure SSO, configure data retention, and additional network settings.

1. On the **Networks** page, select the network name to navigate to that network.

   You're redirected to the Wickr Admin Console for the selected network.
2. The following user management options are available. For more information about configuring these settings, see Manage your AWS Wickr network.

- **Security Group** — Manage security groups and their settings, such as password complexity policies, messaging preferences, calling features, security features and external federation. For more information, see Security groups for AWS Wickr .
- **Single Sign-on (SSO) Configuration** — Configure SSO and view the endpoint address for your Wickr network. Wickr supports SSO providers who use OpenID Connect (OIDC) only. Providers who use Security Assertion Markup Language (SAML) are not supported. For more information, see Single sign-on configuration for AWS Wickr.

Step 3: Create and invite users

You can create users in your Wickr network using the following methods:

- **Single sign-on** — If you configure SSO, you can invite users by sharing your Wickr company ID. End users register for Wickr using the provided company ID and their work email address. For more information, see Single sign-on configuration for AWS Wickr.
- **Invitation** — You can manually create users in the AWS Management Console for Wickr and have an email invitation sent to them. End users can register for Wickr by choosing the link in the email.

> **Note**
>
> You can also enable guest users for your Wickr network. For more information, see Guest users in AWS Wickr network

Complete the following procedures to create or invite users.

> **Note**
>
> Administrators are also considered users and must invite themselves to SSO or non-SSO Wickr networks.

To create Wickr users and send invitations with SSO:

Write and send an email to the SSO users who should sign up for Wickr. Include the following information in your email:

- Your Wickr company ID. You specify a company ID for your Wickr network when you configure SSO. For more information, see Configure SSO in AWS Wickr.
- The email address they should use to sign up.
- The URL to download the Wickr client. Users can download the Wickr clients from the AWS Wickr downloads page at https://aws.amazon.com/wickr/download/.

> **Note**
>
> If you created your Wickr network in AWS GovCloud (US-West), instruct your users to download and install the WickrGov client. For all other AWS Regions, instruct your users to download and install the standard Wickr client. For more information about AWS WickrGov, see AWS WickrGov in the *AWS GovCloud (US) User Guide*.

As users register for your Wickr network, they are added to the Wickr team directory with a status of **active**.

To manually create Wickr users and send invitations:

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.

   You're redirected to the Wickr network. In the Wickr network, you can add users, add security groups, configure SSO, configure data retention, and adjust additional settings.
3. In the navigation pane, choose **User management**.
4. On the **User management** page, under the **Team directory** tab, choose **Invite user**.

   You can also bulk invite users by choosing the drop-down arrow next to **Invite user**. On the **Bulk invite users** page, select **Download template** to download a CSV template that you can edit and upload with your list of users.
5. Enter the user's first name, last name, country code, phone number, and email address. Email address is the only field that is required. Be sure to choose the appropriate security group for the user.
6. Choose **Invite**.

Wickr sends an invitation email to the address you specify for the user. The email provides download links for the Wickr client applications, and a link to register for Wickr. For more information about what this end user experience looks like, see Download the Wickr app and accept your invitation in the *AWS Wickr User Guide*.

As users register for Wickr using the link in the email, their status in the Wickr team directory will change from **Pending** to **Active**.

Next steps

You completed the getting started steps. To manage Wickr, see the following:

- Manage your AWS Wickr network
- Manage users in AWS Wickr

Manage your AWS Wickr network

In the AWS Management Console for Wickr you can manage your Wickr network name, security groups, SSO configuration, and data retention settings.

Topics

Network details for AWS Wickr

You can edit the name of your Wickr network and view your network ID in the **Network details** section of the AWS Management Console for Wickr.

Topics

View network details in AWS Wickr

You can view the details of your Wickr network, including your network name and network ID.

Complete the following procedure to view your Wickr network profile and network ID.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, find the network you want to view.
3. On the right-hand side of the network you want to view, select the vertical ellipsis icon (three dots), and then choose **View details**.

   The **Network home** page displays your Wickr network name and network ID in the **Network details** section. You can use the network ID to configure federation.

Edit network name in AWS Wickr

You can edit the name of your Wickr network.

Complete the following procedure to edit your Wickr network name.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to the Wickr Admin Console for that network.
3. On the **Network home** page, in the **Network details** section, choose **Edit**.
4. Enter your new network name into the **Network Name** text box.
5. Choose **Save** to save your new network name.

Delete network in AWS Wickr

You can delete your AWS Wickr network.

> **Note**
>
> If you delete a premium free trial network, you won't be able to create another one.

To delete your Wickr network on the Networks home page, complete the following procedure.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, find the network you want to delete.
3. On the right-hand side of the network you want to delete, select the vertical ellipsis icon (three dots), and then choose **Delete network**.

4. Type **confirm** in the pop-up window, and then choose **Delete**.

   It can take a few minutes for the network to delete.

To delete your Wickr network while in the network, complete the following procedure.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network you want to delete.
3. Near the top right corner of the **Network home** page, choose **Delete network**.
4. Type **confirm** in the pop-up window, and then choose **Delete**.

   It can take a few minutes for the network to delete.

> **Note**
>
> Data retained by your data retention configuration (if enabled) will not be deleted when you delete your network. For more information, see Data retention for AWS Wickr.

Security groups for AWS Wickr

In the **Security Groups** section of the AWS Management Console for Wickr, you can manage security groups and their settings, such as password complexity policies, messaging preferences, calling features, security features and network federation.

Topics

- View security groups in AWS Wickr
- Create a security group in AWS Wickr
- Edit a security group in AWS Wickr
- Delete a security group in AWS Wickr

View security groups in AWS Wickr

You can view the details of your Wickr security groups.

Complete the following procedure to view security groups.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Security groups**.

   The **Security groups** page displays your current Wickr security groups and gives you the option to create a new group.

   On the **Security groups** page, select the security group you want to view. The page will display the current details for that security group.

Create a security group in AWS Wickr

You can create a new Wickr security group.

Complete the following procedure to create a security group.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Security groups**.
4. On the **Security groups** page, choose **Create security group** to create a new security group.

   > **Note**
   >
   > A new security group with a default name is automatically added to the security groups list.

5. On the **Create security group** page, enter the name of your security group.
6. Choose **Create security group**.

   For more information about editing the new security group, see Edit a security group in AWS Wickr.

Edit a security group in AWS Wickr

You can edit the details of your Wickr security group.

Complete the following procedure to edit a security group.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Security groups**.
4. Select the name of the security group that you want to edit.

   The security group details page displays the settings for the security group in different tabs.
5. The following tabs and corresponding settings are available:

   - **Security group details** — Choose **Edit** in the **Security group details** section to edit the name.
   - **Messaging** — Manage messaging features for members of the group.
     - **Burn-on-read** — Controls the maximum value that users can set for their burn-on-read timers in their Wickr clients. For more information, see Set message expiration and burn timers in the Wickr client.
     - **Expiration timer** — Controls the maximum value that users can set for their message expiration timer in their Wickr clients. For more information, see Set message expiration and burn timers in the Wickr client.
     - **Quick responses** — Set a list of quick responses for users to respond to messages.
     - **Secure shredder intensity** — Configure how frequently the secure shredder control runs for users. For more information, see Messaging.
   - **Calling** — Manage calling features for members of the group.
     - **Enable audio calling** — Users can initiate audio calls.
     - **Enable video calling and screen sharing** — Users can start video calls or share screen during call.
     - **TCP calling** — Enabling (or forcing) TCP calling is typically used when standard VoIP UDP ports are disallowed by an organization's IT or security department. If TCP calling is disabled, and UDP ports are not available for use, Wickr clients will try UDP first and fallback to TCP.
   - **Media and links** — Manage settings related to media and links for members of the group.

     File download size — Select **Best quality transfer** to allow users to transfer files and attachments in their original encrypted form. If you select **Low bandwidth transfer**, file attachments sent by users in Wickr will be compressed by the Wickr file transfer service.
   - **Location** — Manage location sharing settings for members of the group.

Location sharing — Users can share their locations using GPS-enabled devices. This feature displays a visual map based on the device's operating system defaults. Users have the option to disable the map view and share a link containing their GPS coordinates instead.

- **Security** — Configure additional security features for the group.
  - **Enable account takeover protection** — Enforce a two-factor authentication when a user adds a new device to their account. To verify a new device, user can generate a Wickr code from their old device, or perform an email verification. This is an additional layer of security to prevent unauthorized access to AWS Wickr accounts.
  - **Enable always re-authenticate** — Force users to always re-authenticate when re-entering the application.
  - **Master recovery key** — Creates a Master recovery key when an account is created. Users can approve the addition of a new device to their account if no other devices are available.
- **Notification and visibility** — Configure notification and visibility settings such as message previews in notifications for members of the group.
- **Wickr open access** — Configure Wickr open access settings for members of the group.
  - **Enable Wickr open access** — Enabling Wickr open access will disguise traffic to protect data on restricted and monitored networks. Based on geographic location, Wickr open access will connect to various global proxy servers that provide the best path and protocols for traffic obfuscation.
  - **Force Wickr open access** — Automatically enables and enforces Wickr open access on all devices.
- **Federation** — Control your users ability to communicate with other Wickr networks.
  - **Local federation** — The ability to federate with AWS users in other networks within the same region. For example, if there are two networks in AWS Canada (Central) Region with local federation enabled, they will be able to communicate with each other.
  - **Global federation** — The ability to federate with either Wickr Enterprise users or AWS users in a different network who belong to other regions. For example, a user on a Wickr network in AWS Canada (Central) Region, and a user in a network in AWS Europe (London) Region will be able to communicate with each other when global federation is turned **ON** for both networks.
  - **Restricted federation** — Allow list specific AWS Wickr or Wickr Enterprise networks that users can federate with. When configured, users can only communicate with external

users in allow listed networks. Both networks must allow list each other to use restricted federation.

For information on guest federation, see Enable or disable guest users in AWS Wickr network.

- **ATAK plugin configuration** — For more information on enabling ATAK, see What is ATAK?.

6. Choose **Save** to save edits you make to the security group details.

Delete a security group in AWS Wickr

You can delete your Wickr security group.

Complete the following procedure to delete a security group.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Security groups**.
4. On the **Security groups** page, find the security group you want to delete.
5. On the right-hand side of the security group you want to delete, select the vertical ellipsis icon (three dots), and then choose **Delete**.
6. Type **confirm** in the pop-up window, and then choose **Delete**.

   When you delete a security group that has assigned users, those users are automatically added to the default security group. To modify the security group assigned to users see Edit users in AWS Wickr network.

Single sign-on configuration for AWS Wickr

In the AWS Management Console for Wickr, you can configure Wickr to use a single sign-on system to authenticate. SSO provides an added layer of security when paired with an appropriate multi-factor authentication (MFA) system. Wickr supports SSO providers who use OpenID Connect (OIDC) only. Providers who use Security Assertion Markup Language (SAML) are not supported.

Topics

- Configure SSO in AWS Wickr
- Grace period for token refresh

View SSO details in AWS Wickr

You can view the details of your single sign-on configuration for your Wickr network and the network endpoint.

Complete the following procedure to view the current single sign-on configuration for your Wickr network, if any.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User Management**.

   On the **User Management** page, the **Single Sign-on** section displays your Wickr network endpoint and current SSO configuration.

Configure SSO in AWS Wickr

To ensure secure access to your Wickr network, you can set up your current single sign-on configuration. Detailed guides are available to assist you with this process.

For more information about configuring SSO, see the following guides:

> ⚠ **Important**
>
> When you configure SSO, you specify a company ID for your Wickr network. Be sure to write down the company ID for your Wickr network. You must provide it to your end users when sending invitation emails. End users must specify the company ID when they register for your Wickr network.

- AWS Wickr Single Sign-on (SSO) setup with Microsoft Entra (Azure AD)
- AWS Wickr Single Sign-on (SSO) setup with Okta
- AWS Wickr Single Sign-on (SSO) setup with Amazon Cognito

Configure AWS Wickr with Microsoft Entra (Azure AD) single sign-on

AWS Wickr can be configured to use Microsoft Entra (Azure AD) as an identity provider. To do so, complete the following procedures in both Microsoft Entra and the AWS Wickr admin console.

> **Warning**
>
> After SSO is enabled on a network it will sign active users out of Wickr and force them to re-authenticate using the SSO provider.

Step 1: Register AWS Wickr as an application in Microsoft Entra

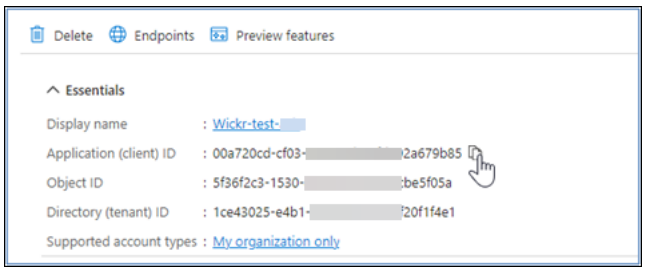
Complete the following procedure to register AWS Wickr as an application in Microsoft Entra.

> **Note**
>
> Refer to the Microsoft Entra documentation for detailed screenshots and troubleshooting. For more information, see Register an application with the Microsoft identity platform

1. In the navigation pane, choose **Applications** and then choose **App Registrations**.
2. On the **App Registrations** page, choose **Register an application**, and then enter an application name.
3. Select **Accounts in this organizational directory only (Default Directory only - Single tenant)**.
4. Under **Redirect URI**, select **Web**, and then enter the following web address: `https://messaging-pro-prod.wickr.com/deeplink/oidc.php`.

   > **Note**
   >
   > The Redirect URI can also be copied from the SSO configuration settings in the AWS Wickr Admin console.

5. Choose **Register**.
6. After registration, copy/save the Application (Client) ID generated.



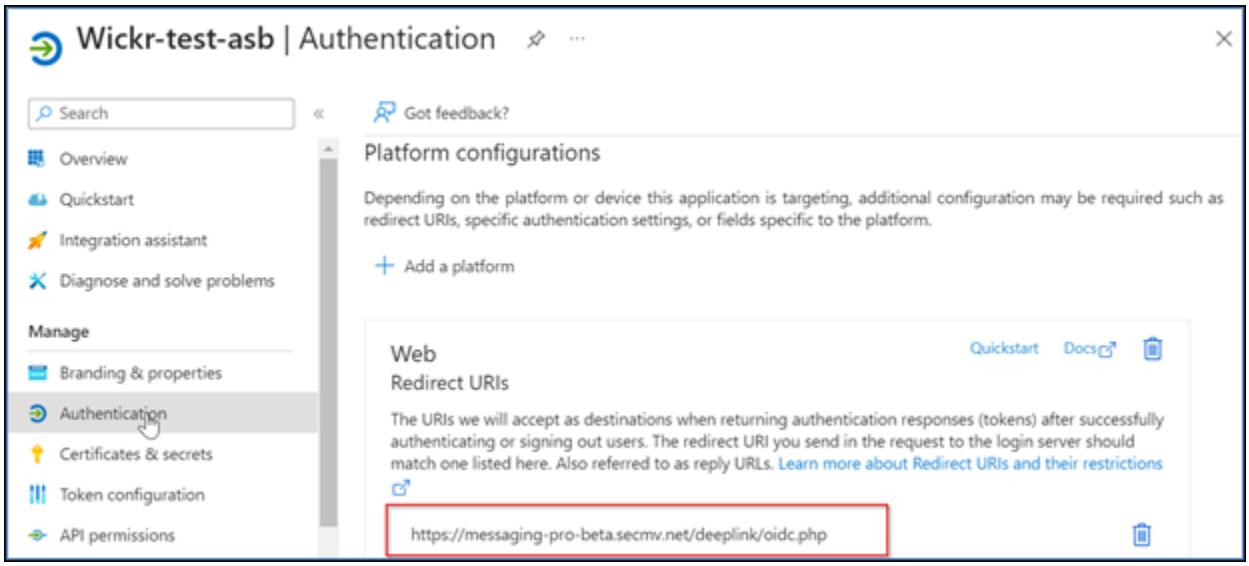
7. Select the **Endpoints** tab to make a note of the following:

    1. Oauth 2.0 authorization endpoint (v2): E.g.: `https://login.microsoftonline.com/1ce43025-e4b1-462d-a39f-337f20f1f4e1/oauth2/v2.0/authorize`
    2. Edit this value to remove the 'oauth2/" and "authorize". E.g. fixed URL will look like this: `https://login.microsoftonline.com/1ce43025-e4b1-462d-a39f-337f20f1f4e1/v2.0/`
    3. This will be referenced as the **SSO Issuer**.

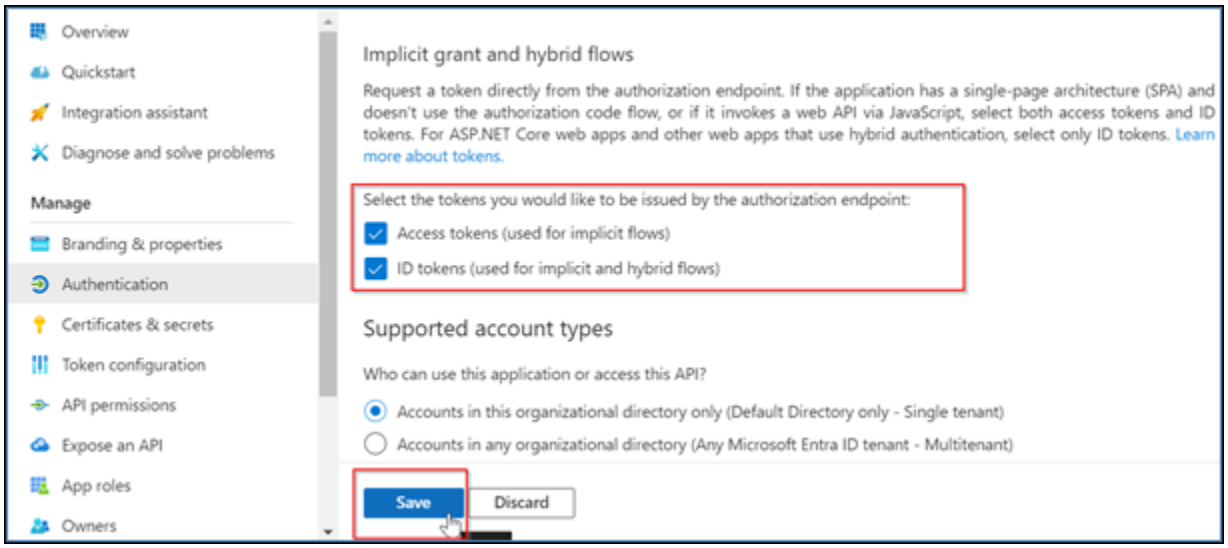
Step 2: Setup authentication

Complete the following procedure to setup authentication in Microsoft Entra.

1. In the navigation pane, choose **Authentication**.
2. On the **Authentication** page, make sure that the **Web Redirect URI** is the same as entered previously (in *Register AWS Wickr as an Application*).



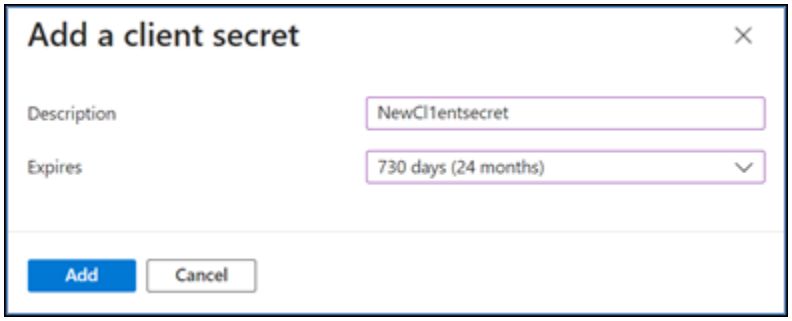
3. Select **Access tokens used for implicit flows** and **ID tokens used for implicit and hybrid flows**.
4. Choose **Save**.



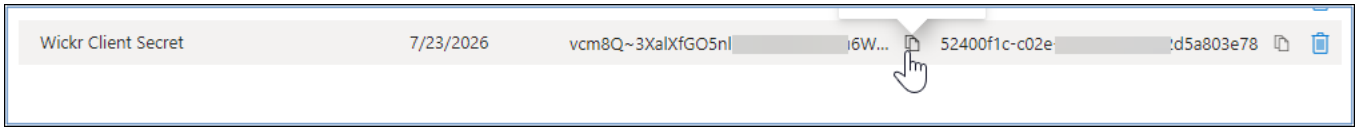
Step 3: Setup certificates and secrets

Complete the following procedure to setup certificates and secrets in Microsoft Entra.

1. In the navigation pane, choose **Certificates & secrets**.
2. On the **Certificates & secrets** page, select the **Client secrets** tab.
3. Under the **Client secrets** tab, select **New client secret**.
4. Enter a description and select an expiration period for the secret.
5. Choose **Add**.



6. After the certificate is created, copy the **Client secret value**.

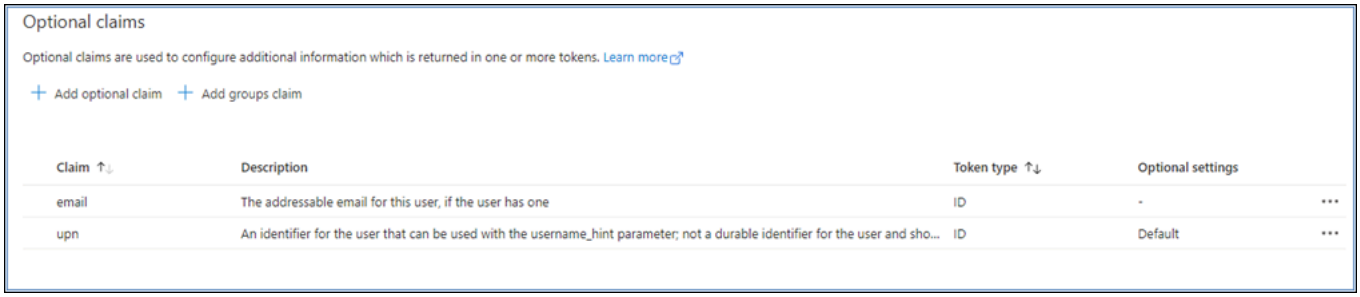


> **Note**
>
> The client secret value (not Secret ID) will be required for your client application code. You may not be able to view or copy the secret value after leaving this page. If you do not copy it now, you will have to go back to create a new client secret.

Step 4: Setup token configuration

Complete the following procedure to setup token configuration in Microsoft Entra.

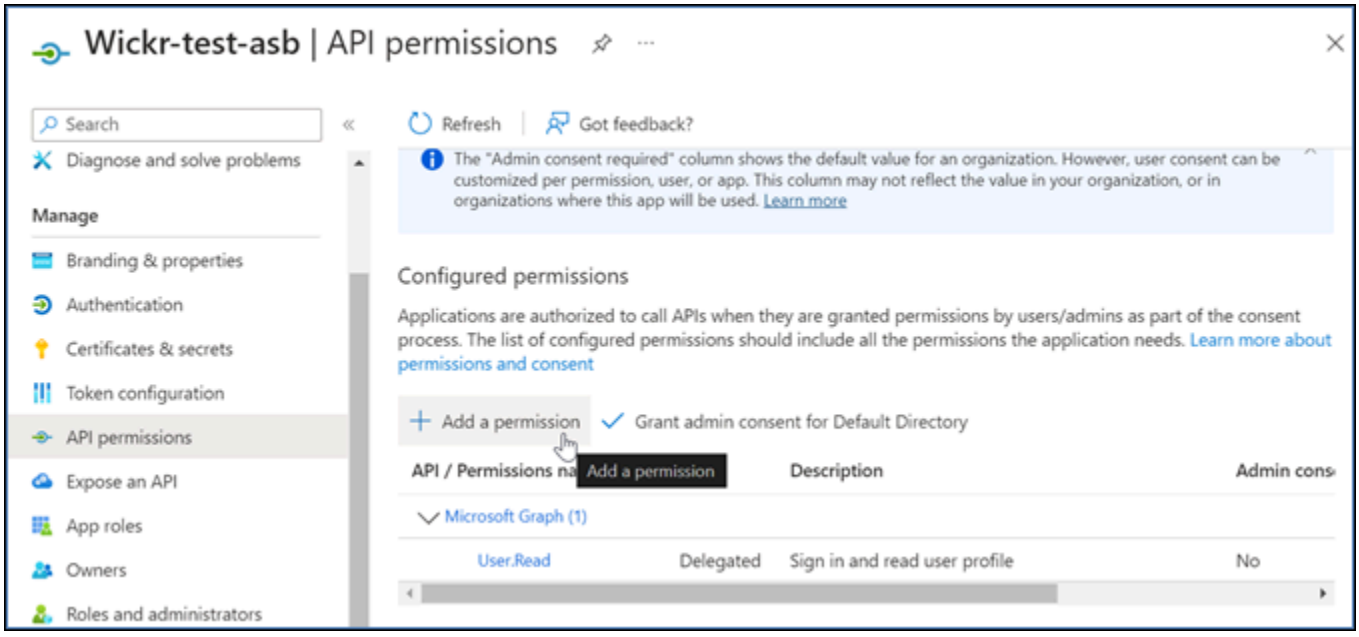
1. In the navigation pane, choose **Token configuration**.
2. On the **Token configuration** page, choose **Add optional claim**.
3. Under **Optional claims**, select the **Token type** as **ID**.
4. After selecting **ID**, under **Claim**, select **email** and **upn**.
5. Choose **Add**.



Step 5: Setup API permissions

Complete the following procedure to setup API permissions in Microsoft Entra.

1. In the navigation pane, choose **API permissions**.
2. On the **API permissions** page, choose **Add a permission**.

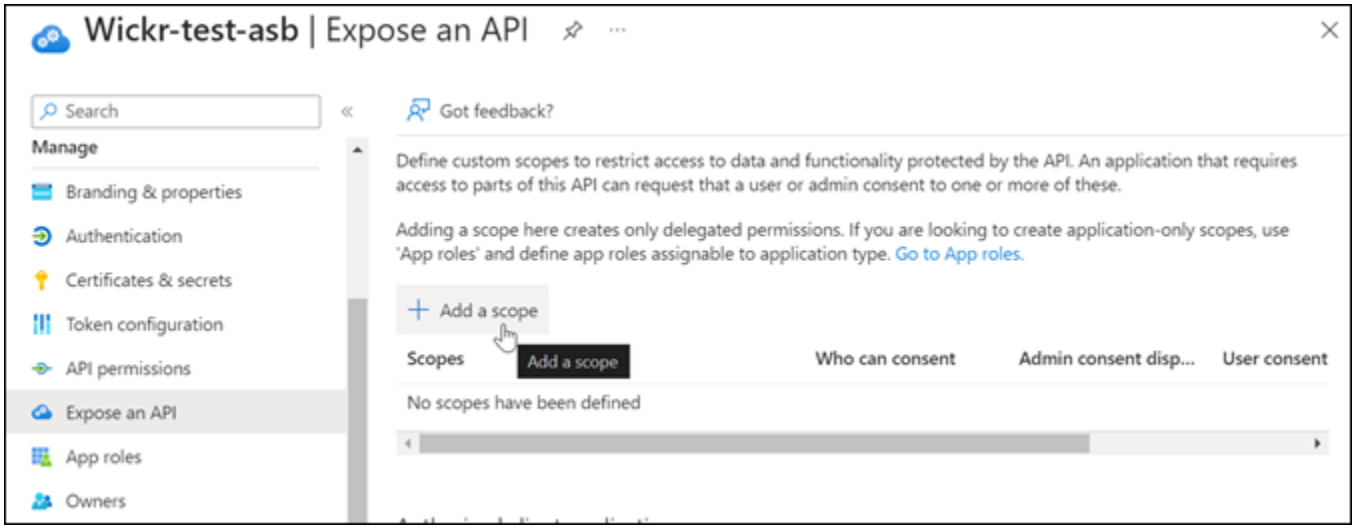


3. Select **Microsoft Graph** and then select **Delegated Permissions** .
4. Select the checkbox for **email** , **offline_access**, **openid**, **profile**.
5. Choose **Add permissions**.

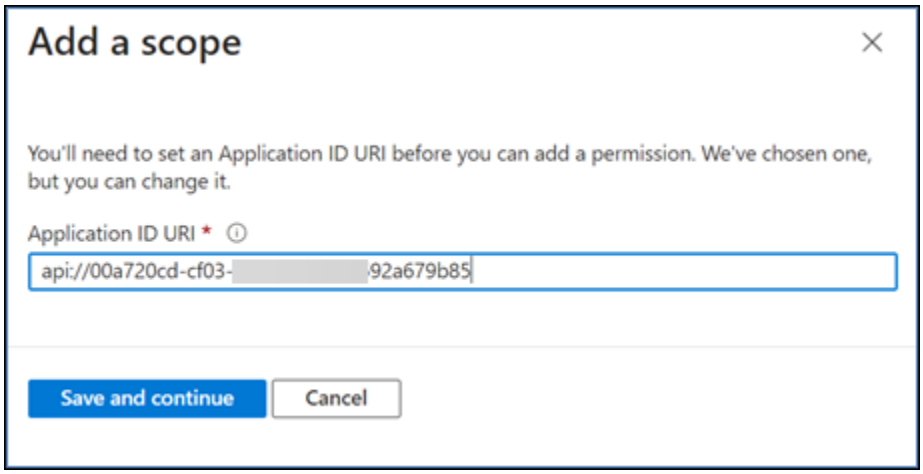
Step 6: Expose an API

Complete the following procedure to expose an API for each of the 4 scopes in Microsoft Entra.

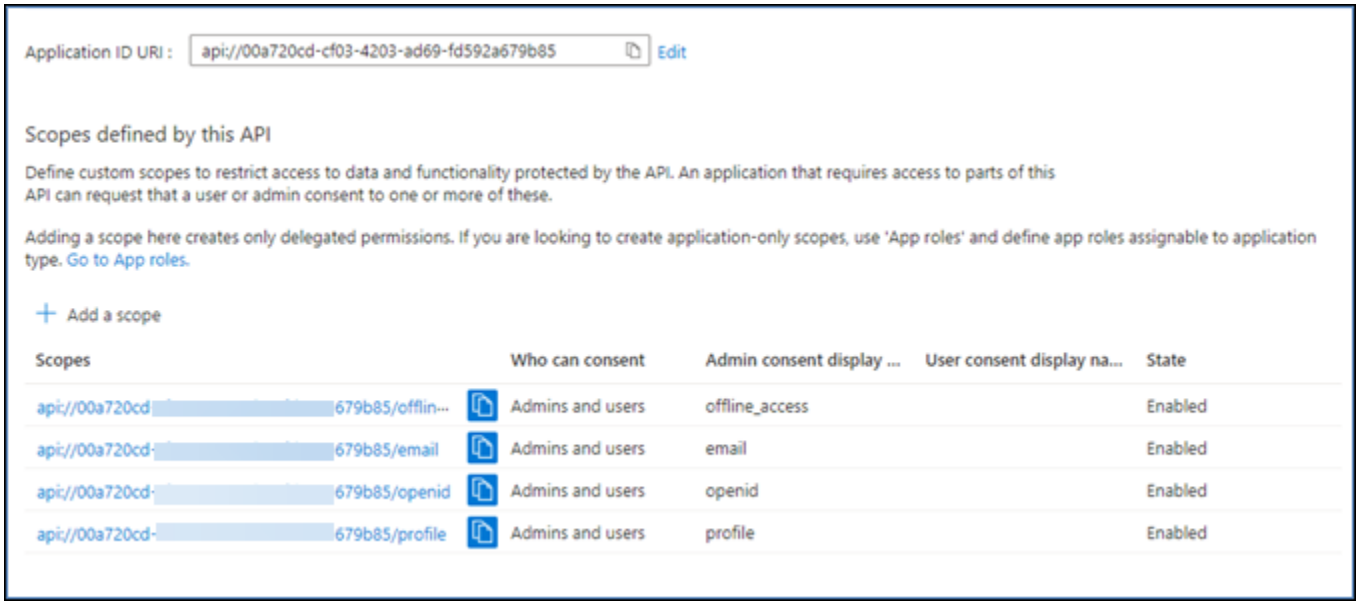
1. In the navigation pane, choose **Expose an API**.
2. On the **Expose an API** page, choose **Add a scope**.



Application ID URI should auto populate, and the ID that follows the URI should match the **Application ID** (created in *Register AWS Wickr as an application*).



3. Choose **Save and continue**.
4. Select the **Admins and users** tag, and then enter the scope name as **offline_access**.
5. Select **State**, and then select **Enable**.
6. Choose **Add scope**.
7. Repeat steps 1—6 of this section to add the following scopes: **email**, **openid**, and **profile**.



8. Under **Authorized client applications**, choose **Add a client application**.
9. Select all four scopes created in the previous step.
10. Enter or verify the **Application (client) ID**.
11. Choose **Add application**.

Step 7: AWS Wickr SSO configuration

Complete the following configuration procedure in the AWS Wickr console.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks page**, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**, and then choose **Configure SSO**.
4. Under **Network endpoint**, make sure the **Redirect URI** matches the following web address (added in step 4 under *Register AWS Wickr as an application*).

   `https://messaging-pro-prod.wickr.com/deeplink/oidc.php`.
5. Enter the following details:

   - **Issuer** — This is the endpoint that was modified previously (E.g. `https://login.microsoftonline.com/1ce43025-e4b1-462d-a39f-337f20f1f4e1/v2.0/`).
   - **Client ID** — This is the **Application (client) ID** from the **Overview** pane.
   - **Client secret (optional)** — This is the **Client secret** from the **Certificates & secrets** pane.
   - **Scopes** — These are the scope names exposed on the **Expose an API** pane. Enter **email**, **profile**, **offline_access**, and **openid**.
   - **Custom username scope (optional)** — Enter **upn**.
   - **Company ID** — This can be a unique text value including alphanumeric and underscore characters. This phrase is what your users will enter when registering on new devices.

   Other fields are optional.
6. Choose **Next**.
7. Verify the details in the **Review and save** page, and then choose **Save changes**.

SSO configuration is complete. To verify, you can now add a user to the application in Microsoft Entra, and login with the user using SSO and Company ID.

For more information on how to invite and onboard users, see Create and invite users.

Troubleshooting

Following are common issues you might encounter and suggestions for resolving them.

- SSO Connection test fails or is unresponsive:

- Make sure the **SSO Issuer** is configured as expected.
- Make sure the required fields in the **SSO Configured** are set as expected.

- Connection test is successful, but the user is unable to login:
  - Make sure the user is added to the Wickr application you registered in Microsoft Entra.
  - Make sure the user is using the correct company ID, including the prefix. *E.g. UE1-DemoNetworkW_drqtva*.
  - The **Client Secret** may not be set correctly in the **AWS Wickr SSO Configuration**. Re-set it by creating another **Client secret** in Microsoft Entra and set the new **Client secret** in the **Wickr SSO Configuration**.

Grace period for token refresh

Occasionally, there may be instances where identity providers encounter temporary or extended outages, which may lead to your users being logged out unexpectedly due to a failed refresh token for their client session. To prevent this problem, you can establish a grace period that allows your users to remain signed in even if their client refresh token fails during such outages.

Here are the available options for the grace period:

- No grace period (default): Users will be signed out immediately after a refresh token failure.
- 30 minute grace period: Users can stay signed in for up to 30 minutes after a refresh token failure.
- 60 minute grace period: Users can stay signed in for up to 60 minutes after a refresh token failure.

Network tags for AWS Wickr

You can apply tags to Wickr networks. You can then use those tags to search and filter your Wickr networks or track your AWS costs. You can configure network tags on the **Network home** page of the AWS Management Console for Wickr.

A tag is a key-value pair applied to a resource to hold metadata about that resource. Each tag is a label consisting of a key and a value. For more information on tags, see also What are tags? and Tagging use cases.

Topics

Manage network tags in AWS Wickr

You can manage network tags for your Wickr network.

Complete the following procedure to manage network tags for your Wickr network.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. On the **Network home** page, in the **Tags** section, choose **Manage tags**.
4. On the **Manage tags** page, you can complete one of the following options:

   - **Add new tags** — Enter new tags in the form of a key and a value pair. Choose **Add new tag** to add multiple key value pairs. Tags are case-sensitive. For more information, see Add a network tag in AWS Wickr.
   - **Edit existing tags** — Select the key or value text for an existing tag, and then enter the modification into the text box. For more information, see Edit a network tag in AWS Wickr.
   - **Remove existing tags** — Choose **Remove** button that is listed next to the tag you want to delete. For more information, see Remove a network tag in AWS Wickr.

Add a network tag in AWS Wickr

You can add a network tag to your Wickr network.

Complete the following procedure to add a tag to your Wickr network. For more information about managing tags, see Manage network tags in AWS Wickr.

1. On the **Network home** page, in the **Tags** section, choose **Add new tag**.
2. On the **Manage tags** page, choose **Add new tag**.
3. In the blank **Key** and **Value** fields that appear, enter the new tag key and value.
4. Choose **Save changes** to save the new tags.

Edit a network tag in AWS Wickr

You can edit a network tag to your Wickr network.

Complete the following procedure to edit a tag associated with your Wickr network. For more information about managing tags, see Manage network tags in AWS Wickr.

1. On the **Manage tags** page, edit the value of a tag.

> **Note**
>
> You can't edit the key of a tag. Instead, remove the key and value pair, and add a new tag using the new key.

2. Choose **Save changes** to save your edits.

Remove a network tag in AWS Wickr

You can remove a network tag to your Wickr network.

Complete the following procedure to remove a tag from your Wickr network. For more information about managing tags, see Manage network tags in AWS Wickr.

1. On the **Manage tags** page, choose **Remove** for the tag you want to remove.
2. Choose **Save changes** to save your edits.

Read receipts for AWS Wickr

Read receipts for AWS Wickr are notifications sent to the sender to show when their message has been read. These receipts are available in one-on-one conversations. A single check mark will appear for sent messages, and a solid circle with a check mark will appear for read messages. To see read receipts on messages during external conversations, both networks should have read receipts enabled.

Administrators can enable or disable read receipts in the administrator panel. This setting will be applied to the entire network.

Complete the following procedure to enable or disable read receipts.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Network polices**.
4. On the **Network polices** page, in the **Messaging** section, choose **Edit**.
5. Select the checkbox to **Enable** or **Disable** read receipts.
6. Choose **Save changes**.

Manage network plan for AWS Wickr

In the AWS Management Console for Wickr, you can manage your network plan based on your business needs.

To manage your network plan, complete the following procedure.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. On the **Network home** page, in the **Network details** section, choose **Edit**.
4. On the **Edit network details** page, choose your desired network plan. You can modify your current network plan by choosing one of the following:

   - **Standard —** For small and large business teams that need administrative controls and flexibility.
   - **Premium** or **Premium Free Trial —** For businesses that require the highest feature limits, granular administrative controls, and data retention.

     Administrators have the option to select a premium free trial, which is available for up to 30 users and lasts for three months. For AWS WickrGov, the premium free trial option allows up to 50 users and also last for three months. This offer is open to new and standard plans. During the premium free trial period, administrators can upgrade or downgrade to Premium or Standard plans

     > **Note**
     >
     > To stop usage and billing on your network, remove all users, including any suspended users from your network.

Premium free trial limitations

The following limitations apply to the premium free trial:

- If a plan has ever been enrolled in a premium free trial before, it will not be eligible for another trial.
- Only one network for each AWS account can be enrolled in a premium free trial.
- The guest user feature is not available during the premium free trial.
- If a standard network has more than 30 users (more than 50 users for AWS WickrGov), it will not be possible to upgrade to a premium free trial.

Data retention for AWS Wickr

AWS Wickr Data retention can retain all conversations in network. This includes direct message conversations and conversations in Groups or Rooms between in-network (internal) members and those with other teams (external) with whom your network is federated. Data retention is only available to AWS Wickr Premium plan users and enterprise customers who opt in for data retention. For more information on the Premium plan, see Wickr Pricing

When a network administrator configures and activates data retention for their network, all messages and files shared in their network are retained in accordance with the organization's compliance policies. These .txt file outputs are accessible by the network administrator in an external location (eg: local storage, Amazon S3 bucket, or any other storage as per user's choice), from where they can be analyzed, erased, or transferred.

> **Note**
>
> Wickr never accesses your messages and files. Therefore, it is your responsibility to configure a data retention system.

Topics

View data retention details in AWS Wickr

Complete the following procedure to view the data retention details for your Wickr network. You can also enable or disable data retention for your Wickr network.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Network polices**.
4. The **Network polices** page displays steps for setting up data retention, and the option to activate or deactivate the data retention feature. For more information about configuring data retention, see Configure data retention for AWS Wickr.

> **Note**
>
> When data retention is activated, a **Data Retention Turned On** message will be visible for all users in your network informing them of the retention-enabled network.

Configure data retention for AWS Wickr

To configure data retention for your AWS Wickr network, you must deploy the data retention bot Docker image to a container on a host, such as a local computer or an instance in Amazon Elastic Compute Cloud (Amazon EC2). After the bot is deployed, you can configure it to store data locally or in an Amazon Simple Storage Service (Amazon S3) bucket. You can also configure the data retention bot to use other AWS services like AWS Secrets Manager (Secrets Manager), Amazon CloudWatch (CloudWatch), Amazon Simple Notification Service (Amazon SNS), and AWS Key Management Service (AWS KMS). The following topics describe how to configure and run the data retention bot for your Wickr network.

Topics

- IAM policy to use data retention with AWS services
- Start the data retention bot for your Wickr network
- Stop the data retention bot for your Wickr network

Prerequisites to configure data retention for AWS Wickr

Before you get started, you must get the data retention bot name (labeled as **Username**) and initial password from the AWS Management Console for Wickr. You must specify both of these values the first time you start the data retention bot. You must also enable data retention in the console. For more information, see View data retention details in AWS Wickr.

Password for data retention bot in AWS Wickr

The first time you start the data retention bot, you specify the initial password using one of the following options:

- The `WICKRIO_BOT_PASSWORD` environment variable. The data retention bot environment variables are outlined in the Environment variables to configure data retention bot in AWS Wickr section later in this guide.
- The **password** value in Secrets Manager identified by the `AWS_SECRET_NAME` environment variable. The Secrets Manager values for the data retention bot are outlined in the Secrets Manager values for AWS Wickr section later in this guide.
- Enter the password when prompted by the data retention bot. You will need to run the data retention bot with interactive TTY access using the `-ti` option.

A new password will be generated when you configure the data retention bot for the first time. If you need to re-install the data retention bot, you use the generated password. The initial password is not valid after the initial installation of the data retention bot.

The new generated password will be displayed as shown in the following example.

> **Important**
>
> Save the password in a safe place. If you lose the password you will not be able to re-install the data retention bot. Don't share this password. It provides the ability to start data retention for your Wickr network.

```
*****************************************************************
**** GENERATED PASSWORD
**** DO NOT LOSE THIS PASSWORD, YOU WILL NEED TO ENTER IT EVERY TIME
**** TO START THE BOT
 "HuEXAMPLERAW4lGgEXAMPLEn"
 *****************************************************************
```

Storage options for AWS Wickr network

After data retention is enabled and the data retention bot is configured for your Wickr network, it will capture all messages and files sent within your network. Messages are saved in files which are limited to a specific size or time limit that can be configured using an environment variable. For more information, see Environment variables to configure data retention bot in AWS Wickr.

You can configure one of the following options for storing this data:

- Store all captured messages and files locally. This is the default option. It's your responsibility to move local files to another system for long-term storage, and to make sure the host disk does not run out of memory or space.
- Store all captured messages and files in an Amazon S3 bucket. The data retention bot will save all decrypted messages and files to the Amazon S3 bucket you specify. The captured messages and files are removed from the host machine after they are successfully saved to the bucket.
- Store all captured messages and files encrypted in an Amazon S3 bucket. The data retention bot will re-encrypt all captured messages and files using a key that you supply and save them to the Amazon S3 bucket you specify. The captured messages and files are removed from the host machine after they are successfully re-encrypted and saved to the bucket. You will need software to decrypt the messages and files.

  For more information about creating an Amazon S3 bucket to use with your data retention bot, see Creating a bucket in the *Amazon S3 User Guide*

Environment variables to configure data retention bot in AWS Wickr

You can use the following environment variables to configure the data retention bot. You set these environment variables using the `-e` option when you run the data retention bot Docker image. For more information, see Start the data retention bot for your Wickr network.

> **Note**
>
> These environment variables are optional unless otherwise specified.

Use the following environment variables to specify the data retention bot credentials:

- `WICKRIO_BOT_NAME` — The name of the data retention bot. This variable is *required* when you run the data retention bot Docker image.
- `WICKRIO_BOT_PASSWORD` — The initial password for the data retention bot. For more information, see Prerequisites to configure data retention for AWS Wickr. This variable is *required* if you don't plan to start the data retention bot with a password prompt or you don't plan to use Secrets Manager to store the data retention bot credentials.

Use the following environment variables to configure the default data retention streaming capabilities:

- `WICKRIO_COMP_MESGDEST` – The path name to the directory where messages will be streamed. The default value is `/tmp/<botname>/compliance/messages`.
- `WICKRIO_COMP_FILEDEST` – The path name to the directory where files will be streamed. The default value is `/tmp/<botname>/compliance/attachments`.
- `WICKRIO_COMP_BASENAME` – The base name for the received messages files. The default value is `receivedMessages`.
- `WICKRIO_COMP_FILESIZE` – The maximum file size for a received messages file in kibibyte (KiB). A new file is started when the max size is reached. The default value is `1000000000`, as in 1024 GiB.
- `WICKRIO_COMP_TIMEROTATE` – The amount of time, in minutes, for which the data retention bot will put received messages into a received messages file. A new file is started when the time limit is reached. You can only use the file size or time to limit the size of the received messages file. The default value is `0`, as in no limit.

Use the following environment variable to define the default AWS Region to use.

- `AWS_DEFAULT_REGION` – The default AWS Region to use for AWS services like Secrets Manager (not used for Amazon S3 or AWS KMS). The `us-east-1` Region is used by default if this environment variable is not defined.

Use the following environment variables to specify the Secrets Manager secret to use when you opt to use Secrets Manager to store the data retention bot credentials and AWS service information. For more information about the values you can store in Secrets Manager see Secrets Manager values for AWS Wickr.

- `AWS_SECRET_NAME` – The name of the Secrets Manager secret that contains the credentials and AWS service information needed by the data retention bot.
- `AWS_SECRET_REGION` – The AWS Region that the AWS secret is located in. If you are using AWS secrets and this value is not defined the `AWS_DEFAULT_REGION` value will be used.

> **Note**
>
> You can store all of the following environment variables as values in Secrets Manager. If you opt to use Secrets Manager, and you store these values there, then you don't need to specify them as environment variables when you run the data retention bot Docker image. You only need to specify the `AWS_SECRET_NAME` environment variable described earlier in this guide. For more information, see Secrets Manager values for AWS Wickr.

Use the following environment variables to specify the Amazon S3 bucket when you opt to store messages and files to a bucket.

- `WICKRIO_S3_BUCKET_NAME` – The name of the Amazon S3 bucket where messages and files will be stored.
- `WICKRIO_S3_REGION` – The AWS Region of the Amazon S3 bucket where messages and files will be stored.
- `WICKRIO_S3_FOLDER_NAME` – The optional folder name in the Amazon S3 bucket where messages and files will be stored. This folder name will be preceded with the key for messages and files saved to the Amazon S3 bucket.

Use the following environment variables to specify the AWS KMS details when you opt to use client side encryption to re-encrypt files when saving them to an Amazon S3 bucket.

- `WICKRIO_KMS_MSTRKEY_ARN` – The Amazon Resource Name (ARN) of the AWS KMS master key used to re-encrypt the message files and files on the data retention bot before they are saved to the Amazon S3 bucket.

- `WICKRIO_KMS_REGION` – The AWS Region where the AWS KMS master key is located.

Use the following environment variable to specify the Amazon SNS details when you opt to send data retention events to an Amazon SNS topic. The events sent include startup, shutdown, as well as error conditions.

- `WICKRIO_SNS_TOPIC_ARN` – The ARN of the Amazon SNS topic that you want data retention events sent to.

Use the following environment variable to send data retention metrics to CloudWatch. If specified, the metrics will be generated every 60 seconds.

- `WICKRIO_METRICS_TYPE` – Set the value of this environment variable to `cloudwatch` to send metrics to CloudWatch.

Secrets Manager values for AWS Wickr

You can use Secrets Manager to store the data retention bot credentials and AWS service information. For more information about creating a Secrets Manager secret, see Create an AWS Secrets Manager secret in the *Secrets Manager User Guide*.

The Secrets Manager secret can have the following values:

- `password` – The data retention bot password.
- `s3_bucket_name` – The name of the Amazon S3 bucket where messages and files will be stored. If not set, the default file streaming will be used.
- `s3_region` – The AWS Region of the Amazon S3 bucket where messages and files will be stored.
- `s3_folder_name` – The optional folder name in the Amazon S3 bucket where messages and files will be stored. This folder name will be preceded with the key for messages and files saved to the Amazon S3 bucket.
- `kms_master_key_arn` – The ARN of the AWS KMS master key used to re-encrypt the message files and files on the data retention bot before they are saved to the Amazon S3 bucket.
- `kms_region` – The AWS Region where the AWS KMS master key is located.
- `sns_topic_arn` – The ARN of the Amazon SNS topic that you want data retention events sent to.

IAM policy to use data retention with AWS services

If you plan to use other AWS services with the Wickr data retention bot, you must ensure the host has the appropriate AWS Identity and Access Management (IAM) role and policy to access them. You can configure the data retention bot to use Secrets Manager, Amazon S3, CloudWatch, Amazon SNS, and AWS KMS. The following IAM policy allows access to specific actions for these services.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "secretsmanager:GetSecretValue",
                "sns:Publish",
                "cloudwatch:PutMetricData",
                "kms:GenerateDataKey"
            ],
            "Resource": "*"
        }
    ]
}
```

You can create an IAM policy that is more strict by identifying the specific objects for each service that you want to allow the containers on your host to access. Remove the actions for the AWS services that you do not intend to use. For example, if you intent to use only an Amazon S3 bucket, then use the following policy, which removes the `secretsmanager:GetSecretValue`, `sns:Publish`, `kms:GenerateDataKey`, and `cloudwatch:PutMetricData` actions.

JSON

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

```
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "*"
        }
    ]
}
```

If you are using an Amazon Elastic Compute Cloud (Amazon EC2) instance to host your data retention bot, create an IAM role using the Amazon EC2 common case and assign a policy using the policy definition from above.

Start the data retention bot for your Wickr network

Before you run the data retention bot, you should determine how you want to configure it. If you plan to run the bot on a host that:

- Will not have access to AWS services, then your options are limited. In that case you will use the default message streaming options. You should decide whether you want to limit the size of the captured message files to a specific size or time interval. For more information, see Environment variables to configure data retention bot in AWS Wickr.
- Will have access to AWS services, then you should create a Secrets Manager secret to store the bot credentials, and AWS service configuration details. After the AWS services are configured, you can proceed to start the data retention bot Docker image. For more information about the details you can store in a Secrets Manager secret, see Secrets Manager values for AWS Wickr

The following sections show example commands to run the data retention bot Docker image. In each of the example commands, replace the following example values with your own:

- *compliance_1234567890_bot* with the name of your data retention bot.
- *password* with the password for your data retention bot.
- *wickr/data/retention/bot* with the name of your Secrets Manager secret to use with your data retention bot.
- *bucket-name* with the name of the Amazon S3 bucket where messages and files will be stored.
- *folder-name* with the folder name in the Amazon S3 bucket where messages and files will be stored.

- *us-east-1* with the AWS Region of the resource you're specifying. For example, the Region of the AWS KMS master key or the Region of the Amazon S3 bucket.
- *arn:aws:kms:us-east-1:111122223333:key/12345678-1234-abcde-a617-abababababab* with the Amazon Resource Name (ARN) of your AWS KMS master key to use to re-encrypt message files and files.

Start the bot with password environment variable (no AWS service)

The following Docker command starts the data retention bot. The password is specified using the WICKRIO_BOT_PASSWORD environment variable. The bot starts using the default file streaming, and using the default values defined in the Environment variables to configure data retention bot in AWS Wickr section of this guide.

```
docker run -v /opt/compliance_1234567890_bot:/tmp/compliance_1234567890_bot \
-d --restart on-failure:5 --name="compliance_1234567890_bot" -ti \
-e WICKRIO_BOT_NAME='compliance_1234567890_bot' \
-e WICKRIO_BOT_PASSWORD='password' \
wickr/bot-compliance-cloud:latest
```

Start the bot with password prompt (no AWS service)

The following Docker command starts the data retention bot. Password is entered when prompted by the data retention bot. It will start using the default file streaming using the default values defined in the Environment variables to configure data retention bot in AWS Wickr section of this guide.

```
docker run -v /opt/compliance_1234567890_bot:/tmp/compliance_1234567890_bot \
-d --restart on-failure:5 --name="compliance_1234567890_bot" -ti \
-e WICKRIO_BOT_NAME='compliance_1234567890_bot' \
wickr/bot-compliance-cloud:latest

docker attach compliance_1234567890_bot
.
.
.
Enter the password:************
Re-enter the password:************
```

Run the bot using the `-ti` option to receive the password prompt. You should also run the `docker attach` *<container ID or container name>* command immediately after starting the

docker image so that you get the password prompt. You should run both of these commands in a script. If you attach to the docker image and don't see the prompt, press **Enter** and you will see the prompt.

Start the bot with 15 minute message file rotation (no AWS service)

The following Docker command starts the data retention bot using environment variables. It also configures it to rotate the received messages files to 15 minutes.

```
docker run -v /opt/compliance_1234567890_bot:/tmp/compliance_1234567890_bot --
network=host \
-d --restart on-failure:5 --name="compliance_1234567890_bot" -ti \
-e WICKRIO_BOT_NAME='compliance_1234567890_bot' \
-e WICKRIO_BOT_PASSWORD='password' \
-e WICKRIO_COMP_TIMEROTATE=15 \
wickr/bot-compliance-cloud:latest
```

Start the bot and specify the initial password with Secrets Manager

You can use the Secrets Manager to identify the data retention bot's password. When you start the data retention bot, you will need to set an environment variable that specifies the Secrets Manager where this information is stored.

```
docker run -v /opt/compliance_1234567890_bot:/tmp/compliance_1234567890_bot --
network=host \
-d --restart on-failure:5 --name="compliance_1234567890_bot" -ti \
-e WICKRIO_BOT_NAME='compliance_1234567890_bot' \
-e AWS_SECRET_NAME='wickrpro/alpha/new-3-bot' \
wickr/bot-compliance-cloud:latest
```

The `wickrpro/compliance/compliance_1234567890_bot` secret has the following secret value in it, shown as plaintext.

```
{
    "password":"password"
}
```

Start the bot and configure Amazon S3 with Secrets Manager

You can use the Secrets Manager to host the credentials, and the Amazon S3 bucket information. When you start the data retention bot, you will need to set an environment variable that specifies the Secrets Manager where this information is stored.

```
docker run -v /opt/compliance_1234567890_bot:/tmp/compliance_1234567890_bot --
network=host \
-d --restart on-failure:5 --name="compliance_1234567890_bot" -ti \
 -e WICKRIO_BOT_NAME='compliance_1234567890_bot' \
 -e AWS_SECRET_NAME='wickrpro/alpha/compliance_1234567890_bot' \
wickr/bot-compliance-cloud:latest
```

The `wickrpro/compliance/compliance_1234567890_bot` secret has the following secret value in it, shown as plaintext.

```
{
    "password":"password",
    "s3_bucket_name":"bucket-name",
    "s3_region":"us-east-1",
    "s3_folder_name":"folder-name"
}
```

Messages and files received by the bot will be put in the `bot-compliance` bucket in the folder named `network1234567890`.

Start the bot and configure Amazon S3 and AWS KMS with Secrets Manager

You can use the Secrets Manager to host the credentials, the Amazon S3 bucket, and AWS KMS master key information. When you start the data retention bot, you will need to set an environment variable that specifies the Secrets Manager where this information is stored.

```
docker run -v /opt/compliance_1234567890_bot:/tmp/compliance_1234567890_bot --
network=host \
-d --restart on-failure:5 --name="compliance_1234567890_bot" -ti \
 -e WICKRIO_BOT_NAME='compliance_1234567890_bot' \
 -e AWS_SECRET_NAME='wickrpro/alpha/compliance_1234567890_bot' \
wickr/bot-compliance-cloud:latest
```

The `wickrpro/compliance/compliance_1234567890_bot` secret has the following secret value in it, shown as plaintext.

```
{
    "password":"password",
    "s3_bucket_name":"bucket-name",
    "s3_region":"us-east-1",
    "s3_folder_name":"folder-name",
    "kms_master_key_arn":"arn:aws:kms:us-east-1:111122223333:key/12345678-1234-abcde-
a617-abababababab",
    "kms_region":"us-east-1"
}
```

Messages and files received by the bot will be encrypted using the KMS key identified by the ARN value, then put in the “bot-compliance'” bucket in the folder named “network1234567890”. Make sure you have the appropriate IAM policy setup.

Start the bot and configure Amazon S3 using environment variables

If you don't want to use Secrets Manager to host the data retention bot credentials, you can start the data retention bot Docker image with the following environment variables. You must identify the name of the data retention bot using the `WICKRIO_BOT_NAME` environment variable.

```
docker run -v /opt/compliance_1234567890_bot:/tmp/compliance_1234567890_bot --
network=host \
-d --restart on-failure:5 --name="compliance_1234567890_bot" -ti \
-e WICKRIO_BOT_NAME='compliance_1234567890_bot' \
-e WICKRIO_BOT_PASSWORD='password' \
-e WICKRIO_S3_BUCKET_NAME='bot-compliance' \
-e WICKRIO_S3_FOLDER_NAME='network1234567890' \
-e WICKRIO_S3_REGION='us-east-1' \
wickr/bot-compliance-cloud:latest
```

You can use environment values to identify the data retention bot’s credentials, information about Amazon S3 buckets, and configuration information for the default file streaming.

Stop the data retention bot for your Wickr network

The software running on the data retention bot will capture `SIGTERM` signals and gracefully shutdown. Use the `docker stop` *<container ID or container name>* command, as shown in the following example, to issue the `SIGTERM` command to the data retention bot Docker image.

```
docker stop compliance_1234567890_bot
```

Get the data retention logs for your Wickr network

The software running on the data retention bot Docker image will output to log files in the `/tmp/<botname>/logs` directory. They will rotate to a maximum of 5 files. You can get the logs by running the following command.

```
docker logs <botname>
```

Example:

```
docker logs compliance_1234567890_bot
```

Data retention metrics and events for your Wickr network

Following are the Amazon CloudWatch (CloudWatch) metrics and Amazon Simple Notification Service (Amazon SNS) events that are currently supported by the 5.116 version of the AWS Wickr data retention bot.

Topics

CloudWatch metrics for your Wickr network

Metrics are generated by the bot in 1 minute intervals and transmitted to the CloudWatch service associated with the account the data retention bot Docker image is running on.

Following are the existing metrics supported by the data retention bot.

Metric	Description
Messages_Rx	Messages received.
Messages_Rx_Failed	Failures to process received messages.
Messages_Saved	Messages saved to the received messages file.

Metric	Description
Messages_Saved_Failed	Failures to save messages to the received messages file.
Files_Saved	Files received.
Files_Saved_Bytes	Number of bytes for files received.
Files_Saved_Failed	Failures to save files.
Logins	Logins (normally this will be 1 for each interval).
Login_Failures	Failures to login (normally this will be 1 for each interval).
S3_Post_Errors	Errors posting message files and files to Amazon S3 bucket.
Watchdog_Failures	Watchdog failures.
Watchdog_Warnings	Watchdog warnings.

Metrics are generated to be consumed by CloudWatch. The namespace used for bots is `WickrIO`. Each metric has an array of dimensions. Following is the list of dimensions that are posted with the above metrics.

Dimension	Value
Id	The bot's username.
Device	Description of specific bot device or instance. Useful if you are running multiple bot devices or instances.
Product	The product for the bot. Can be `WickrPro_` or `WickrEnterprise_` with `Alpha`, `Beta`, or `Production` appended.

Dimension	Value
BotType	The bot type. Labeled as **Compliance** for the compliance bots.
Network	The ID of the associated network.

Amazon SNS events for your Wickr network

The following events are posted to the Amazon SNS topic defined by the Amazon Resource Name (ARN) value identified using the `WICKRIO_SNS_TOPIC_ARN` environment variable or the `sns_topic_arn` Secrets Manager secret value. For more information, see Environment variables to configure data retention bot in AWS Wickr and Secrets Manager values for AWS Wickr.

Events generated by the data retention bot are sent as JSON strings. The following values are included in the events as of the 5.116 version of the data retention bot.

Name	Value
complianceBot	The username of the data retention bot.
dataTime	The date and time when the event occurred.
device	A description of the specific bot device or instance. Useful if you are running multiple bot instances.
dockerImage	The Docker image associated with the bot.
dockerTag	The tag or version of the Docker image.
message	The event message. For more information see Critical events and Normal events.
notificationType	This value will be `Bot Event`.
severity	The severity of the event. Can be `normal` or `critical`.

You must subscribe to the Amazon SNS topic so that you can receive the events. If you subscribe using an email address, an email will be sent to you containing information similar to the following example.

```
{
"complianceBot": "compliance_1234567890_bot",
 "dateTime": "2022-10-12T13:05:39",
 "device": "Desktop 1234567890ab",
 "dockerImage": "wickr/bot-compliance-cloud",
 "dockerTag": "5.116.13.01",
 "message": "Logged in",
 "notificationType": "Bot Event",
 "severity": "normal"
}
```

Critical events

These events will cause the bot to stop or restart. The number of restarts is limited to avoid causing other issues.

Login failures

Following are the possible events that can be generated when the bot fails to login. Each message will indicate the reason for the login failure.

Event type	Event message
failedlogin	Bad credentials. Check the password.
failedlogin	User not found.
failedlogin	Account or device is suspended.
provisioning	User exited the command.
provisioning	Bad password for the `config.wickr` file.
provisioning	Cannot read the `config.wickr` file.
failedlogin	Logins all failed.
failedlogin	New user but database already exists.

More critical events

Event type	Event messages
Suspended Account	WickrIOClientMain::slotAdminUserSuspend: code(%1): reason: %2“
BotDevice Suspended	Device is suspended!
WatchDog	The SwitchBoard system is down for more than <*N*> minutes
S3 Failures	Failed to put file <*file-name* ≫ on S3 bucket. Error: <*AWS-error* >
Fallback Key	SERVER SUBMIITED FALLBACK KEY: Is not a recognized client active fallback key. Please submit logs to desktop engineering.

Normal events

Following are the events that warn you about normal operating occurrences. Too many occurrences of these types of events within a specific time period may be cause for concern.

Device added to account

This event is generated when a new device is added to the data retention bot account. Under some circumstances, this can be an important indication that someone has created an instance of the data retention bot. Following is the message for this event.

```
A device has been added to this account!
```

Bot logged in

This event is generated when the bot has successfully logged in. Following is the message for this event.

```
Logged in
```

Shutting down

This event is generated when the bot is shutting down. If the user did not explicitly initiate this, it could be an indication of a problem. Following is the message for this event.

```
Shutting down
```

Updates available

This event is generated when the data retention bot is started and it identifies that there is a newer version of the associated Docker image available. This event is generated when the bot starts, and on a daily basis. This event includes the `versions` array field which identifies the new versions that are available. Following is an example of what this event looks like.

```
{
  "complianceBot": "compliance_1234567890_bot",
  "dateTime": "2022-10-12T13:05:55",
  "device": "Desktop 1234567890ab",
  "dockerImage": "wickr/bot-compliance-cloud",
  "dockerTag": "5.116.13.01",
  "message": "There are updates available",
  "notificationType": "Bot Event",
  "severity": "normal",
  "versions": [
    "5.116.10.01"
  ]
}
```

What is ATAK?

The Android Team Awareness Kit (ATAK)—or Android Tactical Assault Kit (also ATAK) for military use—is a smart phone geospatial infrastructure and situational awareness application that enables safe collaboration over geography. While it was initially designed for use in combat zones, ATAK has been adapted to fit the missions of local, state, and federal agencies.

Topics

- Enable ATAK in the Wickr Network Dashboard
- Additional information about ATAK
- Install and pair the Wickr plugin for ATAK

- Unpair the Wickr Plugin for ATAK
- Dial and receive a call in ATAK
- Send a file in ATAK
- Send a secure voice message (Push-to-talk) in ATAK
- Pinwheel (Quick Access) for ATAK
- Navigation for ATAK

Enable ATAK in the Wickr Network Dashboard

AWS Wickr supports many agencies that use Android Tactical Assault Kit (ATAK). However, until now, ATAK operators that use Wickr have had to leave the application in order to do so. To help reduce disruptions and operational risk, Wickr has developed a plugin that enhances ATAK with secure communication features. With the Wickr plugin for ATAK, users can message, collaborate, and transfer files on Wickr within the ATAK application. This eliminates interruptions, and the complexity of configuration with ATAK's chat features.

Enable ATAK in the Wickr Network Dashboard

Complete the following procedure to enable ATAK in the Wickr Network Dashboard.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Security groups**.
4. On the **Security groups** page, select the desired security group for which you want to enable ATAK.
5. On the **Integration** tab, in the **ATAK plugin** section, choose **Edit**.
6. On the **Edit ATAK plugin** page, select the checkbox **Enable ATAK plugin**.
7. Choose **Add new package**
8. Enter the package name in the **Packages** text box. You can enter one of the following values depending on the version of the ATAK that your users will install and use:

    - `com.atakmap.app.civ` — Enter this value into the **Packages** text box if your Wickr end users are going to install and use the civilian version of the ATAK application on their Android devices.

- `com.atakmap.app.mil` — Enter this value into the **Packages** text box if your Wickr end users are going to install and use the military version of the ATAK application on their Android devices.

9. Choose **Save**.

   ATAK is now enabled for the selected Wickr Network, and the selected Security Group. You should ask the Android users in the security group for which you enabled the ATAK functionality to install the Wickr plugin for ATAK. For more information, see Install and pair the Wickr ATAK plugin.

Additional information about ATAK

For more information about the Wickr plugin for ATAK, see the following:

- Wickr ATAK Plugin Overview
- Additional Wickr ATAK Plugin Information

Install and pair the Wickr plugin for ATAK

The Android Team Awareness Kit (ATAK) is an Android solution used by the US military, state, and governmental agencies that require situational awareness capabilities for mission planning, execution, and incident response. ATAK has a plugin architecture which allows developers to add functionality. It enables users to navigate using GPS and geospatial map data overlaid with real-time situational awareness of ongoing events. In this document, we show you how to install the Wickr plugin for ATAK on an Android device and pair it with the Wickr client. This allows you to message and collaborate on Wickr without exiting the ATAK application.

Install the Wickr plugin for ATAK

Complete the following procedure to install the Wickr plugin for ATAK on an Android device.

1. Go to the Google Play store, and install the Wickr for ATAK plugin.
2. Open the ATAK application on your Android device.

3. In the ATAK application, choose the menu icon
   ()
   at the top-right of the screen, and then choose **Plugins**.
4. Choose **Import**.
5. On the **Select Import Type** pop-up, choose **Local SD** and navigate to where you saved the Wickr plugin for ATAK .apk file.
6. Choose the plugin file and follow the prompts to install it.

   > **Note**
   >
   > If you are asked to send the plugin file for scanning, choose **No**.

7. The ATAK application will ask if you would like to load the plugin. Choose **OK**.

The Wickr plugin for ATAK is now installed. Continue to the following Pair ATAK with Wickr section to finish the process.

Pair ATAK with Wickr

Complete the following procedure to pair the ATAK application with Wickr after you successfully installed the Wickr plugin for ATAK.

1. In the ATAK application, choose the menu icon
   ()
   at the top-right of the screen, and then choose **Wickr Plugin**.
2. Choose **Pair Wickr**.

   A notification prompt will appear asking you to review permissions for the Wickr plugin for ATAK. If the notification prompt doesn't appear, open the Wickr client and go to **Settings**, then **Connected Apps**. You should see the plugin under the **Pending** section of the screen.
3. Choose **Approve** to pair.
4. Choose **Open Wickr ATAK Plugin** button to go back to the ATAK application.

   You have now successfully paired the ATAK plugin and Wickr, and can use the plugin to send messages and collaborate using Wickr without exiting the ATAK application.

Unpair the Wickr Plugin for ATAK

You can unpair the Wickr plugin for ATAK.

Complete the following procedure to unpair the ATAK plugin with Wickr.

1. In the native app, choose **Settings**, and then choose **Connected Apps**.
2. On the **Connected Apps** screen, choose **Wickr ATAK Plugin**.
3. On the **Wickr ATAK Plugin** screen, choose **Remove** at the bottom of the screen.

   You have now successfully unpaired the Wickr plugin for ATAK.

Dial and receive a call in ATAK

You can dial and receive a call in the Wickr plugin for ATAK.

Complete the following procedure to dial and receive a call.

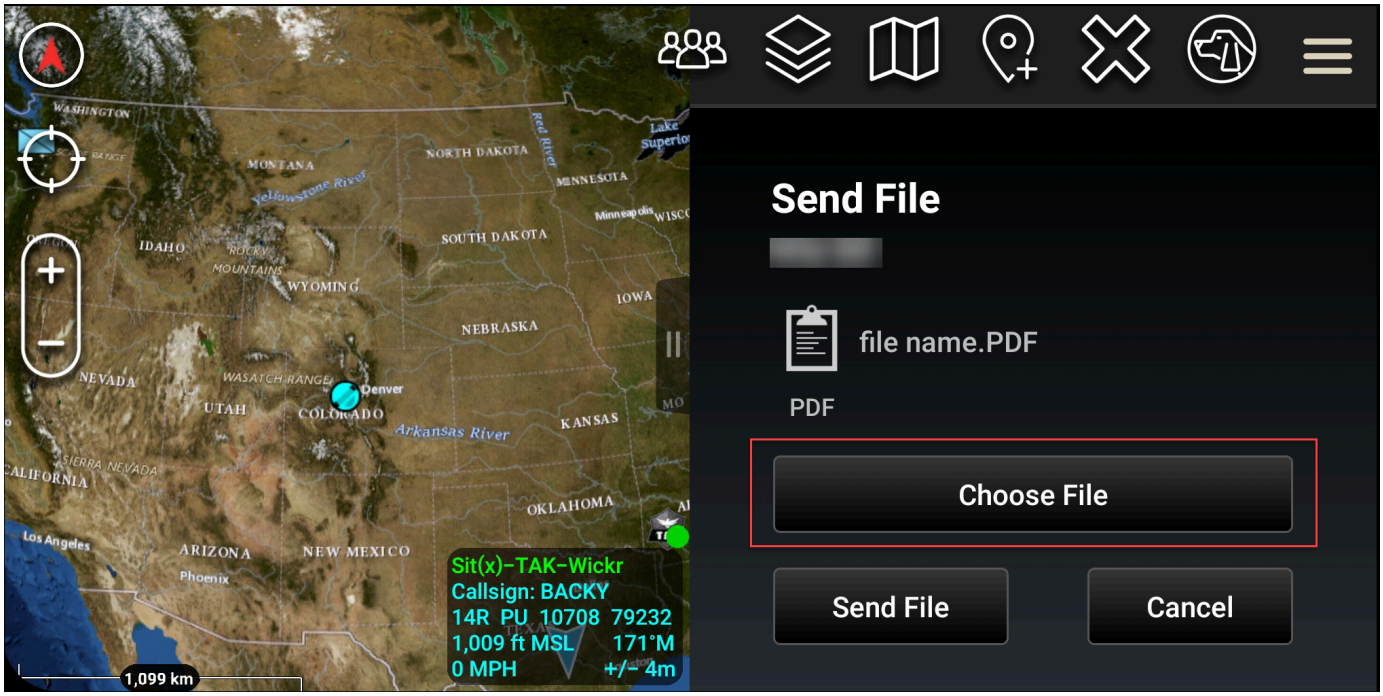
1. Open a chat window.
2. In the **Map** view, choose the icon for the user you want to call.
3. Choose the phone icon at the top-right of the screen.
4. Once connected, you can return to the ATAK plugin view and receive a call.

Send a file in ATAK

You can send a file in the Wickr plugin for ATAK.

Complete the following procedure to send a file.

1. Open a chat window.
2. In the **Map** view, search for the user that you want to send a file.
3. When you find the user that you want to send a file, select their name.
4. On the **Send File** screen, select **Choose File**, and then navigate to the file that you want to send.



5. On the browser window, choose the desired file.
6. On the **Send File screen**, choose **Send File**.

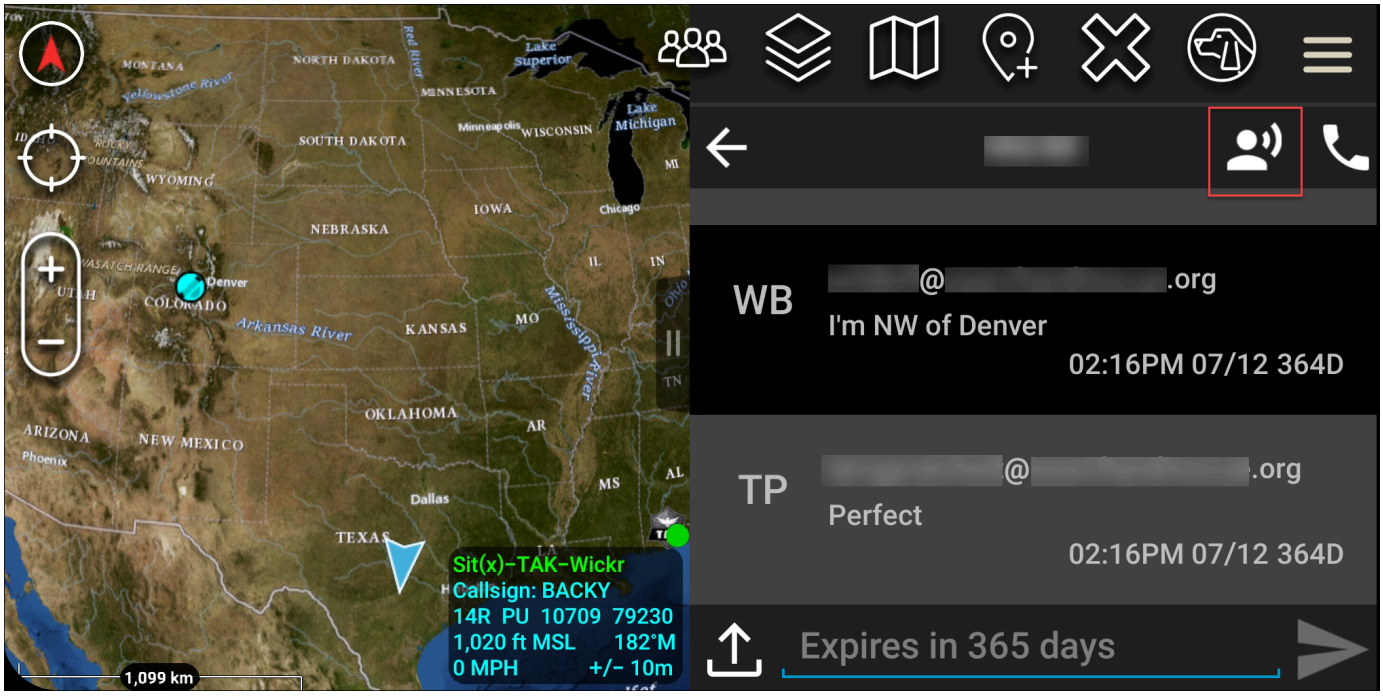
   The download icon displays, indicating the file you selected is being downloaded.

Send a secure voice message (Push-to-talk) in ATAK

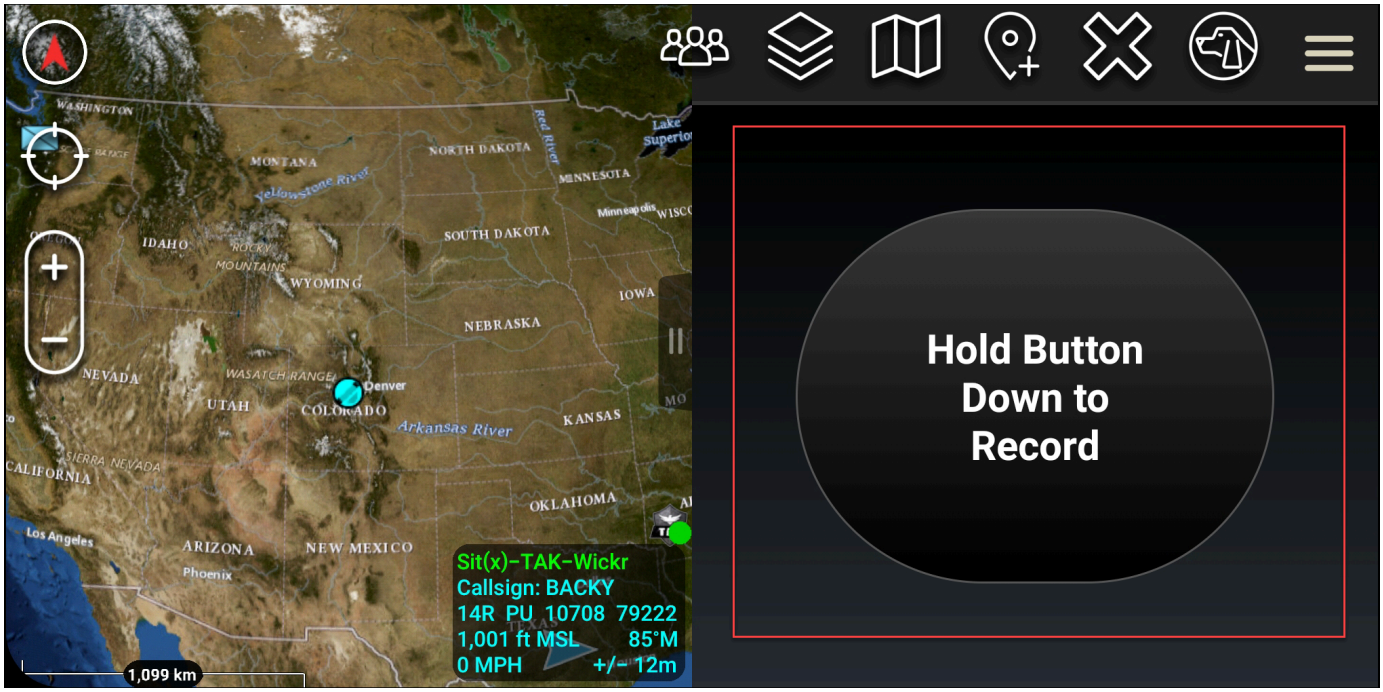
You can send a secure voice message (Push-to-talk) in the Wickr plugin for ATAK.

Complete the following procedure to send a secure voice message.

1. Open a chat window.
2. Choose the Push-to-Talk icon at the top of the screen, indicated by an icon of a person talking.



3. Select and hold the **Hold Button Down to Record** button.



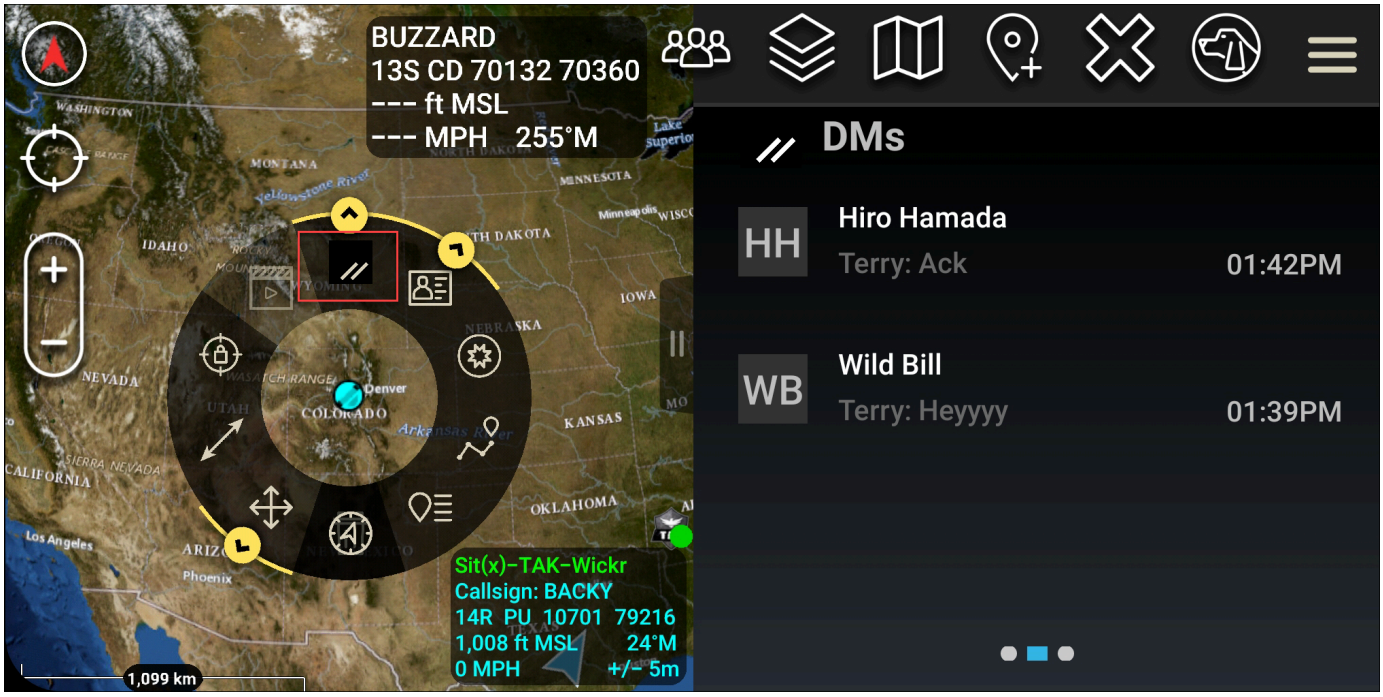
4. Record your message.
5. After you record your message, release the button to send.

Pinwheel (Quick Access) for ATAK

The pinwheel or quick access feature is used for one-one-one conversations or direct messages.

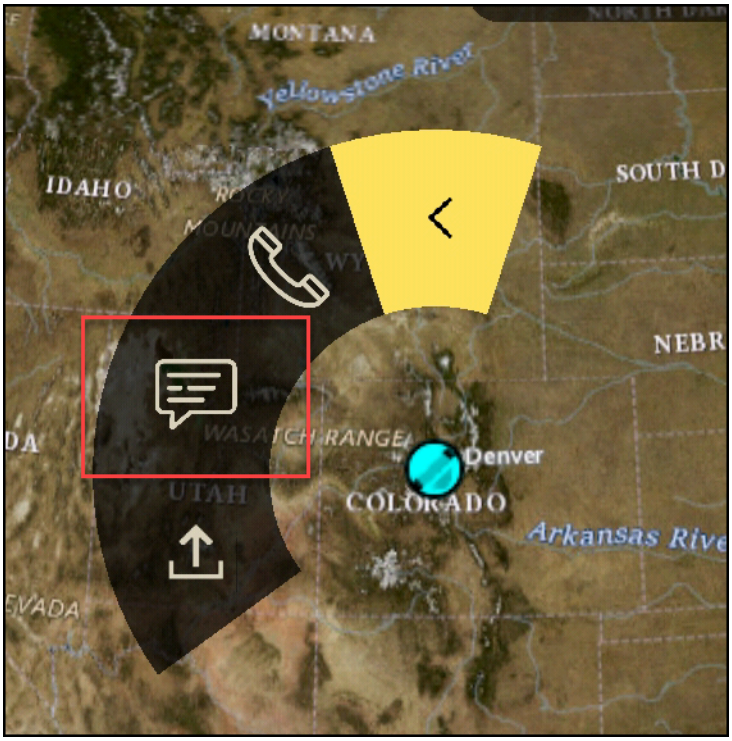
Complete the following procedure to use the pinwheel.

1. Open the split screen view of the ATAK map and the Wickr for ATAK plugin simultaneously. The map displays your teammates or assets on the map view.
2. Choose the user icon to open the pinwheel.
3. Choose the Wickr icon to view the available options for the selected user.

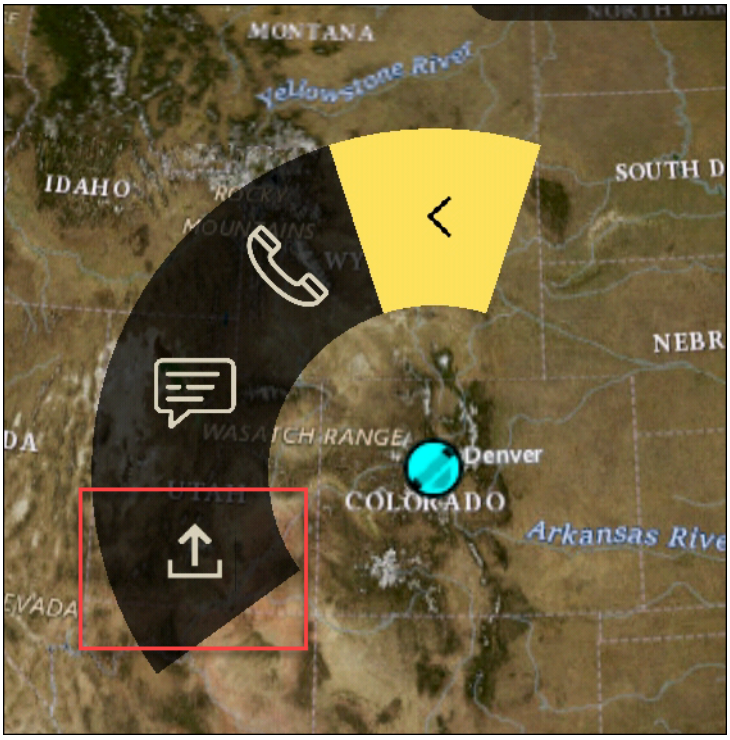


4. On the pinwheel, choose one of the following icons:
   - **Phone**: Choose to call.

- **Message**: Choose to chat.



- **File send**: Choose to send a file.



Navigation for ATAK

The plugin UI contains three plugin views that are indicated by the blue and white shapes at the bottom-right of the screen. Swipe left and right to navigate between the views.

- **Contacts view**: Create a direct message group or room conversation.
- **DMs view**: Create a one-to-one conversation. Chat functionality works as in the Wickr native app. This functionality allows you to remain in the Map view and communicate with others on the plugin.
- **Rooms view**: The existing rooms in the native app are ported over. Anything done in the plugin reflects in the Wickr native app.

> **Note**
>
> Certain functions, such as deleting a room, can only be performed in the native app and in person to prevent unintended modification by users and interference cause by field equipment.

Ports and domains to allow list for your Wickr network

Allow list the following ports to ensure Wickr functions correctly:

Ports

- TCP port 443 (for messages and attachments)
- UDP ports 16384-16584 (for calling)

Domains and addresses to allowlist by Region

If you need to allowlist all possible calling domains and server IP addresses, see the following list of potential CIDRs by Region. Check this list periodically, as it is subject to change.

> **Note**
>
> Registration and verification emails are sent from donotreply@wickr.email.

US East (N. Virginia)

Domains:	• gw-pro-prod.wickr.com • api.messaging.wickr.us-east-1.amazonaws.com
Calling CIDR addresses:	• 44.211.195.0/27 • 44.213.83.32/28
Calling IP addresses:	• 44.211.195.0 • 44.211.195.1 • 44.211.195.2 • 44.211.195.3 • 44.211.195.4 • 44.211.195.5 • 44.211.195.6 • 44.211.195.7

	• 44.211.195.8 • 44.211.195.9 • 44.211.195.10 • 44.211.195.11 • 44.211.195.12 • 44.211.195.13 • 44.211.195.14 • 44.211.195.15 • 44.211.195.16 • 44.211.195.17 • 44.211.195.18 • 44.211.195.19 • 44.211.195.20 • 44.211.195.21 • 44.211.195.22 • 44.211.195.23 • 44.211.195.24 • 44.211.195.25 • 44.211.195.26 • 44.211.195.27 • 44.211.195.28 • 44.211.195.29 • 44.211.195.30 • 44.211.195.31 • 44.213.83.32 • 44.213.83.33 • 44.213.83.34 • 44.213.83.35 • 44.213.83.36 • 44.213.83.37

	• 44.213.83.38 • 44.213.83.39 • 44.213.83.40 • 44.213.83.41 • 44.213.83.42 • 44.213.83.43 • 44.213.83.44 • 44.213.83.45 • 44.213.83.46 • 44.213.83.47

Asia Pacific (Malaysia)

Domains:	• gw-pro-prod.wickr.com • api.messaging.wickr.ap-southeast-5.amazonaws.com
Calling CIDR addresses:	• 43.216.226.160/28
Calling IP addresses:	• 43.216.226.160 • 43.216.226.161 • 43.216.226.162 • 43.216.226.163 • 43.216.226.164 • 43.216.226.165 • 43.216.226.166 • 43.216.226.167 • 43.216.226.168 • 43.216.226.169 • 43.216.226.170 • 43.216.226.171 • 43.216.226.172

	• 43.216.226.173 • 43.216.226.174 • 43.216.226.175

Asia Pacific (Singapore)

Domain:	• gw-pro-prod.wickr.com • api.messaging.wickr.ap-southeast-1.amazonaws.com
Calling CIDR addresses:	• 47.129.23.144/28
Calling IP addresses:	• 47.129.23.144 • 47.129.23.145 • 47.129.23.146 • 47.129.23.147 • 47.129.23.148 • 47.129.23.149 • 47.129.23.150 • 47.129.23.151 • 47.129.23.152 • 47.129.23.153 • 47.129.23.154 • 47.129.23.155 • 47.129.23.156 • 47.129.23.157 • 47.129.23.158 • 47.129.23.159

Asia Pacific (Sydney)

Domain:	• gw-pro-prod.wickr.com

	• api.messaging.wickr.ap-southeast-2.amazonaws.com
Calling CIDR addresses:	• 3.27.180.208/28
Calling IP addresses:	• 3.27.180.208 • 3.27.180.209 • 3.27.180.210 • 3.27.180.211 • 3.27.180.212 • 3.27.180.213 • 3.27.180.214 • 3.27.180.215 • 3.27.180.216 • 3.27.180.217 • 3.27.180.218 • 3.27.180.219 • 3.27.180.220 • 3.27.180.221 • 3.27.180.222 • 3.27.180.223

Asia Pacific (Tokyo)

Domain:	• gw-pro-prod.wickr.com • api.messaging.wickr.ap-northeast-1.amazonaws.com
Calling CIDR addresses:	• 57.181.142.240/28
Calling IP addresses:	• 57.181.142.240 • 57.181.142.241 • 57.181.142.242

	• 57.181.142.243 • 57.181.142.244 • 57.181.142.245 • 57.181.142.246 • 57.181.142.247 • 57.181.142.248 • 57.181.142.249 • 57.181.142.250 • 57.181.142.251 • 57.181.142.252 • 57.181.142.253 • 57.181.142.254 • 57.181.142.255

Canada (Central)

Domain:	• gw-pro-prod.wickr.com • api.messaging.wickr.ca-central-1.amazonaws.com
Calling CIDR addresses:	• 15.156.152.96/28
Calling IP addresses:	• 15.156.152.96 • 15.156.152.97 • 15.156.152.98 • 15.156.152.99 • 15.156.152.100 • 15.156.152.101 • 15.156.152.102 • 15.156.152.103 • 15.156.152.104 • 15.156.152.105

	• 15.156.152.106 • 15.156.152.107 • 15.156.152.108 • 15.156.152.109 • 15.156.152.110 • 15.156.152.111

Europe (Frankfurt)

Domain:	• gw-pro-prod.wickr.com • api.messaging.wickr.eu-central-1.amazonaws.com
Calling CIDR addresses:	• 3.78.252.32/28
Calling IP addresses:	• 3.78.252.32 • 3.78.252.33 • 3.78.252.34 • 3.78.252.35 • 3.78.252.36 • 3.78.252.37 • 3.78.252.38 • 3.78.252.39 • 3.78.252.40 • 3.78.252.41 • 3.78.252.42 • 3.78.252.43 • 3.78.252.44 • 3.78.252.45 • 3.78.252.46 • 3.78.252.47

Messaging IP addresses:	• 3.163.236.183 • 3.163.238.183 • 3.163.251.183 • 3.163.232.183 • 3.163.241.183 • 3.163.245.183 • 3.163.248.183 • 3.163.234.183 • 3.163.237.183 • 3.163.243.183 • 3.163.247.183 • 3.163.240.183 • 3.163.242.183 • 3.163.244.183 • 3.163.246.183 • 3.163.249.183 • 3.163.252.183 • 3.163.235.183 • 3.163.250.183 • 3.163.239.183 • 3.163.233.183

Europe (London)

Domain:	• gw-pro-prod.wickr.com • api.messaging.wickr.eu-west-2.amazonaws.com
Calling CIDR addresses:	• 13.43.91.48/28
Calling IP addresses:	• 13.43.91.48 • 13.43.91.49

	• 13.43.91.50 • 13.43.91.51 • 13.43.91.52 • 13.43.91.53 • 13.43.91.54 • 13.43.91.55 • 13.43.91.56 • 13.43.91.57 • 13.43.91.58 • 13.43.91.59 • 13.43.91.60 • 13.43.91.61 • 13.43.91.62 • 13.43.91.63

Europe (Stockholm)

Domain:	• gw-pro-prod.wickr.com • api.messaging.wickr.eu-north-1.amazonaws.com
Calling CIDR addresses:	• 13.60.1.64/28
Calling IP addresses:	• 13.60.1.64 • 13.60.1.65 • 13.60.1.66 • 13.60.1.67 • 13.60.1.68 • 13.60.1.69 • 13.60.1.70 • 13.60.1.71 • 13.60.1.72

	• 13.60.1.73 • 13.60.1.74 • 13.60.1.75 • 13.60.1.76 • 13.60.1.77 • 13.60.1.78 • 13.60.1.79

Europe (Zurich)

Domain:	• gw-pro-prod.wickr.com • api.messaging.wickr.eu-central-2.amazonaws.com
Calling CIDR addresses:	• 16.63.106.224/28
Calling IP addresses:	• 16.63.106.224 • 16.63.106.225 • 16.63.106.226 • 16.63.106.227 • 16.63.106.228 • 16.63.106.229 • 16.63.106.230 • 16.63.106.231 • 16.63.106.232 • 16.63.106.233 • 16.63.106.234 • 16.63.106.235 • 16.63.106.236 • 16.63.106.237 • 16.63.106.238 • 16.63.106.239

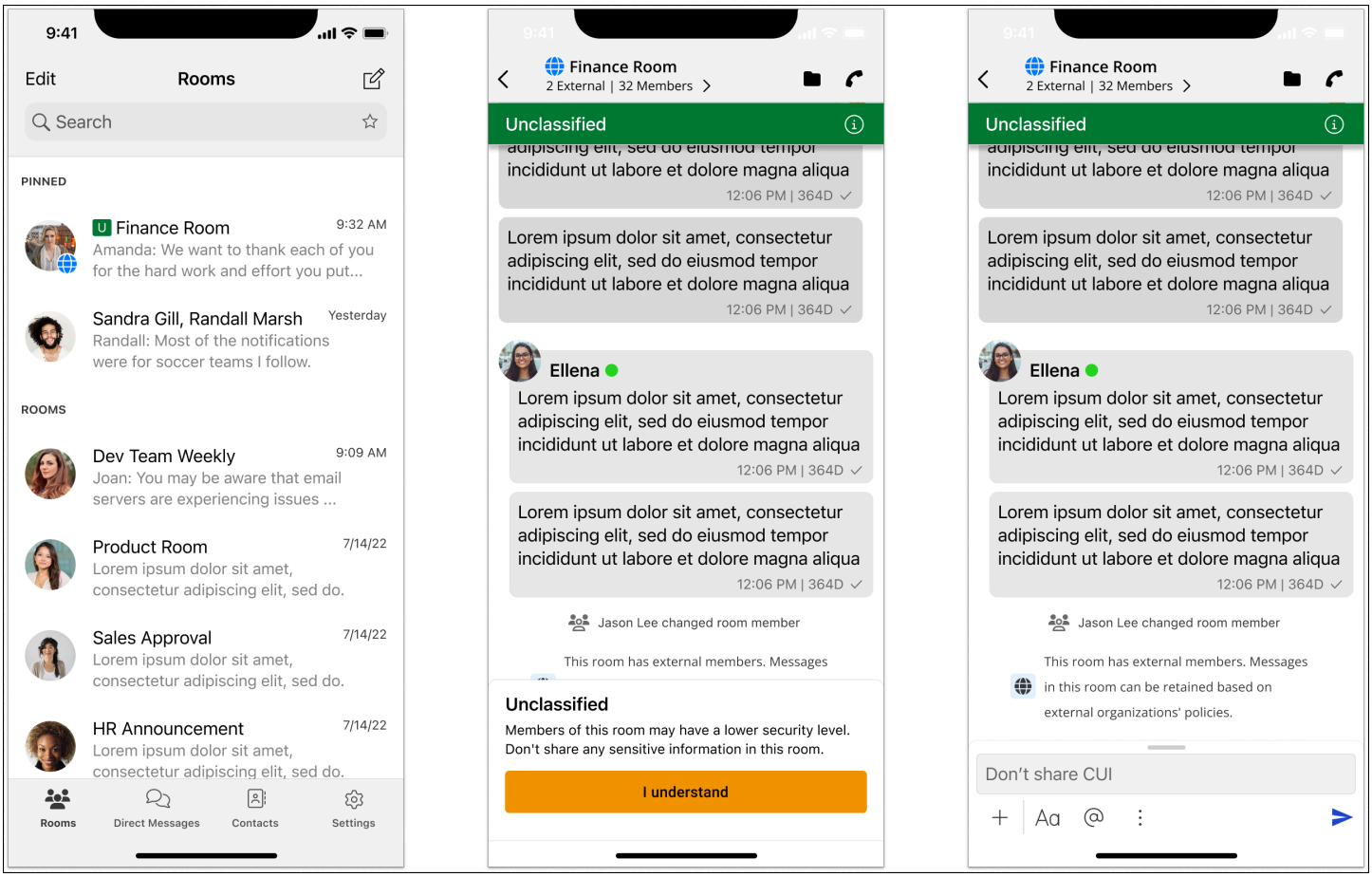
AWS GovCloud (US-West)

Domain:	• gw-pro-prod.wickr.com • api.messaging.wickr.us-gov-west-1.amazonaws.com
Calling CIDR addresses:	• 3.30.186.208/28
Calling IP addresses:	• 3.30.186.208 • 3.30.186.209 • 3.30.186.210 • 3.30.186.211 • 3.30.186.212 • 3.30.186.213 • 3.30.186.214 • 3.30.186.215 • 3.30.186.216 • 3.30.186.217 • 3.30.186.218 • 3.30.186.219 • 3.30.186.220 • 3.30.186.221 • 3.30.186.222 • 3.30.186.223

GovCloud cross boundary classification and federation

AWS Wickr offers WickrGov client tailored for GovCloud users. The GovCloud Federation allows communication between GovCloud users and commercial users. The cross boundary classification feature enables user interface changes to conversations for GovCloud users. As a GovCloud user, you must adhere to strict guidelines concerning government defined classification. When GovCloud users engage in conversations with commercial users (Enterprise, AWS Wickr, Guest users), they will see the following unclassified warnings displayed:

- A U tag in the room list
- An unclassified acknowledgment on the message screen
- An unclassified banner on top of the conversation



> **Note**
>
> These warnings will only be shown when a GovCloud user is in conversation or part of a room with external users. They will disappear if the external users leave the conversation. No warnings will be shown in conversations between GovCloud users.

File preview for AWS Wickr

Organizations using the Wickr Premium tier (including Premium free trial), can now manage file download permissions at the security group level.

File downloads are enabled by default in security groups. Administrators can enable or disable file downloads through the administrator panel. This setting is applied to the entire Wickr network.

To enable or disable file download, complete the following procedure.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Security groups**.
4. Select the name of the security group that you want to edit.

   The security group details page displays the settings for the security group in different tabs.
5. Under the **Messaging** tab, in the **Media and links** section, choose **Edit**.
6. On the **Edit media and links** page, check or uncheck the **File downloads** option.
7. Choose **Save changes**.

When file downloads are enabled for a security group, users can download files shared in direct messages and rooms. If downloads are disabled, they can only preview these files and upload to the **Files** tab, but cannot download them. Users are also restricted from taking screenshots; attempts will result in a black screen.

> **Note**
>
> When File downloads are disabled, all the users in that security group will need to be on Wickr versions 6.54 and above for this file setting to apply.

> **Note**
>
> In rooms where users from different networks (due to federation) and security groups are present, the ability of each user to preview or download files is based on their specific security group settings. As a result, some users can download files in a room while others can only preview them.

Manage users in AWS Wickr

In the **User management** section of the AWS Management Console for Wickr you can view current Wickr users and bots, and modify their details.

Topics

Team directory in AWS Wickr network

You can view current Wickr users and modify their details in the **User management** section of the AWS Management Console for Wickr.

Topics

View users in AWS Wickr network

You can view the details of users registered to your Wickr network.

Complete the following procedure to view users registered to your Wickr network.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.

   The **Team directory** tab displays users registered to your Wickr network, including their name, email address, assigned security group, and current status. For current users, you can view their devices, edit their details, suspend, delete, and switch them to another Wickr network.

Invite a user in AWS Wickr network

You can invite a user in your Wickr network.

Complete the following procedure to invite a user in your Wickr network.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. In the **Team directory** tab, choose **Invite user**.
5. On the **Invite user** page, enter the user's email address and security group. Email address and security group are the only fields that is required. Be sure to choose the appropriate security group for the user. Wickr will send an invitation email to the address you specify for the user.
6. Choose **Invite user**.

   An email is sent to the user. The email provides download links for the Wickr client applications, and a link to register for Wickr. As users register for Wickr using the link in the email, their status in the Wickr team directory will change from **Pending** to **Active**.

Edit users in AWS Wickr network

You can edit users in your Wickr network.

Complete the following procedure to edit a user.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. In the **Team directory** tab, select the vertical ellipsis (three dots) icon of the user you want to edit.
5. Choose **Edit**.
6. Edit the user information, and then choose **Save changes**.

Delete a user in AWS Wickr network

You can delete a user in your Wickr network.

Complete the following procedure to delete a user.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. In the **Team directory** tab, select the vertical ellipsis (three dots) icon of the user you want to delete.
5. Choose **Delete** to delete the user.

   When you delete a user, that user is no longer able to sign in to your Wickr network in the Wickr client.
6. In the pop-up window, choose **Delete**.

Bulk delete users in AWS Wickr network

You can bulk delete Wickr network users in the **User management** section in the AWS Management Console for Wickr.

> **Note**
>
> The option to bulk delete users only applies when SSO is not enabled.

To bulk delete your Wickr network users using a CSV template, complete the following procedure.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. The **Team directory** tab displays users registered to your Wickr network.
5. In the **Team directory** tab, choose **Manage users**, and then choose **Bulk delete**.
6. On the **Bulk delete users** page, download the sample CSV template. To download the sample template, choose **Download template**.
7. Complete the template by adding the email of the users you want to bulk delete from your network.

8. Upload the completed CSV template. You can drag and drop the file into the upload box, or select **choose a file**.
9. Select the check box, **I understand that deleting user is not reversible**.
10. Choose **Delete users**.

> **Note**
>
> This action will immediately start deleting users and may take several minutes. Deleted users will no longer able to sign in to your Wickr network in the Wickr client.

To bulk delete your Wickr network users by downloading a CSV of your team directory, complete the following procedure.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. The **Team directory** tab displays users registered to your Wickr network.
5. In the **Team directory** tab, choose **Manage users**, and then choose **Download as CSV**.
6. After you download the team directory CSV template, remove the rows of users who don't need to be deleted.
7. In the **Team directory** tab, choose **Manage users**, and then choose **Bulk delete**.
8. On the **Bulk delete users** page, upload the team directory CSV template. You can drag and drop the file into the upload box, or select **Choose a file**.
9. Select the check box, **I understand that deleting user is not reversible**.
10. Choose **Delete users**.

> **Note**
>
> This action will immediately start deleting users and may take several minutes. Deleted users will no longer able to sign in to your Wickr network in the Wickr client.

Bulk suspend users in AWS Wickr network

You can bulk suspend Wickr network users in the **User management** section in the AWS Management Console for Wickr.

> **Note**
>
> The option to bulk suspend users only applies when SSO is not enabled.

To bulk suspend your Wickr network users, complete the following procedure.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. The **Team directory** tab displays users registered to your Wickr network.
5. In the **Team directory** tab, choose **Manage users**, and then choose **Bulk suspend**.
6. On the **Bulk suspend users** page, download the sample CSV template. To download the sample template, choose **Download template**.
7. Complete the template by adding the email of the users you want to bulk suspend from your network.
8. Upload the completed CSV template. You can drag and drop the file into the upload box, or select **choose a file**.
9. Choose **Suspend users**.

> **Note**
>
> This action will immediately start suspending users and may take several minutes. Suspended users can't sign in to your Wickr network in the Wickr client. When you suspend a user who is currently signed in to your Wickr network in the client, that user is automatically signed out.

Guest users in AWS Wickr network

The Wickr guest user feature allows individual guest users to sign in to the Wickr client and collaborate with Wickr network users. Wickr administrators can enable or disable guest users for their Wickr networks.

After the feature is enabled, guest users invited to your Wickr network can interact with users in your Wickr network. A fee will be applied to your AWS account for the guest user feature. For more information about pricing for the guest user feature, see Wickr pricing page under **Pricing** Add-ons.

Topics

Enable or disable guest users in AWS Wickr network

You can enable or disable guest users in your Wickr network.

Complete the following procedure to enable or disable guest users for your Wickr network.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Security groups**.
4. Select the name for a specific security group.

> **Note**
>
> You can enable guest users for individual security groups only. To enable guest users for all security groups in your Wickr network, you must enable the feature for each security group in your network.

5. Choose the **Federation** tab in the security group.
6. There are two locations where the option to enable guest users are available:

- **Local federation —** For networks in US East (Northern Virginia), choose **Edit** in the **Local federation** section of the page.
- **Global federation —** For all other networks in other regions, choose **Edit** in the **Global federation** section of the page.

7. On the **Edit federation** page, select **Enable federation**.
8. Choose **Save changes** to save the change and make it effective for the security group.

   Registered users in the specific security group in your Wickr network can now interact with guest users. For more information, see Guest users in the *Wickr User Guide*.

View guest user count in AWS Wickr network

You can view the guest user count in your Wickr network.

Complete the following procedure to view the guest user count for your Wickr network.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.

   The **User management** page, displays a count of guest users in your Wickr network.

View monthly usage in AWS Wickr network

You can view the number of guest users your network has communicated with during a billing period.

Complete the following procedure to view your monthly usage for your Wickr network.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. Select the **Guest users** tab.

   The **Guest users** tab displays the guest users monthly usage.

> **Note**
>
> Guest billing data is updated every 24 hours.

View guest users in AWS Wickr network

You can view the guest users a network user has communicated with during a specific billing period.

Complete the following procedure to view guest users a network user communicated with during a specific billing period.

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. Select the **Guest users** tab.

   The **Guest users** tab displays the guest users in your network.

Block a guest user in AWS Wickr network

You can block and unblock a guest user in your Wickr network. Blocked users can't communicate with anyone in your network.

To block a guest user

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. Select the **Guest users** tab.

   The **Guest users** tab displays the guest users in your network.
5. In the **Guest Users** section, find the email of the guest user you want to block.
6. On the right-hand side of the guest user's name, select the three dots, and choose **Block guest user**.

7. Choose **Block** on the pop-up window.
8. To view the list of blocked users in your Wickr network, select the **Status** drop-down menu, and then select **Blocked**.

To unblock a guest user

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **User management**.
4. Select the **Guest users** tab.

   The **Guest users** tab displays the guest users in your network.
5. Select the **Status** drop-down menu, and then select **Blocked**.
6. In the **Blocked** section, find the email of the guest user you want to unblock.
7. On the right-hand side of the guest user's name, select the three dots, and choose **Unblock user**.
8. Choose **Unblock** on the pop-up window.

Security in AWS Wickr

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 Wickr, 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 Wickr. The following topics show you how to configure Wickr to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Wickr resources.

Topics

Data protection in AWS Wickr

The AWS shared responsibility model applies to data protection in AWS Wickr. 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 the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

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 Wickr 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.

Identity and access management for AWS Wickr

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 Wickr resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience for AWS Wickr

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Wickr.

Service user – If you use the Wickr service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Wickr features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Wickr, see Troubleshooting AWS Wickr identity and access.

Service administrator – If you're in charge of Wickr resources at your company, you probably have full access to Wickr. It's your job to determine which Wickr features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Wickr, see How AWS Wickr works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Wickr. To view example Wickr identity-based policies that you can use in IAM, see Identity-based policy examples for AWS Wickr.

Authenticating with identities for AWS Wickr

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

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and AWS Multi-factor authentication in IAM in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

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

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. To temporarily assume an IAM role in the AWS Management Console, you can switch from a user to an IAM role (console). You can assume a

role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Methods to assume a role in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – 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*.

- **Service-linked role** – 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.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Managing access using policies for AWS Wickr

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

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**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

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*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, 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 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.

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.

Access control lists (ACLs)

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.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role).

You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. 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*.

AWS managed policies for AWS Wickr

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.

AWS managed policy: AWSWickrFullAccess

You can attach the `AWSWickrFullAccess` policy to your IAM identities. This policy grants full administrative permission to the Wickr service, including the AWS Management Console for Wickr

in the AWS Management Console. For more information about attaching policies to an identity, see Adding and removing IAM identity permissions in the *AWS Identity and Access Management User Guide*.

Permissions details

This policy includes the following permissions.

- `wickr` – Grants full administrative permission to the Wickr service.

JSON

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

Wickr updates to AWS managed policies

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

Change	Description	Date
AWSWickrFullAccess – New policy	Wickr added a new policy that grants full administr ative permission to the Wickr service, including the Wickr administrator console in the AWS Management Console.	November 28, 2022

Change	Description	Date
Wickr started tracking changes	Wickr started tracking changes for its AWS managed policies.	November 28, 2022

How AWS Wickr works with IAM

Before you use IAM to manage access to Wickr, learn what IAM features are available to use with Wickr.

IAM features you can use with AWS Wickr

IAM feature	Wickr support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	No
Policy condition keys	No
ACLs	No
ABAC (tags in policies)	No
Temporary credentials	No
Principal permissions	No
Service roles	No
Service-linked roles	No

To get a high-level view of how Wickr 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 Wickr

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. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. 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 Wickr

To view examples of Wickr identity-based policies, see Identity-based policy examples for AWS Wickr.

Resource-based policies within Wickr

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. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by

attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for Wickr

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. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Wickr actions, see Actions Defined by AWS Wickr in the *Service Authorization Reference*.

Policy actions in Wickr use the following prefix before the action:

```
wickr
```

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

```
"Action": [
      "wickr:action1",
      "wickr:action2"
         ]
```

To view examples of Wickr identity-based policies, see Identity-based policy examples for AWS Wickr.

Policy resources for Wickr

Supports policy resources: No

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. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

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

To see a list of Wickr resource types and their ARNs, see Resources Defined by AWS Wickr in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Wickr.

To view examples of Wickr identity-based policies, see Identity-based policy examples for AWS Wickr.

Policy condition keys for Wickr

Supports service-specific policy condition keys: No

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 (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. 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.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Wickr condition keys, see Condition Keys for AWS Wickr in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions Defined by AWS Wickr.

To view examples of Wickr identity-based policies, see Identity-based policy examples for AWS Wickr.

ACLs in Wickr

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 Wickr

Supports ABAC (tags in policies): No

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

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 Wickr

Supports temporary credentials: No

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switch from a user to an IAM role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Wickr

Supports forward access sessions (FAS): No

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for Wickr

Supports service roles: No

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 Wickr functionality. Edit service roles only when Wickr provides guidance to do so.

Service-linked roles for Wickr

Supports service-linked roles: No

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 Wickr

By default, a brand new IAM user has no permissions to do anything. An IAM administrator must create and assign IAM policies that give users permission to administer the AWS Wickr service. The following shows an example of a permissions policy.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "wickr:CreateAdminSession",
                "wickr:ListNetworks"
            ],
            "Resource": "*"
        }
    ]
}
```

This sample policy gives users permissions to create, view, and manage Wickr networks using the AWS Management Console for Wickr. To learn more about the elements within an IAM policy statement, see Identity-based policies for Wickr. To learn how to create an IAM policy using these example JSON policy documents, see Creating policies on the JSON tab in the *IAM User Guide*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Wickr 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 AWS 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 Management Console for Wickr

Attach the `AWSWickrFullAccess` AWS managed policy to your IAM identities to grant them full administrative permission to the Wickr service, including the Wickr administrator console in the AWS Management Console. For more information, see AWS managed policy: AWSWickrFullAccess.

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": "*"
        }
    ]
}
```

Troubleshooting AWS Wickr identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Wickr and IAM.

Topics

- I am not authorized to perform an administrative action in the AWS Management Console for Wickr

I am not authorized to perform an administrative action in the AWS Management Console for Wickr

If the AWS Management Console for Wickr tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your sign-in credentials.

The following example error occurs when the `mateojackson` IAM user tries to use the AWS Management Console for Wickr to create, manage, or view Wickr networks in the AWS Management Console for Wickr but does not have the `wickr:CreateAdminSession` and `wickr:ListNetworks` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 wickr:ListNetworks
```

In this case, Mateo asks his administrator to update his policies to allow him to access the AWS Management Console for Wickr using the `wickr:CreateAdminSession` and `wickr:ListNetworks` actions. For more information, see Identity-based policy examples for AWS Wickr and AWS managed policy: AWSWickrFullAccess.

Compliance validation

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. 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 Wickr is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – AWS Config; assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS Wickr

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, Wickr offers several features to help support your data resiliency and backup needs. For more information, see Data retention for AWS Wickr.

Infrastructure Security in AWS Wickr

As a managed service, AWS Wickr is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

Configuration and vulnerability analysis in AWS Wickr

Configuration and IT controls are a shared responsibility between AWS and you, our customer. For more information, see the AWS shared responsibility model.

It is your responsibility to configure Wickr according to specifications and guidelines, to periodically instruct your users to download the latest version of the Wickr client, to ensure you are running the latest version of the Wickr data retention bot, and to monitor Wickr usage by your users.

Security best practices for AWS Wickr

Wickr 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.

To prevent potential security events associated with your use of Wickr, follow these best practices:

- Implement least privilege access and create specific roles to be used for Wickr actions. Use IAM templates to create a role. For more information, see AWS managed policies for AWS Wickr.
- Access the AWS Management Console for Wickr by authenticating to the AWS Management Console first. Don't share your personal console credentials. Anyone on the internet can browse to the console, but they can't sign in or start a session unless they have valid credentials to the console.

Monitoring AWS Wickr

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

- *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. For more information about logging Wickr API calls using CloudTrail, see Logging AWS Wickr API calls using AWS CloudTrail.

Logging AWS Wickr API calls using AWS CloudTrail

AWS Wickr is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Wickr. CloudTrail captures all API calls for Wickr as events. The calls captured include calls from the AWS Management Console for Wickr and code calls to the Wickr API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Wickr. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Wickr, the IP address from which the request was made, who made the request, when it was made, and additional details. To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Wickr information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Wickr, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail Event history.

For an ongoing record of events in your AWS account, including events for Wickr, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally,

you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple regions and Receiving CloudTrail log files from multiple accounts

All Wickr actions are logged by CloudTrail. For example, calls to the `CreateAdminSession`, and `ListNetworks` actions generate entries in the CloudTrail log files.

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 or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding Wickr log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateAdminSession` action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "<principal-id>",
```

```
        "arn": "<arn>",
        "accountId": "<account-id>",
        "accessKeyId": "<access-key-id>",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "<principal-id>",
                "arn": "<arn>",
                "accountId": "<account-id>",
                "userName": "<user-name>"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-03-10T07:53:17Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-03-10T08:19:24Z",
    "eventSource": "wickr.amazonaws.com",
    "eventName": "CreateAdminSession",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "<ip-address>",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/110.0.0.0 Safari/537.36",
    "requestParameters": {
        "networkId": 56019692
    },
    "responseElements": {
        "sessionCookie": "***",
        "sessionNonce": "***"
    },
    "requestID": "39ed0e6f-36e9-460d-8a6e-f24be0ec11c5",
    "eventID": "98ccb633-0e6c-4325-8996-35c3043022ac",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "<account-id>",
    "eventCategory": "Management"
}
```


The following example shows a CloudTrail log entry that demonstrates the `CreateNetwork` action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "<principal-id>",
        "arn": "<arn>",
        "accountId": "<account-id>",
        "accessKeyId": "<access-key-id>",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "<principal-id>",
                "arn": "<arn>",
                "accountId": "<account-id>",
                "userName": "<user-name>"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-03-10T07:53:17Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-03-10T07:54:09Z",
    "eventSource": "wickr.amazonaws.com",
    "eventName": "CreateNetwork",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "<ip-address>",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/110.0.0.0 Safari/537.36",
    "requestParameters": {
        "networkName": "BOT_Network",
        "accessLevel": "3000"
    },
    "responseElements": null,
    "requestID": "b83c0b6e-73ae-45b6-8c85-9910f64d33a1",
    "eventID": "551277bb-87e0-4e66-b2a0-3cc1eff303f3",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "<account-id>",
    "eventCategory": "Management"
}
```

The following example shows a CloudTrail log entry that demonstrates the `ListNetworks` action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "<principal-id>",
        "arn": "<arn>",
        "accountId": "<account-id>",
        "accessKeyId": "<access-key-id>",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "<principal-id>",
                "arn": "<arn>",
                "accountId": "<account-id>",
                "userName": "<user-name>"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-03-10T12:19:39Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-03-10T12:29:32Z",
    "eventSource": "wickr.amazonaws.com",
    "eventName": "ListNetworks",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "<ip-address>",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/110.0.0.0 Safari/537.36",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "b9800ba8-541a-43d1-9c8e-efd94d5f2115",
    "eventID": "5fbc83d7-771b-457d-9329-f85163a6a428",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "<account-id>",
    "eventCategory": "Management"
}
```

The following example shows a CloudTrail log entry that demonstrates the `UpdateNetworkdetails` action.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "<principal-id>",
        "arn": "<arn>",
        "accountId": "<account-id>",
        "accessKeyId": "<access-key-id>",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "<principal-id>",
                "arn": "<arn>",
                "accountId": "<account-id>",
                "userName": "<user-name>"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-03-08T22:42:15Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-03-08T22:42:58Z",
    "eventSource": "wickr.amazonaws.com",
    "eventName": "UpdateNetworkDetails",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "<ip-address>",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36",
    "requestParameters": {
        "networkName": "CloudTrailTest1",
        "networkId": <network-id>
    },
    "responseElements": null,
    "requestID": "abced980-23c7-4de1-b3e3-56aaf0e1fdbb",
    "eventID": "a4dc3391-bdce-487d-b9b0-6f76cedbb198",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
```

```
    "recipientAccountId": "<account-id>",
    "eventCategory": "Management"
}
```

The following example shows a CloudTrail log entry that demonstrates the `TagResource` action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "<principal-id>",
        "arn": "<arn>",
        "accountId": "<account-id>",
        "accessKeyId": "<access-key-id>",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "<principal-id>",
                "arn": "<arn>",
                "accountId": "<account-id>",
                "userName": "<user-name>"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-03-08T22:42:15Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-03-08T23:06:04Z",
    "eventSource": "wickr.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "<ip-address>",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36",
    "requestParameters": {
        "resource-arn": "<arn>",
        "tags": {
            "some-existing-key-3": "value 1"
        }
    },
    "responseElements": null,
```

```
    "requestID": "4ff210e1-f69c-4058-8ac3-633fed546983",
    "eventID": "26147035-8130-4841-b908-4537845fac6a",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "<account-id>",
    "eventCategory": "Management"
}
```

The following example shows a CloudTrail log entry that demonstrates the `ListTagsForResource` action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "<principal-id>",
        "arn": "<arn>",
        "accountId": "<account-id>",
        "accessKeyId": "<access-key-id>",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "<access-key-id>",
                "arn": "<arn>",
                "accountId": "<account-id>",
                "userName": "<user-name>"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-03-08T18:50:37Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-03-08T18:50:37Z",
    "eventSource": "wickr.amazonaws.com",
    "eventName": "ListTagsForResource",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "<ip-address>",
    "userAgent": "axios/0.27.2",
    "errorCode": "AccessDenied",
    "requestParameters": {
```

```
        "resource-arn": "<arn>"
    },
    "responseElements": {
        "message": "User: <arn> is not authorized to perform: wickr:ListTagsForResource
 on resource: <arn> with an explicit deny"
    },
    "requestID": "c7488490-a987-4ca2-a686-b29d06db89ed",
    "eventID": "5699d5de-3c69-4fe8-b353-8ae62f249187",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "<account-id>",
    "eventCategory": "Management"
}
```


Analytics dashboard in AWS Wickr

You can use the analytics dashboard to view how your organization is utilizing AWS Wickr. The following procedure explains how to access the analytics dashboard by using the AWS Wickr console.

To access the analytics dashboard

1. Open the AWS Management Console for Wickr at https://console.aws.amazon.com/wickr/.
2. On the **Networks** page, select the network name to navigate to that network.
3. In the navigation pane, choose **Analytics**.

   The **Analytics** page displays the metrics for your network in different tabs.

On the **Analytics** page, you will find a time frame filter at the top right corner of each tab. This filter applies to the entire page. Additionally, at the top right corner of each tab, you can export the data points for the selected time range by choosing the **Export** option available.

> **Note**
>
> The time selected is in UTC (Universal Time Coordinated).

The following tabs are available:

- **Overview** displays:
  - **Registered —** The total number of registered users, including active and suspended users on the network in the selected time. It does not include pending or invited users.
  - **Pending —** The total number of pending users on the network in the selected time.
  - **User Registration —** The graph displays the total number of users registered in the selected time range.
  - **Devices —** The number of devices where the app has been active.
  - **Client Versions —** The number of active devices categorized by their client versions.

- **Members** displays:
  - **Status —** Active users on the network within the time period selected.
  - **Active users —**
    - The graph displays the count of active users over time and can be aggregated by daily, weekly or monthly (within the above selected time range).
    - The active user count can be broken down by **Platform**, **Client Version**, or **Security Group**. If a security group was deleted, the total count will be shown as **Deleted#**.

- **Messages** displays:
  - **Messages sent —** The count of unique messages sent by all users and bots on the network in the selected time period.
  - **Calls —** Number of unique calls made by all users in the network.
  - **Files —** Number of files sent by users in the network (includes voice memos).
  - **Devices —** The pie chart displays the number of active devices categorized by their operating system.
  - **Client Versions —** The number of active devices categorized by their client versions.

Document history

The following table describes the documentation releases for Wickr.

Change	Description	Date
File preview is now available	Wickr administrators now have the ability to enable or disable file downloads. For more information, see File preview for AWS Wickr.	May 29, 2025
Newly redesigned Wickr administrator console is now available	Wickr has enhanced the Wickr administrator console for better navigation and improved accessibility for administrators.	March 13, 2025
Wickr is now available in the Asia Pacific (Malaysia) AWS Region	Wickr is now available in the Asia Pacific (Malaysia) AWS Region. For more information, see Regional availability.	November 20, 2024
Delete network is now available	Wickr administrators now have the ability to delete an AWS Wickr network. For more information, see Delete network in AWS Wickr.	October 4, 2024
Configuring AWS Wickr with Microsoft Entra (Azure AD) SSO is now available	AWS Wickr can be configured to use Microsoft Entra (Azure AD) as an identity provider. For more information, see Configure AWS Wickr with Microsoft Entra (Azure AD) single sign-on.	September 18, 2024

Wickr is now available in the Europe (Zurich) AWS Region	Wickr is now available in the Europe (Zurich) AWS Region. For more information, see Regional availability.	August 12, 2024
Cross Boundary classification and federation is now available	The cross boundary classification feature enables user interface changes to conversations for GovCloud users. For more information, see GovCloud cross boundary classification and federation.	June 25, 2024
The read receipt feature is now available	Wickr administrators can now enable or disable the read receipt feature in the Administrator Console. For more information, see Read receipts.	April 23, 2024
Global Federation now supports restricted federation and administrators can view usage analytics in the Administrator Console	Global Federation now supports restricted federation. This works for Wickr networks in other AWS Regions. For more information, see Security groups. Additionally, administrators can now view their usage analytics on the Analytics dashboard in the Admin Console. For more information, see Analytics dashboard.	March 28, 2024

A three-month free trial of AWS Wickr's Premium plan is now available	Wickr administrators can now choose a three-month free trial Premium plan for up to 30 users. During the free trial, all Standard and Premium plan features are available , including unlimited admin controls and data retention. The guest user feature is not available during the Premium free trial. For more informati on, see Manage plan.	February 9, 2024
The guest user feature is generally available and more administrator controls have been added	Wickr administrators can now access a range of new features, including list of guest users, the ability to bulk delete or suspend users, and the option to block guest users from communicating in your Wickr network. For more information, see Guest users.	November 8, 2023
Wickr is now available in the Europe (Frankfurt) AWS Region	Wickr is now available in the Europe (Frankfurt) AWS Region. For more information, see Regional availability.	October 26, 2023
Wickr networks now have the ability to federate across AWS Regions	Wickr networks now have the ability to federate across AWS Regions. For more informati on, see Security groups.	September 29, 2023
Wickr is now available in the Europe (London) AWS Region	Wickr is now available in the Europe (London) AWS Region. For more information, see Regional availability.	August 23, 2023

Wickr is now available in the Canada (Central) AWS Region	Wickr is now available in the Canada (Central) AWS Region. For more information, see Regional availability.	July 3, 2023
The guest user feature now available for preview	Guest users can sign in to the Wickr client and collaborate with Wickr network users. For more information, see Guest users (preview).	May 31, 2023
AWS Wickr is now integrated with AWS CloudTrail, and is now available in AWS GovCloud (US-West) as WickrGov	AWS Wickr is now integrated with AWS CloudTrail. For more information, see Logging AWS Wickr API calls using AWS CloudTrail. Additionally, Wickr is now available in AWS GovCloud (US-West) as WickrGov. For more information, see AWS WickrGov in the *AWS GovCloud (US) User Guide*.	March 30, 2023
Tagging and multiple network creation	Tagging now supported in AWS Wickr. For more information, see Network tags. Multiple networks can now be created in Wickr. For more information, see Create a network.	March 7, 2023
Initial release	Initial release of the Wickr Administration Guide	November 28, 2022

Release notes

To help you keep track of the ongoing updates and improvements to Wickr, we publish release notices that describe recent changes.

May 2025

- File preview is now available. When file downloads are disabled by the admin in the admin console for a security group, users will only be able to view a list of supported files in Messaging and Files tabs.

March 2025

- Redesigned Wickr administrator console is now available.

October 2024

- Wickr now supports delete network. For more information, see Delete network in AWS Wickr.

September 2024

- Administrators can now configure AWS Wickr with Microsoft Entra (Azure AD) single sign-on. For more information, see Configure AWS Wickr with Microsoft Entra (Azure AD) single sign-on.

August 2024

- Enhancements
  - Wickr is now available in the Europe (Zurich) AWS Region.

June 2024

- Cross Boundary classification and federation is now available for GovCloud users. For more information, see GovCloud cross boundary classification and federation.

April 2024

- Wickr now supports read receipts. For more information, see Read receipts.

March 2024

- Global Federation now supports restricted federation, where global federation can be enabled only for selected networks that are added under restricted federation. This works for Wickr networks in other AWS Regions. For more information, see Security groups.
- Administrators can now view their usage analytics on the Analytics dashboard in the Admin Console. For more information, see Analytics dashboard.

February 2024

- AWS Wickr is now offering a three-month free trial of its Premium plan for up to 30 users. Changes and limitations include:
  - All Standard and Premium plan features such as unlimited admin controls and data retention are now available in the Premium free trial. The guest user feature is not available during the Premium free trial.
  - The previous Free trial is no longer available. You can upgrade your existing Free trial or Standard plan to a Premium free trial if you haven't already used the Premium free trial. For more information, see Manage plan.

November 2023

- Guest users feature is now generally available. Changes and additions include:
  - Ability to report abuse by other Wickr users.
  - Administrators can view a list of guest users a network has interacted with, and monthly usage counts.
  - Administrators can block guest users from communicating with their network.
  - Add-on pricing for guest users.

- Admin control enhancements

- Ability to bulk delete/suspend users.
- Additional SSO setting to configure a grace period for token refresh.

October 2023

- Enhancements
  - Wickr is now available in the Europe (Frankfurt) AWS Region.

September 2023

- Enhancements
  - Wickr networks now have the ability to federate across AWS Regions. For more information, see Security groups.

August 2023

- Enhancements
  - Wickr is now available in the Europe (London) AWS Region.

July 2023

- Enhancements
  - Wickr is now available in the Canada (Central) AWS Region.

May 2023

- Enhancements
  - Added support for guest users. For more information, see Guest users in AWS Wickr network.

March 2023

- Wickr is now integrated with AWS CloudTrail. For more information, see Logging AWS Wickr API calls using AWS CloudTrail.
- Wickr is now available in AWS GovCloud (US-West) as WickrGov. For more information, see AWS WickrGov in the *AWS GovCloud (US) User Guide*.
- Wickr now supports tagging. For more information, see Network tags for AWS Wickr. Multiple networks can now be created in Wickr. For more information, see Step 1: Create a network.

February 2023

- Wickr now supports the Android Tactical Assault Kit (ATAK). For more information, see Enable ATAK in the Wickr Network Dashboard.

January 2023

- Single sign-on (SSO) can now be configured on all plans, including Free Trial and Standard.