# Table of Contents

What is AWS AppFabric? ........................................................................................................................................ 1
What is OCSF? .................................................................................................................................................. 1
Benefits ............................................................................................................................................................ 1
Benefits of OCSF ............................................................................................................................................... 1
Use cases .......................................................................................................................................................... 1
Use cases of OCSF ........................................................................................................................................... 1
How AppFabric works ...................................................................................................................................... 1
How OCSF event categories and classes work ............................................................................................... 1
Accessing AppFabric ....................................................................................................................................... 2
Accessing OCSF event categories and classes ............................................................................................... 2
Related services ............................................................................................................................................... 2
Related services of OCSF ............................................................................................................................... 2
Pricing ............................................................................................................................................................... 3
Pricing of OCSF ............................................................................................................................................... 3
Availability .................................................................................................................................................... 3
Availability of OCSF ........................................................................................................................................ 3
Terminology and concepts ............................................................................................................................... 4
OCSF schema .................................................................................................................................................. 6
OCSF-based schema in AppFabric .................................................................................................................. 6
OCSF event categories and classes ................................................................................................................ 6
Prerequisites and recommendations ................................................................................................................ 7
Sign up for an AWS account ............................................................................................................................ 7
Create an administrative user .......................................................................................................................... 7
(Required) Complete application prerequisites ............................................................................................ 8
(Optional) Create an output location ............................................................................................................... 8
Amazon S3 .................................................................................................................................................... 8
Amazon Kinesis Data Firehose ....................................................................................................................... 9
(Optional) Create an AWS KMS key ............................................................................................................... 9
Getting started ............................................................................................................................................... 11
Prerequisites ................................................................................................................................................... 11
Step 1: Create app bundle .............................................................................................................................. 11
Step 2: Authorize applications ...................................................................................................................... 12
Step 3: Set up audit log ingestions ................................................................................................................ 13
Step 4: Use the user access tool .................................................................................................................... 15
Step 5: Connect AppFabric data in security tools and other destinations .................................................... 16
Supported applications .................................................................................................................................. 17
Asana ............................................................................................................................................................... 17
AppFabric support for Asana .......................................................................................................................... 17
Connecting AppFabric to your Asana account ............................................................................................... 18
Atlassian Jira suite ......................................................................................................................................... 18
AppFabric support for the Jira suite .............................................................................................................. 19
Connecting AppFabric to your Jira account ................................................................................................. 19
Dropbox ......................................................................................................................................................... 21
AppFabric support for Dropbox ...................................................................................................................... 21
Connecting AppFabric to your Dropbox account .......................................................................................... 22
GitHub ............................................................................................................................................................ 23
AppFabric support for GitHub ....................................................................................................................... 23
Connecting AppFabric to your GitHub account ........................................................................................... 24
Google Workspace ....................................................................................................................................... 26
AppFabric support for Google Workspace ................................................................................................... 26
Connecting AppFabric to your Google Workspace account ..................................................................... 26
Microsoft 365 .............................................................................................................................................. 28
AppFabric support for Microsoft 365 ........................................................................................................... 28
Connecting AppFabric to your Microsoft 365 account ............................................................................... 29
Miro ............................................................................................................................................................... 30
AppFabric support for Miro .......................................................................................................................... 31
Connecting AppFabric to your Miro account .............................................................................................. 31
Okta ............................................................................................................................................................... 33
AppFabric support for Okta .......................................................................................................................... 33
Connecting AppFabric to your Okta account ............................................................................................. 33
ServiceNow .................................................................................................................................................... 35
AppFabric support for ServiceNow ............................................................................................................... 35
Data delay considerations .......................................................... 36
Connecting AppFabric to your ServiceNow account .......................... 36
Slack .................................................................................. 37
AppFabric support for Slack ...................................................... 38
Connecting AppFabric to your Slack account ................................ 38
Smartsheet ........................................................................... 40
AppFabric support for Smartsheet ................................................ 41
Connecting AppFabric to your Smartsheet account ......................... 41
Webex by Cisco ...................................................................... 42
AppFabric support for Webex .................................................... 43
Connecting AppFabric to your Webex account ............................... 43
Zendesk ................................................................................ 45
AppFabric support for Zendesk .................................................. 45
Connecting AppFabric to your Zendesk account ............................ 46
Zoom .................................................................................... 47
AppFabric support for Zoom ..................................................... 47
Connecting AppFabric to your Zoom account ............................... 48
Compatible security tools .......................................................... 50
QuickSight ............................................................................ 50
AppFabric audit log ingestion considerations ................................ 50
Security Lake ........................................................................ 51
AppFabric audit log ingestion considerations ................................ 51
Logz.io .................................................................................. 55
AWS AppFabric audit log ingestion considerations ......................... 55
Netskope ................................................................................. 56
AWS AppFabric audit log ingestion considerations ......................... 56
NetWitness ............................................................................. 56
AWS AppFabric audit log ingestion considerations ......................... 57
Rapid7 .................................................................................... 57
AWS AppFabric audit log ingestion considerations ......................... 57
Splunk ..................................................................................... 58
AWS AppFabric audit log ingestion considerations ......................... 58
Delete resources ..................................................................... 60
Delete an ingestion destination ................................................... 60
Delete an ingestion ................................................................. 60
Delete an app authorization ....................................................... 60
Delete an app bundle ............................................................... 61
Security .................................................................................. 62
Data protection ....................................................................... 62
Encryption at rest .................................................................... 63
Encryption in transit ................................................................... 63
Key management ..................................................................... 63
Key policy .............................................................................. 63
How AppFabric uses grants in AWS KMS .................................. 65
Monitoring your encryption keys for AppFabric ............................. 65
Identity and access management ................................................ 67
Audience ................................................................................ 67
Authenticating with identities ..................................................... 67
Managing access using policies ................................................ 70
How AWS AppFabric works with IAM ....................................... 71
Identity-based policy examples ................................................ 76
Using service-linked roles ........................................................ 79
AWS managed policies ........................................................... 80
Troubleshooting ..................................................................... 83
Compliance validation ............................................................... 85
Security best practices .............................................................. 85
Monitor for application without admin access ................................. 85
Monitor for AppFabric events ................................................................. 86
Resilience .......................................................................................... 86
Infrastructure security ................................................................. 86
Configuration and vulnerability analysis .................................. 86
Monitoring ...................................................................................... 87
Monitoring with CloudWatch ...................................................... 87
CloudTrail logs ............................................................................. 88
AppFabric information in CloudTrail ........................................ 88
Understanding AppFabric log file entries ................................. 88
Quotas ............................................................................................. 90
Document history .......................................................................... 91
What is AWS AppFabric?

AWS AppFabric quickly connects software as a service (SaaS) applications across your organization, so IT and security teams can easily manage and secure applications using a standard schema, and employees will soon be able to complete everyday tasks faster using generative AI.

Topics
- Benefits (p. 1)
- Use cases (p. 1)
- How AppFabric works (p. 1)
- Accessing AppFabric (p. 2)
- Related services (p. 2)
- Pricing (p. 3)
- Availability (p. 3)

Benefits

You can use AppFabric to do the following:
- Connect your applications in minutes, and reduce operational costs.
- Increase visibility across SaaS application data to elevate your security posture.
- Automatically facilitate tasks across applications with generative AI, soon.

Use cases

You can use AppFabric to:
- Connect your SaaS applications quickly
  - AppFabric natively connects top SaaS productivity and security applications to each other, providing a fully managed SaaS interoperability solution.
- Elevate your security posture
  - Application data is automatically normalized, enabling administrators to set common policies, standardize security alerts, and easily manage user access across multiple applications.
- Reimagine productivity
  - With a common generative AI assistant, AppFabric empowers employees to get answers quickly, automate task management, and generate insights across their SaaS productivity applications.

How AppFabric works

AppFabric quickly connects multiple SaaS applications with no coding required for increased productivity and security. The following diagram shows the benefits of AppFabric.
Accessing AppFabric

AppFabric is available in the US East (N. Virginia), Europe (Ireland), and Asia Pacific (Tokyo) AWS Regions. For more information about AWS Regions, see AWS service endpoints in the AWS General Reference.

In each Region, you can access AppFabric in any of the following ways:

**AWS Management Console**

The AWS Management Console is a browser-based interface that you can use to create and manage AWS resources. The AppFabric console provides access to your AppFabric resources. You can use the AppFabric console to create and manage all AppFabric resources.

**AppFabric API**

To access AppFabric programmatically, use the AppFabric API, and issue HTTPS requests directly to the service. For more information, see the AWS AppFabric API Reference.

**AWS Command Line Interface (AWS CLI)**

With the AWS CLI, you can issue commands at your system’s command line to interact with AppFabric and other AWS services. If you want to build scripts that perform tasks, the command line tools are also useful. For information about installing and using the AWS CLI, see the AWS Command Line Interface User Guide for Version 2. For information about the AWS CLI commands for AppFabric, see the AppFabric section of the AWS CLI Reference.

**Related services**

You can use the following AWS services with AppFabric:

**Amazon Kinesis Data Firehose**

Amazon Kinesis Data Firehose is an extract, transform, and load (ETL) service that reliably captures, transforms, and delivers streaming data to data lakes, data stores, and analytics services. When you use AppFabric, you can choose to output your Open Cybersecurity Schema Framework (OCSF) normalized or raw audit logs in JSON format to a Kinesis Data Firehose stream as your destination. For more information, see Create an output location in Kinesis Data Firehose (p. 9).
Amazon Security Lake

Amazon Security Lake automatically centralizes security data from AWS environments, SaaS providers, on premises and cloud sources into a purpose-built data lake stored in your account. You can integrate AppFabric audit log data with Security Lake by selecting Amazon Kinesis Data Firehose as a destination and configuring Kinesis Data Firehose to deliver data in the correct format and path in Security Lake. For more information, see Collecting data from custom sources in the Amazon Security Lake User Guide.

Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. When you use AppFabric, you can choose to output your OCSF normalized (JSON or Apache Parquet) or raw (JSON) audit logs to a new or existing Amazon S3 bucket as your destination. For more information, see Create an output location in Amazon S3 (p. 8).

Amazon QuickSight

Amazon QuickSight powers data-driven organizations with unified business intelligence (BI) at hyperscale. With QuickSight, all users can meet varying analytic needs from the same source of truth through modern interactive dashboards, paginated reports, embedded analytics, and natural language queries. You can analyze AppFabric audit log data in QuickSight, by choosing the Amazon S3 bucket where your AppFabric logs are stored as your source. For more information, see Creating a dataset using Amazon S3 files in the Amazon QuickSight User Guide. You can also import AppFabric data in Amazon S3 to Amazon Athena and select Amazon Athena as the data source in QuickSight. For more information, see Creating a dataset using Amazon Athena data in the Amazon QuickSight User Guide.

AWS Key Management Service

With AWS Key Management Service (AWS KMS), you can create, manage, and control cryptographic keys across your applications and AWS services. When you create an app bundle in AppFabric, you set up an encryption key to securely protect your authorized application data. This key encrypts your data within the AppFabric service. AppFabric can use an AWS owned key created and managed by AppFabric on your behalf, or a customer managed key that you create and manage in AWS KMS. For more information, see Create an AWS KMS key (p. 9).

Pricing

For AppFabric pricing details and examples, see AWS AppFabric Pricing.

Availability

To view the currently supported AWS Regions and endpoints for AppFabric, see AWS AppFabric endpoints and quotas in the AWS General Reference.
Terminology and concepts

This topic describes the key terminology and concepts in AWS AppFabric to help you get started.

**App bundle**

An AppFabric app bundle stores all of your AppFabric app authorizations and ingestions (see the following definition of ingestions). You can create one app bundle per AWS account per AWS Region.

**App authorization**

An app authorization grants AppFabric permission to connect and interact with your applications. It allows ingestion of audit logs from your applications, with OAuth (Open Authorization - an open standard for access delegation to grant applications access) or personal access token (PAT) credentials. You can set up multiple app authorizations (up to 50) per app bundle. This allows AppFabric to ingest audit logs from multiple tenants of applications, by repeating the app authorization creation step as needed for each tenant of the application. The credentials that are shared are encrypted with an AWS owned key or a customer managed key from the AWS Key Management Service (AWS KMS), and are stored in AppFabric.

**Ingestion**

An AppFabric ingestion uses an app authorization to pull audit logs from an application through the application's public APIs. It then delivers the audit logs to one or more (up to five) destinations.

**Client ID**

When you create an app authorization to connect with an application that uses the OAuth flow, AppFabric might ask you for the client ID and client secret. The client ID and client secret can be found in your application's authentication app. For instructions on where to find the client ID in a given authentication app, see Supported applications (p. 17). The client ID and client secret that are shared are encrypted with an AWS owned key or a customer managed key AWS KMS key and stored in AppFabric.

**Client secret**

When you create an app authorization to connect with an application that uses the OAuth flow, AppFabric might ask you for the client ID and client secret. The client ID and client secret can be found in your application's authentication app. For instructions on where to find the client secret in a given authentication app, see Supported applications (p. 17). The client ID and client secret that are shared are encrypted with an AWS owned key or a customer managed key AWS KMS key and stored in AppFabric.

**Ingestion destination**

An ingestion destination defines where the audit logs pulled from an ingestion should be stored. Each ingestion can deliver audit logs to one or more destinations (up to five), which are an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose in your AWS account. For each destination, you can define whether you would like the logs to be in raw form or normalized into an Open Cybersecurity Schema Framework (OCSF) schema. When you select the OCSF schema, you can define the format of the logs (JSON or Apache Parquet). The Apache Parquet format can be used only if Amazon S3 is selected as the destination.

**OAuth**

OAuth is an open protocol to allow secure authorization in a simple and standard method from web, mobile, and desktop applications. AppFabric uses OAuth to create some app authorizations.
Open Cybersecurity Schema Framework (OCSF)

The Open Cybersecurity Schema Framework (OCSF) is an open-source project delivering an extensible framework for developing schemas, along with a vendor-agnostic core security schema. Vendors and other data producers can adopt and extend the schema for their specific domains. The goal is to provide an open standard, adopted in any environment, application, or solution, while complementing existing security standards and processes. AppFabric has extended this schema to create a software as a service (SaaS)-centric event structure that all SaaS app audit logs supported by AppFabric will be normalized to. For more information, see Open Cybersecurity Schema Framework (p. 6).

Personal access token (PAT)

A personal access token (PAT) is a string of characters that can be used to access a computer system instead of the usual password. When you create an app authorization to connect with an application that uses the PAT flow, AppFabric might ask you for a PAT. The PAT can be found in your application’s authentication app. For instructions on where to find the PAT in a specific authentication app, see Supported applications (p. 17). The service account tokens that are shared are encrypted with an AWS owned key or a customer managed key AWS KMS key and stored in AppFabric.

Service account token

When you create an AppFabric app authorization to connect with an application, some applications will require a service account to be created for application authentication. AppFabric might ask for the service account token as part of the app authorization process. For instructions on where to find the service account token in a given authentication app, see Supported applications (p. 17). The service account tokens that are shared are encrypted with an AWS owned key or a customer managed key AWS KMS key and stored in AppFabric.

Tenant ID

When you create an app authorization, AppFabric might ask you for the tenant ID and tenant name of your app. The tenant ID is a unique identifier for your application tenant. Each application might have different terms for a tenant such as Workspace ID for Slack or Domain ID for Asana. For instructions on where to find the tenant ID in a specific application, see Supported applications (p. 17).

Tenant name

When you create an app authorization, AppFabric might ask you for the tenant ID and tenant name of your app. The tenant name is a unique name that you give to the tenant ID, to be used within an app bundle. This value is used to label the app authorization and any related ingestion.
Open Cybersecurity Schema Framework

The Open Cybersecurity Schema Framework (OCSF) is a collaborative, open-source effort by AWS and leading partners in the cybersecurity industry. OCSF provides a standard schema for common security events, defines versioning criteria to facilitate schema evolution, and includes a self-governance process for security log producers and consumers. The public source code for OCSF is hosted on GitHub.

OCSF-based schema in AppFabric

The AWS AppFabric OCSF 1.0.0-rc.3 based schema is tailored specifically to address your needs for normalized, consistent, low-effort observability of their software as a service (SaaS) portfolio. AppFabric, in collaboration with the OCSF open source community, introduced new OCSF event categories, event classes, activities, and objects so that OCSF is applicable to SaaS application events. AppFabric automatically normalizes audit events that it receives from SaaS applications and delivers this data to the Amazon Simple Storage Service (Amazon S3) or Amazon Kinesis Data Firehose services in your AWS account. For an Amazon S3 destination, you can choose between two normalization options (OCSF or Raw) and two data format options (JSON or Parquet). When delivering to Kinesis Data Firehose, you can also choose between two normalization options (OCSF or Raw) but the data format is limited to JSON.

OCSF event categories and classes

AppFabric uses the following two OCSF event categories:

- **Identity and Access Management** – AppFabric uses the following event classes within this category:
  - Account Change
  - Authentication
  - User Access Management
  - Group Management
- **Application Activity** – AppFabric uses the following event classes within this category:
  - Web Resources Activity
  - Web Resource Access Activity
Prerequisites and recommendations

If you're a new AWS customer, complete the setup prerequisites that are listed on this page before you start using AWS AppFabric. 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 an AWS account (p. 7)
- Create an administrative user (p. 7)
- (Required) Complete application prerequisites (p. 8)
- (Optional) Create an output location (p. 8)
- (Optional) Create an AWS KMS key (p. 9)

Sign up for an AWS account

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

To sign up for an AWS account

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call 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 an administrative user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing My Account.

Create an administrative user

After you sign up for an AWS account, create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing Root user and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the AWS Sign-In User Guide.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the IAM User Guide.
Create an administrative user

- For your daily administrative tasks, grant administrative access to an administrative user in AWS IAM Identity Center.

For instructions, see Getting started in the AWS IAM Identity Center User Guide.

Sign in as the administrative user

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the AWS Sign-In User Guide.

(Required) Complete application prerequisites

To use AppFabric to receive user information and audit logs from applications, many applications require that you have specific role and plan types. Ensure that you have reviewed the prerequisites for each application that you want to authorize with AppFabric, and that you have the proper plans and roles. For more information about the application-specific prerequisites, see Supported Applications (p. 17), or choose one of the following application-specific topics.

- Asana (p. 17)
- Atlassian Jira suite (p. 18)
- Dropbox (p. 21)
- GitHub (p. 23)
- Google Workspace (p. 26)
- Microsoft 365 (p. 28)
- Miro (p. 30)
- Okta (p. 33)
- ServiceNow (p. 35)
- Slack (p. 37)
- Smartsheet (p. 40)
- Webex by Cisco (p. 42)
- Zendesk (p. 45)
- Zoom (p. 47)

(Optional) Create an output location

AppFabric supports Amazon Simple Storage Service (Amazon S3) and Amazon Kinesis Data Firehose as audit log ingestion destinations.

Amazon S3

You can create a new Amazon S3 bucket using the AppFabric console when you create an ingestion destination. You can also create a bucket using the Amazon S3 service. If you choose to create your bucket using the Amazon S3 service, you must create the bucket before creating the AppFabric ingestion destination, and then select the bucket when you create the ingestion destination. You can choose to use
an existing Amazon S3 bucket in your AWS account, as long as it meets the following requirements for existing buckets:

- AppFabric requires that your Amazon S3 bucket be in the same AWS Region as your Amazon S3 resources.
- You can encrypt your bucket using server-side encryption with Amazon S3 managed keys (SSE-S3) or server-side encryption with AWS Key Management Service (AWS KMS) keys (SSE-KMS).

**Amazon Kinesis Data Firehose**

You can choose to use Amazon Kinesis Data Firehose as your ingestion destination for AppFabric data. To use Kinesis Data Firehose, you can create the Kinesis Data Firehose delivery stream in your AWS account before creating an ingestion or while you’re creating an ingestion destination in AppFabric. You can create a Kinesis Data Firehose delivery stream using the AWS Management Console, AWS CLI, or the AWS APIs or SDKs. For stream configuration instructions, see the following topics:

- AWS Management Console instructions – [Creating an Amazon Kinesis Data Firehose Delivery Stream](https://docs.aws.amazon.com/kinesis/latest/dev/firehose-ds-streams-quickstart.html) in the *Amazon Kinesis Data Firehose Developer Guide*
- AWS CLI instructions – `create-delivery-stream` in the *AWS CLI Command Reference*
- AWS APIs and SDKs instructions – `CreateDeliveryStream` in the *Amazon Kinesis Data Firehose API Reference*

The requirements when using Amazon Kinesis Data Firehose as the AppFabric output destination are as follows:

- You must create the stream in the same AWS Region as your AppFabric resources.
- You must select Direct PUT as the source.
- Attach `AmazonKinesisFirehoseFullAccess` AWS managed policy to your user, or attach the following permissions to your user:

```json
{
    "Sid": "TagFirehoseDeliveryStream",
    "Effect": "Allow",
    "Action": ["firehose:TagDeliveryStream"],
    "Condition": {
        "ForAllValues:StringEquals": {"aws:TagKeys": "AWSAppFabricManaged"},
        "Resource": "arn:aws:firehose:*:*:deliverystream/*"
    }
}
```

Kinesis Data Firehose supports integration with a variety of third-party security tools, such as Splunk and Logz.io. For information about how to properly configure Amazon Kinesis so that it outputs data to these tools, see [Destination Settings](https://docs.aws.amazon.com/kinesis/latest/dev/firehose-ds-streams-quickstart.html) in the *Amazon Kinesis Data Firehose Developer Guide*.

**Optional) Create an AWS KMS key**

In the process of creating an AppFabric app bundle, you will select or set up an encryption key to securely protect your data from all authorized applications. This key will be used to encrypt your data within the AppFabric service.

AppFabric encrypts data by default. AppFabric can use an AWS owned key created and managed by AppFabric on your behalf or a customer managed key that you create and manage in AWS Key

---

9
Management Service (AWS KMS). AWS owned keys are a collection of AWS KMS keys that an AWS service owns and manages for use in multiple AWS accounts. Customer managed keys are AWS KMS keys in your AWS account that you create, own, and manage. For more information about AWS owned keys and customer managed keys, see Customer keys and AWS keys in the AWS Key Management Service Developer Guide.

If you want to use a customer managed key to encrypt your data, such as authorization tokens, within AppFabric, you can create one with AWS KMS. For more information about the permissions policy that grants access to your customer managed key in AWS KMS, see the Key policy (p. 63) section of this guide.
Getting started with AWS AppFabric

To get started with AWS AppFabric, you must first create an app bundle and then authorize and connect applications to your app bundle. After app authorizations are connected to applications, you can use AppFabric features such as audit log ingestions and user access.

This section explains how to start using AppFabric in the AWS Management Console.

Topics

- Prerequisites (p. 11)
- Step 1: Create app bundle (p. 11)
- Step 2: Authorize applications (p. 12)
- Step 3: Set up audit log ingestions (p. 13)
- Step 4: Use the user access tool (p. 15)
- Step 5: Connect AppFabric data in security tools and other destinations (p. 16)

Prerequisites

Before you get started, you must first create an AWS account and an administrative user. For more information, see Sign up for an AWS account (p. 7) and Create an administrative user (p. 7).

Step 1: Create app bundle

An app bundle stores all of your AppFabric app authorizations and ingestions. To create an app bundle, set up an encryption key to securely protect your authorized application data.

2. In the Select a Region selector in the upper-right corner of the page, select an AWS Region. AppFabric is available in the US East (N. Virginia), Europe (Ireland), and Asia Pacific (Tokyo) Regions only.
3. Choose Getting started.
5. In the Encryption section, set up an encryption key to securely protect your data from all authorized applications. This key is used to encrypt your data within the AppFabric service.

AppFabric encrypts data by default. AppFabric can use an AWS owned key created and managed by AppFabric on your behalf or a customer managed key that you create and manage in AWS Key Management Service (AWS KMS).

6. For AWS KMS Key, choose either Use AWS owned key or Customer managed key.

If you choose to use a customer managed key, enter either the Amazon Resource Name (ARN) or the key ID of the existing key that you want to use, or choose Create an AWS KMS key.

Consider the following when choosing an AWS owned key or a customer managed key:

- AWS owned keys are a collection of AWS Key Management Service (AWS KMS) keys that an AWS service owns and manages for use in multiple AWS accounts. Although AWS owned keys are not in your AWS account, an AWS service can use an AWS owned key to protect the resources in your account. AWS owned keys don't count against the AWS KMS quotas for your account. You don't
need to create or maintain the key or its key policy. The rotation of AWS owned keys varies across services. For information about the rotation of an AWS owned key for AppFabric, see Encryption at rest (p. 63).

- Customer managed keys are KMS keys in your AWS account that you create, own, and manage. You have full control over these AWS KMS keys. You can establish and maintain their key policies, AWS Identity and Access Management (IAM) policies, and grants. You can enable and disable them, rotate their cryptographic material, add tags, create aliases that refer to the AWS KMS keys, and schedule the AWS KMS keys for deletion. Customer managed keys appear on the Customer managed keys page of the AWS Management Console for AWS KMS.

To definitively identify a customer managed key, use the DescribeKey operation. For customer managed keys, the value of the KeyManager field of the DescribeKey response is CUSTOMER. You can use your customer managed key in cryptographic operations and audit usage in AWS CloudTrail logs. With many AWS services that integrate with AWS KMS, you can specify a customer managed key to protect the data stored and managed for you. Customer managed keys incur a monthly fee and a fee for use in excess of the AWS Free Tier. Customer managed keys count against the AWS KMS quotas for your account.

For more information about AWS owned keys and customer managed keys, see Customer keys and AWS keys in the AWS Key Management Service Developer Guide.

Note
When an app bundle is created, AppFabric also creates a special IAM role in your AWS account called a service-linked role (SLR) for AppFabric. It allows the service to send metrics to Amazon CloudWatch. After you add an audit log destination, the SLR allows the AppFabric service access to your AWS resources (Amazon S3 buckets, Amazon Kinesis Data Firehose delivery streams). For more information, see Using service-linked roles for AppFabric (p. 79).

7. (Optional) For Tags, you have the option to add tags to your app bundle. Tags are key-value pairs that assign metadata to resources that you create. For more information, see Tagging your AWS resources in the AWS Tag Editor User Guide.

8. To create your app bundle, choose Create app bundle.

Step 2: Authorize applications

After your app bundle is created successfully, you can now authorize AppFabric to connect and interact with each of your applications. Authorized applications are encrypted and stored in your app bundle. To set up multiple app authorizations per app bundle, repeat the app authorization step as needed for each application.

Before you begin the steps to authorize applications, review and verify prerequisites for each application, such as the plan type needed, in Supported applications (p. 17).

1. On the Getting started page, for Step 2. Authorize applications, choose Create app authorization.

2. In the App authorization section, select the application that you want to grant permission for AppFabric to connect to from the Application dropdown. The applications shown are those that are currently supported by AppFabric.

3. When you select an application, required fields of information appear. These fields include tenant ID and tenant name and might also include client ID, client secret, or personal access token. The input values for these fields varies by application. For detailed application-specific instructions on how to find these values, see Supported applications (p. 17).

4. (Optional) For Tags, you have the option to add tags to your app authorization. Tags are key-value pairs that assign metadata to resources that you create. For more information, see Tagging your AWS resources in the AWS Tag Editor User Guide.
5. Choose **Create app authorization**.

6. If a pop-up window appears (dependent upon the application that is being connected), select **Allow** to authorize AppFabric to connect with your application.

If your app authorization was successful, you will see a success message of **App authorization connected** on the **Getting Started** page.

7. You can check the status of your app authorization at any time on the **App authorizations** page listed in the navigation pane, under status for each application. A **Connected** status means that your app authorization has been granted for AppFabric to connect to the application and is complete.

8. Possible app authorization statuses are shown in the following table, including troubleshooting steps that you can take to fix related errors.

<table>
<thead>
<tr>
<th>Status name</th>
<th>Status description</th>
<th>Troubleshooting steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pending</td>
<td>A status of Pending means that an app authorization for the application is created, but AppFabric isn't yet connected to the application.</td>
<td>When you see this status, select <strong>Connect</strong> from the <strong>Actions</strong> dropdown of the <strong>App authorization</strong> page to initiate a connection. If this error persists, check if your browser's pop-up blocker is disabled. If there is any error message, like <strong>400 Bad Request</strong> in the pop-up window, check that all the information, such as tenant ID, client ID, and client secret, is correctly entered. It's also possible that the app authorization of the application isn't created correctly. For more information, see <strong>Supported applications</strong> (p. 17).</td>
</tr>
<tr>
<td>Connection validation failed</td>
<td>A status of Connection validation failed means that AppFabric can't validate the connection of the app authorization with an application.</td>
<td>Check that all the information, such as tenant ID, client ID and client secret, is entered correctly for the app authorization.</td>
</tr>
<tr>
<td>Token auto-rotation failed</td>
<td>A status of token auto-rotation failed means that the OAuth refresh token has failed after the app authorization was successfully connected.</td>
<td>If this error persists, check the authentication application of the application. For more information, see <strong>Supported applications</strong> (p. 17).</td>
</tr>
</tbody>
</table>

9. To authorize additional applications, repeat steps 1 through 8 as needed.

**Step 3: Set up audit log ingestions**

After you have at least one app authorization created in your app bundle, you can now set up an audit log ingestion. An audit log ingestion consumes audit logs from an authorized application and normalizes them into the Open Cybersecurity Schema Framework (OCSF). It then delivers them to one or more destinations within AWS. You can also choose to deliver raw JSON files to your destinations.
1. On the **Getting started** page, for the **Step 3. Set up audit log ingestions** section, select **Ingestions quick setup**.

   **Note**
   For faster setup, use the **Ingestions quick setup** page, accessible from the **Getting started** page only, to create ingestions for multiple app authorizations at a time, with the same ingestion destination. For example, the same Amazon S3 bucket or Amazon Kinesis Data Firehose data stream.
   You can also create ingestions from the **Ingestions** page, accessible from the navigation pane. On the **Ingestions** page, you can set up one ingestion at a time to distinct destinations. On the **Ingestions** page, you can also create a tag for an ingestion. The following instructions are for the **Ingestions quick setup** page.

2. For **Select app authorizations**, select the app authorizations that you want to create an audit log ingestions for. The tenant names that appear in the **App authorizations** dropdown are the tenant names of applications that you have previously created an app authorization for with AppFabric.

3. For **Add destination**, select a destination for the audit log ingestions of the applications that you selected. Destination options include **Amazon S3 - Existing Bucket**, **Amazon S3 - New Bucket**, or **Amazon Kinesis Data Firehose**. If you select multiple tenant names, the destination you choose is applied to each ingestion of an app authorization.

4. When you choose a destination, additional required fields appear.
   a. If you choose **Amazon S3 — New bucket** as your destination, you must enter the name of the S3 bucket that you want to create. For more instructions on how to create an Amazon S3 bucket, see **Create an output destination** (p. 8).
   b. If you choose **Amazon S3 — Existing bucket** as your destination, select the name of the Amazon S3 bucket that you want to use.
   c. If you choose **Amazon Kinesis Data Firehose** as your destination, select the name of the delivery stream from the Firehose delivery stream name dropdown list. For more instructions on how to create an Amazon Kinesis Data Firehose delivery stream, see **Create an output destination** (p. 8), and note the permissions policy required for AppFabric.

5. For **Schema & Format**, you can choose to store your audit logs in **Raw - JSON**, **OCSF - JSON**, **OCSF - Parquet for Amazon S3 buckets**, or **Raw - JSON or OCSF-JSON for Kinesis Data Firehose**.

   The Raw data format provides your audit log data converted to JSON from a string of data. The OCSF data format normalizes your audit log data to the AppFabric Open Cybersecurity Schema Framework (OCSF) schema. For more information about how AppFabric uses OCSF, see **Open Cybersecurity Schema Framework** (p. 6). You can select only one schema and format data type at a time for an ingestion. If you want to add an additional schema and format data type, you can set up an additional ingestion destination by repeating the ingestion creation process.

6. (Optional) If you want to add a tag to an ingestion, go to the **Ingestions** page from the navigation pane. To go to the ingestion details page, select the tenant name. For **Tags**, you have the option to add tags to your ingestion. Tags are key-value pairs that assign metadata to resources that you create. For more information, see **Tagging your AWS resources** in the **AWS Tag Editor User Guide**.

7. Choose **Set up ingestions**.

   When you successfully set up an ingestion, you will see a success message of **Ingestion created** on the **Getting Started** page.

8. You can also check the state of your ingestions and status of your ingestion destinations at any time on the **Ingestions** page from the navigation pane. On this page, you can see the tenant name created upon creating app authorization, destination, and state of your ingestions. A state of **Enabled** for your ingestion means that your ingestion is enabled. If you choose the tenant name of an app authorization on this page, you can see a detail page for that app authorization, including destination details and status. A status of **Active** for your ingestion destination means that the destination is set up properly and active. If the app authorization has the **Connected** status and the ingestion destination status is **Active**, then the audit log should be processed and delivered. If the app authorization status or the ingestion destination status are any of the failed states, the audit log
will not be processed or delivered even if the ingestion status is enabled. To fix an app authorization failure, see Step 2. Authorize applications (p. 12).

9. Possible ingestion and ingestion destination statuses are shown in the following table, with troubleshooting steps that you can take to fix any error status.

<table>
<thead>
<tr>
<th>State or status name</th>
<th>Description</th>
<th>Troubleshooting steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>A Disabled state for the ingestion means that your ingestion is disabled.</td>
<td>You can enable the ingestion by selecting Enable from the Actions dropdown of the Ingestions page.</td>
</tr>
<tr>
<td>Failed</td>
<td>A Failed state for the ingestion destination means that the ingestion destination isn't accepting the audit log. For example, this status might occur because of a full storage location.</td>
<td>To fix these issues, go to the Amazon S3 or Kinesis Data Firehose consoles.</td>
</tr>
</tbody>
</table>

**Step 4: Use the user access tool**

Using the AppFabric user access tool, security and IT Admin teams can quickly see who has access to specific applications by running a simple search using the employee’s corporate email address. This approach can be helpful in reducing time spent on tasks like user deprovisioning that might require manually checking or auditing a user’s access across SaaS applications. If a user is found, AppFabric provides the user’s name in the application and their in-app user status (for example, Active) if provided by the application. AppFabric searches all authorized applications in an app bundle to return a list of applications that the user has access to.

1. On the Getting Started page, for Step 4. Use the user access tool, choose Look up user.
2. In the Email address field, type a user’s email address, and choose Search.
3. In the Search results section, you see a list of all authorized applications that the user has access to. To show the user’s name in the application and their status (if available), select a search result.
4. A message of User found in the search results column means that the user can access the app listed. The following table shows the possible search results, errors, and the actions that you can take to address the errors.

<table>
<thead>
<tr>
<th>Search result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user not found</td>
<td>No user is found with the email address used.</td>
</tr>
<tr>
<td>An authorization token was not found. Connect the app authorization for the application.</td>
<td>Check that all the information, such as tenant ID, client ID, and client secret, is entered correctly for the app authorization.</td>
</tr>
<tr>
<td>The authorization token was revoked. Connect the app authorization for the application.</td>
<td>Check that all the information, such as tenant ID, client ID, and client secret, is entered correctly for the app authorization.</td>
</tr>
<tr>
<td>We were unable to rotate the authorization token. Connect the app authorization for the application.</td>
<td>The OAuth refresh token has failed after the app authorization was successfully connected. If this error persists, check the authentication application of the application.</td>
</tr>
</tbody>
</table>
Step 5: Connect AppFabric data in security tools and other destinations

Normalized (or raw) application data from AppFabric is compatible with any tool that supports data ingestion from Amazon S3 and integration with Kinesis Data Firehose, including security tools like Logz.io, Netskope, NetWitness, Rapid7, and Splunk, or your proprietary security solution. To get normalized (or raw) application data from AppFabric, follow the previous steps 1 through 3. For more details on how to set up specific security tools and services, see Compatible security tools and services (p. 50).
Supported applications

AWS AppFabric supports integration with the following applications. Choose the name of an application for more information about how to set up AppFabric to connect to it.

Topics

• Asana (p. 17)
• Atlassian Jira suite (p. 18)
• Dropbox (p. 21)
• GitHub (p. 23)
• Google Workspace (p. 26)
• Microsoft 365 (p. 28)
• Miro (p. 30)
• Okta (p. 33)
• ServiceNow (p. 35)
• Slack (p. 37)
• Smartsheet (p. 40)
• Webex by Cisco (p. 42)
• Zendesk (p. 45)
• Zoom (p. 47)

Asana

Asana is a work management platform that helps individuals, teams, and organizations orchestrate work, from daily tasks to cross-functional strategic initiatives. It provides a living system of clarity where everyone can communicate, collaborate, and coordinate work. With Asana, teams integrate critical business tools into one place so work moves forward no matter where it happens.

You can use AWS AppFabric to receive audit logs and user data from Asana, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

Topics

• AppFabric support for Asana (p. 17)
• Connecting AppFabric to your Asana account (p. 18)

AppFabric support for Asana

AppFabric supports receiving user information and audit logs from Asana.

Prerequisites

To use AppFabric to transfer audit logs from Asana to supported destinations, you must meet the following requirements:

• You must have an Enterprise account with Asana. For more information about creating or upgrading to an Asana Enterprise account, see Asana Enterprise on the Asana website.
• You must have a user with the Super Admin role in your Asana account. For more information about roles, see Admin and super admin roles in Asana on the Asana website.

Rate limit considerations

Asana imposes rate limits on the Asana API. For more information about the Asana API rate limits, see Rate limits on the Asana Developers Guide website. If the combination of AppFabric and your existing Asana applications exceed the limit, audit logs appearing in AppFabric might be delayed.

Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.

Connecting AppFabric to your Asana account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Asana. To find the information required to authorize Asana with AppFabric, use the following steps.

App authorizations

Tenant ID

AppFabric will request your tenant ID. The tenant ID in AppFabric is called the domain ID in Asana. To find the domain ID, use the following instructions from the Asana home screen:

1. Choose your account profile picture and select Admin Console.
2. Then select Settings.
3. Scroll to Domain Settings.
4. Enter the domain ID from this section into the AppFabric Tenant ID configuration.

Tenant name

Enter a name that identifies this unique Asana organization. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

Service account token

You must have a service account token from an Asana service account to enter into the AppFabric Asana app authorization. If you don't have a service account token, use the following instructions:

1. To create a service account, follow the instructions in Service Accounts on the Asana Guide website.
2. Copy and save the token from the bottom of the Add service account page the first time you view the Add service account page.
3. If you close the Add service account page before saving the token, you must edit your service account, generate a new token, and save it.

Atlassian Jira suite

Atlassian unleashes the potential of every team. Their agile and DevOps, IT service management and work management software helps teams organize, discuss, and complete shared work. The majority of
the Fortune 500 and over 240,000 companies of all sizes worldwide - including NASA, Kiva, Deutsche Bank, and Salesforce - rely on Atlassian solutions to help their teams work better together and deliver quality results on time. Learn more about Atlassian products, including Jira Software, Confluence, Jira Service Management, Trello, Bitbucket, and Jira Align at Atlassian.

You can use AWS AppFabric to receive audit logs and user data from the Jira suite (other than Jira Align), normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

Topics
- AppFabric support for the Jira suite (p. 19)
- Connecting AppFabric to your Jira account (p. 19)

AppFabric support for the Jira suite

AppFabric supports receiving user information and audit logs from the Jira suite, with the exception of Jira Align.

Prerequisites

To use AppFabric to transfer audit logs from the Jira suite to supported destinations, you must meet the following requirements:

- You must have a Jira Standard Plan or higher. For more information about the capabilities of the Jira plans, see Jira Software, Jira Service Management, Jira Work Management, and Jira Product Discovery pricing pages.
- You must have a user with the Organization admin role in your Jira account. For more information about roles, see Give users admin permissions on the Atlassian Support website.

Rate limit considerations

The Jira suite imposes rate limits on the Jira API. For more information about the Jira suite API rate limits, see Rate limiting on the Atlassian Developers Guide website. If the combination of AppFabric and your existing Jira API applications exceed the limit, audit logs appearing in AppFabric might be delayed.

Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.

Connecting AppFabric to your Jira account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Jira. To find the information required to authorize Jira with AppFabric, use the following steps.

Create an OAuth application

AppFabric integrates with the Jira suite using OAuth. To create an OAuth application in Jira, use the following steps:

1. Navigate to the Atlassian Developer Console.
2. Next to My apps, choose Create, OAuth 2.0 integration.
3. Give your app a name and choose Create.
4. Navigate to the Authorization section and choose Add next to OAuth 2.0.
5. Use a URL with the following format in the Callback URL field and choose Save changes:

   https://<region>.console.aws.amazon.com/appfabric/oauth2

   In this URL, <region> is the code for the AWS Region in which you configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.
6. Navigate to the Settings section, copy your client ID and client secret, and save it to use for the AppFabric app authorization.

**Required scopes**

You must add the following scopes to your Jira OAuth application’s Permissions page:

- Under Classic Scopes:
  - Jira API > read:jira-user
- Under Granular Scopes:
  - Jira API > read:audit-log:jira
  - Jira API > read:user:jira

**App authorizations**

**Tenant ID**

AppFabric will request your tenant ID. The tenant ID in AppFabric is your Jira instance subdomain. You can find your Jira instance subdomain in your browser’s address bar between https:// and .atlassian.net.

**Tenant name**

Enter a name that identifies this unique Jira server. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

**Client ID**

AppFabric will request your client ID. To find your client ID in Jira, use the following steps:

1. Navigate to the Atlassian Developer Console.
2. Select the OAuth application that you use to connect AppFabric.
3. Enter the client ID from the Settings page into the client ID field in AppFabric.

**Client secret**

AppFabric will request your client secret. The Client secret in AppFabric is the Secret in Jira. To find your Secret in Jira, use the following steps:

1. Navigate to the Atlassian Developer Console.
2. Select the OAuth application that you use to connect AppFabric.
3. Enter the secret from the **Settings** page into the **Client Secret** field in AppFabric.

**Approve authorization**

After creating the app authorization in AppFabric you will receive a pop-up window from Jira to approve the authorization. To approve the AppFabric authorization, choose **Allow**.

**Dropbox**

Dropbox helps your organization get better work done faster by bringing your people together - no matter what they're working on, where they're working, or what kind of tools they happen to be using. It enables users to accelerate innovation and efficiency by providing a simple, secure way to share content. Dropbox is one place to keep life organized and keep work moving. With more than 700 million registered users across 180 countries, Dropbox is on a mission to design a more enlightened way of working.

You can use AWS AppFabric to receive audit logs and user data from Dropbox, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

**Topics**

- AppFabric support for Dropbox (p. 21)
- Connecting AppFabric to your Dropbox account (p. 22)

**AppFabric support for Dropbox**

AppFabric supports receiving user information and audit logs from Dropbox.

**Prerequisites**

To use AppFabric to transfer audit logs from Dropbox to supported destinations, you must meet the following requirements:

- You must have a Dropbox Business account. For more information about creating or upgrading to a Dropbox Business account, see Dropbox Business on the Dropbox website.
- You must have a user with the Team Admin role in your Dropbox account. For more information about roles, see How to change admin rights for your Dropbox team on the Dropbox Help Center website.

**Rate limit considerations**

Dropbox imposes rate limits on the Dropbox API. For more information about the Dropbox API rate limits, see Rate limits on the Dropbox Performance Guide website. If the combination of AppFabric and your existing Dropbox API applications exceed the limit, audit logs appearing in AppFabric might be delayed.

**Data delay considerations**

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.
Connecting AppFabric to your Dropbox account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Dropbox. To find the information required to authorize Dropbox with AppFabric, use the following steps.

Create an OAuth application

AppFabric integrates with Dropbox using OAuth. To create an OAuth application in Dropbox, use the following steps:

2. On the new application configuration page, choose Scoped access for the API.
3. Next, select Full Dropbox for the type of access.
4. Name your OAuth application, and then chose Create app to complete the initial OAuth application setup.
5. On the application info page, add a redirect URL with the following format in the OAuth2 redirect URIs field.

   https://<region>.console.aws.amazon.com/appfabric/oauth2

   In this URL, <region> is the code for the AWS Region in which you configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.
6. Choose Add.
7. Copy and save your app key and app secret for use in the AppFabric app authorization.
8. You can leave all other fields on the Settings tab with their default values.

Required scopes

You must add the following scopes to your Dropbox app using the Permissions tab on the app info screen:

- account_info.read
- team_data.member
- events.read
- members.read
- team_info.read

Choose Submit after you are done.

App authorizations

Tenant ID

AppFabric will request your tenant ID. Enter any value that uniquely identifies your Dropbox account, such as team name.

Tenant name

Enter a name that identifies this unique Dropbox account. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.
Client ID

AppFabric will request a client ID. The client ID in AppFabric is your Dropbox app key. To find your Dropbox app key, use the following steps:

2. Find the app that you use to connect AppFabric.
3. Find the app key in the Status section of the app’s info page.
4. Enter the app key for your Dropbox app into the Client ID field in AppFabric.

Client secret

AppFabric will request a client secret. The client secret in AppFabric is your Dropbox app secret. To find your Dropbox app secret, use the following steps:

2. Find the app that you use to connect AppFabric.
3. Find the app secret in the Status section of the app’s info page.
4. Enter the app secret for your Dropbox app into the Client Secret field in AppFabric.

Approve authorization

After creating the app authorization in AppFabric, you will receive a pop-up window from Dropbox to approve the authorization. To approve the AppFabric authorization, choose Allow.

GitHub

GitHub is a platform and cloud-based service for software development and version control using Git, allowing developers to store and manage their code. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project. You can use AWS AppFabric to receive audit logs and user data from GitHub, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

Topics

- AppFabric support for GitHub (p. 23)
- Connecting AppFabric to your GitHub account (p. 24)

AppFabric support for GitHub

AppFabric supports receiving user information and audit logs from GitHub.

Prerequisites

To use AppFabric to transfer audit logs from GitHub to supported destinations, you must meet the following requirements:

- To access the Audit logs you need to have an enterprise account.
- To access the Enterprise audit logs you need to have Administrator role for your enterprise account.
- To get audit logs from organization, you need to be Organization owner.
Rate limit considerations

GitHub imposes rate limits on the GitHub API. For more information about the GitHub API rate limits, see API Request Limits and Allocations on the GitHub website. If the combination of AppFabric and your existing GitHub API applications exceed GitHub’s limits, audit logs appearing in AppFabric may be delayed.

Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.

Connecting AppFabric to your GitHub account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with GitHub. To find the information required to authorize GitHub with AppFabric, use the following steps.

Create an OAuth application

AppFabric integrates with the GitHub using OAuth. Use the following steps to create an OAuth application in GitHub. For more information, see Creating an OAuth App on the GitHub website.

1. Choose your profile photo located in the top-right corner of the page, and then choose Settings.
2. Choose Developer settings in the left navigation pane.
3. Choose OAuth apps in the left navigation pane.
   Note
   This button will be labeled Register a new application if you haven't previously created an OAuth app.
5. Enter the name of your app in the Application name text box.
6. Enter the full application instance URL in the Homepage URL text box.
7. (Optional) Enter a description for your app in the Application description text box. Users will see this description.
8. Enter a URL with the following format in the Authorization callback URL text box.

   https://<region>.console.aws.amazon.com/appfabric/oauth2

   In this URL, <region> is the code for the AWS Region in which you configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.
9. Choose Enable Device Flow if your OAuth app will use device flow to identify and authorize users. For more information about device flow, see Authorizing OAuth apps on the GitHub website.
10. Choose Register application.

App authorizations

Tenant ID

AppFabric will request your tenant ID. The tenant ID should be provided in either of the following formats:
Enterprise audit log:

Use the enterprise's audit log if you want to know aggregated actions from all of the organizations owned by your enterprise account.

To use the enterprise audit log, the tenant ID is your account's enterprise ID. You can find your enterprise ID in the address bar of your browser. For example, exampleenterprise is the enterprise ID in the following URL https://github.com/settings/enterprises/exampleenterprise.

When you specify the tenant ID for enterprise audit log, you must prefix it with enterprise:. Therefore, specify the previous example as enterprise:exampleenterprise.

Organization audit log:

Use the organization's audit log as an organization admin if you want to know the actions performed by members of your organization. It includes details such as who performed the action, what the action was, and when it was performed.

To use organization audit log, the tenant ID is your organization ID. You can find your organization ID in the address bar of your browser. For example, exampleorganization is the organization ID in the following URL https://github.com/settings/organizations/exampleorganization.

When you specify the tenant ID for organization audit log, you must prefix it with organization:. Therefore, specify the previous example as organization:exampleorganization.

Tenant name

Enter a name that identifies this unique GitHub enterprise or organization. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

Client ID

AppFabric will request a client ID. Use the following steps to find your client ID in GitHub,

1. Choose your profile photo located in the top-right corner of the page, and then choose Settings.
2. Choose Developer settings in the left navigation pane.
3. Choose OAuth apps in the left navigation pane.
4. Choose the specific OAuth app, and then look for the Client ID value.

Client secret

AppFabric will request a client secret. Use the following steps to find your client secret in GitHub.

1. Choose your profile photo located in the top-right corner of the page, and then choose Settings.
2. Choose Developer settings in the left navigation pane.
3. Choose OAuth apps in the left navigation pane.
4. Choose the specific OAuth app, and then look for the Client Secret value. If you are unable to find an existing client secret, then you might need to generate a new one.

Approve authorization

After creating the app authorization in AppFabric, you will receive a pop-up window from GitHub to approve the authorization. To approve the AppFabric authorization, choose Allow.
Make sure that your organizations have [granted access](#) to the OAuth app, if [OAuth App access restrictions](#) are enabled.

**Google Workspace**

Google Workspace is a collection of cloud computing, productivity and collaboration tools, software and products developed and marketed by Google.

You can use AWS AppFabric to receive audit logs and user data from Google Workspace, normalize the data into [Open Cybersecurity Schema Framework (OCSF)](#) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

**Topics**

- [AppFabric support for Google Workspace](#)
- [Connecting AppFabric to your Google Workspace account](#)

---

**AppFabric support for Google Workspace**

AppFabric supports receiving user information and audit logs from Google Workspace.

**Prerequisites**

To use AppFabric to transfer audit logs from Google Workspace to supported destinations, you must meet the following requirements:

- You must subscribe to the Google Workspace Enterprise Standard plan. For more information about creating or upgrading to the Google Workspace Enterprise Standard plan, see the [Google Workspace Plans](#) website.
- You must have a user with the Administrator role in your Google Workspace.
- For AppFabric to deliver logs, you need to enable Google Admin SDK API on your Google Cloud project. For more information, see [Enable Google Workspace APIs](#) in the [Google Workspace Developer Guide](#).

**Rate limit considerations**

Google Workspace imposes rate limits on the Google Workspace API. For more information about Google Workspace API rate limits, see [Limits and Quotas](#) on the [Google Workspace Admin Guide](#) on the Google Workspace website. If the combination of AppFabric and your existing Google Workspace API applications exceed the limit, audit logs appearing in AppFabric might be delayed.

**Data delay considerations**

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact [AWS Support](#).

**Connecting AppFabric to your Google Workspace account**

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Google Workspace. To find the information required to authorize Google Workspace with AppFabric, use the following steps.
Create an OAuth application

AppFabric integrates with Google Workspace using OAuth. To create an OAuth application in Google Workspace, use the following steps:

1. To configure your OAuth consent screen, follow the instructions in Configure the OAuth consent screen in the Google Workspace Developer Guide on the Google Workspace website.

   Choose Internal for the User type.

2. To configure OAuth credentials for AppFabric, follow the instructions in the OAuth client ID credentials section of the Create access credentials page in the Google Workspace Developer Guide.

3. Use a redirect URL with the following format.

   https://<region>.console.aws.amazon.com/appfabric/oauth2

   In this URL, <region> is the code for the AWS Region in which you've configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.

Required scopes

You must add the following scopes to your Google Workspace OAuth application:

- https://www.googleapis.com/auth/admin.reports.audit.readonly
- https://www.googleapis.com/auth/admin.directory.user

If you don't see these scopes, add the Admin SDK API to your Google Cloud API library.

App authorizations

Tenant ID

AppFabric will request your tenant ID. The tenant ID in AppFabric is your Google Workspace project ID. To find your project ID, see Locate the project ID on the Google API Console Help website.

Tenant name

Enter a name that identifies this unique Google Workspace. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

Client ID

AppFabric will request your client ID. To find your client ID, use the following steps:

1. Find your client ID using the information in the View Credentials section of the Manage Credentials page in the Google Workspace Developer Guide.
2. Enter the client ID for your OAuth client into the Client ID field in AppFabric.

Client secret

AppFabric will request your client secret. To find your client secret, use the following steps:
1. Find your client secret using the information in the View Credentials section of the Manage Credentials page on the Google Workspace Developer Guide.

2. If you need to reset your client secret, use the instructions in the Reset Client Secret section of the Manage Credentials page on the Google Workspace Developer Guide.

3. Enter the your client secret into the Client secret field in AppFabric.

Approve authorization

After creating the app authorization in AppFabric you will receive a pop-up window from Google Workspace to approve the authorization. To approve the AppFabric authorization, choose allow.

Microsoft 365

Microsoft 365 is a product family of productivity software, collaboration, and cloud-based services owned by Microsoft.

You can use AWS AppFabric to receive audit logs and user data from Microsoft 365, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

Topics

- AppFabric support for Microsoft 365 (p. 28)
- Connecting AppFabric to your Microsoft 365 account (p. 29)

AppFabric support for Microsoft 365

AppFabric supports receiving user information and audit logs from Microsoft 365.

Prerequisites

To use AppFabric to transfer audit logs from Microsoft 365 to supported destinations, you must meet the following requirements:

- You must subscribe to a Microsoft 365 Enterprise plan. For more information about creating or upgrading to a Microsoft 365 Enterprise plan, see Microsoft 365 Enterprise Plans on the Microsoft website.
- You must have a user with Administrator permissions in your Microsoft 365 account.
- You must turn on audit logging for your organization. For more information, see Turn auditing on or off on the Microsoft website.

Rate limit considerations

Microsoft 365 imposes rate limits on the Microsoft 365 API. For more information about Microsoft 365 API rate limits, see Microsoft Graph service-specific throttling limits in the Microsoft Graph documentation on the Microsoft website. If the combination of AppFabric and your existing Microsoft 365 API applications exceed the limit, audit logs appearing in AppFabric might be delayed.

Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.
Connecting AppFabric to your Microsoft 365 account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Microsoft 365. To find the information required to authorize Microsoft 365 with AppFabric, use the following steps.

Create an OAuth application

AppFabric integrates with Microsoft 365 using OAuth. To create an OAuth application in Microsoft 365, use the following steps:

1. Follow the instructions in the Register an application section in the Azure Active Directory Developer Guide on the Microsoft website.
   
   Choose Accounts in this organizational directory only in the Supported Account Types configuration.

2. Follow the instructions in the Add a redirect URI section in the Azure Active Directory Developer Guide.
   
   Choose the Web platform.

   https://<region>.console.aws.amazon.com/appfabric/oauth2

   In this URL, <region> is the code for the AWS Region in which you've configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.

   You can skip the other input fields for the Web platform.

3. Follow the instructions in the Add a client secret section of the Azure Active Directory Developer Guide.

Required permissions

You must add the following permissions to your OAuth application. To add permissions, follow the instructions in the Add permissions to access your web API section of the Azure Active Directory Developer Guide.

- Microsoft Graph API > User.Read (automatically added)
- Office 365 Management APIs > ActivityFeed.Read (Select Delegated Type)
- Office 365 Management APIs > ActivityFeed.ReadDlp (Select Delegated Type)
- Office 365 Management APIs > ServiceHealth.Read (Select Delegated Type)

After you've added the permissions, to grant admin consent for the permissions, follow the instructions in the Admin consent button section of the Azure Active Directory Developer Guide.

App authorizations

AppFabric supports receiving user information and audit logs from your Microsoft 365 account. To receive both audit logs and user data from Microsoft 365, you must create two app authorizations, one that is named Microsoft 365 in the app authorization drop-down list, and another that is named Microsoft 365 Audit Log in the app authorization drop-down list. You can use the same tenant ID, client
ID and client secret for both app authorizations. To receive audit logs from Microsoft 365, you need both the Microsoft 365 and Microsoft 365 Audit Log app authorizations. To use the user access tool alone, only the Microsoft 365 app authorization is required.

**Tenant ID**

AppFabric will request your tenant ID. The tenant ID in AppFabric is your Azure Active Directory tenant ID. To find your Azure Active Directory tenant ID, see How to find your Azure Active Directory tenant ID in the Azure Product Documentation on the Microsoft website.

**Tenant name**

Enter a name that identifies this unique Microsoft 365 account. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

**Client ID**

AppFabric will request your client ID. The client ID in AppFabric is the Microsoft 365 application (client) ID. To find your Microsoft 365 application (client) ID, use the following steps:

1. Open the overview page for the OAuth application that you use with AppFabric.
2. The application (client) ID appears under Essentials.
3. Enter the application (client) ID for your OAuth client into the Client ID field in AppFabric.

**Client secret**

AppFabric will request your client secret. Microsoft 365 provides this value only when you initially create the client secret for your OAuth application. To generate a new client secret if you don't have one, use the following steps:

1. To create a client secret, follow the instructions in the Add a client secret section of the Azure Active Directory Developer Guide.
2. Enter the contents of the Value field into the Client secret field in AppFabric.

**Approve authorization**

After creating the app authorization in AppFabric, you will receive a pop-up window from Microsoft 365 to approve the authorization. To approve the AppFabric authorization, choose allow.

---

**Miro**

Miro is an online workspace for innovation that enables distributed teams of any size to build the next big thing. The platform's infinite canvas enables teams to lead engaging workshops and meetings, design products, brainstorm ideas, and more. Miro, co-headquartered in San Francisco and Amsterdam, serves more than 50M users worldwide, including 99% of the Fortune 100. Miro was founded in 2011 and currently has more than 1,500 employees in 12 hubs around the world. To learn more, visit Miro.

Miro includes a full suite of collaborative capabilities designed for innovation including diagramming, wireframing, real-time data visualization, workshop facilitation, and built-in support for agile practices, workshops, and interactive presentations. Miro recently announced Miro AI which extends Miro's capabilities, with AI-driven mapping and diagramming, clustering and summarization, and content generation. Miro enables organizations to reduce the number of standalone tools, reducing information fragmentation and cost.
You can use AWS AppFabric to receive audit logs and user data from Miro, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

Topics

- AppFabric support for Miro (p. 31)
- Connecting AppFabric to your Miro account (p. 31)

AppFabric support for Miro

AppFabric supports receiving user information and audit logs from Miro.

Prerequisites

To use AppFabric to transfer audit logs from Miro to supported destinations, you must meet the following requirements:

- You must have a Miro Enterprise Plan. For more information about the Miro plan types, see the Miro pricing page on the Miro website.
- You must have a user with the Company Admin role in your Miro account. For more information about roles, see the Company level section of Roles in Miro on the Miro Help Center website.
- You must have an Enterprise Developer team in your Miro account. For information about creating developer teams, see Enterprise Developer teams on the Miro Help Center website.

Rate limit considerations

Miro imposes rate limits on the Miro API. For more information about the Miro API rate limits, see Rate Limiting in the Miro Developers Guide on the Miro website. If the combination of AppFabric and your existing Miro API applications exceed the limit, audit logs appearing in AppFabric might be delayed.

Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.

Connecting AppFabric to your Miro account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Miro. To find the information required to authorize Miro with AppFabric, use the following steps.

Create an OAuth application

AppFabric integrates with Miro using OAuth. To create an OAuth application in Miro, use the following steps:

1. To create an OAuth application, follow the instructions in the Creating and installing apps section of the Enterprise Developer teams article on the Miro Help Center website.
2. On the app creation dialog, select the Expire user authorization token check box after you select a developer team on the enterprise organization.

   Note
   You must do this before creating the app because you can't change this option after you create the app.
3. On the app page, enter a URL with the following format in the Redirect URI for OAuth 2.0 section.

   https://<region>.console.aws.amazon.com/appfabric/oauth2

   In this URL, `<region>` is the code for the AWS Region in which you've configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.


**Required scopes**

You must add the following scopes on the Permissions section of your Miro OAuth app page:

- auditlogs:read
- organizations:read

**App authorizations**

**Tenant ID**

AppFabric will request your tenant ID. The tenant ID in AppFabric is your Miro Team ID. For information about how to find your Miro Team ID, see the Frequently Asked Questions section of I am a new Miro Admin. Where to start? on the Miro Help Center website.

**Tenant name**

Enter a name that identifies this unique Miro organization. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

**Client ID**

AppFabric will request your client ID. To find your client ID, use the following steps:

1. Navigate to your Miro profile settings.
2. Select the Your apps tab.
3. Select the app that you use to connect with AppFabric.
4. Enter the client ID from the App Credentials section into the Client ID field in AppFabric.

**Client secret**

AppFabric will request your client secret. To find your client secret, use the following steps:

1. Navigate to your Miro profile settings.
2. Select the Your apps tab.
3. Select the app that you use to connect with AppFabric.
4. Enter the client secret from the App Credentials section into the Client secret field in AppFabric.

**Approve authorization**

After creating the app authorization in AppFabric, you will receive a pop-up window from Miro to approve the authorization. To approve the AppFabric authorization, choose Allow.
Okta

Okta is the World's Identity Company. As the leading independent Identity partner, Okta frees everyone to safely use any technology—anywhere, on any device or app. The most trusted brands trust Okta to enable secure access, authentication, and automation. With flexibility and neutrality at the core of the Okta Workforce Identity and Customer Identity Clouds, business leaders and developers can focus on innovation and accelerate digital transformation, thanks to customizable solutions and more than 7,000 pre-built integrations. Okta is building a world where Identity belongs to you. Learn more at okta.com.

You can use AWS AppFabric to receive audit logs and user data from Okta, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

Topics
- AppFabric support for Okta (p. 33)
- Connecting AppFabric to your Okta account (p. 33)

AppFabric support for Okta

AppFabric supports receiving user information and audit logs from Okta.

Prerequisites

To use AppFabric to transfer audit logs from Okta to supported destinations, you must meet the following requirements:

- You can use AppFabric with any Okta plan type.
- You must have a user with the Super Admin role in your Okta account.
- The user approving the app authorization in AppFabric must also have the Super Admin role in your Okta account.

Rate limit considerations

Okta imposes rate limits on the Okta API. For more information about the Okta API rate limits, see Rate limits in the Okta Developer Guide on the Okta website. If the combination of AppFabric and your existing Okta API applications exceed Okta's limits, audit logs appearing in AppFabric might be delayed.

Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.

Connecting AppFabric to your Okta account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Okta. To find the information required to authorize Okta with AppFabric, use the following steps.

Create an OAuth application

AppFabric integrates with Okta using OAuth. To create an OAuth application to connect with AppFabric, follow the instructions in Create OIDC app integrations on the Okta Help Center website. Following are configuration considerations for AppFabric:
1. For **Application Type**, choose **Web application**.
2. For **Grant type**, choose **Authorization Code** and **Refresh Token**.
3. Use a redirect URL with the following format as the **Sign-in redirect URI** and **Sign-out redirect URI**.

   ```
   https://<region>.console.aws.amazon.com/appfabric/oauth2
   ```

   In this URL, `<region>` is the code for the AWS Region in which you've configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.

4. You can skip the **Trusted Origins** configuration.
5. Grant access to everyone in your Okta organization in the **Controlled access** configuration.

   **Note**
   If you skip this step during initial OAuth application creation, you can assign everyone in your organization as a group using the **Assignments** tab on the application configuration page.

6. You can leave all other options with their default values.

**Required scopes**

You must add the following scopes to your Okta OAuth application:

- `okta.logs.read`
- `okta.users.read`

**App authorizations**

**Tenant ID**

AppFabric will request a tenant ID. The tenant ID in AppFabric is your Okta domain. For more information about finding your Okta domain, see [Find your Okta domain](#) in the Okta Developer Guide on the Okta website.

**Tenant name**

Enter a name that identifies this unique Okta organization. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

**Client ID**

AppFabric will request a client ID. To find your client ID in Okta, use the following steps:

1. Navigate to the Okta developer console.
2. Choose the **Applications** tab.
3. Choose your application and then choose the **General** tab.
4. Scroll to the **Client Credentials** section.
5. Enter the client ID from your OAuth client into the **Client ID** field in AppFabric.

**Client secret**

AppFabric will request a client secret. To find your client secret in Okta, use the following steps:
1. Navigate to the Okta developer console.
2. Choose the **Applications** tab.
3. Choose your application and then choose the **General** tab.
4. Scroll to the **Client Credentials** section.
5. Enter the client secret from your OAuth application into the **Client Secret** field in AppFabric.

**Approve authorization**

After creating the app authorization in AppFabric, you will receive a pop-up window from Okta to approve the authorization. To approve the AppFabric authorization, choose **allow**. The user approving the Okta authorization must have **Super Admin** permission in Okta.

**ServiceNow**

ServiceNow is a leading provider of cloud-based services that automate enterprise IT operations. ServiceNow’s ITOM gives enterprises complete visibility and control of their entire IT environment — including virtualized and cloud infrastructure. It simplifies service mapping, delivery and assurance, consolidating IT service and infrastructure data into a single system of record. It also automates and streamlines key processes — including event, incident, problem, configuration and change management. You can use AWS AppFabric to receive audit logs and user data from ServiceNow, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

**Topics**

- [AppFabric support for ServiceNow](#)
- [Data delay considerations](#)
- [Connecting AppFabric to your ServiceNow account](#)

**AppFabric support for ServiceNow**

AppFabric supports receiving user information and audit logs from ServiceNow.

**Prerequisites**

To use AppFabric to transfer audit logs from ServiceNow to supported destinations, you must meet the following requirements:

- You can use AppFabric with any ServiceNow plan type.
- You must have a user with the Administrator role in your ServiceNow account.
- You must have a ServiceNow instance.

**Rate limit considerations**

ServiceNow imposes rate limits on the ServiceNow API. For more information about the ServiceNow API rate limits, see [Inbound REST API rate limiting](#) on the ServiceNow website. If the combination of AppFabric and your existing ServiceNow API applications exceed the limits, audit logs appearing in AppFabric may be delayed.
Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.

Connecting AppFabric to your ServiceNow account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with ServiceNow. Use the following steps to find the information required to authorize ServiceNow with AppFabric.

Create an OAuth application

The Now Platform supports OAuth 2.0 - Authorization Grant type for public clients to generate an access token.

1. Register your OAuth application. This requires the following three steps. For more information on completing these steps, see the Register your application with ServiceNow on the ServiceNow website.
   a. Register the app and make sure the Auth Scope has access to the Table API, with a REST API PATH of now/table, and an HTTP Method of GET as shown in the following example.

   ![REST API Auth Scope](image)

   b. Generate an authorization code.
   c. Generate a bearer token using the authorization code.

2. Use a redirect URL with the following format.

   https://<region>.console.aws.amazon.com/appfabric/oauth2

   In this URL, <region> is the code for the AWS Region in which you configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.
App authorizations

Tenant ID

AppFabric will request a tenant ID. The tenant ID in AppFabric is your instance name. You can find your tenant ID in the address bar of your browser. For example, `example` is the tenant ID in the following URL: `https://example.service-now.com`.

Tenant name

Enter a name that identifies this unique ServiceNow organization. AppFabric uses the tenant's name to label the app authorizations and any ingestions created from the app authorization.

Client ID

AppFabric will request a client ID. Use the following steps to find your client ID in ServiceNow.

1. Navigate to the ServiceNow console.
2. Choose System OAuth, and then choose the Application Registry tab.
3. Choose your application.
4. Enter the client ID from your OAuth client into the Client ID field in AppFabric.

Client secret

AppFabric will request a client secret. Use the following steps to find your client secret in ServiceNow.

1. Navigate to the ServiceNow console.
2. Choose System OAuth, and then choose the Application Registry tab.
3. Choose your application.
4. Enter the client secret from your OAuth application into the Client Secret field in AppFabric.

Approve authorization

After creating the app authorization in AppFabric, you will receive a pop-up window from ServiceNow to approve the authorization. Choose Allow to approve the AppFabric authorization.

Slack

Slack is on a mission to make people's working lives simpler, more pleasant, and more productive. It is the productivity platform for customer companies that improves performance by empowering everyone with no-code automation, making search and knowledge sharing seamless, and keeping teams connected and engaged as they move work forward together. As part of Salesforce, Slack is deeply integrated into the Salesforce Customer 360, supercharging productivity across sales, service and marketing teams. To learn more and get started with Slack for free, visit `slack.com`.

You can use AWS AppFabric to receive audit logs and user data from Slack, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

Topics
AppFabric support for Slack

AppFabric supports receiving user information and audit logs from Slack.

Prerequisites

To use AppFabric to transfer audit logs from Slack to supported destinations, you must meet the following requirements:

• You must have an Enterprise Grid plan with Slack. For more information, see An introduction to Slack Enterprise Grid on the Slack website.
• You must have a user with the Org Owner role in your Slack account. For more information about roles, see Types of roles in Slack in the Slack Help Center on the Slack website.

Rate limit considerations

Slack imposes rate limits on the Slack API. For more information about Slack API rate limits, see Rate limits in the Slack API Usage Guide on the Slack website. If the combination of AppFabric and your existing Slack API applications exceed the limit, audit logs appearing in AppFabric might be delayed.

Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.

Connecting AppFabric to your Slack account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Slack. To find the information required to authorize Slack with AppFabric, use the following steps.

Create an OAuth application

AppFabric integrates with Slack using OAuth. There are two ways to create an OAuth app: Using an app manifest or From scratch. To create an OAuth application in Slack, use the following steps.

Using an app manifest

1. Navigate to the Slack App Management UI in your browser.
3. Choose From an app manifest.
4. Choose the workspace for which you want to authorize AppFabric.
5. In the Enter app manifest below box, choose JSON and replace the existing JSON with the following. Replace <region> with the appropriate AWS Region (for example, us-east-1).

```json
{
   "display_information": {
      "name": "AppFabric"
   }
}
```
6. Copy and save the client ID and client secret from the **Basic Information** page.

7. For the `auditLogs:read` scope, you must enable public distribution of your app. For more information, see [Enabling public distribution](#) on the Slack website.

### From scratch

1. Choose **From scratch** on the **Create an app** screen.
2. Name your app and choose a workspace.
3. Copy and save the client ID and client secret from the **Basic Information** page.
4. On the **OAuth & Permissions** page, opt in to the **Advanced token security via token rotation** option.
5. Add a URL with the following format in the **Redirect URLs** section of the **OAuth & Permissions** page.

   ```
   https://<region>.console.aws.amazon.com/appfabric/oauth2
   ```

   In this URL, `<region>` is the code for the AWS Region in which you've configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is `https://us-east-1.console.aws.amazon.com/appfabric/oauth2`.

6. For the `auditLogs:read` scope, you must enable public distribution of your app. For more information, see [Enabling public distribution](#) on the Slack website.

### Required scopes

**Note**

This section is only applicable if you chose to create the OAuth app from scratch. Skip this section if you chose to use app manifest to create an application authorization.

You must add the following user token scopes on the **OAuth & Permissions** page of your Slack OAuth application:

- `auditlogs:read`
- `users:read.email`
- `users:read`
App authorizations

Tenant ID

AppFabric will request your tenant ID. The tenant ID in AppFabric is your Slack workspace ID. To get your tenant ID, following the instructions in Locate your Slack URL in the Slack Help Center on the Slack website. Your Slack workspace URL has a format similar to examplecorp.slack.com or examplecorp.enterprise.slack.com. The tenant ID you need is examplecorp without .slack.com or .enterprise.slack.com.

Tenant name

Enter a name that identifies your Slack workspace ID. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization

Client ID

AppFabric will request the client ID from your Slack OAuth application. To find the client ID, use the following steps:

1. Navigate to the Slack App Management UI in your browser.
2. Choose the OAuth application that you use with AppFabric.
3. Enter the client ID from the Basic Information page into the Client ID field in AppFabric.

Client secret

AppFabric will request the client secret from your Slack OAuth application. To find the client secret, use the following steps:

1. Navigate to the Slack App Management UI in your browser.
2. Choose your the OAuth application that you use with AppFabric.
3. Enter the client secret from the Basic Information page into the Client secret field in AppFabric.

Approve authorization

After creating the app authorization in AppFabric, you will receive a pop-up window from Slack to approve the authorization. To approve the AppFabric authorization, choose allow.

Smartsheet

Smartsheet is a work management platform that helps you align work, people, and technology across your enterprise. Smartsheet offers a robust set of enterprise-grade capabilities to empower everyone to manage projects, automate workflows, and rapidly build solutions at scale, creating an environment for innovation while maintaining security and compliance.

You can use AWS AppFabric to receive audit logs and user data from Smartsheet, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

Topics

• AppFabric support for Smartsheet (p. 41)
AppFabric support for Smartsheet

AppFabric supports receiving user information and audit logs from Smartsheet.

Prerequisites

To use AppFabric to transfer audit logs from Smartsheet to supported destinations, you must meet the following requirements:

- You must have a Smartsheet Business, Enterprise, or Advance account. For more information about creating or upgrading your Smartsheet account, see either Smartsheet pricing or Smartsheet Advance on the Smartsheet website.
- You must complete the Smartsheet developer registration process.

Rate limit considerations

Smartsheet imposes rate limits on the Smartsheet API. For more information about the Smartsheet API rate limits, see Rate limiting in the Smartsheet API Reference on the Smartsheet website.

Data delay considerations

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact AWS Support.

Connecting AppFabric to your Smartsheet account

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Smartsheet. To find the information required to authorize Smartsheet with AppFabric, use the following steps.

Create an OAuth application

AppFabric integrates with Smartsheet using OAuth. To create an OAuth application in Smartsheet, use the following steps:

1. Navigate to the developer tools in your Smartsheet account.
2. Choose Create New App from the developer tools screen.
3. Complete all of the input fields on the Create New App screen.
4. Use any unique value for App URL and App Contact/support.
5. Use a redirect URL with the following format as the App redirect URL.

   https://<region>.console.aws.amazon.com/appfabric/oauth2

In this URL, <region> is the code for the AWS Region in which you've configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.
6. Choose **Save**.
7. Copy and save the app client ID and app secret.

### Required scopes

Smartsheet does not require you to explicitly add scopes to your OAuth configuration. AppFabric will request the following scopes in the authorization request to your Smartsheet account:

- READ_EVENTS
- READ_USERS

### App authorizations

#### Tenant ID

AppFabric will request your tenant ID. The tenant ID in AppFabric is your Smartsheet account ID.

#### Tenant name

AppFabric will request your tenant ID. Enter any value that uniquely identifies your Smartsheet account.

#### Client ID

AppFabric will request your client ID. The client ID in AppFabric is your Smartsheet app client ID. To find your app client ID in Smartsheet, use the following steps:

1. Navigate to the developer tools in your Smartsheet account.
2. Select the OAuth application that you use to connect with AppFabric.
3. Enter the app client ID from the **App Profile** screen into the **Client ID** field in AppFabric.

#### Client secret

AppFabric will request your client secret. The client secret in AppFabric is your Smartsheet app secret. To find your app secret in Smartsheet, use the following steps:

1. Navigate to the developer tools in your Smartsheet account.
2. Select the OAuth application that you use to connect with AppFabric.
3. Enter the app secret from the **App Profile** screen into **Client Secret** field in AppFabric.

#### Approve authorization

After creating the app authorization in AppFabric, you will receive a pop-up window from Smartsheet to approve the authorization. To approve the AppFabric authorization, choose **Allow**.
**About Webex by Cisco**

Webex is a leading provider of cloud-based collaboration solutions which includes video meetings, calling, messaging, events, customer experience solutions like contact center and purpose-built collaboration devices. Webex’s focus on delivering inclusive collaboration experiences fuels innovation, which leverages AI and Machine Learning, to remove the barriers of geography, language, personality, and familiarity with technology. Its solutions are underpinned with security and privacy by design. Webex works with the world’s leading business and productivity apps – delivered through a single application and interface. Learn more at [webex.com](http://webex.com).

You can use AWS AppFabric to receive audit logs and user data from Webex, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

**Topics**
- [AppFabric support for Webex](#) (p. 43)
- [Connecting AppFabric to your Webex account](#) (p. 43)

**AppFabric support for Webex**

AppFabric supports receiving user information and audit logs from Webex.

**Prerequisites**

To use AppFabric to transfer audit logs from Webex to supported destinations, you must meet the following requirements:

- You must have a Collaboration Flex plan, Meet Plan, Call Plan, or higher. For more information about creating or upgrading to the applicable Webex plan type, see [Webex pricing for all features](http://webex.com) on the Webex website.
- Your account must have the **Pro Pack** license to access Security Audit Events provided by one of the Cisco AuditLog APIs.
- You must have a user with the **Organizational Administrator** > **Full Administrator** role.
- The **Administrator Roles** configuration for your **Full Administrator** must have the **Compliance Officer** option enabled.

**Rate limit considerations**

Webex imposes rate limits on the Webex API. For more information about the Webex API rate limits, see [Rate limits](http://webex.com) in the Webex Developers Guide on the Webex website. If the combination of AppFabric and your existing Webex API applications exceed the limit, audit logs appearing in AppFabric might be delayed.

**Data delay considerations**

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact [AWS Support](http://aws.amazon.com/support).

**Connecting AppFabric to your Webex account**

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Webex. To find the information required to authorize Webex with AppFabric, use the following steps.
Create an OAuth application

AppFabric integrates with Webex using OAuth. To create an OAuth application in Webex, use the following steps:

1. Follow the instructions in the Registering your Integration section in the Integrations & Authorization page of the Webex Developers Guide.
2. Use a redirect URL with the following format.
   
   https://<region>.console.aws.amazon.com/appfabric/oauth2

   In this URL, <region> is the code for the AWS Region in which you've configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is us-east-1. For that Region, the redirect URL is https://us-east-1.console.aws.amazon.com/appfabric/oauth2.

Required scopes

You must add the following scopes to your Webex OAuth application:

- spark-compliance:events_read
- audit:events_read
- spark-admin:people_read

App authorizations

Tenant ID

AppFabric will request your tenant ID. The tenant ID in AppFabric is your Webex organization ID. For information about how to find your Webex organization ID, see Look Up Your Organization ID in CiscoWebex Control Hub on the Webex Help Center website.

Tenant name

Enter a name that identifies this unique Webex instance. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

Client ID

AppFabric will request your Webex client ID. To find your Webex client ID, use the following steps:

2. Choose your avatar at the top right.
3. Choose My Webex Apps.
4. Choose the OAuth2 application that you use for AppFabric.
5. Enter the client ID on this page into the Client ID field in AppFabric.

Client secret

AppFabric will request your Webex client secret. Webex only presents your client secret once when you initially create your OAuth application. To generate a new client secret if you didn't save the initial client secret, use the following steps:
2. Choose your avatar at the top right.
3. Choose My Webex Apps.
4. Choose the OAuth2 application that you use for AppFabric.
5. On this page, generate a new client secret.
6. Enter the new client secret into the Client secret field in AppFabric.

**Approve authorization**

After creating the app authorization in AppFabric you will receive a pop-up window from Webex to approve the authorization. To approve the AppFabric authorization, choose accept.

---

**Zendesk**

Zendesk started the customer experience revolution in 2007 by enabling any business around the world to take their customer service online. Today, Zendesk is the champion of great service everywhere for everyone, and powers billions of conversations, connecting more than 100,000 brands with hundreds of millions of customers over telephony, chat, email, messaging, social channels, communities, review sites, and help centers. Zendesk products are built with love to be loved. The company was conceived in Copenhagen, Denmark, built and grown in California, and today employs more than 6,000 people across the world.

You can use AWS AppFabric to receive audit logs and user data from Zendesk, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

**Topics**

- AppFabric support for Zendesk (p. 45)
- Connecting AppFabric to your Zendesk account (p. 46)

**AppFabric support for Zendesk**

AppFabric supports receiving user information and audit logs from Zendesk.

**Prerequisites**

To use AppFabric to transfer audit logs from Zendesk to supported destinations, you must meet these requirements:

- You must have a Zendesk Suite Enterprise or Enterprise Plus account or a Zendesk Support Enterprise account. For more information about creating or upgrading to a Zendesk Enterprise account, see Checking your plan type Zendesk on the Zendesk website.
- You must have a user with the Administrator role in your Zendesk account. For more information about roles, see Understanding Zendesk Support user roles on the Zendesk website.

**Rate limit considerations**

Zendesk imposes rate limits on the Zendesk API. For more information about the Zendesk API rate limits, see Rate limits in the Zendesk Developers Guide on the Zendesk website. If the combination of AppFabric
and your existing Zendesk API applications exceed the limit, audit logs appearing in AppFabric might be delayed.

**Data delay considerations**

You might see up to 30-minute delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss. However, this might be customizable at an account-level. For assistance, contact [AWS Support](https://aws.amazon.com/support/).

**Connecting AppFabric to your Zendesk account**

After you create your app bundle within the AppFabric service, you must authorize AppFabric with Zendesk. To find the information required to authorize Zendesk with AppFabric, use the following steps.

**Create an OAuth application**

AppFabric integrates with Zendesk using OAuth. In Zendesk, you must create an OAuth application with the following settings:

1. Follow the instructions in the [Registering your application with Zendesk](https://support.zendesk.com/hc/en-us/articles/360055590934) section of the [Using OAuth authentication with your application](https://support.zendesk.com/hc/en-us/articles/360055603150) article on the Zendesk Support website.
2. Use a redirect URL with the following format: 

   ```
   https://<region>.console.aws.amazon.com/appfabric/oauth2
   ```

   In this URL, `<region>` is the code for the AWS Region in which you've configured your AppFabric app bundle. For example, the code for the US East (N. Virginia) Region is `us-east-1`. For that Region, the redirect URL is `https://us-east-1.console.aws.amazon.com/appfabric/oauth2`.

**App authorizations**

**Tenant ID**

AppFabric will request your Tenant ID. The Tenant ID in AppFabric is your Zendesk subdomain. For more information about finding your Zendesk subdomain, see [Where can I find my Zendesk subdomain](https://support.zendesk.com/hc/en-us/articles/360055629812) on the Zendesk Support website.

**Tenant name**

Enter a name that identifies this unique Zendesk organization. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

**Client ID**

AppFabric will request a client ID. The client ID in AppFabric is your Zendesk API unique identifier. To find your Zendesk unique identifier, use the following steps:

1. Navigate to the [Admin Center](https://admin.zendesk.com) in your Zendesk account.
2. Choose Apps and integrations.
3. Choose APIs, Zendesk APIs.
4. Choose the OAuth Clients tab.
5. Choose the OAuth application that you created for AppFabric.
6. Enter the unique identifier for your OAuth client into the **Client ID** field in AppFabric.

**Client secret**

AppFabric will request a client secret. The client secret in AppFabric is your Zendesk secret token. Zendesk presents your secret token only once when you first create your Zendesk OAuth application. To generate a new secret token if you didn’t save the initial secret token, use the following steps:

1. Navigate to the **Admin Center** in your Zendesk account.
2. Choose **Apps and integrations**.
3. Choose **APIs, Zendesk APIs**.
4. Choose the **OAuth Clients** tab.
5. Choose the OAuth application that you created for AppFabric.
6. Choose the **Regenerate** button next to the **Secret token** field.
7. Enter the new secret token into the **Client secret** field in AppFabric.

**Approve authorization**

After creating the app authorization in AppFabric, you will receive a pop-up window from Zendesk to approve the authorization. To approve the AppFabric authorization, choose **Allow**.

**Zoom**

Zoom is an all-in-one intelligent collaboration platform that makes connecting easier, more immersive, and more dynamic for businesses and individuals. Zoom technology puts people at the center, enabling meaningful connections, facilitating modern collaboration, and driving human innovation through solutions like team chat, phone, meetings, omnichannel cloud contact center, smart recordings, whiteboard, and more, in one offering.

You can use AWS AppFabric to receive audit logs and user data from Zoom, normalize the data into Open Cybersecurity Schema Framework (OCSF) format, and output the data to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose stream.

**Topics**

- AppFabric support for Zoom (p. 47)
- Connecting AppFabric to your Zoom account (p. 48)

**AppFabric support for Zoom**

AppFabric supports receiving user information and audit logs from Zoom.

**Prerequisites**

To use AppFabric to transfer audit logs from Zoom to supported destinations, you must meet the following requirements:

- You must have a Zoom Pro, Business, Education, or Enterprise plan.
- Your Zoom **Admin** role must have permission to create server-to-server OAuth applications. For information about enabling server-to-server OAuth applications, see the Enable permissions section of the Server-to-Server OAuth page in the Zoom Developers Guide on the Zoom website.
Your Zoom Admin role must have permission to view admin activity logs and sign in/sign out audit activity. For more information about enabling permission to view audit activity, see Using role management and Using Admin Activity Logs on the Zoom Support website.

Rate limit considerations
Zoom imposes rate limits on the Zoom API. For more information about Zoom API rate limits, see Rate limits in the Zoom Developers Guide. If the combination of AppFabric and your existing Zoom applications exceed the limit, audit logs appearing in AppFabric might be delayed.

Data delay considerations
You might see an approximately 24-hour delay for an audit event to be delivered to your destination. This is due to delay in audit events made available by the application as well as due to precautions taken to reduce data loss.

Connecting AppFabric to your Zoom account
After you create your app bundle within the AppFabric service, then you must authorize AppFabric with Zoom. To find the information required to authorize Zoom with AppFabric, use the following steps.

Create a server-to-server OAuth application
AppFabric uses server-to-server OAuth with app credentials to integrate with Zoom. To create a server-to-server OAuth application in Zoom, follow the instructions in Create a Server-to-Server OAuth app in the Zoom Developers Guide. AppFabric does not support Zoom webhooks, and you can skip the section for adding webhook subscriptions.

Required scopes
You must add the following scopes to your Zoom server-to-server OAuth application:

- user:read:admin
- user:write:admin
- account:read:admin
- account:write:admin
- group:read:admin
- group:write:admin
- report:read:admin
- role:read:admin
- role:write:admin

App authorizations
Tenant ID
AppFabric will request your tenant ID. The tenant ID in AppFabric is the Zoom account ID. To find your Zoom account ID, use the following steps:

1. Navigate to the Zoom marketplace.
2. Choose Manage.
3. Choose the server-to-server OAuth application that you use for AppFabric.
4. Enter the account ID from the **App Credentials** page into the **Tenant ID** field in AppFabric.

**Tenant name**

Enter a name that identifies this unique Zoom organization. AppFabric uses the tenant name to label the app authorizations and any ingestions created from the app authorization.

**Client ID**

AppFabric will request your client ID. To find your Zoom client ID, use the following steps:

1. Navigate to the Zoom marketplace.
2. Choose **Manage**.
3. Choose the server-to-server OAuth application that you use for AppFabric.
4. Enter the client ID from the **App Credentials** page into the **Client ID** field in AppFabric.

**Client secret**

AppFabric will request your client secret. To find your Zoom client secret, use the following steps:

1. Navigate to the Zoom marketplace.
2. Choose **Manage**.
3. Choose the server-to-server OAuth application that you use for AppFabric.
4. Enter the client secret from the **App Credentials** page into the **Client secret** field in AppFabric.

**Audit log delivery**

Zoom makes audit logs available by accessing the API every 24 hours. When viewing audit logs with AppFabric, the data that you see for Zoom is for the previous day’s activities.
Compatible security tools and services

AWS AppFabric supports integration with the following security tools and services. Choose the name of a service for more information about how to set up AppFabric to connect to it.

Topics
- Amazon QuickSight (p. 50)
- Amazon Security Lake (p. 51)
- Logz.io (p. 55)
- Netskope (p. 56)
- NetWitness (p. 56)
- Rapid7 (p. 57)
- Splunk (p. 58)

Amazon QuickSight

Amazon QuickSight powers data-driven organizations with unified business intelligence (BI) at hyperscale. With QuickSight, all users can meet varying analytic needs from the same source of truth through modern interactive dashboards, paginated reports, embedded analytics, and natural language queries. You can analyze AWS AppFabric audit log data in QuickSight, by choosing your Amazon Simple Storage Service (Amazon S3) bucket where your AppFabric logs are stored as your source.

AppFabric audit log ingestion considerations

The following sections describe the AppFabric output schema, output formats, and output destinations to use with Amazon QuickSight.

Schema and formats

QuickSight supports the following AppFabric output schema and formats:
- Raw - JSON
  - AppFabric outputs data in the original schema used by the source application in the JSON format.
- OCSF - JSON
  - AppFabric normalizes the data using the Open Cybersecurity Schema Framework (OCSF) and outputs the data in the JSON format.

Output locations

QuickSight supports the following AppFabric output locations:
• Amazon S3
  • You can ingest data from Amazon S3 directly into QuickSight by Creating a dataset using Amazon S3 files. To verify that your target file set doesn't exceed QuickSight data source quotas, see Data source quotas in the Amazon QuickSight User Guide.
  • If your file set exceeds QuickSight quotas for an Amazon S3 data source, you can ingest your data in Amazon S3 using Amazon Athena and AWS Glue tables. Using Athena in your QuickSight dataset will incur additional costs. For more information about Athena pricing, see the Athena pricing page.

To use Athena:
1. Follow the instructions in Using AWS Glue to connect to data sources in Amazon S3 in the Athena User Guide.
2. Follow the instructions in Creating a dataset using Athena data in the Amazon QuickSight User Guide.

Amazon Security Lake

Amazon Security Lake automatically centralizes security data from AWS environments, software as a service (SaaS) providers, on premises and cloud sources into a purpose-built data lake stored in your AWS account. With Security Lake, you can get a more complete understanding of your security data across your entire organization. Security Lake has adopted the Open Cybersecurity Schema Framework (OCSF), an open source security event schema. With OCSF support, the service normalizes and combines security data from AWS and a broad range of enterprise security data sources.

AppFabric audit log ingestion considerations

You can get your SaaS audit logs into Amazon Security Lake in your AWS account by adding a custom source to Security Lake. The following sections describe the AppFabric output schema, output format, and output destinations to use with Security Lake.

Schema and format

Security Lake supports the following AppFabric output schema and format:

• OCSF - JSON
  • AppFabric normalizes the data using the Open Cybersecurity Schema Framework (OCSF) and outputs the data in JSON format.

Output locations

Security Lake supports AppFabric as a custom source using an Amazon Kinesis Data Firehose delivery stream as the AppFabric ingestion output location. To configure the AWS Glue table and Kinesis Data Firehose delivery stream, and to set up a custom source in Security Lake, use the following procedures.

Create an AWS Glue table

1. Navigate to Amazon Simple Storage Service (Amazon S3) and create a bucket with a name of your choice.
2. Navigate to the AWS Glue console.
3. For Data Catalog, go to the Tables section, and choose Add Table.
4. Enter a name of your choice for this table.
5. Select the Amazon S3 bucket that you created in step 1.
6. For the data format, select JSON, and choose Next.
7. On the Choose or define schema page, choose Edit schema as JSON.
8. Enter the following schema, and complete the AWS Glue table creation process.

```json
[
  {
    "Name": "activity_id",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "activity_name",
    "Type": "string",
    "Comment": ""
  },
  {
    "Name": "actor",
    "Type":
    "struct<session:struct<created_time:bigint,uid:string,issuer:string>,user:struct<uid:string,email_addr:string,credential_uid:string,name:string,type:string>>",
    "Comment": ""
  },
  {
    "Name": "user",
    "Type":
    "struct<uid:string,email_addr:string,credential_uid:string,name:string,type:string>"
  },
  {
    "Name": "group",
    "Type":
    "struct<uid:string,desc:string,name:string,type:string,privileges:array<string>>"
  },
  {
    "Name": "privileges",
    "Type": "array<string>",
    "Comment": ""
  },
  {
    "Name": "web_resources",
    "Type":
    "array<struct<type:string,uid:string,name:string,data:struct<current_value:string,previous_value:string>>>"
  },
  {
    "Name": "http_request",
    "Type": "struct<http_method:string,user_agent:string,url:string>"
  },
  {
    "Name": "auth_protocol"
  },
  {
    "Name": "auth_protocol_id",
    "Type": "int"
  },
  {
    "Name": "category_name",
    "Type": "string"
  }
],
```
```json
{
    "Name": "category_uid",
    "Type": "string",
    "Comment": ""
},
{
    "Name": "class_name",
    "Type": "string",
    "Comment": ""
},
{
    "Name": "class_uid",
    "Type": "string",
    "Comment": ""
},
{
    "Name": "is_mfa",
    "Type": "boolean",
    "Comment": ""
},
{
    "Name": "raw_data",
    "Type": "string",
    "Comment": ""
},
{
    "Name": "severity",
    "Type": "string",
    "Comment": ""
},
{
    "Name": "severity_id",
    "Type": "int",
    "Comment": ""
},
{
    "Name": "status",
    "Type": "string",
    "Comment": ""
},
{
    "Name": "status_detail",
    "Type": "string",
    "Comment": ""
},
{
    "Name": "status_id",
    "Type": "int",
    "Comment": ""
},
{
    "Name": "time",
    "Type": "bigint",
    "Comment": ""
},
{
    "Name": "type_name",
    "Type": "string",
    "Comment": ""
},
{
    "Name": "type_uid",
    "Type": "string",
    "Comment": ""
}
```
Create a custom source in Security Lake

1. Navigate to the Amazon Security Lake console.
2. Select Custom sources in the navigation pane.
3. Choose Create custom source.
4. Enter a name for your custom source and select an applicable OCSF event class.
   
   **Note**
   
5. For both AWS account ID and External ID, enter your AWS account ID. Then, choose Create.
6. Save the Amazon S3 location of the custom source. You will use it to set up an Amazon Kinesis Data Firehose delivery stream.

Create a delivery stream in Kinesis Data Firehose

1. Navigate to the Amazon Kinesis Data Firehose console.
2. Choose Create a delivery stream.
3. For Source, select Direct PUT.
4. For Destination, choose S3.
5. In the Transform and convert records section, choose Enable record format conversion and choose Apache Parquet as the output format.
6. For AWS Glue table, choose the AWS Glue table that you created in the previous procedure, and choose the latest version.
7. For Destination settings, choose the Amazon S3 bucket that you created with the Security Lake custom source.
8. For Dynamic Partitioning, choose Enabled.
9. For Inline parsing for JSON, choose Enabled.
   
   - For Keyname, enter eventDayValue.
   - For JQ Expression, enter (.time/1000)|strftime("%Y%m%d").
10. For the S3 bucket prefix, enter the following value.
Replace `<region>` and `<account_id>` with your AWS Region and AWS account ID.

11. For the **S3 bucket error output prefix**, enter the following value.

```
ext/AppFabric/error/
```

12. For the **Retry duration**, select **300**.
13. For the **Buffer size**, select **128 MiB**.
14. For the **Buffer interval**, select **60s**.
15. Complete the creation process for the Kinesis Data Firehose delivery stream.

---

### Create AppFabric ingestions

To send data to Amazon Security Lake, you must create an ingestion in the AppFabric console that uses the Kinesis Data Firehose delivery stream that you created earlier as the output location. For more information about configuring AppFabric ingestions to use Kinesis Data Firehose as an output location, see the [Create an output location](#) (p. 8).

---

### Logz.io

Logz.io helps cloud native businesses monitor and secure their environments via the Logz.io Open 360 Platform – transforming observability and security from a high-cost, low-value burden into a high-value, cost-efficient enabler of better business outcomes.

Logz.io Cloud SIEM directly addresses today's leading security challenges – from data overload to the omnipresent cyber skills gap – via fast querying, multidimensional detection and deep customizable security content to help monitor and investigate across the full-expanse of your cloud environment – with no performance degradation, regardless of data volumes.

The Logz.io solution was purpose-built to deliver advanced threat analysis and investigation with less complexity and cost. Customers are backed by dedicated security analysts, threat content as a service and AI-backed capabilities purpose-built to help reduce noisy data and focus on the information that enables your team to rapidly prioritize real world threats.

### AWS AppFabric audit log ingestion considerations

The following sections describe the AppFabric output schema, output formats, and output destinations to use with Logz.io.

#### Schema and format

Logz.io supports the following AppFabric output schema and formats:

- **Raw - JSON**
  - AppFabric outputs data in the original schema used by the source application in the JSON format.
- **OCSF - JSON**
  - AppFabric normalizes the data using the Open Cybersecurity Schema Framework (OCSF) and outputs the data in the JSON format.
Output locations

Logz.io supports the following AppFabric output locations:

- Amazon Kinesis Data Firehose
  - To configure your Kinesis Data Firehose delivery stream so that it sends data to Logz.io, follow the instructions in [Choose Logz.io for Your Destination](https://docs.aws.amazon.com/kinesis/data-frehose/index.html#configure) in the Amazon Kinesis Data Firehose Developer Guide.
- Amazon Simple Storage Service (Amazon S3)
  - To configure Logz.io to receive data from the Amazon S3 bucket that contains your audit logs, follow the instructions in [Configure an Amazon S3 bucket](https://docs.logz.io/guides/monitoring/configure Amazon S3) on the Logz.io website.

Netskope

Netskope, a global cybersecurity leader, is redefining cloud, data, and network security to help organizations apply zero trust principles to protect data. Fast and easy to use, the Netskope platform provides optimized access and zero trust security for people, devices, and data anywhere they go. Netskope helps customers reduce risk, accelerate performance, and get unrivaled visibility into any cloud, web, and private application activity. Thousands of customers, including more than 25 of the Fortune 100, trust Netskope and its powerful NewEdge network to address evolving threats, new risks, technology shifts, organizational and network changes, and new regulatory requirements. Learn how Netskope helps customers be ready for anything on their SASE journey, visit [netskope.com](https://www.netskope.com).

AWS AppFabric audit log ingestion considerations

The following sections describe the AppFabric output schema, output formats, and output destinations to use with Netskope.

Schema and format

Netskope supports the following AppFabric output schema and formats:

- Raw - JSON
  - AppFabric outputs data in the original schema used by the source application in the JSON format.
- OCSF - JSON
  - AppFabric normalizes the data using the Open Cybersecurity Schema Framework (OCSF) and outputs the data in the JSON format.

Output locations

Netskope supports the following AppFabric output location:

- Amazon Simple Storage Service (Amazon S3)
  - To configure Netskope to receive data from the Amazon S3 bucket that contains your audit logs, follow the instructions in [Data Protection for Amazon Web Services S3](https://docs.aws.netskope.com/) on the Netskope website.

NetWitness

NetWitness is a leading developer of extended detection and response (XDR) software. Their global base of highly security-conscious customers relies on NetWitness XDR to defend against sophisticated and
aggressive adversaries. With the industry’s most complete, integrated, and mature platform to detect, investigate, and respond to digital attacks, NetWitness XDR is the unifying foundation of a modern and effective SOC.

Due to its highly modular architecture, NetWitness XDR detects threats wherever they occur — in the cloud, on-premises, with mobile and remote workers, or anywhere in between. The NetWitness Platform XDR delivers complete visibility combined with applied threat intelligence and user behavior analytics to detect threats, prioritize activities, investigate, and automate response. All this empowers security analysts with better, faster efficiency to keep security operations well ahead of business-impacting threats.

AWS AppFabric audit log ingestion considerations

The following sections describe the AppFabric output schema, output formats, and output destinations to use with NetWitness.

Schema and format

NetWitness supports the following AppFabric output schema and formats:

- Raw - JSON
  - AppFabric outputs data in the original schema used by the source application in the JSON format.
- OCSF - JSON
  - AppFabric normalizes the data using the Open Cybersecurity Schema Framework (OCSF) and outputs the data in the JSON format.

Output locations

NetWitness supports the following AppFabric output location:

- Amazon Simple Storage Service (Amazon S3)
  - To configure NetWitness to receive data from the Amazon S3 bucket that contains your audit logs, follow the instructions in S3 Universal Connector Event Source Log Configuration Guide on the NetWitness Platform Integrations page on the NetWitness website.

Rapid7

Rapid7, Inc. is on a mission to create a safer digital world by making cybersecurity simpler and more accessible. Rapid7 empowers security professionals to manage a modern attack surface through best-in-class technology, leading-edge research, and broad, strategic expertise. Rapid7’s comprehensive security solutions help more than 10,000 global customers unite cloud risk management and threat detection to reduce attack surfaces and eliminate threats with speed and precision.

AWS AppFabric audit log ingestion considerations

The following sections describe the AppFabric output schema, output format, and output destinations to use with Rapid7.

Schema and format

Rapid7 supports the following AppFabric output schema and formats:
Output locations

Rapid7 supports the following AppFabric output location:

- Amazon Simple Storage Service (Amazon S3)
  - To configure Rapid7 to receive data from the Amazon S3 bucket that contains your audit logs, follow the instructions in the [How to Monitor Your Amazon S3 Activity with InsightIDR](https://www.rapid7.com/blog) blog post on the Rapid7 Blog website.

Splunk

Splunk helps make organizations more resilient. Leading organizations use Splunk's unified security and observability platform to keep their digital systems secure and reliable. Organizations trust Splunk to prevent security, infrastructure, and application issues from becoming major incidents, absorb shocks from digital disruptions and accelerate digital transformation.

AWS AppFabric audit log ingestion considerations

The following sections describe the AppFabric output schema, output formats, and output destinations to use with Splunk.

Schema and format

Splunk supports the following AppFabric output schema and formats:

- Raw - JSON
  - AppFabric outputs data in the original schema used by the source application in the JSON format.
- OCSF - JSON
  - AppFabric normalizes the data using the Open Cybersecurity Schema Framework (OCSF) and outputs the data in the JSON format.
- OCSF - Parquet
  - AppFabric normalizes the data using the Open Cybersecurity Schema Framework (OCSF) and outputs the data in the Apache Parquet format.

Output locations

Splunk supports the following AppFabric output locations:

- Amazon Kinesis Data Firehose
  - To configure Splunk to receive audit logs from the Kinesis Data Firehose stream that contains your audit logs, follow the instructions in [Splunk Add-on for Amazon Kinesis Data Firehose](https://www.splunk.com) on the Splunk website.
- Amazon Simple Storage Service (Amazon S3)
• To configure Splunk to receive data from the Amazon S3 bucket that contains your audit logs, follow the instructions in Configure SQS-based S3 inputs for the Splunk Add-on for AWS on the Splunk website.
Delete AWS AppFabric resources

If you don’t want to continue using AWS AppFabric, be sure to delete the data in the output locations you created during setup and your AppFabric resources to avoid incurring additional charges. To clean up your AppFabric resources, you must delete the resources in the reverse order in which you created them for each software as a service (SaaS) application: Ingestion destinations > Ingestions > App authorization > App bundles

After you’ve deleted your final app authorization, you can delete the app bundle.

Topics
• Delete an ingestion destination (p. 60)
• Delete an ingestion (p. 60)
• Delete an app authorization (p. 60)
• Delete an app bundle (p. 61)

Delete an ingestion destination

If you select an output location when you create an ingestion, AppFabric creates ingestion destinations on your behalf. To delete an ingestion destination, use the following steps:

2. From the Getting started page, expand the menu on the left.
3. Choose Ingestions.
4. Choose an app authorization.
5. Select the option button next to the destination that you want to delete and choose Delete.
6. Choose Delete on the delete destination dialog box to confirm.
7. Repeat the above steps for all of your destinations.

Delete an ingestion

To delete an ingestion, use the following steps:

1. From the Getting started page, expand the menu on the left.
2. Choose Ingestions.
3. Select the option button that is next to your app authorization.
4. Choose the Actions dropdown menu.
5. Choose Delete.
6. Choose Delete on the delete ingestion dialog box to confirm.

Delete an app authorization

To delete an app authorization, use the following steps:

1. From the Getting started page, expand the menu on the left.
2. Choose **App authorizations**.
3. Select the option button next to the app authorization that you want to delete.
4. Choose the **Actions** dropdown menu.
5. Choose **Delete**.
6. Choose **Delete** on the delete ingestion dialog box to confirm.

Delete an app bundle

To delete your app bundle, use the following steps:

1. From the **Getting started** page, expand the menu on the left.
2. Choose **App bundle**.
3. Choose the **Delete** button.
4. Type **delete** to confirm, and then choose **Delete**.
Security in AWS AppFabric

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

**Topics**

- Data protection in AWS AppFabric (p. 62)
- Identity and access management for AWS AppFabric (p. 67)
- Compliance validation for AWS AppFabric (p. 85)
- Security best practices for AWS AppFabric (p. 85)
- Resilience in AWS AppFabric (p. 86)
- Infrastructure security in AWS AppFabric (p. 86)
- Configuration and vulnerability analysis in AWS AppFabric (p. 86)

Data protection in AWS AppFabric

The AWS shared responsibility model applies to data protection in AWS AppFabric. 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. This content includes 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.
- 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 140-2 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) 140-2.

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 AppFabric 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.

Encryption at rest

AWS AppFabric supports encryption at rest, a server-side encryption feature in which AppFabric transparently encrypts all data related to your app bundles when it is persisted to disk, and decrypts them when you access the data. By default, AppFabric encrypts your data using an AWS owned key from AWS Key Management Service (AWS KMS). You can also choose to encrypt your data using your own customer managed key from AWS KMS.

When you delete an app bundle, all its metadata is permanently deleted.

Encryption in transit

When you configure an app bundle, you can choose either an AWS owned key or a customer managed key. When collecting and normalizing the data for an audit log ingestion, AppFabric stores data temporarily in an intermediate Amazon Simple Storage Service (Amazon S3) bucket and encrypts it using this key. This intermediate bucket is deleted after 30 days, using a bucket lifecycle policy.

AppFabric secures all data in transit using TLS 1.2 and signs API requests for AWS services with AWS Signature V4.

Key management

AppFabric supports encrypting data with an AWS owned key or a customer managed key. We recommend that you use a customer managed key because it puts you in full control over your encrypted data. When you choose a customer managed key, AppFabric attaches a resource policy to the customer managed key that grants it access to the customer managed key.

Customer managed key

To create a customer managed key, follow the steps for Creating symmetric encryption KMS keys in the AWS KMS Developer Guide.

Key policy

Key policies control access to your customer managed keys. Every customer managed key must have exactly one key policy, which contains statements that determine who can use the key and how they can use it. When you create your customer managed key, you can specify a key policy. For information about creating a key policy, see Creating a key policy in the AWS KMS Developer Guide.

To use a customer managed key with AppFabric, the AWS Identity and Access Management (IAM) user or role creating your AppFabric resources must have permission to use your customer managed key. We
recommend that you create a key that you use only with AppFabric and add your AppFabric users as users of the key. This approach limits the scope of access to your data. The permissions your users require are as follows:

- `kms:DescribeKey`
- `kms:CreateGrant`
- `kms:GenerateDataKey`
- `kms:Decrypt`

The AWS KMS console guides you through creating a key with the appropriate key policy. For more information about key policies, see Key policies in AWS KMS in the AWS KMS Developer Guide.

Following is an example key policy that permits:

- The AWS account root user full control of the key.
- Users permitted to use AppFabric to use your customer managed key with AppFabric.
- A key policy for an app bundle setup in us-east-1.

```json
{
    "Id": "key-consolepolicy-3",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow access for key administrators",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
            "Action": ["kms:*"],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key_ID"
        },
        {
            "Sid": "Allow read-only access to key metadata to the account",
            "Effect": "Allow",
            "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
            "Action": ["kms:Describe*", "kms:Get*", "kms:List*", "kms:RevokeGrant"],
            "Resource": "*"
        },
        {
            "Sid": "Allow access to principals authorized to use AWS AppFabric",
            "Effect": "Allow",
            "Principal": {"AWS": "iam-role/user-creating-appfabric-resources"},
            "Action": ["kms:Decrypt", "kms:GenerateDataKey", "kms:DescribeKey", "kms:CreateGrant", "kms:ListAliases"],
            "Resource": "*",
            "Condition": {"StringEquals": {"kms:ViaService": "appfabric.us-east-1.amazonaws.com", "kms:CallerAccount": "111122223333"}}
        }
    ]
}
```
How AppFabric uses grants in AWS KMS

AppFabric requires a grant to use your customer managed key. For more information, see [Grants in AWS KMS](https://docs.aws.amazon.com/kms/latest/developerguide/arns.html) in the AWS KMS Developer Guide.

When you create an app bundle, AppFabric creates a grant on your behalf by sending a CreateGrant request to AWS KMS. Grants in AWS KMS are used to give AppFabric access to an AWS KMS key in a customer account. AppFabric requires that the grant to use your customer managed key for the following internal operations:

- Send **GenerateDataKey** requests to AWS KMS to generate data keys encrypted by your customer managed key.
- Send **Decrypt** requests to AWS KMS to decrypt the encrypted data keys so that they can be used to encrypt your data and to decrypt application access tokens in transit.
- Send **Encrypt** requests to AWS KMS to encrypt application access tokens in transit.

Following is an example of a grant.

```
{
    "KeyId": "arn:aws:kms:us-east-1:111122223333:key/ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "GrantId": "0ab0ac0d0b000f00ea00cc000f00000000c0bc0a0000000Sample",
    "Name": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "CreationDate": "2022-10-11T20:35:39+00:00",
    "GranteePrincipal": "appfabric.us-east-1.amazonaws.com",
    "RetiringPrincipal": "appfabric.us-east-1.amazonaws.com",
    "IssuingAccount": "arn:aws:iam::111122223333:root",
    "Operations": [
        "Decrypt",
        "Encrypt",
        "GenerateDataKey"
    ],
    "Constraints": {
        "EncryptionContextSubset": {
            "appBundleArn": "arn:aws:fabric:us-east-1:111122223333:appbundle/ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE"
        }
    }
}
```

When you delete an app bundle, AppFabric retires issued grants on your customer managed key.

Monitoring your encryption keys for AppFabric

When you use AWS KMS customer managed keys with AppFabric, you can use AWS CloudTrail logs to track requests that AppFabric sends to AWS KMS.

Following is an example of an CloudTrail event logged when AppFabric uses CreateGrant for your customer managed key.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTEXAMPLE:SampleUser",
        "arn": "arn:aws:iam::111122223333:role/appFabricRoleSample",
        "accountId": "111122223333",
        "accessKeyId": "AROAIGDTESTEXAMPLE:SampleUser",
        "sessionContext": {
            "sessionIssuer": {
                "arn": "arn:aws:iam::111122223333:root",
                "principalId": "AROAIGDTESTEXAMPLE:SampleUser",
                "accountId": "111122223333",
                "awsAccount": "111122223333",
                "sourceAccount": "111122223333",
                "sourcePrincipalId": "A86385D2C49B4ECA83B9B97ED531B905"}
        }
    }
```
Monitoring your encryption keys for AppFabric

```
"arn": "arn:aws:sts::111122223333:assumed-role/AssumedRole/SampleUser",
"accountId": "111122223333",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "AROAIGDTESTANDEXAMPLE",
    "arn": "arn:aws:iam::111122223333:role/AssumedRole",
    "accountId": "111122223333",
    "userName": "SampleUser"
  },
  "webIdFederationData": {},
  "attributes": {
    "creationDate": "2023-04-28T14:01:33Z",
    "mfaAuthenticated": "false"
  }
},
"eventTime": "2023-04-28T14:05:48Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-east-1",
"sourceIPAddress": "appfabric.amazonaws.com",
"userAgent": "appfabric.amazonaws.com",
"requestParameters": {
  "granteePrincipal": "appfabric.us-east-1.amazonaws.com",
  "constraints": {
    "encryptionContextSubset": {
      "appBundleArn": "arn:aws:appfabric:us-east-1:111122223333:appbundle/ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE"
    }
  },
  "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLEID",
  "retiringPrincipal": "appfabric.us-east-1.amazonaws.com",
  "operations": [
    "Encrypt",
    "Decrypt",
    "GenerateDataKey"
  ],
  "responseElements": {
    "grantId": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/KEY_ID"
  },
  "additionalEventData": {
    "grantId": "0ab0ac0d0b000f00ea00ccaa0e00fc00bd0ce000c000f0000000c0bc0a0000aaafSAMPLE"
  },
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "AWS Internal",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-east-1:111122223333:appbundle/ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_256_GCM_SHA384",
    "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
  }
}
```
Identity and access management for AWS AppFabric

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

Topics
- Audience (p. 67)
- Authenticating with identities (p. 67)
- Managing access using policies (p. 70)
- How AWS AppFabric works with IAM (p. 71)
- Identity-based policy examples for AWS AppFabric (p. 76)
- Using service-linked roles for AppFabric (p. 79)
- AWS managed policies for AWS AppFabric (p. 80)
- Troubleshooting AWS AppFabric identity and access (p. 83)

Audience

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

Service user – If you use the AppFabric service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more AppFabric 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 AppFabric, see Troubleshooting AWS AppFabric identity and access (p. 83).

Service administrator – If you’re in charge of AppFabric resources at your company, you probably have full access to AppFabric. It’s your job to determine which AppFabric 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 AppFabric, see How AWS AppFabric works with IAM (p. 71).

IAM administrator – If you’re an IAM administrator, you might want to learn details about how you can write policies to manage access to AppFabric. To view example AppFabric identity-based policies that you can use in IAM, see Identity-based policy examples for AWS AppFabric (p. 76).

Authenticating with identities

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 Signing AWS 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 Using multi-factor authentication (MFA) in AWS 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 When to create an IAM user (instead of a role) 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. You can temporarily assume an IAM role in the AWS Management Console by switching roles. 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 Using IAM roles 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 Creating a role for a third-party Identity Provider 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 How IAM roles differ from resource-based policies 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.

- **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, resources, and condition keys for AWS AppFabric in the Service Authorization Reference.

- **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 Creating 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 Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the IAM User Guide.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the IAM User Guide.
Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy 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 Creating IAM 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 Choosing 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 an 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.

• **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the AWS Organizations 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.

### How AWS AppFabric works with IAM

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

### IAM features you can use with AWS AppFabric

<table>
<thead>
<tr>
<th>IAM feature</th>
<th>AppFabric support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity-based policies (p. 72)</td>
<td>Yes</td>
</tr>
<tr>
<td>Resource-based policies (p. 72)</td>
<td>No</td>
</tr>
<tr>
<td>Policy actions (p. 72)</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy resources (p. 73)</td>
<td>Yes</td>
</tr>
<tr>
<td>Policy condition keys (p. 74)</td>
<td>No</td>
</tr>
<tr>
<td>ACLs (p. 74)</td>
<td>No</td>
</tr>
<tr>
<td>ABAC (tags in policies) (p. 74)</td>
<td>Yes</td>
</tr>
<tr>
<td>Temporary credentials (p. 75)</td>
<td>No</td>
</tr>
<tr>
<td>Principal permissions (p. 75)</td>
<td>Yes</td>
</tr>
<tr>
<td>Service roles (p. 75)</td>
<td>No</td>
</tr>
<tr>
<td>Service-linked roles (p. 75)</td>
<td>Yes</td>
</tr>
</tbody>
</table>
To get a high-level view of how AppFabric 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 AppFabric

| 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 Creating IAM 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 AppFabric

To view examples of AppFabric identity-based policies, see Identity-based policy examples for AWS AppFabric (p. 76).

Resource-based policies within AppFabric

| 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 How IAM roles differ from resource-based policies in the IAM User Guide.

Policy actions for AppFabric

| 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 AppFabric actions, see [Actions defined by AWS AppFabric](#) in the *Service Authorization Reference*.

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

```
appfabric
```

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

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

You can specify multiple actions using wildcard characters (*). For example, to specify all actions that begin with the word List, include the following action.

```
"Action": "appfabric:List*"
```

To view examples of AppFabric identity-based policies, see [Identity-based policy examples for AWS AppFabric](#) (p. 76).

### Policy resources for AppFabric

<table>
<thead>
<tr>
<th>Supports policy resources</th>
<th>Yes</th>
</tr>
</thead>
</table>

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 AppFabric resource types and their ARNs, see [Resource types defined by AWS AppFabric](#) in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see [Actions defined by AWS AppFabric](#).

To view examples of AppFabric identity-based policies, see [Identity-based policy examples for AWS AppFabric](#) (p. 76).
Policy condition keys for AppFabric

| 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 AppFabric condition keys, see Condition keys for AWS AppFabric in the Service Authorization Reference. To learn with which actions and resources you can use a condition key, see Actions defined by AWS AppFabric.

To view examples of AppFabric identity-based policies, see Identity-based policy examples for AWS AppFabric (p. 76).

ACLs in AppFabric

| 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 AppFabric

| Supports ABAC (tags in policies) | Yes |

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**.


### Using temporary credentials with AppFabric

| 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](https://docs.aws.amazon.com/IAM/latest/UserGuide/list_services_supportediam.html) 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 [Switching to a role (console)](https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial-switching-iam-roles.html) 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](https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorials-create-temporary-credentials.html).

### Cross-service principal permissions for AppFabric

| Supports principal permissions | Yes |

When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see [Actions, resources, and condition keys for AWS AppFabric](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference-service-context-variables.html) in the Service Authorization Reference.

### Service roles for AppFabric

| 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 [Creating a role to delegate permissions to an AWS service](https://docs.aws.amazon.com/IAM/latest/UserGuide/reference-service-context-variables.html) in the IAM User Guide.

**Warning**

Changing the permissions for a service role might break AppFabric functionality. Edit service roles only when AppFabric provides guidance to do so.

### Service-linked roles for AppFabric

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

For details about creating or managing AppFabric service-linked roles, see Using service-linked roles for AppFabric (p. 79).

Identity-based policy examples for AWS AppFabric

By default, users and roles don’t have permission to create or modify AppFabric resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. 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.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the IAM User Guide.

For details about actions and resource types defined by AppFabric, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS AppFabric in the Service Authorization Reference.

Topics
- Policy best practices (p. 76)
- Using the AppFabric console (p. 77)
- Allow access to app bundles (p. 77)
- Restrict access to app bundles (p. 77)
- Restrict deleting or stopping ingestions (p. 78)
- Allow users to view their own permissions (p. 78)

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AppFabric 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...
Identity-based policy examples

IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation 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 Configuring MFA-protected API access 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 AppFabric console**

Attach the AWSAppFabricReadOnlyAccess AWS managed policy to your IAM identities to grant them read-only permission to the AppFabric service, including the AppFabric console in the AWS Management Console. Or, you can attach the AWSAppFabricFullAccess AWS managed policy to your IAM identities to grant them full administrative permission to the AppFabric service. For more information, see AWS managed policies for AWS AppFabric (p. 80).

**Allow access to app bundles**

The following policy example grants access to app bundles in the AppFabric service.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "appfabric:StartUserAccessTasks",
        "appfabric:BatchGetUserAccessTasks"
      ],
      "Resource": ["arn:aws:appfabric:*:*:appbundle/*"]
    },
    "Version": "2012-10-17"
  ]
}
```

**Restrict access to app bundles**

The following policy example restricts access to app bundles in the AppFabric service.

```
{
  "Statement": [
    {
      "Action": ["appfabric:*"],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "appfabric:StartUserAccessTasks",
        "appfabric:BatchGetUserAccessTasks"
      ],
      "Resource": ["arn:aws:appfabric:*:*:appbundle/*"]
    }
  ],
  "Version": "2012-10-17"
}
```
**Restrict deleting or stopping ingestions**

The following policy example restricts the deletion or stopping of ingestions in the AppFabric service.

```json
{
  "Statement": [
    {
      "Action": ["appfabric:*"],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "appfabric:StopIngestion",
        "appfabric:DeleteIngestion",
        "appfabric:DeleteIngestionDestination"
      ],
      "Resource": ["arn:aws:appfabric::*:*:appbundle/**"]
    }
  ],
  "Version": "2012-10-17"
}
```

**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.

```json
{
  "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": "*"
    }
  ]
}
```
Using service-linked roles for AppFabric

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

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

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

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

Service-linked role permissions for AppFabric

AppFabric uses the service-linked role named AWSServiceRoleForAppFabric – Allows AppFabric to put data in the an ingestion destination resource, such as an Amazon S3 bucket or an Amazon Kinesis Data Firehose delivery stream. It also allows AppFabric to put CloudWatch metric data in the AWS/AppFabric namespace.

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

- `appfabric.amazonaws.com`

The role permissions policy named AWSAppFabricServiceRolePolicy allows AppFabric to complete the following actions on the specified resources:

- Action: `cloudwatch:PutMetricData` in the AWS/AppFabric namespace. This action grants permission for AppFabric to put metric data into the Amazon CloudWatch AWS/AppFabric namespace. For more information about the AppFabric metrics available in CloudWatch, see Monitoring AWS AppFabric with Amazon CloudWatch (p. 87).
- Action: `s3:PutObject` in an Amazon S3 bucket. This action grants permission for AppFabric to put ingested data into an Amazon S3 bucket that you specify.
- Action: `firehose:PutRecordBatch` in an Amazon Kinesis Data Firehose delivery stream. This action grants permission for AppFabric to put ingested data into an Amazon Kinesis Data Firehose delivery stream that you specify.

For more information, see AWS managed policies for AppFabric (p. 80).

You must configure permissions to allow your users, groups, or roles to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the IAM User Guide.
Creating a service-linked role for AppFabric

You don't need to manually create a service-linked role. When you create an AppFabric app bundle in the AWS Management Console, the AWS CLI, or the AWS API, AppFabric creates the service-linked role for you.

Editing a service-linked role for AppFabric

AppFabric doesn't allow you to edit the AWSServiceRoleForAppFabric service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a service-linked role in the IAM User Guide.

Deleting a service-linked role for AppFabric

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must delete all of your AppFabric app bundles before you can delete the service-linked role.

Cleaning up a service-linked role

Before you can use IAM to delete a service-linked role, you must first delete any resources used by the role. App bundles that you create in AppFabric are used by the role. For more information, see Delete AWS AppFabric resources (p. 60).

Note

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

Manually delete the service-linked role

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForAppFabric service-linked role. For more information, see Deleting a service-linked role in the IAM User Guide.

Supported Regions for AppFabric service-linked roles

AppFabric supports using service-linked roles in all of the AWS Regions where the service is available. For more information, see AppFabric endpoints and quotas in the AWS General Reference.

AWS managed policies for AWS AppFabric

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 don't remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

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

AWS managed policy: AWSAppFabricReadOnlyAccess

You can attach the AWSAppFabricReadOnlyAccess policy to your IAM identities. This policy grants read-only permissions to the AppFabric service.

Permissions details

This policy includes the following permissions:

- appfabric – Grants permission to get an app bundle, list app bundles, get an app authorization, list app authorizations, get an ingestion, list ingestions, get an ingestion destination, list ingestion destinations, and list resource tags.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "appfabric:GetAppAuthorization",
                "appfabric:GetAppBundle",
                "appfabric:GetIngestion",
                "appfabric:GetIngestionDestination",
                "appfabric:ListAppAuthorizations",
                "appfabric:ListAppBundles",
                "appfabric:ListIngestionDestinations",
                "appfabric:ListIngestions",
                "appfabric:ListTagsForResource"
            ],
            "Resource": "*
        }
    ]
}
```

AWS managed policy: AWSAppFabricFullAccess

You can attach the AWSAppFabricFullAccess policy to your IAM identities. This policy grants administrative permissions to the AppFabric service.

Permissions details

This policy includes the following permissions:

- appfabric – Grants full administrative permission to AppFabric.
- kms – Grants permission to list aliases.
- s3 – Grants permission to list all of your Amazon S3 buckets, and get bucket location.
- firehose – Grants permission to list Amazon Kinesis Data Firehose delivery streams, and describe delivery streams.
- iam – Grants permission to create the AWSServiceRoleForAppFabric service-linked role for AppFabric. For more information, see Using service-linked roles for AppFabric (p. 79).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```
AWS AppFabric Administration Guide

AWS managed policies

AWS managed policy: AWSAppFabricServiceRolePolicy

You can't attach the AWSAppFabricServiceRolePolicy policy to your IAM entities. This policy is attached to a service-linked role that allows AppFabric to perform actions on your behalf. For more information, see Using service-linked roles for AppFabric (p. 79).

Permissions details

This policy includes the following permissions:

- **cloudwatch** – Grants permission for AppFabric to put metric data into the Amazon CloudWatch AWS/AppFabric namespace. For more information about the AppFabric metrics available in CloudWatch, see Monitoring AWS AppFabric with Amazon CloudWatch (p. 87).
- **s3** – Grants permission for AppFabric to put ingested data into an Amazon S3 bucket that you specify.
- **firehose** – Grants permission for AppFabric to put ingested data into an Amazon Kinesis Data Firehose delivery stream that you specify.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        
```
AppFabric updates to AWS managed policies

View details about updates to AWS managed policies for AppFabric since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the AppFabric Document history (p. 91) page.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWSAppFabricReadOnlyAccess</strong></td>
<td>AppFabric added a new policy to grant read-only permissions to the AppFabric service.</td>
<td>June 27, 2023</td>
</tr>
<tr>
<td>(p. 81)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– New policy</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AWSAppFabricFullAccess</strong></td>
<td>AppFabric added a new policy to grant administrative permissions to the AppFabric service.</td>
<td>June 27, 2023</td>
</tr>
<tr>
<td>(p. 81)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– New policy</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AWSAppFabricServiceRolePolicy</strong></td>
<td>AppFabric added a new policy for the AWSServiceRoleForAppFabric service-linked role.</td>
<td>June 27, 2023</td>
</tr>
<tr>
<td>(p. 81)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>– New policy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AppFabric started tracking changes</td>
<td>AppFabric started tracking changes for its AWS managed policies.</td>
<td>June 27, 2023</td>
</tr>
</tbody>
</table>

Troubleshooting AWS AppFabric identity and access

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

- I am not authorized to perform an action in AppFabric (p. 84)
- I am not authorized to perform iam:PassRole (p. 84)
- I want to allow people outside of my AWS account to access my AppFabric resources (p. 84)

I am not authorized to perform an action in AppFabric

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

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

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

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

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

I am not authorized to perform iam:PassRole

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

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

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

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

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

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

I want to allow people outside of my AWS account to access my AppFabric resources

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

To learn more, consult the following:

- To learn whether AppFabric supports these features, see How AWS AppFabric works with IAM (p. 71).
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the IAM User Guide.
Compliance validation

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

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

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. 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 baseline environments on AWS that are security and compliance focused.
- **Architecting for HIPAA Security and Compliance on Amazon Web Services** – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

  **Note**
  Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- **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 – The AWS Config service 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. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- **AWS Audit Manager** – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Security best practices for AWS AppFabric

AWS AppFabric provides several 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.

Monitor for application without admin access

With the read-only AWS Identity and Access Management (IAM) permission, anyone can integrate AppFabric with Amazon QuickSight and other security information and event management (SIEM) tools.
such as Splunk. To monitor application security, data is delivered to an Amazon Simple Storage Service (Amazon S3) bucket or an Amazon Kinesis Data Firehose delivery stream.

**Monitor for AppFabric events**

You can monitor AppFabric using Amazon CloudWatch metrics. CloudWatch collects data from AppFabric every minute and processes it into metrics. You can set alarms that set off notifications when metrics match specified thresholds. For more information, see *Monitoring AWS AppFabric with Amazon CloudWatch* (p. 87).

**Resilience in AWS AppFabric**

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*.

**Infrastructure security in AWS AppFabric**

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

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

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

**Configuration and vulnerability analysis in AWS AppFabric**

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

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

- **Amazon CloudWatch** monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the [Amazon CloudWatch User Guide](#).

- **Amazon CloudWatch Logs** enables you to monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the [Amazon CloudWatch Logs User Guide](#).

- **AWS CloudTrail** captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the [AWS CloudTrail User Guide](#).

### Monitoring AWS AppFabric with Amazon CloudWatch

You can monitor AWS AppFabric using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met. For more information, see the [Amazon CloudWatch User Guide](#).

The AppFabric service reports the following metrics in the `AWS/AppFabric` namespace.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume of Ingested Data</td>
<td>The size of data that is delivered to Amazon Simple Storage Service (Amazon S3) or Amazon Kinesis Data Firehose.</td>
</tr>
<tr>
<td>Ingestion Destination Status</td>
<td>The status of the ingestion destination (1 for active; 0 for any other).</td>
</tr>
</tbody>
</table>

The following dimension is supported for AppFabric metrics.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingestion Destination Arn</td>
<td>The Amazon Resource Name (ARN) of the ingestion destination.</td>
</tr>
</tbody>
</table>
Logging AWS AppFabric API calls using AWS CloudTrail

AWS AppFabric is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AppFabric. CloudTrail captures all API calls for AppFabric as events. The calls captured include calls from the AppFabric console and code calls to the AppFabric API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AppFabric. 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 AppFabric, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the AWS CloudTrail User Guide.

AppFabric information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AppFabric, 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 in the AWS CloudTrail User Guide.

For an ongoing record of events in your AWS account, including events for AppFabric, 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 topics in the AWS CloudTrail User Guide:

- 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 AppFabric actions are logged by CloudTrail and are documented in the AWS AppFabric API Reference. For example, calls to the CreateAppBundle, UpdateAppBundle, and GetAppBundle 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 CloudTrail userIdentity element in the AWS CloudTrail User Guide.

Understanding AppFabric 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 `CreateAppBundle` action.

```json
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser",
        "arn": "arn:aws:sts::111122223333:assumed-role/AssumedRole/SampleUser",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAXUFER33B4FVC2GCYR",
                "arn": "arn:aws:iam::111122223333:role/AssumedRole",
                "accountId": "111122223333",
                "userName": "SampleUser"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-31T21:11:15Z",
                "mfaAuthenticated": "false"
            }
        },
        "eventTime": "2023-05-31T21:22:16Z",
        "eventSource": "appfabric.amazonaws.com",
        "eventName": "CreateAppBundle",
        "awsRegion": "us-east-1",
        "sourceIpAddress": "3.90.81.91",
        "userAgent": "Coral/Apache-HttpClient5",
        "requestParameters": {
            "clientToken": "64d9069f-e565-49a4-9374-6dc8631142e2"
        },
        "responseElements": {
            "appId": {
                "arn": "arn:aws:appfabric:us-east-1:111122223333:appbundle/6aa92da0-5eeb-4ff4-aabf-4db7fd022ad1",
                "idpClientConfiguration": {
                    "saml1Audience": "urn:amazon:cognito:sp:us-east-1_GEdGiavzr",
                    "samlRedirect": "https://6aa92da0-5eeb-4ff4-aabf-4db7fd022ad1.auth.us-east-1.amazoncognito.com/saml2/idpresponse",
                    "oidcRedirect": "https://6aa92da0-5eeb-4ff4-aabf-4db7fd022ad1.auth.us-east-1.amazoncognito.com/oauth2/idpresponse"
                }
            }
        },
        "requestID": "17e15a5d-8c66-46c7-ad5b-f521004fa9c2",
        "eventID": "ba1dd847-86f6-4386-85be-0398e844a358",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management",
        "tlsDetails": {
            "clientProvidedHostHeader": "frontend.fabric.us-east-1.amazonaws.com"
        }
    }
}
```
Quotas for AWS AppFabric

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

To view the quotas for AppFabric, open the Service Quotas console. In the navigation pane, choose AWS services and select AppFabric.

To request a quota increase, see Requesting a quota increase in the Service Quotas User Guide. If the quota is not yet available in Service Quotas, use the limit increase form.

The quotas related to AppFabric that are in your AWS account are shown in the following table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Default</th>
<th>Adjustable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application bundles</td>
<td>Each supported Region: 1</td>
<td>No</td>
<td>The maximum number of application bundles that you can create in an account in the current AWS Region.</td>
</tr>
<tr>
<td>Application authorizations</td>
<td>Each supported Region: 50</td>
<td>No</td>
<td>The maximum number of application authorizations that you can create in an account in the current AWS Region.</td>
</tr>
<tr>
<td>Ingestions</td>
<td>Each supported Region: 50</td>
<td>No</td>
<td>The maximum number of ingestions that you can create in an account in the current AWS Region.</td>
</tr>
<tr>
<td>Ingestion destinations</td>
<td>Each supported Region: 5</td>
<td>No</td>
<td>The maximum number of ingestion destinations that you can create per ingestion in an account in the current AWS Region.</td>
</tr>
</tbody>
</table>
# Document history for the AppFabric Administration Guide

The following table describes the documentation releases for AWS AppFabric.

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added GitHub and ServiceNow as supported applications. (p. 91)</td>
<td>For more information about the new supported applications, see Supported applications in AWS AppFabric.</td>
<td>October 31, 2023</td>
</tr>
<tr>
<td>Started tracking AWS managed policies for AWS AppFabric (p. 91)</td>
<td>For more information about the AWS managed policies for AppFabric, see AWS managed policies for AWS AppFabric.</td>
<td>June 27, 2023</td>
</tr>
<tr>
<td>Initial release (p. 91)</td>
<td>Initial release of the AWS AppFabric Administration Guide.</td>
<td>June 27, 2023</td>
</tr>
</tbody>
</table>