User Guide

Amazon Security Lake

Amazon Security Lake: User Guide

Table of Contents

What is Amazon Security Lake?

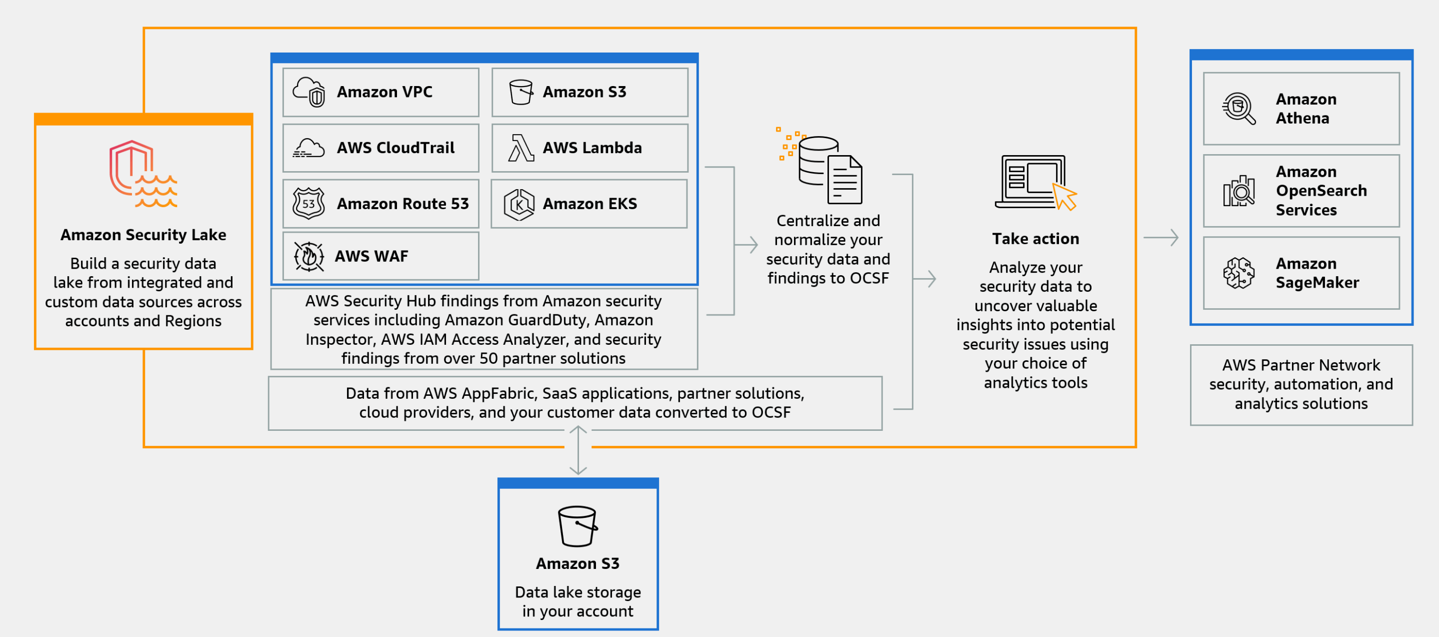
Amazon Security Lake is a fully managed security data lake service. You can use Security Lake to automatically centralize security data from AWS environments, SaaS providers, on premises, cloud sources, and third-party sources into a purpose-built data lake that's stored in your AWS account. Security Lake helps you analyze security data, so you can get a more complete understanding of your security posture across the entire organization. With Security Lake, you can also improve the protection of your workloads, applications, and data.

The data lake is backed by Amazon Simple Storage Service (Amazon S3) buckets, and you retain ownership over your data.

Security Lake automates the collection of security-related log and event data from integrated AWS services and third-party services. It also helps you manage the lifecycle of data with customizable retention and replication settings. Security Lake converts ingested data into Apache Parquet format and a standard open-source schema called the Open Cybersecurity Schema Framework (OCSF). With OCSF support, Security Lake normalizes and combines security data from AWS and a broad range of enterprise security data sources.

Other AWS services and third-party services can subscribe to the data that's stored in Security Lake for incident response and security data analytics.

Overview of Security Lake


Amazon Security Lake
Build a security data lake from integrated and custom data sources across accounts and Regions
Amazon VPC
Amazon S3
AWS CloudTrail
AWS Lambda
Amazon Route 53
Amazon EKS
AWS WAF
AWS Security Hub findings from Amazon security services including Amazon GuardDuty, Amazon Inspector, AWS IAM Access Analyzer, and security findings from over 50 partner solutions
Data from AWS AppFabric, SaaS applications, partner solutions, cloud providers, and your customer data converted to OCSF
Centralize and normalize your security data and findings to OCSF
Take action
Analyze your security data to uncover valuable insights into potential security issues using your choice of analytics tools
Amazon Athena
Amazon OpenSearch Services
Amazon SageMaker
AWS Partner Network security, automation, and analytics solutions
Amazon S3
Data lake storage in your account

Features of Security Lake

Here are some key ways that Security Lake helps you centralize, manage, and subscribe to security-related log and event data.

Data aggregation into your account

Security Lake creates a purpose-built security data lake in your account. Security Lake collects log and event data from cloud, on premises, and custom data sources across accounts and Regions. The data lake is backed by Amazon Simple Storage Service (Amazon S3) buckets, and you retain ownership over your data.

Variety of supported log and event sources

Security Lake collects security logs and events from multiple sources, including on-premises, AWS services, and third-party services. After ingesting logs, regardless of the source, you can access them centrally, and manage their lifecycle. For details about sources from which logs and events are collected by Security Lake, see Source management in Security Lake

Data transformation and normalization

Security Lake automatically partitions incoming data from natively supported AWS services and converts it to a storage- and query-efficient Parquet format. It also transforms data from natively supported AWS services to the Open Cybersecurity Schema Framework (OCSF) open-source schema. This makes the data compatible with other AWS services and third-party providers without the need for post-processing. Since Security Lake normalizes data, many security solutions can consume this data in parallel.

Multiple levels of access for subscribers

Subscribers consume data stored in Security Lake. You can choose a subscriber's level of access to your data. Subscribers may consume data only from the sources, and in the AWS Regions, that you specify. Subscribers may be automatically notified about new objects as they're written to the data lake. Or, subscribers can query data from the data lake. Security Lake automatically creates and exchanges the credentials needed between Security Lake and the subscriber.

Multi-account and multi-Region data management

You can centrally enable Security Lake across all Regions where it's available, and across multiple AWS accounts. In Security Lake, you can also designate rollup Regions to consolidate security log and event data from multiple Regions. This can help you comply with data residency compliance requirements.

Configurable and customizable

Security Lake is a configurable and customizable service. You can specify which sources, accounts, and Regions you want to configure log collection for. You can also specify a subscriber's level of access to the data lake.

Data lifecycle management and optimization

Security Lake manages the lifecycle of your data with customizable retention settings and storage costs with automated storage tiering. Security Lake automatically partitions and converts incoming security data to a storage and query efficient Apache Parquet format.

Accessing Security Lake

For a list of Regions where Security Lake is currently available, see Security Lake Regions and endpoints. To learn more about Regions, see AWS service endpoints in the *AWS General Reference*.

In each Region, you can access Security Lake 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 Security Lake console provides access to your Security Lake account and resources. You can perform most Security Lake tasks by using the Security Lake console.

Security Lake API

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

AWS Command Line Interface (AWS CLI)

With the AWS CLI, you can issue commands at your system's command line to perform Security Lake tasks and AWS tasks. Using the command line can be faster and more convenient than using the console. The command line tools are also useful if you want to build scripts that perform tasks. For information about installing and using the AWS CLI, see the AWS Command Line Interface.

AWS SDKs

AWS provides SDKs that consist of libraries and sample code for various programming languages and platforms, such as Java, Go, Python, C++, and .NET. The SDKs provide convenient, programmatic access to Security Lake and other AWS services. They also handle tasks such as cryptographically signing requests, managing errors, and retrying requests automatically. For information about installing and using the AWS SDKs, see Tools to Build on AWS.

Related services

The following are other AWS services that Security Lake uses:

- Amazon EventBridge – Security Lake uses EventBridge to notify subscribers when objects are written to the data lake.
- AWS Glue – Security Lake uses AWS Glue crawlers to create the AWS Glue Data Catalog tables and send newly written data to the Data Catalog. Security Lake also stores partition metadata for AWS Lake Formation tables in the Data Catalog.
- AWS Lake Formation – Security Lake creates a separate Lake Formation table for each source that contributes data to Security Lake. Lake Formation tables contain information about data from each source, including schema, partition, and data location information. Subscribers have the option to consume data by querying the Lake Formation tables.
- AWS Lambda – Security Lake uses Lambda functions to support extract, transform, and load (ETL) jobs on raw data and to register partitions for source data in AWS Glue.
- Amazon S3 – Security Lake stores your data as Amazon S3 objects. Storage classes and retention settings are based on Amazon S3 offerings. Security Lake doesn't support Amazon S3 Select.
- Amazon Simple Queue Service – Security Lake uses Amazon SQS to enable event-driven processing and manage notifications.

Security Lake collects data from custom sources in addition to the following AWS services:

- AWS CloudTrail management and data events (S3, Lambda)
- Amazon Elastic Kubernetes Service (Amazon EKS) Audit Logs
- Amazon Route 53 resolver query logs
- AWS Security Hub findings

- Amazon Virtual Private Cloud (Amazon VPC) Flow Logs
- AWS WAFv2 Logs

For more information about these sources, see Collecting data from AWS services in Security Lake. You can consume the Amazon S3 objects in your security data lake by creating a subscriber that can read data in the OCSF schema. You can also query data by using Amazon Athena, Amazon Redshift, and third-party subscription services that integrate with AWS Glue.

Concepts and terminology

This section describes the key concepts and terms to help you use Amazon Security Lake.

Contributing Region

One or more AWS Regions that contribute data to a rollup Region.

Data lake

Your persistent data that is stored in Amazon Simple Storage Service (Amazon S3) and managed by Security Lake. Security Lake uses AWS Glue to send newly written data to the Data Catalog. Security Lake also creates a AWS Lake Formation table for each source that contributes data to the data lake. A data lake typically stores the following:

- Structured and unstructured data
- Raw and transformed data

Security Lake is a data lake service that's designed to collect security-related logs and events.

Open Cybersecurity Schema Framework (OCSF)

A standardized open-source schema for security logs and events. It was developed by AWS and other security industry leaders across various security domains. Security Lake automatically converts the logs and events that it collects from AWS services into the OCSF schema. Custom sources convert their logs and events into OCSF before sending them to Security Lake.

Rollup Region

An AWS Region that consolidates security logs and events from one or more contributing Regions. Specifying one or more rollup Regions can help you comply with regional compliance requirements.

Source

A set of logs and events generated from a single system that matches a specific event class in OCSF. Security Lake can collect data from a source. A source may be another AWS service or a third-party service. For third-party sources, you must convert the data to the OCSF schema before sending it to Security Lake.

Subscriber

A service that consumes logs and events from Security Lake. A subscriber may be another AWS service or a third-party service.

Getting started with Amazon Security Lake

The topics in this section explain how to enable and start using Security Lake. You'll learn how to configure your data lake settings and set up log collection. You can enable and use Security Lake through the AWS Management Console or programmatically. Whichever method you use, you must first set up an AWS account and an administrative user. The steps after that differ based on the method of access.

The Security Lake console offers a streamlined process for getting started, and creates all necessary AWS Identity and Access Management (IAM) roles that you need to create your data lake.

If you access Security Lake programmatically, it's necessary to create some AWS Identity and Access Management (IAM) roles in order to configure your data lake.

> ⚠ **Important**
>
> Security Lake does not support backfilling of existing AWS raw log source events that were generated before enabling Security Lake.

Topics

Setting up your AWS account

Before you can enable Amazon Security Lake, you must have an AWS account. If you do not have an AWS account, complete the following steps to create one.

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

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

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

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

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 a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and 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 a user with administrative access

1. Enable IAM Identity Center.

For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- 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*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

Identify the account that you'll use to enable Security Lake

Security Lake integrates with AWS Organizations to manage log collection across multiple accounts in an organization. If you want to use Security Lake for an organization, you must use your Organizations management account to designate a delegated Security Lake administrator. Then, you must use the credentials of the delegated administrator to enable Security Lake, add member accounts, and enable Security Lake for them. For more information, see Managing multiple accounts with AWS Organizations in Security Lake.

Alternatively, you can use Security Lake without the Organizations integration for a standalone account that's not part of an organization.

Considerations when enabling Security Lake

Before enabling Security Lake, consider the following:

- Security Lake provides cross-region management features, which means you can create your data lake and configure log collection across AWS Regions. To enable Security Lake in all supported Regions, you can choose any supported Regional endpoint. You can also add rollup Regions to aggregate data from multiple regions to a single Region.
- We recommend activating Security Lake in all of the supported AWS Regions. If you do this, Security Lake can collect data that's connected to unauthorized or unusual activity even in Regions that you aren't actively using. If Security Lake is not activated in all supported Regions, its ability to collect data from other services that you use in multiple Regions is reduced.
- When you enable Security Lake for the first time in any Region, it creates a service-linked role for your account called `AWSServiceRoleForSecurityLake`. This role includes the permissions to call other AWS services on your behalf and operate the security data lake. For more information about how service-linked roles work, see Using service-linked roles in the *IAM User Guide*. If you enable Security Lake as the delegated Security Lake administrator, Security Lake creates the service-linked role in each member account in the organization.
- Security Lake doesn't support Amazon S3 Object Lock. When the data lake buckets are created, S3 Object Lock is disabled by default. Enabling Object Lock on a bucket interrupts the delivery of normalized log data to the data lake.
- If you are re-enabling Security Lake in a region, you must delete the region's corresponding AWS Glue database from your previous use of Security Lake.

Enabling Security Lake using the console

This tutorial explains how to enable and configure Security Lake through the AWS Management Console. As part of the AWS Management Console, the Security Lake console offers a streamlined process for getting started, and creates all necessary AWS Identity and Access Management (IAM) roles that you need to create your data lake.

Step 1: Configure sources

Security Lake collects log and event data from a variety of sources and across your AWS accounts and AWS Regions. Follow these instructions to identify which data you want Security Lake to collect. You can only use these instructions to add a natively-supported AWS service as a source.

For information about adding a custom source, see Collecting data from custom sources in Security Lake.

To configure log source collection

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. By using the AWS Region selector in the upper-right corner of the page, select a Region. You can enable Security Lake in the current Region and other Regions while onboarding.
3. Choose **Get started**.
4. For **Select log and event sources**, choose one of the following options:

   a. **Ingest default AWS sources –** When you choose the recommended option, CloudTrail - S3 data events isn't included for ingestion. This is because ingesting high volume of CloudTrail - S3 data events might impact the usage cost significantly. To ingest this source, select the **Ingest specific AWS sources** option.

   b. **Ingest specific AWS sources –** With this option, you can select one or more log and event sources that you want to ingest.

   > **Note**
   >
   > When you enable Security Lake in an account for the first time, all the selected log and event sources will be a part of a 15-day free trial period. For more information about usage statistics, see Reviewing usage and estimated costs.

5. For **Versions**, chose the version of data source from which you want to ingest log and event sources.

   > **Important**
   >
   > If you don't have the required role permissions to enable the new version of the AWS log source in the specified Region, contact your Security Lake administrator. For more information, see Update role permissions.

6. For **Select Regions**, choose whether to ingest log and event sources from all supported Regions or specific Regions. If you choose **Specific Regions**, select which Regions to ingest data from.

7. For **Service access**, create a new IAM role or use an existing IAM role that gives Security Lake permission to collect data from your sources and add them to your data lake. One role is used across all Regions in which you enable Security Lake.
8. Choose **Next**.

Step 2: Define storage settings and rollup Regions (optional)

You can specify the Amazon S3 storage class in which you want Security Lake to store your data and for how long. You can also specify a rollup Region to consolidate data from multiple Regions. These are optional steps. For more information, see Lifecycle management in Security Lake.

To configure storage and rollup settings

1. If you want to consolidate data from multiple contributing Regions to a rollup Region, for **Select rollup Regions**, choose **Add rollup Region**. Specify the rollup Region and the Regions that will contribute to it. You can set up one or more rollup Regions.
2. For **Select storage classes**, choose an Amazon S3 storage class. The default storage class is **S3 Standard**. Provide a retention period (in days) if you want the data to transition to another storage class after that time, and choose **Add transition**. After the retention period ends, the objects expire and Amazon S3 deletes them. For more information about Amazon S3 storage classes and retention, see Retention management.
3. If you selected a rollup Region in the first step, for **Service access**, create a new IAM role or use an existing IAM role that gives Security Lake permission to replicate data across multiple Regions.
4. Choose **Next**.

Step 3: Review and create data lake

Review the sources that Security Lake will collect data from, your rollup Regions, and your retention settings. Then, create your data lake.

To review and create the data lake

1. While enabling Security Lake, review **Log and event sources**, **Regions**, **Rollup Regions**, and **Storage classes**.
2. Choose **Create**.

After creating your data lake, you will see the **Summary** page on the Security Lake console. This page provides an overview of the number of **Regions** and **Rollup Regions**, information about subscribers, and **Issues**.

The **Issues** menu shows you a summary of issues from the last 14 days that are impacting the Security Lake service or your Amazon S3 buckets. For additional details about each issue, you can go to the **Issues** page of the Security Lake console.

Step 4: View and query your own data

After creating your data lake, you can use Amazon Athena or similar services to view and query your data from AWS Lake Formation databases and tables. When you use the console, Security Lake automatically grants database view permissions to the role that you use to enable Security Lake. At a minimum, the role must have *Data analyst* permissions. For more information on permission levels, see Lake Formation personas and IAM permissions reference. For instructions on granting SELECT permissions, see Granting Data Catalog permissions using the named resource method in the *AWS Lake Formation Developer Guide*.

Step 5: Create subscribers

After creating your data lake, you can add subscribers to consume your data. Subscribers can consume data by directly accessing objects in your Amazon S3 buckets or by querying the data lake. For more information about subscribers, see Subscriber management in Security Lake.

Enabling Security Lake programmatically

This tutorial explains how to enable and start using Security Lake programmatically. The Amazon Security Lake API gives you comprehensive, programmatic access to your Security Lake account, data, and resources. Alternatively, you can use AWS command line tools— the AWS Command Line Interface or the AWS Tools for PowerShell—or the AWS SDKs to access Security Lake.

Step 1: Create IAM roles

If you access Security Lake programmatically, it's necessary to create some AWS Identity and Access Management (IAM) roles in order to configure your data lake.

⚠ **Important**

It's not necessary to create these IAM roles if you use the Security Lake console to enable and configure Security Lake.

You must create roles in IAM if you'll be taking one or more of the following actions (choose the links to see more information about IAM roles for each action):

- Creating a custom source – Custom sources are sources other than natively-supported AWS services that send data to Security Lake.
- Creating a subscriber with data access – Subscribers with permissions can directly access S3 objects from your data lake.
- Creating a subscriber with query access – Subscribers with permissions can query data from Security Lake using services like Amazon Athena.
- Configuring a rollup Region – A rollup Region consolidates data from multiple AWS Regions.

After creating the roles previously mentioned, attach the AmazonSecurityLakeAdministrator AWS managed policy to the role that you're using to enable Security Lake. This policy grants administrative permissions that allow a principal to onboard to Security Lake and access all Security Lake actions.

Attach the AmazonSecurityLakeMetaStoreManager AWS managed policy to create your data lake or query data from Security Lake. This policy is necessary for Security Lake to support extract, transform, and load (ETL) jobs on raw log and event data that it receives from sources.

Step 2: Enable Amazon Security Lake

To enable Security Lake programmatically, use the CreateDataLake operation of the Security Lake API. If you're using the AWS CLI, run the create-data-lake command. In your request, use the `region` field of the `configurations` object to specify the Region code for the Region in which to enable Security Lake. For a list of Region codes, see Amazon Security Lake endpoints in the *AWS General Reference*.

Example 1

The following example command enables Security Lake in the `us-east-1` and `us-east-2` Regions. In both Regions, this data lake is encrypted with Amazon S3 managed keys. Objects expire after 365 days, and objects transition to the ONEZONE_IA S3 storage class after 60 days. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake create-data-lake \
--configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":[{"days":60,"storageClass":"ONEZONE_IA"}]}},
 {"encryptionConfiguration": {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
--meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-role/
AmazonSecurityLakeMetaStoreManager"
```

Example 2

The following example command enables Security Lake in the us-east-2 Region. This data lake is encrypted with a customer managed key that was created in AWS Key Management Service (AWS KMS). Objects expire after 500 days, and objects transition to the GLACIER S3 storage class after 30 days. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake create-data-lake \
--configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}}]' \
--meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-role/
AmazonSecurityLakeMetaStoreManager"
```

Note

If you've already enabled Security Lake and want to update the configuration settings for a Region or source, use the UpdateDataLake operation, or if using the AWS CLI, the update-data-lake command. Don't use the CreateDataLake operation.

Step 3: Configure sources

Security Lake collects log and event data from a variety of sources and across your AWS accounts and AWS Regions. Follow these instructions to identify which data you want Security Lake to collect. You can only use these instructions to add a natively-supported AWS service as a source. For information about adding a custom source, see Collecting data from custom sources in Security Lake.

To define one or more collection sources programmatically, use the CreateAwsLogSource operation of the Security Lake API. For each source, specify a Regionally unique value for the `sourceName` parameter. Optionally use additional parameters to limit the scope of the source to specific accounts (`accounts`) or a specific version (`sourceVersion`).

> **Note**
>
> If you don't include an optional parameter in your request, Security Lake applies your request to all accounts or all versions of the specified source, depending on the parameter that you exclude. For example, if you're the delegated Security Lake administrator for an organization and you exclude the `accounts` parameter, Security Lake applies your request to all the accounts in your organization. Similarly, if you exclude the `sourceVersion` parameter, Security Lake applies your request to all versions of the specified source.

If your request specifies a Region in which you haven't enabled Security Lake, an error occurs. To address this error, ensure that the `regions` array specifies only those Regions in which you've enabled Security Lake. Alternatively, you can enable Security Lake in the Region, and then submit your request again.

When you enable Security Lake in an account for the first time, all the selected log and event sources will be a part of a 15-day free trial period. For more information about usage statistics, see Reviewing usage and estimated costs.

Step 4: Configure storage settings and rollup Regions (optional)

You can specify the Amazon S3 storage class in which you want Security Lake to store your data and for how long. You can also specify a rollup Region to consolidate data from multiple Regions. These are optional steps. For more information, see Lifecycle management in Security Lake.

To define a target objective programmatically when you enable Security Lake, use the CreateDataLake operation of the Security Lake API. If you've already enabled Security Lake and want to define a target objective, use the UpdateDataLake operation, not the `CreateDataLake` operation.

For either operation, use the supported parameters to specify the configuration settings that you want:

- To specify a rollup Region, use the `region` field to specify the Region that you want to contribute data to the rollup Regions. In the `regions` array of the `replicationConfiguration` object, specify the Region code for each rollup Region. For a list of Region codes, see Amazon Security Lake endpoints in the *AWS General Reference*.
- To specify retention settings for your data, use the `lifecycleConfiguration` parameters:
  - For `transitions`, specify the total number of days (`days`) that you want to store S3 objects in a particular Amazon S3 storage class (`storageClass`).
  - For `expiration`, specify the total number of days that you want to store objects in Amazon S3, using any storage class, after objects are created. When this retention period ends, objects expire and Amazon S3 deletes them.

  Security Lake applies the specified retention settings to the Region that you specify in the `region` field of the `configurations` object.

For example, the following command creates a data lake with `ap-northeast-2` as a rollup Region. The `us-east-1` Region will contribute data to the `ap-northeast-2` Region. This example also establishes a 10-day expiration period for objects that are added to the data lake.

```
$ aws securitylake create-data-lake \
--configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","replicationConfiguration":
 {"regions": ["ap-northeast-2"],"roleArn":"arn:aws:iam::123456789012:role/service-
role/AmazonSecurityLakeS3ReplicationRole"},"lifecycleConfiguration": {"expiration":
{"days":10}}}]' \
--meta-store-manager-role-arn "arn:aws:iam::123456789012:role/service-role/
AmazonSecurityLakeMetaStoreManager"
```

You have now created your data lake. Use the ListDataLakes operation of the Security Lake API to verify enablement of Security Lake and your data lake settings in each Region.

If issues or errors arise in the creation of your data lake, you can view a list of exceptions by using the ListDataLakeExceptions operation, and notify users of exceptions with the CreateDataLakeExceptionSubscription operation. For more information, see Troubleshooting data lake status.

Step 5: View and query your own data

After creating your data lake, you can use Amazon Athena or similar services to view and query your data from AWS Lake Formation databases and tables. When you programmatically enable Security Lake, database view permissions aren't granted automatically. The data lake administrator account in AWS Lake Formation must grant SELECT permissions to the IAM role you want to use to query the relevant databases and tables. At a minimum, the role must have *Data analyst* permissions. For more information on permission levels, see Lake Formation personas and IAM permissions reference. For instructions on granting SELECT permissions, see Granting Data Catalog permissions using the named resource method in the *AWS Lake Formation Developer Guide*.

Step 6: Create subscribers

After creating your data lake, you can add subscribers to consume your data. Subscribers can consume data by directly accessing objects in your Amazon S3 buckets or by querying the data lake. For more information about subscribers, see Subscriber management in Security Lake.

Managing multiple accounts with AWS Organizations in Security Lake

You can use Amazon Security Lake to collect security logs and events from multiple AWS accounts. To help automate and streamline the management of multiple accounts, we strongly recommend that you integrate Security Lake with AWS Organizations.

In Organizations, the account that you use to create the organization is called the management account. To integrate Security Lake with Organizations, the management account must designate a delegated Security Lake administrator account for the organization.

The delegated Security Lake administrator can enable Security Lake and configure Security Lake settings for member accounts. The delegated administrator can collect logs and events across the organization in all AWS Regions where Security Lake is enabled (regardless of which Regional endpoint they're currently using). The delegated administrator can also configure Security Lake to automatically collect log and event data for new organization accounts.

The delegated Security Lake administrator has access to log and event data for associated member accounts. Accordingly, they can configure Security Lake to collect data owned by associated member accounts. They can also grant subscribers permission to consume data owned by associated member accounts.

To enable Security Lake for multiple accounts in an organization, the organization management account must first designate a delegated Security Lake administrator account for the organization. The delegated administrator can then enable and configure Security Lake for the organization.

> ⚠ **Important**
>
> Use Security Lake's RegisterDataLakeDelegatedAdministrator API to allow Security Lake access to your Organization and register Organizations's delegated administrator.
> If you use Organizations' APIs to register a delegated administrator, service-linked roles for the Organizations might not be created successfully. To ensure full functionality, use the Security Lake APIs.

For information about setting up Organizations, see Creating and managing an organization in the *AWS Organizations User Guide*.

Important considerations for delegated Security Lake administrators

Take note of the following factors that define how a delegated administrator behaves in Security Lake:

The delegated administrator is the same in all Regions.

When you create the delegated administrator, it becomes the delegated administrator for every Region in which you enable Security Lake.

We recommend setting the Log Archive account as the Security Lake delegated administrator.

The Log Archive account is an AWS account that is dedicated to ingesting and archiving all security-related logs. Access to this account is typically limited to a few users, such as auditors and security teams for compliance investigations. We recommend setting the Log Archive account as the Security Lake delegated administrator so that you can view security-related logs and events with minimal context switching.

In addition, we recommend that only a minimal set of users have direct access to the Log Archive account. Outside of this select group, if a user needs access to the data that Security Lake collects, you can add them as a Security Lake subscriber. For information about adding a subscriber, see Subscriber management in Security Lake.

If you don't use the AWS Control Tower service, you may not have a Log Archive account. For more information about the Log Archive account, see Security OU – Log Archive account in the *AWS Security Reference Architecture*.

An organization can have only one delegated administrator.

You can have only one delegated Security Lake administrator for each organization.

The organization management account cannot be the delegated administrator.

Based on AWS Security best practices and the principle of least privilege, your organization management account cannot be the delegated administrator.

The delegated administrator must be part of an active organization.

When you delete an organization, the delegated administrator account can no longer manage Security Lake. You must designate a delegated administrator from a different organization or use Security Lake with a standalone account that's not part of an organization.

IAM permissions required to designate the delegated administrator

When designating the delegated Security Lake administrator, you must have permissions to enable Security Lake and use certain AWS Organizations API operations listed in the following policy statement.

You can add the following statement to the end of an AWS Identity and Access Management (IAM) policy to grant these permissions.

```
{
    "Sid": "Grant permissions to designate a delegated Security Lake administrator",
    "Effect": "Allow",
    "Action": [
        "securitylake:RegisterDataLakeDelegatedAdministrator",
        "organizations:EnableAWSServiceAccess",
        "organizations:RegisterDelegatedAdministrator",
        "organizations:ListAccounts",
        "organizations:ListDelegatedAdministrators",
        "organizations:ListAWSServiceAccessForOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization"
    ],
    "Resource": "*"
}
```

Designating the delegated Security Lake administrator and adding member accounts

Choose your access method to designate the delegated Security Lake administrator account for your organization. Only the organization management account can designate the delegated administrator account for their organization. The organization management account cannot be the delegated administrator account for their organization.

Note

- The organization management account should use the Security Lake `RegisterDataLakeDelegatedAdministrator` operation to designate the delegated Security Lake administrator account. Designating the delegated Security Lake administrator through Organizations isn't supported.
- If you want to change the delegated administrator for the organization, you must first remove the current delegated administrator. You can then designate a new delegated administrator.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

   Sign in using the credentials of the management account for your organization.
2. - If Security Lake is not yet enabled, select **Get Started**, and then designate the delegated Security Lake administrator on the **Enable Security Lake** page.
   - If Security Lake is already enabled, designate the delegated Security Lake administrator on the **Settings** page.
3. Under **Delegate administration to another account**, select the account that already serves as the delegated administrator for other AWS security services (recommended). Alternatively, enter the 12-digit AWS account ID of the account that you want to designate as the delegated Security Lake administrator.
4. Choose **Delegate**. If Security Lake is not already enabled, designating the delegated administrator will enable Security Lake for that account in your current Region.

API

To designate the delegated administrator programmatically, use the RegisterDataLakeDelegatedAdministrator operation of the Security Lake API. You must invoke the operation from the organization management account. If you're using the AWS CLI, run the register-data-lake-delegated-administrator command from the organization management account. In your request, use the `accountId` parameter to specify the 12-digit account ID of the AWS account to designate as the delegated administrator account for the organization.

For example, the following AWS CLI command designates the delegated administrator. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake register-data-lake-delegated-administrator \
--account-id 123456789012
```

The delegated administrator can also choose to automate the collection of AWS log and event data for new organization accounts. With this configuration, Security Lake is automatically enabled in new accounts when the accounts are added to the organization in AWS Organizations. As the delegated administrator, you can enable this configuration by using the CreateDataLakeOrganizationConfiguration operation of the Security Lake API or, if you're using the AWS CLI, by running the create-data-lake-organization-configuration command. In your request, you can also specify certain configuration settings for new accounts.

For example, the following AWS CLI command automatically enables Security Lake and the collection of Amazon Route 53 resolver query logs, AWS Security Hub findings, and Amazon Virtual Private Cloud (Amazon VPC) Flow Logs in new organization accounts. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake create-data-lake-organization-configuration \
--auto-enable-new-account '[{"region":"us-east-1","sources":
[{"sourceName":"ROUTE53"},{"sourceName":"SH_FINDINGS"},{"sourceName":"VPC_FLOW"}]}]'
```

After the organization management account designates the delegated administrator, the administrator can enable and configure Security Lake for the organization. This includes enabling and configuring Security Lake to collect AWS log and event data for individual accounts in the organization. For more information, see Collecting data from AWS services in Security Lake.

You can use the GetDataLakeOrganizationConfiguration operation to get details about your organization's current configuration for new member accounts.

Removing the delegated Security Lake administrator

Only the organization management account can remove the delegated Security Lake administrator for their organization. If you want to change the delegated administrator for the organization, remove the current delegated administrator, and then designate the new delegated administrator.

> **Important**
>
> Removing the delegated Security Lake administrator deletes your data lake and disables Security Lake for the accounts in your organization.

You can't change or remove the delegated administrator by using the Security Lake console. These tasks can only be performed programmatically.

To remove the delegated administrator programmatically, use the DeregisterDataLakeDelegatedAdministrator operation of the Security Lake API. You must invoke the operation from the organization management account. The If you're using the AWS CLI, run the deregister-data-lake-delegated-administrator command from the organization management account.

For example, the following AWS CLI command removes the delegated Security Lake administrator.

```
$ aws securitylake deregister-data-lake-delegated-administrator
```

To keep the delegated administrator designation but change the automatic configuration settings of new member accounts, use the DeleteDataLakeOrganizationConfiguration operation of the Security Lake API, or, if you're using the AWS CLI, the delete-data-lake-organization-configuration command. Only the delegated administrator can change these settings for the organization.

For example, the following AWS CLI command stops the automatic collection of Security Hub findings from new member accounts that join the organization. New member accounts won't contribute Security Hub findings to the data lake after the delegated administrator invokes this operation. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake delete-data-lake-organization-configuration \
--auto-enable-new-account '[{"region":"us-east-1","sources":
[{"sourceName":"SH_FINDINGS"}]}]'
```

Security Lake trusted access

After you set up Security Lake for an organization, the AWS Organizations management account can enable trusted access with Security Lake. Trusted access allows Security Lake to create an IAM service-linked role and perform tasks in your organization and its accounts on your behalf. For more information, see Using AWS Organizations with other AWS services in the *AWS Organizations User Guide*.

As a user of the organization management account, you can disable trusted access for Security Lake in AWS Organizations. For instructions on disabling trusted access, see How to enable or disable trusted access in the *AWS Organizations User Guide*.

We recommend disabling trusted access if the delegated administrator's AWS account is suspended, isolated, or closed.

Managing Regions in Security Lake

Amazon Security Lake can collect security logs and events across AWS Regions in which you've enabled the service. For each Region, your data is stored in a different Amazon S3 bucket. You can specify different data lake configurations (for example, different sources and retention settings) for different Regions. You can also define one or more rollup Regions to consolidate data from multiple Regions.

Checking Region status

Security Lake can collect data across multiple AWS Regions. To track the state of your data lake, it can be helpful to understand how each Region is currently configured. Choose your preferred access method, and follow these steps to get the current status of a Region.

Console

To check Region status

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. In the navigation pane, choose **Regions**. The **Regions** page appears, providing an overview of the Regions in which Security Lake is currently enabled.
3. Select a Region, and then choose **Edit** to see details for that Region.

API

To get the status of log collection in the current Region, use the GetDataLakeSources operation of the Security Lake API. If you're using the AWS CLI, run the get-data-lake-sources command. For the `accounts` parameter, specify one or more AWS account IDs as a list. If your request succeeds, Security Lake returns a snapshot for those accounts in the current Region, including which AWS sources Security Lake is collecting data from and the status of each source. If you don't include the `accounts` parameter, the response includes the status of log collection for all accounts in which Security Lake is configured in the current Region.

For example, the following AWS CLI command retrieves log collection status for the specified accounts in the current Region. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake get-data-lake-sources \
--accounts "123456789012" "111122223333"
```

The following AWS CLI command lists log collection status for all accounts and enabled sources in the specified Region. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake get-data-lake-sources \
--regions "us-east-1" \
--query 'dataLakeSources[].[account,sourceName]'
```

To determine whether you've enabled Security Lake for a Region, use the ListDataLakes operation. If you're using the AWS CLI, run the list-data-lakes command. For the `regions` parameter, specify the Region code for the Region—for example, `us-east-1` for the US East (N. Virginia) Region. For a list of Region codes, see Amazon Security Lake endpoints in the *AWS General Reference*. The `ListDataLakes` operation returns the data lake configuration settings for each Region that you specify in your request. If you don't specify a Region, Security Lake returns the status and configuration settings of your data lake in each Region in which Security Lake is available.

For example, the following AWS CLI command shows the status and configuration settings of your data lake in the `eu-central-1` Region. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake list-data-lakes \
--regions "us-east-1" "eu-central-1"
```

Changing Region settings

Choose your preferred method, and follow these instructions to update settings for your data lake in one or more AWS Regions.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. In the navigation pane, choose **Regions**.
3. Select a Region, and then choose **Edit**.

4. Select the check box for **Override sources for all accounts in *<Region>*** to confirm that your selections here override previous selections for this Region.
5. For **Select storage classes**, choose **Add transition** to add new storage classes for your data.
6. For **Tags**, optionally assign or edit the tags for the Region. A *tag* is a label that you can define and assign to certain types of AWS resources, including the data lake configuration for your AWS account in a particular Region. To learn more, see Tagging Security Lake resources.
7. To turn a Region into a rollup Region, choose **Rollup Regions** (under **Settings**) in the navigation pane. Then choose **Modify**. In the **Select rollup Regions** section, choose **Add rollup Region**. Select the contributing Regions, and provide Security Lake with permission to replicate data across multiple Regions. When you finish, choose **Save** to save your changes.

API

To update Region settings for your data lake programmatically, use the UpdateDataLake operation of the Security Lake API. If you're using the AWS CLI, run the update-data-lake command. For the `region` parameter, specify the Region code for the Region that you want to change the settings for—for example, `us-east-1` for the US East (N. Virginia) Region. For a list of Region codes, see Amazon Security Lake endpoints in the *AWS General Reference*.

Use additional parameters to specify a new value for each setting that you want to change —for example, the encryption key (`encryptionConfiguration`) and retention settings (`lifecycleConfiguration`).

For example, the following AWS CLI command updates the data expiration and storage class transition settings for the `us-east-1` Region. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ update-data-lake \
--configurations '[{"region":"us-east-1","lifecycleConfiguration": {"expiration":
{"days":500},"transitions":[{"days":45,"storageClass":"ONEZONE_IA"}]}}]'
```

Configuring rollup Regions in Security Lake

A rollup Region consolidates data from one or more contributing Regions. Specifying a rollup Region can help you comply with Regional compliance requirements.

Due to limitations in Amazon S3, replication from Customer Managed Key (CMK) encrypted regional data lake to S3 managed encrypted (default encryption) regional data lake is not supported.

> **Important**
>
> If you created a custom source, to ensure that custom source data is replicated properly to the destination, Security Lake recommends following the best practices described in Best practices for ingesting custom sources. Replication cannot be performed on data that does not follow the S3 partition data path format as described on the page.

Before adding a rollup Region, you first need to create two different roles in AWS Identity and Access Management (IAM):

- IAM role for data replication
- IAM role to register AWS Glue partitions

> **Note**
>
> Security Lake creates these IAM roles or uses existing roles on your behalf when you use the Security Lake console. However, you must create these roles when using the Security Lake API or AWS CLI.

IAM role for data replication

This IAM role grants permission to Amazon S3 to replicate source logs and events across multiple Regions.

To grant these permissions, create an IAM role that starts with the prefix `SecurityLake`, and attach the following sample policy to the role. You'll need the Amazon Resource Name (ARN) of the role when you create a rollup Region in Security Lake. In this policy, `sourceRegions` are contributing Regions, and `destinationRegions` are rollup Regions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowReadS3ReplicationSetting",
      "Action": [
        "s3:ListBucket",
        "s3:GetReplicationConfiguration",
        "s3:GetObjectVersionForReplication",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging",
        "s3:GetObjectRetention",
        "s3:GetObjectLegalHold"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::aws-security-data-lake-[[sourceRegions]]*",
        "arn:aws:s3:::aws-security-data-lake-[[sourceRegions]]*/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:ResourceAccount": [
            "{{bucketOwnerAccountId}}"
          ]
        }
      }
    },
    {
      "Sid": "AllowS3Replication",
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ReplicateTags",
        "s3:GetObjectVersionTagging"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::aws-security-data-lake-[[destinationRegions]]*/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:ResourceAccount": [
            "{{bucketOwnerAccountId}}"
          ]
        }
      }
    }
  ]
}
```

Attach the following trust policy to your role to permit Amazon S3 to assume the role:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3ToAssume",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

If you use a customer managed key from AWS Key Management Service (AWS KMS) to encrypt your Security Lake data lake, you must grant the following permissions in addition to the permissions in the data replication policy.

```
{
    "Action": [
        "kms:Decrypt"
    ],
    "Effect": "Allow",
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "s3.{sourceRegion1}.amazonaws.com",
                "s3.{sourceRegion2}.amazonaws.com"
                ],
            "kms:EncryptionContext:aws:s3:arn": [
                "arn:aws:s3:::aws-security-data-lake-{sourceRegion1}*",
                "arn:aws:s3:::aws-security-data-lake-{sourceRegion2}*"
            ]
        }
    },
    "Resource": [
        "{sourceRegion1KmsKeyArn}",
        "{sourceRegion2KmsKeyArn}"
    ]
},
{
    "Action": [
        "kms:Encrypt"
    ],
    "Effect": "Allow",
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
            "s3.{destinationRegion1}.amazonaws.com",
            ],
            "kms:EncryptionContext:aws:s3:arn": [
                "arn:aws:s3:::aws-security-data-lake-{destinationRegion1}*",
            ]
        }
    },
    "Resource": [
            "{destinationRegionKmsKeyArn}"
    ]
}
```

For more information about replication roles, see Setting up permissions in the *Amazon Simple Storage Service User Guide*.

IAM role to register AWS Glue partitions

This IAM role grants permissions for a partition updater AWS Lambda function used by Security Lake to register AWS Glue partitions for the S3 objects that were replicated from other regions. Without creating this role, subscribers can't query events from those objects.

To grant these permissions, create a role named `AmazonSecurityLakeMetaStoreManager` (you may have already created this role while onboarding to Security Lake). For more information about this role, including a sample policy, see Step 1: Create IAM roles.

In the Lake Formation console, you must also grant `AmazonSecurityLakeMetaStoreManager` permissions as a data lake administrator by following these steps:

1. Open the Lake Formation console at https://console.aws.amazon.com/lakeformation/.

2. Sign in as an administrative user.
3. If a **Welcome to Lake Formation** window appears, choose the user that you created or selected in Step 1, and then choose Get started.
4. If you don't see a **Welcome to Lake Formation** window, then perform the following steps to configure a Lake Formation Administrator.

    1. In the navigation pane, under **Permissions**, choose **Administrative Roles and tasks**. In the **Data lake administrators** section of the console page, choose **Choose administrators**.
    2. In the **Manage data lake administrators** dialog box, for IAM users and roles, choose the **AmazonSecurityLakeMetaStoreManager** IAM role that you created, and then choose **Save**.

For more information about changing permissions for data lake administrators, see Create a data lake administrator in the *AWS Lake Formation Developer Guide*.

Adding rollup Regions

Choose your preferred access method, and follow these steps to add a rollup Region.

> **Note**
>
> A Region can contribute data to multiple rollup Regions. However, a rollup Region cannot be a contributing Region for another rollup Region.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. In the navigation pane, under **Settings**, choose **Rollup Regions**.
3. Choose **Modify**, and then choose **Add rollup Region**.
4. Specify the rollup Region and contributing Regions. Repeat this step if you want to add multiple rollup Regions.
5. If this is your first time adding a rollup Region, for **Service access**, create a new IAM role or use an existing IAM role that gives Security Lake permission to replicate data across multiple Regions.
6. When you finish, choose **Save**.

You can also add a rollup Region when you onboard to Security Lake. For more information, see Getting started with Amazon Security Lake.

API

To add a rollup Region programmatically, use the UpdateDataLake operation of the Security Lake API. If you're using the AWS CLI, run the update-data-lake command. In your request, use the `region` field to specify the Region that you want to contribute data to the rollup Region. In the `regions` array of the `replicationConfiguration` parameter, specify the Region code for each rollup Region. For a list of Region codes, see Amazon Security Lake endpoints in the *AWS General Reference*.

For example, the following command sets `ap-northeast-2` as a rollup Region. The `us-east-1` Region will contribute data to the `ap-northeast-2` Region. This example also establishes a 365-day expiration period for objects that are added to the data lake. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake update-data-lake \
--configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","replicationConfiguration":
 {"regions": ["ap-northeast-2"],"roleArn":"arn:aws:iam::123456789012:role/service-
role/AmazonSecurityLakeS3ReplicationRole"},"lifecycleConfiguration": {"expiration":
{"days":365}}}]'
```

You can also add a rollup Region when you onboard to Security Lake. To do this, use the CreateDataLake operation (or, if using the AWS CLI, the create-data-lake command). For more information about configuring rollup Regions during onboarding, see Getting started with Amazon Security Lake.

Updating or removing rollup Regions

Choose your preferred access method, and follow these steps to update or remove rollup Regions in Security Lake.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. In the navigation pane, under **Settings**, choose **Rollup Regions**.

3. Choose **Modify**.
4. To change the contributing Regions for a rollup Region, specify the updated contributing Regions in the row for rollup Region.
5. To remove a rollup Region, choose **Remove** in the row for rollup Region.
6. When you finish, choose **Save**.

API

To configure rollup Regions programmatically, use the UpdateDataLake operation of the Security Lake API. If you're using the AWS CLI, run the update-data-lake command. In your request, use the supported parameters to specify the rollup settings:

- To add a contributing Region, use the `region` field to specify the Region code for the Region to add. In the `regions` array of the `replicationConfiguration` object, specify the Region code for each rollup Region to contribute data to. For a list of Region codes, see Amazon Security Lake endpoints in the *AWS General Reference*.
- To remove a contributing Region, use the `region` field to specify the Region code for the Region to remove. For the `replicationConfiguration` parameters, don't specify any values.

For example, the following command configures both `us-east-1` and `us-east-2` as contributing Regions. Both Regions will contribute data to the `ap-northeast-3` rollup Region. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake update-data-lake \
--configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","replicationConfiguration":
 {"regions": ["ap-northeast-3"],"roleArn":"arn:aws:iam::123456789012:role/service-
role/AmazonSecurityLakeS3ReplicationRole"},"lifecycleConfiguration": {"expiration":
{"days":365}}},
{"encryptionConfiguration": {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-
east-2","replicationConfiguration": {"regions": ["ap-
northeast-3"],"roleArn":"arn:aws:iam::123456789012:role/service-role/
AmazonSecurityLakeS3ReplicationRole"}, "lifecycleConfiguration": {"expiration":
{"days":500},"transitions":[{"days":60,"storageClass":"ONEZONE_IA"}]}}]'
```

Source management in Security Lake

Sources are logs and events generated from a single system that match a specific event class in the Open Cybersecurity Schema Framework (OCSF) in Security Lake schema. Amazon Security Lake can collect logs and events from a variety of sources, including natively supported AWS services and third-party custom sources.

Security Lake runs extract, transform, and load (ETL) jobs on raw source data, and converts the data to Apache Parquet format and the OCSF schema. After processing, Security Lake stores source data in an Amazon Simple Storage Service (Amazon S3) bucket in your AWS account in the AWS Region that the data was generated in. Security Lake creates a different Amazon S3 bucket for each Region in which you enable the service. Each source gets a separate prefix in your S3 bucket, and Security Lake organizes data from each source in a separate set of AWS Lake Formation tables.

Topics

- Collecting data from AWS services in Security Lake
- Collecting data from custom sources in Security Lake

Collecting data from AWS services in Security Lake

Amazon Security Lake can collect logs and events from the following natively-supported AWS services:

- AWS CloudTrail management and data events (S3, Lambda)
- Amazon Elastic Kubernetes Service (Amazon EKS) Audit Logs
- Amazon Route 53 resolver query logs
- AWS Security Hub findings
- Amazon Virtual Private Cloud (Amazon VPC) Flow Logs
- AWS WAFv2 logs

Security Lake automatically transforms this data into the Open Cybersecurity Schema Framework (OCSF) in Security Lake and Apache Parquet format.

Tip

To add one or more of the preceding services as a log source in Security Lake, you *don't* need to separately configure logging in these services, except CloudTrail management events. If you do have logging configured in these services, you *don't* need to change your logging configuration to add them as log sources in Security Lake. Security Lake pulls data directly from these services through an independent and duplicated stream of events.

Prerequisite: Verify permissions

To add an AWS service as a source in Security Lake, you must have the necessary permissions. Verify that the AWS Identity and Access Management (IAM) policy attached to the role that you use to add a source has permission to perform the following actions:

- `glue:CreateDatabase`
- `glue:CreateTable`
- `glue:GetDatabase`
- `glue:GetTable`
- `glue:UpdateTable`
- `iam:CreateServiceLinkedRole`
- `s3:GetObject`
- `s3:PutObject`

It is recommended for the role to have the following conditions and resource scope for the `S3:getObject` and `s3:PutObject` permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUpdatingSecurityLakeS3Buckets",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::aws-security-data-lake*",
              "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
    }
    ]
}
```

These actions allow you to collect logs and events from the an AWS service and send them to the correct AWS Glue database and table.

If you use a AWS KMS key for server-side encryption of your data lake, you also need permission for `kms:DescribeKey`.

Adding an AWS service as a source

After you add an AWS service as a source, Security Lake automatically starts collecting security logs and events from it. These instructions tell you how to add a natively-supported AWS service as a source in Security Lake. For instructions on adding a custom source, see Collecting data from custom sources in Security Lake.

Console

To add an AWS log source (console)

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Choose **Sources** from the navigation pane.
3. Select the AWS service that you want to collect data from, and choose **Configure**.
4. In the **Source settings** section, enable the source and select the **Version** of data source that you want to use for data ingestion. By default, the latest version of data source is ingested by Security Lake.

> ⚠ **Important**
>
> If you don't have the required role permissions to enable the new version of the AWS log source in the specified Region, contact your Security Lake administrator. For more information, see Update role permissions.

For your subscribers to ingest the selected version of the data source, you must also update your subscriber settings. For the details on how to edit a subscriber, see Subscriber management in Amazon Security Lake.

Optionally, you can choose to ingest the latest version only and disable all previous source versions used for data ingestion.

5. In the **Regions** section, select the Regions in which you want to collect data for the source. Security Lake will collect data from the source from *all* accounts in the selected Regions.
6. Choose **Enable**.

API

To add an AWS log source (API)

To add an AWS service as a source programmatically, use the CreateAwsLogSource operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the create-aws-log-source command. The `sourceName` and `regions` parameters are required. Optionally, you can limit the scope of the source to specific `accounts` or a specific `sourceVersion`.

> ⚠ **Important**
>
> When you don't provide a parameter in your command, Security Lake assumes that the missing parameter refers to the entire set. For example, if you don't provide the `accounts` parameter , the command applies to the entire set of accounts in your organization.

The following example adds VPC Flow Logs as a source in the designated accounts and Regions. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

Note

If you apply this request to a Region in which you haven't enabled Security Lake, you'll receive an error. You can resolve the error by enabling Security Lake in that Region or by using the `regions` parameter to specify only those Regions in which you've enabled Security Lake.

```
$ aws securitylake create-aws-log-source \
--sources sourceName=VPC_FLOW,accounts='["123456789012",
 "111122223333"]',regions=["us-east-2"],sourceVersion="2.0"
```

Getting the status of source collection

Choose your access method, and follow the steps to get a snapshot of the accounts and sources for which log collection is enabled in the current Region.

Console

To get the status of log collection in the current Region

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. On the navigation pane, choose **Accounts**.
3. Hover the cursor over the number in the **Sources** column to see which logs are enabled for the selected account.

API

To get the status of log collection in the current Region, use the GetDataLakeSources operation of the Security Lake API. If you're using the AWS CLI, run the get-data-lake-sources command. For the `accounts` parameter, you can specify one or more AWS account IDs as a list. If your request succeeds, Security Lake returns a snapshot for those accounts in the current Region, including which AWS sources Security Lake is collecting data from and the status of each source.

If you don't include the `accounts` parameter, the response includes the status of log collection for all accounts in which Security Lake is configured in the current Region.

For example, the following AWS CLI command retrieves log collection status for the specified accounts in the current Region. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake get-data-lake-sources \
--accounts "123456789012" "111122223333"
```

Updating role permissions in Security Lake

If you don't have the required role permissions or resources—new AWS Lambda function and Amazon Simple Queue Service (Amazon SQS) queue—to ingest data from a new version of the data source, you must update your `AmazonSecurityLakeMetaStoreManagerV2` role permissions and create a new set of resources to process data from your sources.

Choose your preferred method, and follow the instructions to update your role permissions and create new resources to process data from a new version of an AWS log source in a specified Region. This is a one-time action, as the permissions and resources are automatically applied to future data source releases.

Console

To update role permissions (console)

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

   Sign in with the credentials of the delegated Security Lake administrator.
2. In the navigation pane, under **Settings**, choose **General**.
3. Choose **Update role permissions**.
4. In the **Service access** section, do one of the following:

   - **Create and use a new service role**— You can use the **AmazonSecurityLakeMetaStoreManagerV2** role created by Security Lake.
   - **Use an existing service role**— You can choose an existing service role from the **Service role name** list.
5. Choose **Apply**.

API

To update role permissions (API)

To update permissions programmatically, use the UpdateDataLake operation of the Security Lake API. To update permissions using the AWS CLI, run the update-data-lake command.

To update your role permissions, you must attach the AmazonSecurityLakeMetastoreManager policy to the role.

Deleting the AmazonSecurityLakeMetaStoreManager role

⚠ **Important**

After you update your role permissions to `AmazonSecurityLakeMetaStoreManagerV2`, confirm that the data lake works correctly before you remove the old `AmazonSecurityLakeMetaStoreManager` role. It is recommended to wait at-least 4 hours before removing the role.

If you decide to remove the role, you must first delete the `AmazonSecurityLakeMetaStoreManager` role from AWS Lake Formation.

Follow these steps to remove the `AmazonSecurityLakeMetaStoreManager` role from the Lake Formation console.

1. Sign in to the AWS Management Console, and open the Lake Formation console at https://console.aws.amazon.com/lakeformation/.
2. In the Lake Formation console, from the navigation pane, choose **Administrative roles and tasks**.
3. Remove `AmazonSecurityLakeMetaStoreManager` from each Region.

Removing an AWS service as a source from Security Lake

Choose your access method, and follow these steps to remove a natively-supported AWS service as a Security Lake source. You can remove a source for one or more Regions. When you remove the source, Security Lake stops collecting data from that source in the specified Regions and accounts, and subscribers can no longer consume new data from the source. However, subscribers can still consume data that Security Lake collected from the source before removal. You can only use these instructions to remove a natively-supported AWS service as a source. For information about removing a custom source, see Collecting data from custom sources in Security Lake.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Choose **Sources** from the navigation pane.
3. Select a source, and choose **Disable**.
4. Select a Region or Regions in which you want to stop collecting data from this source. Security Lake will stop collecting data from the source from *all* accounts in the selected Regions.

API

To remove an AWS service as a source programmatically, use the DeleteAwsLogSource operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the delete-aws-log-source command. The `sourceName` and `regions` parameters are required. Optionally, you can limit the scope of the removal to specific `accounts` or a specific `sourceVersion`.

> **Important**
>
> When you don't provide a parameter in your command, Security Lake assumes that the missing parameter refers to the entire set. For example, if you don't provide the `accounts` parameter , the command applies to the entire set of accounts in your organization.

The following example removes VPC Flow Logs as a source in the designated accounts and Regions.

```
$ aws securitylake delete-aws-log-source \
--sources sourceName=VPC_FLOW,accounts='["123456789012",
 "111122223333"]',regions='["us-east-1", "us-east-2"]',sourceVersion="2.0"
```

The following example removes Route 53 as a source in the designated account and Regions.

```
$ aws securitylake delete-aws-log-source \
--sources sourceName=ROUTE53,accounts='["123456789012"]',regions='["us-east-1", "us-
east-2"]',sourceVersion="2.0"
```

The preceding examples are formatted for Linux, macOS, or Unix, and they use the backslash (\) line-continuation character to improve readability.

CloudTrail event logs in Security Lake

AWS CloudTrail provides you with a history of AWS API calls for your account, including API calls made using the AWS Management Console, the AWS SDKs, the command line tools, and certain AWS services. CloudTrail also allows you to identify which users and accounts called AWS APIs for services that support CloudTrail, the source IP address that the calls were made from, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Security Lake can collect logs associated with CloudTrail management events and CloudTrail data events for S3 and Lambda. CloudTrail management events, S3 data events, and Lambda data events are three separate sources in Security Lake. As a result, they have different values for `sourceName` when you add one of these as an ingested log source. Management events, also known as control plane events, provide insight into management operations that are performed on resources in your AWS account. CloudTrail data events, also known as data plane operations, show the resource operations performed on or within resources in your AWS account. These operations are often high-volume activities.

To collect CloudTrail management events in Security Lake, you must have at least one CloudTrail multi-Region organization trail that collects read and write CloudTrail management events. Logging must be enabled for the trail. If you do have logging configured in the other services, you don't need to change your logging configuration to add them as log sources in Security Lake. Security Lake pulls data directly from these services through an independent and duplicated stream of events.

A multi-Region trail delivers log files from multiple Regions to a single Amazon Simple Storage Service (Amazon S3) bucket for a single AWS account. If you already have a multi-Region trail managed through CloudTrail console or AWS Control Tower, no further action is required.

- For information about creating and managing a trail through CloudTrail, see Creating a trail for an organization in the *AWS CloudTrail User Guide*.

- For information about creating and managing a trail through AWS Control Tower, see Logging AWS Control Tower actions with AWS CloudTrail in the *AWS Control Tower User Guide*.

When you add CloudTrail events as a source, Security Lake immediately starts collecting your CloudTrail event logs. It consumes CloudTrail management and data events directly from CloudTrail through an independent and duplicated stream of events.

Security Lake doesn't manage your CloudTrail events or affect your existing CloudTrail configurations. To manage access and retention of your CloudTrail events directly, you must use the CloudTrail service console or API. For more information, see Viewing events with CloudTrail Event history in the *AWS CloudTrail User Guide*.

The following list provides GitHub repository links to the mapping reference for how Security Lake normalizes CloudTrail events to OCSF.

GitHub OCSF repository for CloudTrail events

- Source version 1 (v1.0.0-rc.2)
- Source version 2 (v1.1.0)

Amazon EKS Audit Logs in Security Lake

When you add Amazon EKS Audit Logs as a source, Security Lake starts collecting in-depth information about the activities performed on the Kubernetes resources running in your Elastic Kubernetes Service (EKS) clusters. EKS Audit Logs help you detect potentially suspicious activities in your EKS clusters within the Amazon Elastic Kubernetes Service.

Security Lake consumes EKS Audit Log events directly from the Amazon EKS control plane logging feature through an independent and duplicative stream of audit logs. This process is designed to not require additional set up or affect existing Amazon EKS control plane logging configurations that you might have. For more information, see Amazon EKS control plane logging in the Amazon EKS User Guide.

Amazon EKS audit logs is supported only in OCSF v1.1.0. For information about how Security Lake normalizes EKS Audit Logs events to OCSF, see the mapping reference in the GitHub OCSF repository for Amazon EKS Audit Logs events (v1.1.0).

Route 53 resolver query logs in Security Lake

Route 53 resolver query logs track DNS queries made by resources within your Amazon Virtual Private Cloud (Amazon VPC). This helps you understand how your applications are operating and spot security threats.

When you add Route 53 resolver query logs as a source in Security Lake, Security Lake immediately starts collecting your resolver query logs directly from Route 53 through an independent and duplicated stream of events.

Security Lake doesn't manage your Route 53 logs or affect your existing resolver query logging configurations. To manage resolver query logs, you must use the Route 53 service console. For more information, see Managing Resolver query logging configurations in the *Amazon Route 53 Developer Guide*.

The following list provides GitHub repository links to the mapping reference for how Security Lake normalizes Route 53 logs to OCSF.

GitHub OCSF repository for Route 53 logs

- Source version 1 (v1.0.0-rc.2)
- Source version 2 (v1.1.0)

Security Hub findings in Security Lake

Security Hub findings help you understand your security posture in AWS and let you check your environment against security industry standards and best practices. Security Hub collects findings from various sources, including integrations with other AWS services, third-party product integrations, and checks against Security Hub controls. Security Hub processes findings in a standard format called AWS Security Finding Format (ASFF).

When you add Security Hub findings as a source in Security Lake, Security Lake immediately starts collecting your findings directly from Security Hub through an independent and duplicated stream of events. Security Lake also transforms the findings from ASFF to the Open Cybersecurity Schema Framework (OCSF) in Security Lake (OCSF).

Security Lake doesn't manage your Security Hub findings or affect your Security Hub settings. To manage Security Hub findings, you must use the Security Hub service console, API, or AWS CLI. For more information, see Findings in AWS Security Hub in the *AWS Security Hub User Guide* .

The following list provides GitHub repository links to the mapping reference for how Security Lake normalizes Security Hub findings to OCSF.

GitHub OCSF repository for Security Hub findings

- Source version 1 (v1.0.0-rc.2)
- Source version 2 (v1.1.0)

VPC Flow Logs in Security Lake

The VPC Flow Logs feature of Amazon VPC captures information about the IP traffic going to and from network interfaces within your environment.

When you add VPC Flow Logs as a source in Security Lake, Security Lake immediately starts collecting your VPC Flow Logs. It consumes VPC Flow Logs directly from Amazon VPC through an independent and duplicate stream of Flow Logs.

Security Lake doesn't manage your VPC Flow Logs or affect your Amazon VPC configurations. To manage your Flow Logs, you must use the Amazon VPC service console. For more information, see Work with Flow Logs in the *Amazon VPC Developer Guide*.

The following list provides GitHub repository links to the mapping reference for how Security Lake normalizes VPC Flow Logs to OCSF.

GitHub OCSF repository for VPC Flow Logs

- Source version 1 (v1.0.0-rc.2)
- Source version 2 (v1.1.0)

AWS WAF logs in Security Lake

When you add AWS WAF as a log source in Security Lake, Security Lake immediately starts collecting the logs. AWS WAF is a web application firewall that you can use to monitor web requests that your end users send to your applications and to control access to your content. Logged information includes the time that AWS WAF received a web request from your AWS resource, detailed information about the request, and details about the rules that the request matched.

Security Lake consumes AWS WAF logs directly from AWS WAF through an independent and duplicate stream of logs. This process is designed to not require additional set up or affect existing AWS WAF configurations that you might have. For more information about how you can use AWS WAF to protect your application resources, see How AWS WAF works in the AWS WAF Developer Guide.

> ⚠ **Important**
>
> If you are using Amazon CloudFront distribution as the resource type in AWS WAF, you must select US East (N.Virginia) to ingest the global logs in Security Lake.

AWS WAF logs is supported only in OCSF v1.1.0. For information about how Security Lake normalizes AWS WAF log events to OCSF, see the mapping reference in the GitHub OCSF repository for AWS WAF logs (v1.1.0).

Removing an AWS service as a source

Choose your access method, and follow these steps to remove a natively-supported AWS service as a Security Lake source. You can remove a source for one or more Regions. When you remove the source, Security Lake stops collecting data from that source in the specified Regions and accounts, and subscribers can no longer consume new data from the source. However, subscribers can still consume data that Security Lake collected from the source before removal. You can only use these instructions to remove a natively-supported AWS service as a source. For information about removing a custom source, see Collecting data from custom sources in Security Lake.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Choose **Sources** from the navigation pane.
3. Select a source, and choose **Disable**.
4. Select a Region or Regions in which you want to stop collecting data from this source. Security Lake will stop collecting data from the source from *all* accounts in the selected Regions.

API

To remove an AWS service as a source programmatically, use the DeleteAwsLogSource operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the delete-aws-log-source command. The `sourceName` and `regions` parameters are required. Optionally, you can limit the scope of the removal to specific `accounts` or a specific `sourceVersion`.

> ⚠ **Important**
>
> When you don't provide a parameter in your command, Security Lake assumes that the missing parameter refers to the entire set. For example, if you don't provide the `accounts` parameter , the command applies to the entire set of accounts in your organization.

The following example removes VPC Flow Logs as a source in the designated accounts and Regions.

```
$ aws securitylake delete-aws-log-source \
--sources sourceName=VPC_FLOW,accounts='["123456789012",
 "111122223333"]',regions='["us-east-1", "us-east-2"]',sourceVersion="2.0"
```

The following example removes Route 53 as a source in the designated account and Regions.

```
$ aws securitylake delete-aws-log-source \
--sources sourceName=ROUTE53,accounts='["123456789012"]',regions='["us-east-1", "us-
east-2"]',sourceVersion="2.0"
```

The preceding examples are formatted for Linux, macOS, or Unix, and they use the backslash (\) line-continuation character to improve readability.

Collecting data from custom sources in Security Lake

Amazon Security Lake can collect logs and events from third-party custom sources. A Security Lake custom source is a third-party service that sends security logs and events to Amazon Security

Lake. Before sending the data, the custom source must convert the logs and events to the Open Cybersecurity Schema Framework (OCSF) and meet the source requirements for Security Lake including partitioning, parquet file format and object size and rate requirements.

For each custom source, Security Lake handles the following:

- Provides a unique prefix for the source in your Amazon S3 bucket.
- Creates a role in AWS Identity and Access Management (IAM) that permits a custom source to write data to the data lake. The permissions boundary for this role is set by an AWS managed policy called `AmazonSecurityLakePermissionsBoundary`.
- Creates an AWS Lake Formation table to organize objects that the source writes to Security Lake.
- Sets up an AWS Glue crawler to partition your source data. The crawler populates the AWS Glue Data Catalog with the table. It also automatically discovers new source data and extracts schema definitions.

> **Note**
>
> You can add up to a maximum of 50 custom log sources in an account.

To add a custom source to Security Lake, it must meet the following requirements. Failure to meet these requirements could have performance impacts, and may impact analytics use cases such as querying.

- **Destination** – The custom source must be able to write data to Security Lake as a set of S3 objects underneath the prefix assigned to the source. For sources that contain multiple categories of data, you should deliver each unique Open Cybersecurity Schema Framework (OCSF) event class as a separate source. Security Lake creates an IAM role that permits the custom source to write to the specified location in your S3 bucket.
- **Format –** Each S3 object that's collected from the custom source should be formatted as an Apache Parquet file.
- **Schema –** The same OCSF event class should apply to each record within a Parquet-formatted object. Security Lake supports versions 1.x and 2.x of Parquet. Data page size should be limited to 1 MB (uncompressed). Row group size should be no larger than 256 MB (compressed). For compression within the Parquet object, zstandard is preferred.

- **Partitioning –** Objects must be partitioned by region, AWS account, eventDay. Objects should be prefixed with `source location/region=region/accountId=accountID/eventDay=yyyyMMdd/`.
- **Object size and rate –** Files sent to Security Lake should be sent in increments between 5 minutes and 1 event day. Customers may send files more often than 5 minutes if files are larger than 256MB in size. The object and size requirement is to optimize Security Lake for Query Performance. Not following the custom source requirements may have an impact on your Security Lake performance.
- **Sorting –** Within each Parquet-formatted object, records should be ordered by time to reduce the cost of querying data.

> **Note**
>
> Use the OCSF Validation tool to verify if the custom source is compatible with the `OCSF Schema`.

Partitioning requirements for ingesting custom sources in Security Lake

To facilitate efficient data processing and querying, we require meeting the partitioning and object and size requirements when adding a custom source to Security Lake:

Partitioning

Objects should be partitioned by source location, AWS Region, AWS account, and date.

- The partition data path is formatted as

  `/ext/custom-source-name/region=region/accountId=accountID/eventDay=YYYYMMDD`.

  A sample partition with example bucket name is `aws-security-data-lake-us-west-2-lake-uid/ext/custom-source-name/region=us-west-2/accountId=123456789012/eventDay=20230428/`.

The following list describes the parameters used in the S3 path partition:

- The name of the Amazon S3 bucket in which Security Lake stores your custom source data.

- `source-location` – Prefix for the custom source in your S3 bucket. Security Lake stores all S3 objects for a given source under this prefix, and the prefix is unique to the given source.
- `region` – AWS Region to which the data is uploaded. For example, you must use `US East (N. Virginia)` to upload data into your Security Lake bucket in the US East (N. Virginia) region.
- `accountId` – AWS account ID that the records in the source partition pertain to. For records pertaining to accounts outside of AWS, we recommend using a string such as `external` or `external_externalAccountId`. By adopting this naming convection, you can avoid ambiguity in naming external account IDs so that they do not conflict with AWS account IDs or external account IDs maintained by other identity management systems.
- `eventDay` – UTC timestamp of the record, truncated to hour formatted as an eight character string (`YYYYMMDD`). If records specify a different timezone in the event timestamp, you must convert the timestamp into UTC for this partition key.

Prerequisites to adding a custom source in Security Lake

When adding a custom source, Security Lake creates an IAM role that permits the source to write data to the correct location in the data lake. The name of the role follows the format `AmazonSecurityLake-Provider-{name of the custom source}-{region}`, where `region` is the AWS Region in which you're adding the custom source. Security Lake attaches a policy to the role that permits access to the data lake. If you've encrypted the data lake with a customer managed AWS KMS key, Security Lake also attaches a policy with `kms:Decrypt` and `kms:GenerateDataKey` permissions to the role. The permissions boundary for this role is set by an AWS managed policy called `AmazonSecurityLakePermissionsBoundary`.

Topics

- Verify permissions
- Create IAM role to permit write access to Security Lake bucket location (API and AWS CLI-only step)

Verify permissions

Before adding a custom source, verify that you have the permissions to perform the following actions.

To verify your permissions, use IAM to review the IAM policies that are attached to your IAM identity. Then, compare the information in those policies to the following list of actions that you must be allowed to perform to add a custom source.

- `glue:CreateCrawler`
- `glue:CreateDatabase`
- `glue:CreateTable`
- `glue:StopCrawlerSchedule`
- `iam:GetRole`
- `iam:PutRolePolicy`
- `iam:DeleteRolePolicy`
- `iam:PassRole`
- `lakeformation:RegisterResource`
- `lakeformation:GrantPermissions`
- `s3:ListBucket`
- `s3:PutObject`

These actions allow you to collect logs and events from a custom source, send them to the correct AWS Glue database and table, and store them in Amazon S3.

If you use an AWS KMS key for server-side encryption of your data lake, you also need permission for `kms:CreateGrant`, `kms:DescribeKey`, and `kms:GenerateDataKey`.

> ⚠ **Important**
>
> If you plan to use the Security Lake console to add a custom source, you can skip the next step and proceed to Adding a custom source in Security Lake. The Security Lake console offers a streamlined process for getting started, and creates all necessary IAM roles or uses existing roles on your behalf.
>
> If you plan to use Security Lake API or AWS CLI to add a custom source, continue with the next step to create an IAM role to permit write access to Security Lake bucket location.

Create IAM role to permit write access to Security Lake bucket location (API and AWS CLI-only step)

If you're using Security Lake API or AWS CLI to add a custom source, add this IAM role to grant AWS Glue permission to crawl your custom source data and identify partitions in the data. These partitions are necessary to organize your data and create and update tables in the Data Catalog.

After creating this IAM role, you will need the Amazon Resource Name (ARN) of the role in order to add a custom source.

You must attach the `arn:aws:iam::aws:policy/service-role/AWSGlueServiceRole` AWS managed policy.

To grant the necessary permissions, you must also create and embed the following inline policy in your role to permit AWS Glue crawler to read data files from the custom source and create/update the tables in AWS Glue Data Catalog.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3WriteRead",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::{{bucketName}}/*"
            ]
        }
    ]
}
```

Attach the following trust policy to permit an AWS account by using which, it can assume the role based on the external ID:

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

If the S3 bucket in the Region where you're adding the custom source is encrypted with a customer-managed AWS KMS key, you must also attach the following policy to the role and to your KMS key policy:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey"
        "kms:Decrypt"
    ],
    "Condition": {
        "StringLike": {
            "kms:EncryptionContext:aws:s3:arn": [
                "arn:aws:s3:::{{name of S3 bucket created by Security Lake}"
            ]
        }
    },
    "Resource": [
        "{{ARN of customer managed key}}"
    ]
}
```

Adding a custom source in Security Lake

After creating the IAM role to invoke the AWS Glue crawler, follow these steps to add a custom source in Security Lake.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region where you want to create the custom source.
3. Choose **Custom sources** in the navigation pane, and then choose **Create custom source**.

4. In the **Custom source details** section, enter a globally unique name for your custom source. Then, select an OCSF event class that describes the type of data that the custom source will send to Security Lake.
5. For **AWS account with permission to write data**, enter the **AWS account ID** and **External ID** of the custom source that will write logs and events to the data lake.
6. For **Service Access**, create and use a new service role or use an existing service role that gives Security Lake permission to invoke AWS Glue.
7. Choose **Create**.

API

To add a custom source programmatically, use the CreateCustomLogSource operation of the Security Lake API. Use the operation in the AWS Region where you want to create the custom source. If you're using the AWS Command Line Interface (AWS CLI), run the create-custom-log-source command.

In your request, use the supported parameters to specify configuration settings for the custom source:

- `sourceName` – Specify a name for the source. The name must be a Regionally unique value.
- `eventClasses` – Specify one or more OCSF event classes to describe the type of data that the source will send to Security Lake. For a list of OCSF event classes supported as source in Security Lake, see Open Cybersecurity Schema Framework (OCSF).
- `sourceVersion` – Optionally, specify a value to limit log collection to a specific version of custom source data.
- `crawlerConfiguration` – Specify the Amazon Resource Name (ARN) of the IAM role that you created to invoke the AWS Glue crawler. For the detailed steps to create an IAM role, see Prerequisites to adding a custom source
- `providerIdentity` – Specify the AWS identity and external ID that the source will use to write logs and events to the data lake.

The following example adds a custom source as a log source in the designated log provider account in designated Regions. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake create-custom-log-source \
--source-name EXAMPLE_CUSTOM_SOURCE \
--event-classes '["DNS_ACTIVITY", "NETWORK_ACTIVITY"]' \
--configuration crawlerConfiguration={"roleArn=arn:aws:iam::XXX:role/service-role/
RoleName"},providerIdentity={"externalId=ExternalId,principal=principal"}  \
--region=[“ap-southeast-2”]
```

Keeping custom source data updated in AWS Glue

After you add a custom source in Security Lake, Security Lake creates an AWS Glue crawler. The crawler connects to your custom source, determines the data structures, and populates the AWS Glue Data Catalog with tables.

We recommend manually running the crawler to keep your custom source schema up to date and maintain query functionality in Athena and other querying services. Specifically, you should run the crawler if either of the following changes occur in your input data set for a custom source:

- The data set has one or more new top-level columns.
- The data set has one or more new fields in a column with a `struct` datatype.

For instructions on running a crawler, see Scheduling an AWS Glue crawler in the *AWS Glue Developer Guide*.

Security Lake can't delete or update existing crawlers in your account. If you delete a custom source, we recommend deleting the associated crawler if you plan to create a custom source with the same name in the future.

Deleting a custom source from Security Lake

Delete a custom source to stop sending data from the source to Security Lake. When you remove the source, Security Lake stops collecting data from that source in the specified Regions and accounts, and subscribers can no longer consume new data from the source. However, subscribers can still consume data that Security Lake collected from the source before removal. You can only use these instructions to remove a custom source. For information about removing a natively-supported AWS service, see Collecting data from AWS services in Security Lake.

When deleting a custom source in Security Lake, you must disable each source outside of the Security Lake console with the source. Failure to disable an integration may result in source integrations continuing to send logs into Amazon S3.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region that you want to remove the custom source from.
3. In the navigation pane, choose **Custom sources**.
4. Select the custom source that you want to remove.
5. Choose **Deregister custom source** and then choose **Delete** to confirm the action.

API

To delete a custom source programmatically, use the DeleteCustomLogSource operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the delete-custom-log-source command. Use the operation in the AWS Region where you want to delete the custom source.

In your request, use the `sourceName` parameter to specify the name of the custom source to delete. Or specify the name of the custom source and use the `sourceVersion` parameter to limit the scope of the deletion to only a specific version of data from the custom source.

The following example deletes a custom log source from Security Lake.

This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake delete-custom-log-source \
--source-name EXAMPLE_CUSTOM_SOURCE
```

Subscriber management in Security Lake

An Amazon Security Lake subscriber consumes logs and events from Security Lake. To control costs and adhere to least privilege access best practices, you provide subscribers access to data on a per-source basis. For more information about sources, see Source management in Security Lake.

Security Lake supports two types of subscriber access:

- **Data access** Subscribers with data access to source data in Amazon Security Lake are notified of new objects for the source as the data is written to the S3 bucket. By default, subscribers are notified about new objects through an HTTPS endpoint that they provide. Alternatively, subscribers can be notified about new objects by polling an Amazon Simple Queue Service (Amazon SQS) queue.
- **Query access** – Subscribers with query access can query data that Security Lake collects. These subscribers directly query AWS Lake Formation tables in your S3 bucket with services like Amazon Athena.

Subscribers only have access to the source data in the AWS Region that you select when you create the subscriber. To give a subscriber access to data from multiple Regions, you can specify the Region where you create the subscriber as a rollup Region and have other Regions contribute data to it. For more information about rollup Regions and contributing Regions, see Managing Regions in Security Lake.

> ⚠ **Important**
>
> The maximum number of sources that Security Lake allows to add per subscriber is 10. This could be a combination of AWS sources and custom sources.

Topics

- Managing data access for Security Lake subscribers
- Managing query access for Security Lake subscribers

Managing data access for Security Lake subscribers

Subscribers with data access to source data in Amazon Security Lake are notified of new objects for the source as the data is written to the S3 bucket. By default, subscribers are notified about new objects through an HTTPS endpoint that they provide. Alternatively, subscribers can be notified about new objects by polling an Amazon Simple Queue Service (Amazon SQS) queue.

Subscribers are notified of new Amazon S3 objects for a source as the objects are written to the Security Lake data lake. Subscribers can directly access the S3 objects and receive notifications of new objects through a subscription endpoint or by polling an Amazon Simple Queue Service (Amazon SQS) queue. This subscription type is identified as `S3` in the `accessTypes` parameter of the CreateSubscriber API.

Topics

- Prerequisites to create a subscriber with data access in Security Lake
- Creating a subscriber with data access in Security Lake
- Updating a data subscriber in Security Lake
- Removing a data subscriber from Security Lake

Prerequisites to create a subscriber with data access in Security Lake

You must complete the following prerequisites before you can create a subscriber with data access in Security Lake.

Verify permissions

To verify your permissions, use IAM to review the IAM policies that are attached to your IAM identity. Then, compare the information in those policies to the following list of (permissions) actions that you must have to notify subscribers when new data is written to the data lake.

You will need permission to perform the following actions:

- `iam:CreateRole`
- `iam:DeleteRolePolicy`
- `iam:GetRole`
- `iam:PutRolePolicy`
- `lakeformation:GrantPermissions`

- `lakeformation:ListPermissions`
- `lakeformation:RegisterResource`
- `lakeformation:RevokePermissions`
- `ram:GetResourceShareAssociations`
- `ram:GetResourceShares`
- `ram:UpdateResourceShare`

In addition to the preceding list, you also need permission to perform the following actions:

- `events:CreateApiDestination`
- `events:CreateConnection`
- `events:DescribeRule`
- `events:ListApiDestinations`
- `events:ListConnections`
- `events:PutRule`
- `events:PutTargets`
- `s3:GetBucketNotification`
- `s3:PutBucketNotification`
- `sqs:CreateQueue`
- `sqs:DeleteQueue`
- `sqs:GetQueueAttributes`
- `sqs:GetQueueUrl`
- `sqs:SetQueueAttributes`

Get the subscriber's external ID

To create a subscriber, apart from the subscriber's AWS account ID, you will also need to get their *external ID*. The external ID is a unique identifier that the subscriber provides to you. Security Lake adds the external ID to the subscriber IAM role that it creates. You use the external ID when you create a subscriber in the Security Lake console, through the API, or AWS CLI.

For more information about external IDs, see How to use an external ID when granting access to your AWS resources to a third party in the *IAM User Guide*.

Important

If you plan to use the Security Lake console to add a subscriber, you can skip the next step and proceed to Creating a subscriber with data access in Security Lake. The Security Lake console offers a streamlined process for getting started, and creates all necessary IAM roles or uses existing roles on your behalf.

If you plan to use Security Lake API or AWS CLI to add a subscriber, continue with the next step to create an IAM role to invoke EventBridge API destinations.

Create IAM role to invoke EventBridge API destinations (API and AWS CLI-only step)

If you're using Security Lake through API or AWS CLI, create a role in AWS Identity and Access Management (IAM) that grants Amazon EventBridge permissions to invoke API destinations and send object notifications to the correct HTTPS endpoints.

After creating this IAM role, you'll need the Amazon Resource Name (ARN) of the role in order to create the subscriber. This IAM role isn't necessary if the subscriber polls data from an Amazon Simple Queue Service (Amazon SQS) queue or directly queries data from AWS Lake Formation. For more information about this type of data access method (access type), see Managing query access for Security Lake subscribers.

Attach the following policy to your IAM role:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowInvokeApiDestination",
            "Effect": "Allow",
            "Action": [
                "events:InvokeApiDestination"
            ],
            "Resource": [
                "arn:aws:events:{us-west-2}:{123456789012}:api-destination/
AmazonSecurityLake*/*"
            ]
        }
    ]
}
```

Attach the following trust policy to your IAM role to permit EventBridge to assume the role:

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

Security Lake automatically creates an IAM role that permits the subscriber to read data from the data lake (or poll events from an Amazon SQS queue if that's the preferred method of notification). This role is protected with an AWS managed policy called `AmazonSecurityLakePermissionsBoundary`.

Creating a subscriber with data access in Security Lake

Choose one of the following access methods to create a subscriber with access to data in the current AWS Region.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region where you want to create the subscriber.
3. In the navigation pane, choose **Subscribers**.
4. On the **Subscribers** page, choose **Create subscriber**.
5. For **Subscriber details**, enter **Subscriber name** and an optional **Description**.

   The **Region** is auto-populated as your currently selected AWS Region and can't be modified.
6. For **Log and event sources**, choose which sources the subscriber is authorized to consume.

7. For **Data access method**, choose **S3** to set up data access for the subscriber.
8. For **Subscriber credentials**, provide the subscriber's AWS account ID and external ID.
9. (Optional) For **Notification details**, if you want Security Lake to create an Amazon SQS queue that the subscriber can poll for object notifications, select **SQS queue**. If you want Security Lake to send notifications through EventBridge to an HTTPS endpoint, select **Subscription endpoint**.

   If you select **Subscription endpoint**, also do the following:

   a. Enter the **Subscription endpoint**. Examples of valid endpoint formats include **`http://example.com`**. Optionally, you can also provide an **HTTPS key name** and **HTTPS key value**.
   b. For **Service Access**, create a new IAM role or use an existing IAM role that gives EventBridge permission to invoke API destinations and send object notifications to the correct endpoints.

      For information about creating a new IAM role, see Create IAM role to invoke EventBridge API destinations.
10. (Optional) For **Tags**, enter as many as 50 tags to assign to the subscriber.

    A *tag* is a label that you can define and assign to certain types of AWS resources. Each tag consists of a required tag key and an optional tag value. Tags can help you identify, categorize, and manage resources in different ways. To learn more, see Tagging Security Lake resources.
11. Choose **Create**.

API

To create a subscriber with data access programmatically, use the CreateSubscriber operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the create-subscriber command.

In your request, use these parameters to specify the following settings for the subscriber:

- For `sources`, specify each source that you want the subscriber to access.
- For `subscriberIdentity`, specify the AWS account ID and external ID that the subscriber will use to access source data.

- For `subscriber-name`, specify the name of the subscriber.
- For `accessTypes`, specify S3.

Example 1

The following example creates a subscriber with access to data in the current AWS Region for the specified subscriber identity for an AWS source.

```
$ aws securitylake create-subscriber \
--subscriber-identity {"accountID": 1293456789123,"externalId": 123456789012} \
--sources [{"awsLogSource": {"sourceName": VPC_FLOW, "sourceVersion": 2.0}}] \
--subscriber-name subscriber name \
--access-types S3
```

Example 2

The following example creates a subscriber with access to data in the current AWS Region for the specified subscriber identity for a custom source.

```
$ aws securitylake create-subscriber \
--subscriber-identity {"accountID": 1293456789123,"externalId": 123456789012} \
--sources [{"customLogSource": {"sourceName": custom-source-name,
 "sourceVersion": 2.0}}] \
--subscriber-name subscriber name
--access-types S3
```

The preceding examples are formatted for Linux, macOS, or Unix, and they use the backslash (\) line-continuation character to improve readability.

(Optional) After you create a subscriber, use the CreateSubscriberNotification operation to specify how to notify the subscriber when new data is written to the data lake for the sources that you want the subscriber to access. If you're using the AWS Command Line Interface (AWS CLI), run the create-subscriber-notification command.

- To override the default notification method (HTTPS endpoint) and create an Amazon SQS queue, specify values for the `sqsNotificationConfiguration` parameters.
- If you prefer notification with an HTTPS endpoint, specify values for the `httpsNotificationConfiguration` parameters.

- For the `targetRoleArn` field, specify the ARN of the IAM role that you created to invoke EventBridge API destinations.

```
$ aws securitylake create-subscriber-notification \
--subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \
--configuration
 httpsNotificationConfiguration={"targetRoleArn"="arn:aws:iam::XXX:role/service-
role/RoleName", "endpoint"="https://account-management.$3.$2.securitylake.aws.dev/
v1/datalake"}
```

To get the `subscriberID`, use the ListSubscribers operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the list-subscriber command.

```
$ aws securitylake list-subscribers
```

To subsequently change the notification method (Amazon SQS queue or HTTPS endpoint) for the subscriber, use the UpdateSubscriberNotification operation or, if you're using the AWS CLI, run the update-subscriber-notification command. You can also change the notification method by using the Security Lake console: select the subscriber on the **Subscribers** page, and then choose **Edit**.

Sample object notification message

The following example shows the event notification in JSON structure format for the `CreateSubscriberNotification` operation.

```
{
  "source": "aws.s3",
  "time": "2021-11-12T00:00:00Z",
  "account": "123456789012",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket"
  ],
  "detail": {
    "bucket": {
      "name": "amzn-s3-demo-bucket"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b57f9512698f4b09e608f4f2a65852e5"
    },
    "request-id": "N4N7GDK58NMKJ12R",
    "requester": "securitylake.amazonaws.com"
  }
}
```

Updating a data subscriber in Security Lake

You can update a subscriber by changing the sources from which the subscriber consumes. You can also assign or edit the tags for a subscriber. A *tag* is a label that you can define and assign to certain types of AWS resources, including subscribers. To learn more, see Tagging Security Lake resources.

Choose one of the access methods, and follow these steps to define new sources for an existing subscription.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. In the navigation pane, choose **Subscribers**.
3. Select the subscriber.
4. Choose **Edit**, and then do any of the following:
   - To update the sources for the subscriber, enter the new settings in the **Log and event sources** section.
   - To assign or edit tags for the subscriber, change the tags as necessary in the **Tags** section.
5. When you finish, choose **Save**.

API

To update data access sources for a subscriber programmatically, use the UpdateSubscriber operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the update-subscriber command. In your request, use the `sources` parameters to specify each source that you want the subscriber to access.

```
$ aws securitylake update-subscriber --subscriber-id subscriber ID
```

For a list of subscribers associated with a specific AWS account or organization, use the ListSubscribers operation. If you're using the AWS Command Line Interface (AWS CLI), run the list-subscribers command.

```
$ aws securitylake list-subscribers
```

To review the current settings for a particular subscriber, use the GetSubscriber operation. run the get-subscriber command. Security Lake then returns the subscriber's name and description, external ID, and additional information. If you're using the AWS Command Line Interface (AWS CLI), run the get-subscriber command.

To update the notification method for a subscriber, use the UpdateSubscriberNotification operation. If you're using the AWS Command Line Interface (AWS CLI), run the update-subscriber-notification command. For example, you can specify a new HTTPS endpoint for the subscriber or switch from an HTTPS endpoint to an Amazon SQS queue.

Removing a data subscriber from Security Lake

If you no longer want a subscriber to consume data from Security Lake, you can remove the subscriber by following these steps.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. In the navigation pane, choose **Subscribers**.
3. Select the subscriber that you want to remove.
4. Choose **Delete** and confirm the action. This will delete the subscriber and all the associated notification settings.

API

Based on your scenario, do one of the following:

- To delete the subscriber and all associated notification settings, use the DeleteSubscriber operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the delete-subscriber command.

- To retain the subscriber but stop future notifications to the subscriber, use the DeleteSubscriberNotification operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the run the delete-subscriber-notification command.

Managing query access for Security Lake subscribers

Subscribers with query access can query data that Security Lake collects. These subscribers directly query AWS Lake Formation tables in your S3 bucket with services like Amazon Athena. Although the primary query engine for Security Lake is Athena you can also use other services, such as Amazon Redshift Spectrum and Spark SQL, that integrate with the AWS Glue Data Catalog.

Subscribers query source data from AWS Lake Formation tables in your S3 bucket by using services like Amazon Athena. This subscription type is identified as `LAKEFORMATION` in the `accessTypes` parameter of the CreateSubscriber API.

> **Note**
>
> This section explains how to grant query access to a third-party subscriber. For information about running queries against your own data lake, see Step 4: View and query your own data.

Topics

Prerequisites to create a subscriber with query access in Security Lake

You must complete the following prerequisites before you can create a subscriber with data access in Security Lake.

Verify permissions

Before creating a subscriber with query access, verify that you have permission to perform the following list of actions.

To verify your permissions, use IAM to review the IAM policies that are attached to your IAM identity. Then, compare the information in those policies to the following list of actions that you must be allowed to perform to create a subscriber with query access.

- `iam:CreateRole`
- `iam:DeleteRolePolicy`
- `iam:GetRole`
- `iam:PutRolePolicy`
- `lakeformation:GrantPermissions`
- `lakeformation:ListPermissions`
- `lakeformation:RegisterResource`
- `lakeformation:RevokePermissions`
- `ram:GetResourceShareAssociations`
- `ram:GetResourceShares`
- `ram:UpdateResourceShare`

> ⚠ **Important**
>
> After you have verified the permissions:
>
> - If you plan to use Security Lake console to add a subscriber with query access, you can skip the next step and proceed to Grant Lake Formation administrator permissions. Security Lake creates all the necessary IAM roles or uses existing roles on your behalf.
> - If you plan to use Security Lake API or CLI to add a subscriber with query access, continue with the next step to create an IAM role to query Security Lake data.

Create IAM role to query Security Lake data (API and AWS CLI-only step)

When using Security Lake API or AWS CLI to grant query access to a subscriber, you'll need to create a role named `AmazonSecurityLakeMetaStoreManager`. Security Lake uses this role to register AWS Glue partitions and update AWS Glue tables. You may have already created this role while Create necessary IAM roles.

Grant Lake Formation administrator permissions

You'll also need to add Lake Formation administrator permissions to the IAM role that you use to access the Security Lake console and add subscribers.

You can grant Lake Formation administrator permissions to your role by following these steps:

1. Open the Lake Formation console at https://console.aws.amazon.com/lakeformation/.
2. Sign in as an administrative user.
3. If a **Welcome to Lake Formation** window appears, choose the user that you created or selected in Step 1, and then choose Get started.
4. If you don't see a **Welcome to Lake Formation** window, then perform the following steps to configure a Lake Formation Administrator.

    1. In the navigation pane, under **Permissions**, choose **Administrative roles and tasks**. In the **Data lake administrators** section, choose **Choose administrators**.
    2. In the **Manage data lake administrators** dialog box, for IAM users and roles, choose the administrator role used when accessing the Security Lake console, and then choose **Save**.

For more information about changing permissions for data lake administrators, see Create a data lake administrator in the *AWS Lake Formation Developer Guide*.

The IAM role must have SELECT privileges on the database and tables that you want to grant a subscriber access to. For instructions on how to do this, see Granting Data Catalog permissions using the named resource method in the *AWS Lake Formation Developer Guide*.

Creating a subscriber with query access in Security Lake

Choose your preferred method to create a subscriber with query access in the current AWS Region. A subscriber can query data only from the AWS Region that it is created in. To create a subscriber, you'll need to have the AWS account ID and external ID of the subscriber. The external ID is a unique identifier that the subscriber provides to you. For more information about external IDs, see How to use an external ID when granting access to your AWS resources to a third party in the *IAM User Guide*.

> **Note**
>
> Security Lake does not support Lake Formation cross-account data sharing version 1. You must update Lake Formation cross-account data sharing to version 2 or version 3. For the steps to update **Cross account version settings** through the AWS Lake Formation console or the AWS CLI, see To enable the new version in the *AWS Lake Formation Developer Guide*.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

   Sign in to the delegated administrator account.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region where you want to create the subscriber.
3. In the navigation pane, choose **Subscribers**.
4. On the **Subscribers** page, choose **Create subscriber**.
5. For **Subscriber details**, enter a **Subscriber name** and an optional **Description**.

   The **Region** is auto-populated as your currently selected AWS Region and can't be modified.
6. For **Log and event sources**, choose which sources you want Security Lake to include when returning query results.
7. For **Data access method**, choose **Lake Formation** to create query access for the subscriber.
8. For **Subscriber credentials**, provide the subscriber's AWS account ID and external ID.
9. (Optional) For **Tags**, enter as many as 50 tags to assign to the subscriber.

   A *tag* is a label that you can define and assign to certain types of AWS resources. Each tag consists of a required tag key and an optional tag value. Tags can help you identify, categorize, and manage resources in different ways. To learn more, see Tagging Security Lake resources.
10. Choose **Create**.

API

To create a subscriber with query access programmatically, use the CreateSubscriber operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the create-subscriber command.

In your request, use these parameters to specify the following settings for the subscriber:

- For `accessTypes`, specify `LAKEFORMATION`.
- For `sources`, specify each source that you want Security Lake to include when returning query results.
- For `subscriberIdentity`, specify the AWS identity and external ID that the subscriber uses to query source data.

The following example creates a subscriber with query access in the current AWS Region for the specified subscriber identity. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake create-subscriber \
--subscriber-identity {"accountID": 129345678912,"externalId": 123456789012} \
--sources [{"awsLogSource": {"sourceName": VPC_FLOW, "sourceVersion": 2.0}}] \
--subscriber-name subscriber name \
--access-types LAKEFORMATION
```

Setting up cross-account table sharing (subscriber step)

Security Lake uses Lake Formation cross-account table sharing to support subscriber query access. When you create a subscriber with query access in the Security Lake console, API, or AWS CLI, Security Lake shares information about the relevant Lake Formation tables with the subscriber by creating a resource share in AWS Resource Access Manager (AWS RAM).

When you make certain types of edits to a subscriber with query access, Security Lake creates a new resource share. For more information, see Editing a subscriber with query access in Security Lake.

The subscriber should follow these steps to consume data from your Lake Formation tables:

1. **Accept the resource share** – The subscriber must accept the resource share that has the `resourceShareArn` and `resourceShareName` that's generated when you create or edit the subscriber. Choose one of the following access methods:

   - For console and AWS CLI, see Accepting a resource share invitation from AWS RAM.
   - For API, invoke the GetResourceShareInvitations API. Filter by `resourceShareArn` and `resourceShareName` to find the correct resource share. Accept the invitation with the AcceptResourceShareInvitation API.

   The resource share invitation expires in 12 hours, so you must validate and accept the invitation within 12 hours. If the invitation expires, you continue to see it in a `PENDING` state, but accepting it won't give you access to the shared resources. When more than 12 hours have passed, delete the Lake Formation subscriber and recreate the subscriber to get a new resource share invitation.

2. **Create a resource link to the shared database** – The subscriber must create a resource link to the shared Lake Formation database in either AWS Lake Formation (if using the console) or AWS Glue (if using API/AWS CLI). This resource link points the subscriber's account to the shared database. Choose one of the following access methods:

   - For console and AWS CLI, see see Creating a resource link to a shared Data Catalog database. in the *AWS Lake Formation Developer Guide*.
   - We recommend that subscribers also create a unique database with the CreateDatabase API to store resource link tables.

3. **Query the shared tables** – Services like Amazon Athena can refer to the tables directly, and new data that Security Lake collects is automatically available to query. Queries run in the subscriber's AWS account, and costs incurred from queries are billed to the subscriber. You can control read access to resources in your own Security Lake account.

For more information about granting cross-account permissions, see Cross-account data sharing in Lake Formation in the *AWS Lake Formation Developer Guide*.

Editing a subscriber with query access in Security Lake

Security Lake supports making edits to a subscriber with query access. You can edit the subscriber's name, description, external ID, principal (AWS account ID), and the log sources that the subscriber is able to consume. Choose your preferred method, and follow the steps to edit a subscriber with query access in the current AWS Region.

Note

Security Lake does not support Lake Formation cross-account data sharing version 1. You must update Lake Formation cross-account data sharing to version 2 or version 3. For the steps to update **Cross account version settings** through the AWS Lake Formation console or the AWS CLI, see To enable the new version in the *AWS Lake Formation Developer Guide*.

Console

Based on the details that you want to edit, follow the steps provided for that action only.

To edit subscriber name

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

   Sign in to the delegated administrator account.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region where you want to edit the subscriber details.
3. In the navigation pane, choose **Subscribers**.
4. On the **Subscribers** page, use the radio button to select the subscriber that you want to edit. The **Data access method** for the selected subscriber must be **LAKEFORMATION**.
5. Choose **Edit**.
6. Enter the new **Subscriber name**, and choose **Save**.

To edit subscriber description

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

   Sign in to the delegated administrator account.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region where you want to edit the subscriber.
3. In the navigation pane, choose **Subscribers**.
4. On the **Subscribers** page, use the radio button to select the subscriber that you want to edit. The **Data access method** for the selected subscriber must be **LAKEFORMATION**.
5. Choose **Edit**.

6. Enter the new description for the subscriber, and choose **Save**.

To edit external ID

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

   Sign in to the delegated administrator account.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region where you want to edit the subscriber details.
3. In the navigation pane, choose **Subscribers**.
4. On the **Subscribers** page, use the radio button to select the subscriber that you want to edit. The **Data access method** for the selected subscriber must be **LAKEFORMATION**.
5. Choose **Edit**.
6. Enter the new **External ID** that the subscriber has provided, and choose **Save**.

   Saving the new external ID automatically removes the previous AWS RAM resource share and creates a new resource share for the subscriber.
7. The subscriber must accept the new resource share by following step 1 in Setting up cross-account table sharing (subscriber step). Make sure the Amazon Resource Name (ARN) that appears in subscriber details is the same as in the Lake Formation console. The resource link to the shared tables remains as is, so the subscriber doesn't have to create a new resource link.

To edit principal (AWS account ID)

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

   Sign in to the delegated administrator account.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region where you want to edit the subscriber details.
3. In the navigation pane, choose **Subscribers**.
4. On the **Subscribers** page, use the radio button to select the subscriber that you want to edit. The **Data access method** for the selected subscriber must be **LAKEFORMATION**.
5. Choose **Edit**.
6. Enter the new **AWS account ID** of the subscriber, and choose **Save**.

Saving the new account ID automatically removes the previous AWS RAM resource share so the previous principal can't consume the log and event sources. Security Lake creates a new resource share.

7. Using the credentials of the new principal, the subscriber must accept the new resource share and create a resource link to the shared tables. This gives the new principal access to the shared resources. For instructions, see steps 1 and 2 in Setting up cross-account table sharing (subscriber step). Make sure the ARN that appears in the subscriber details is the same as in the Lake Formation console.

To edit log and event sources

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

   Sign in to the delegated administrator account.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region where you want to edit the subscriber details.
3. In the navigation pane, choose **Subscribers**.
4. On the **Subscribers** page, use the radio button to select the subscriber that you want to edit. The **Data access method** for the selected subscriber must be **LAKEFORMATION**.
5. Choose **Edit**.
6. Deselect existing sources or select sources that you want to add. If you deselect a source, no further action is required from your end. If you select to add a source, no new resource share invitation is created. However, Security Lake updates the shared Lake Formation tables based on the added sources. The subscriber must create a resource link to the updated shared tables so that they can query the source data. For instructions, see step 2 in Setting up cross-account table sharing (subscriber step).
7. Choose **Save**.

API

To edit a subscriber with query access programmatically, use the UpdateSubscriber operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the update-subscriber command. In your request, use the supported parameters to specify the following settings for the subscriber:

- For `subscriberName`, specify the new subscriber name.
- For `subscriberDescription`, specify the new description.
- For `subscriberIdentity`, specify the principal (AWS account ID) and external ID that the subscriber will use to query source data. You must provide both the principal and external ID. If you want to keep one of these values the same, pass in the current value.
    - **Updating only external ID –** This action removes the previous AWS RAM resource share and creates a new resource share for the subscriber. The subscriber must accept the new resource share by following step 1 in Setting up cross-account table sharing (subscriber step). The resource link to the shared tables remains as is, so the subscriber doesn't have to create a new resource link.
    - **Updating only principal –** This action removes the previous AWS RAM resource share so the previous principal can't consume the log and event sources. Security Lake creates a new resource share. Using the credentials of the new principal, the subscriber must accept the new resource share and create a resource link to the shared tables. This gives the new principal access to the shared resources. For instructions, see steps 1 and 2 in Setting up cross-account table sharing (subscriber step).

  To update the external ID *and* principal, follow steps 1 and 2 in Setting up cross-account table sharing (subscriber step).
- For `sources`, remove existing sources or specify sources that you want to add. If you remove a source, no further action is required from your end. If you add a source, no new resource share invitation is created. However, Security Lake updates the shared Lake Formation tables based on the added sources. The subscriber must create a resource link to the updated shared tables so that they can query the source data. For instructions, see step 2 in Setting up cross-account table sharing (subscriber step).

Security Lake queries

You can query the data that Security Lake stores in AWS Lake Formation databases and tables. You can also create third-party subscribers in the Security Lake console, API, or AWS CLI. Third-party subscribers can also query Lake Formation data from the sources that you specify.

The Lake Formation data lake administrator must grant SELECT permissions on the relevant databases and tables to the IAM identity that queries the data. A subscriber must also be created in Security Lake before it can query data. For more information about how to create a subscriber with query access, see Managing query access for Security Lake subscribers.

Topics

Security Lake queries for AWS source version 1 (OCSF 1.0.0-rc.2)

The following section provides guidance on querying data from Security Lake and includes some query examples for natively-supported AWS sources for AWS source version 1. These queries are designed to retrieve data in a specific AWS Region. These examples use us-east-1 (US East (N. Virginia)). In addition, the example queries use a LIMIT 25 parameter, which returns up to 25 records. You can omit this parameter or adjust it based on your preferences. For more examples, see the Amazon Security Lake OCSF Queries GitHub directory.

Log source table

When you query Security Lake data, you must include the name of the Lake Formation table in which the data resides.

```
SELECT *
   FROM
 amazon_security_lake_glue_db_DB_Region.amazon_security_lake_table_DB_Region_SECURITY_LAKE_TABL
   WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
```

```
   LIMIT 25
```

Common values for the log source table include the following:

- `cloud_trail_mgmt_1_0` – AWS CloudTrail management events
- `lambda_execution_1_0` – CloudTrail data events for Lambda
- `s3_data_1_0` – CloudTrail data events for S3
- `route53_1_0` – Amazon Route 53 resolver query logs
- `sh_findings_1_0` – AWS Security Hub findings
- `vpc_flow_1_0` – Amazon Virtual Private Cloud (Amazon VPC) Flow Logs

Example: All Security Hub findings in table `sh_findings_1_0` from us-east-1 Region

```
SELECT *
   FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
   WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
   LIMIT 25
```

Database Region

When you query Security Lake data, you must include the name of the database Region from which you're querying the data. For a complete list of database Regions where Security Lake is currently available, see [Amazon Security Lake endpoints](#).

Example: List AWS CloudTrail activity from source IP

The following example lists all the CloudTrail activities from the source IP *`192.0.2.1`* that were recorded after *`20230301`* (March 01, 2023), in the table *`cloud_trail_mgmt_1_0`* from the *`us-east-1`* DB_Region.

```
SELECT *
```

```
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay > '20230301' AND src_endpoint.ip = '192.0.2.1'
    ORDER BY time desc
    LIMIT 25
```

Partition date

By partitioning your data, you can restrict the amount of data scanned by each query, thereby improving performance and reducing cost. Security Lake implements partitioning through `eventDay`, `region`, and `accountid` parameters. `eventDay` partitions use the format YYYYMMDD.

This is an example query using the `eventDay` partition:

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay > '20230301'
    AND src_endpoint.ip = '192.0.2.1'
    ORDER BY time desc
```

Common values for `eventDay` include the following:

Events occurring in the last 1 year

```
> cast(date_format(current_timestamp - INTERVAL '1' year, '%Y%m%d%H') as varchar)
```

Events occurring in the last 1 month

```
> cast(date_format(current_timestamp - INTERVAL '1' month, '%Y%m%d%H') as varchar)
```

Events occurring in the last 30 days

```
> cast(date_format(current_timestamp - INTERVAL '30' day, '%Y%m%d%H') as varchar)
```

Events occurring in the last 12 hours

```
> cast(date_format(current_timestamp - INTERVAL '12' hour, '%Y%m%d%H')
as varchar)
```

Events occurring in the last 5 minutes

```
> cast(date_format(current_timestamp - INTERVAL '5' minute, '%Y%m%d%H')
as varchar)
```

Events occurring between 7–14 days ago

```
BETWEEN cast(date_format(current_timestamp - INTERVAL '14' day, '%Y%m%d
%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '7'
day, '%Y%m%d%H') as varchar)
```

Events occurring on or after a specific date

```
>= '20230301'
```

Example: List of all CloudTrail activity from source IP `192.0.2.1` on or after March 1, 2023 in table `cloud_trail_mgmt_1_0`

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay >= '20230301'
    AND src_endpoint.ip = '192.0.2.1'
    ORDER BY time desc
    LIMIT 25
```

Example: List of all CloudTrail activity from source IP `192.0.2.1` in the last 30 days in table `cloud_trail_mgmt_1_0`

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay > cast(date_format(current_timestamp - INTERVAL '30' day, '%Y%m%d
%H') as varchar)
```

```
    AND src_endpoint.ip = '192.0.2.1'
    ORDER BY time desc
    LIMIT 25
```

Example Security Lake queries for CloudTrail data

AWS CloudTrail tracks user activity and API usage in AWS services. Subscribers can query CloudTrail data to learn the following types of information:

Here are some example queries of CloudTrail data for AWS source version 1:

Unauthorized attempts against AWS services in the last 7 days

```
SELECT
      time,
      api.service.name,
      api.operation,
      api.response.error,
      api.response.message,
      unmapped['responseElements'],
      cloud.region,
      actor.user.uuid,
      src_endpoint.ip,
      http_request.user_agent
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND api.response.error in (
        'Client.UnauthorizedOperation',
        'Client.InvalidPermission.NotFound',
        'Client.OperationNotPermitted',
        'AccessDenied')
    ORDER BY time desc
    LIMIT 25
```

List of all CloudTrail activity from source IP `192.0.2.1` in the last 7 days

```
SELECT
      api.request.uid,
      time,
      api.service.name,
      api.operation,
      cloud.region,
      actor.user.uuid,
      src_endpoint.ip,
      http_request.user_agent
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
    AND src_endpoint.ip = '127.0.0.1.'
    ORDER BY time desc
    LIMIT 25
```

List of all IAM activity in the last 7 days

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND api.service.name = 'iam.amazonaws.com'
    ORDER BY time desc
    LIMIT 25
```

Instances where the credential `AIDACKCEVSQ6C2EXAMPLE` was used in the last 7 days

```
SELECT
      actor.user.uid,
      actor.user.uuid,
      actor.user.account_uid,
      cloud.region
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND actor.user.credential_uid = 'AIDACKCEVSQ6C2EXAMPLE'
      LIMIT 25
```

List of failed CloudTrail records in the last 7 days

```
SELECT
      actor.user.uid,
      actor.user.uuid,
      actor.user.account_uid,
      cloud.region
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE status='failed' and eventDay BETWEEN cast(date_format(current_timestamp -
 INTERVAL '7' day, '%Y%m%d%H') as varchar) and cast(date_format(current_timestamp -
 INTERVAL '0' day, '%Y%m%d%H') as varchar)
    ORDER BY time DESC
    LIMIT 25
```

Example Security Lake queries for Route 53 resolver query logs

Amazon Route 53 resolver query logs track DNS queries made by resources within your Amazon VPC. Subscribers can query Route 53 resolver query logs to learn the following types of information:

Here are some example queries of Route 53 resolver query logs for AWS source version 1:

List of DNS queries from CloudTrail in the last 7 days

```
SELECT
      time,
      src_endpoint.instance_uid,
      src_endpoint.ip,
      src_endpoint.port,
      query.hostname,
      rcode
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_route53_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
    ORDER BY time DESC
    LIMIT 25
```

List of DNS queries that match `s3.amazonaws.com` in the last 7 days

```
SELECT
      time,
      src_endpoint.instance_uid,
      src_endpoint.ip,
      src_endpoint.port,
      query.hostname,
      rcode,
      answers
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_route53_1_0
    WHERE query.hostname LIKE 's3.amazonaws.com.' and eventDay BETWEEN
 cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m%d%H') as varchar) and
 cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d%H') as varchar)
    ORDER BY time DESC
    LIMIT 25
```

List of DNS queries that didn't resolve in the last 7 days

```
SELECT
      time,
      src_endpoint.instance_uid,
      src_endpoint.ip,
      src_endpoint.port,
      query.hostname,
      rcode,
      answers
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_route53_1_0
    WHERE cardinality(answers) = 0 and eventDay BETWEEN
 cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m%d%H') as varchar) and
 cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d%H') as varchar)
    LIMIT 25
```

List of DNS queries that resolved to `192.0.2.1` in the last 7 days

```
SELECT
      time,
      src_endpoint.instance_uid,
      src_endpoint.ip,
      src_endpoint.port,
      query.hostname,
      rcode,
      answer.rdata
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_route53_1_0
    CROSS JOIN UNNEST(answers) as st(answer)
    WHERE answer.rdata='192.0.2.1' and eventDay BETWEEN
 cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m%d%H') as varchar) and
 cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d%H') as varchar)
    LIMIT 25
```

Example Security Lake queries for Security Hub findings

Security Hub provides you with a comprehensive view of your security state in AWS and helps you check your environment against security industry standards and best practices. Security Hub produces findings for security checks and receives findings from third-party services.

Here are some example queries of Security Hub findings:

New findings with severity greater than or equal to `MEDIUM` in the last 7 days

```
SELECT
      time,
      finding,
      severity
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0_fi
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND severity_id >= 3
      AND state_id = 1
    ORDER BY time DESC
    LIMIT 25
```

Duplicate findings in the last 7 days

```
SELECT
    finding.uid,
    MAX(time) AS time,
    ARBITRARY(region) AS region,
    ARBITRARY(accountid) AS accountid,
    ARBITRARY(finding) AS finding,
    ARBITRARY(vulnerabilities) AS vulnerabilities
```

```
FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m%d
%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d%H')
 as varchar)
GROUP BY finding.uid
LIMIT 25
```

All non-informational findings in the last 7 days

```
SELECT
      time,
      finding.title,
      finding,
      severity
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
    WHERE severity != 'Informational' and eventDay BETWEEN
 cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m%d%H') as varchar) and
 cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d%H') as varchar)
    LIMIT 25
```

Findings where the resource is an Amazon S3 bucket (no time restriction)

```
SELECT *
   FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
   WHERE any_match(resources, element -> element.type = 'amzn-s3-demo-bucket')
   LIMIT 25
```

Findings with a Common Vulnerability Scoring System (CVSS) score greater than 1 (no time restriction)

```
SELECT *
   FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
   WHERE any_match(vulnerabilities, element -> element.cve.cvss.base_score > 1.0)
   LIMIT 25
```

Findings that match Common Vulnerabilities and Exposures (CVE) CVE-*0000-0000* (no time restriction)

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
    WHERE any_match(vulnerabilities, element -> element.cve.uid = 'CVE-0000-0000')
    LIMIT 25
```

Count of products that are sending findings from Security Hub in the last 7 days

```
SELECT
      metadata.product.feature.name,
      count(*)
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
    GROUP BY metadata.product.feature.name
    ORDER BY metadata.product.feature.name DESC
    LIMIT 25
```

Count of resource types in findings in the last 7 days

```
SELECT
      count(*),
      resource.type
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
        CROSS JOIN UNNEST(resources) as st(resource)
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
    GROUP BY resource.type
    LIMIT 25
```

Vulnerable packages from findings in the last 7 days

```
SELECT
      vulnerability
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0,
    UNNEST(vulnerabilities) as t(vulnerability)
    WHERE vulnerabilities is not null
    LIMIT 25
```

Findings that have changed in the last 7 days

```
SELECT
    finding.uid,
    finding.created_time,
    finding.first_seen_time,
    finding.last_seen_time,
    finding.modified_time,
    finding.title,
    state
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_sh_findings_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
    LIMIT 25
```

Example Security Lake queries for Amazon VPC Flow Logs

Amazon Virtual Private Cloud (Amazon VPC) provides details about IP traffic going to and from network interfaces in your VPC.

Here are some example queries of Amazon VPC Flow Logs for AWS source version 1:

Traffic in specific AWS Regions in the last 7 days

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND region in ('us-east-1','us-east-2','us-west-2')
    LIMIT 25
```

List of activity from source IP `192.0.2.1` and source port 22 in the last 7 days

```
SELECT *
```

```
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND src_endpoint.ip = '192.0.2.1'
      AND src_endpoint.port = 22
    LIMIT 25
```

Count of distinct destination IP addresses in the last 7 days

```
SELECT
    COUNT(DISTINCT dst_endpoint.ip)
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
    LIMIT 25
```

Traffic originating from 198.51.100.0/24 in the last 7 days

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
    AND split_part(src_endpoint.ip,'.', 1)='198'AND split_part(src_endpoint.ip,'.',
 2)='51'
    LIMIT 25
```

All HTTPS traffic in the last 7 days

```
SELECT
      dst_endpoint.ip as dst,
      src_endpoint.ip as src,
      traffic.packets
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
```

```
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND dst_endpoint.port = 443
    GROUP BY
      dst_endpoint.ip,
      traffic.packets,
      src_endpoint.ip
    ORDER BY traffic.packets DESC
    LIMIT 25
```

Order by packet count for connections destined to port 443 in the last 7 days

```
SELECT
      traffic.packets,
      dst_endpoint.ip
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND dst_endpoint.port = 443
    GROUP BY
      traffic.packets,
      dst_endpoint.ip
    ORDER BY traffic.packets DESC
    LIMIT 25
```

All traffic between IP 192.0.2.1 and 192.0.2.2 in the last 7 days

```
SELECT
      start_time,
      end_time,
      src_endpoint.interface_uid,
      connection_info.direction,
      src_endpoint.ip,
      dst_endpoint.ip,
      src_endpoint.port,
      dst_endpoint.port,
      traffic.packets,
      traffic.bytes
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
```

```
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND(
        src_endpoint.ip = '192.0.2.1'
        AND dst_endpoint.ip = '192.0.2.2')
      OR (
        src_endpoint.ip = '192.0.2.2'
        AND dst_endpoint.ip = '192.0.2.1')
    ORDER BY start_time ASC
    LIMIT 25
```

All inbound traffic in the last 7 days

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND connection_info.direction = 'ingress'
    LIMIT 25
```

All outbound traffic in the last 7 days

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND connection_info.direction = 'egress'
    LIMIT 25
```

All rejected traffic in the last 7 days

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_vpc_flow_1_0
```

```
    WHERE eventDay BETWEEN cast(date_format(current_timestamp - INTERVAL '7' day, '%Y%m
%d%H') as varchar) and cast(date_format(current_timestamp - INTERVAL '0' day, '%Y%m%d
%H') as varchar)
      AND type_uid = 400105
    LIMIT 25
```

Security Lake queries for AWS source version 2 (OCSF 1.1.0)

The following section provides guidance on querying data from Security Lake and includes some query examples for natively-supported AWS sources for AWS source version 2. These queries are designed to retrieve data in a specific AWS Region. These examples use us-east-1 (US East (N. Virginia)). In addition, the example queries use a `LIMIT 25` parameter, which returns up to 25 records. You can omit this parameter or adjust it based on your preferences. For more examples, see the Amazon Security Lake OCSF Queries GitHub directory.

You can query the data that Security Lake stores in AWS Lake Formation databases and tables. You can also create third-party subscribers in the Security Lake console, API, or AWS CLI. Third-party subscribers can also query Lake Formation data from the sources that you specify.

The Lake Formation data lake administrator must grant `SELECT` permissions on the relevant databases and tables to the IAM identity that queries the data. A subscriber must also be created in Security Lake before it can query data. For more information about how to create a subscriber with query access, see Managing query access for Security Lake subscribers.

Log source table

When you query Security Lake data, you must include the name of the Lake Formation table in which the data resides.

```
SELECT *
FROM
 "amazon_security_lake_glue_db_DB_Region"."amazon_security_lake_table_DB_Region_SECURITY_LAKE_T
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
LIMIT 25
```

Common values for the log source table include the following:

- `cloud_trail_mgmt_2_0` – AWS CloudTrail management events
- `lambda_execution_2_0` – CloudTrail data events for Lambda

- `s3_data_2_0` – CloudTrail data events for S3
- `route53_2_0` – Amazon Route 53 resolver query logs
- `sh_findings_2_0` – AWS Security Hub findings
- `vpc_flow_2_0` – Amazon Virtual Private Cloud (Amazon VPC) Flow Logs
- `eks_audit_2_0` – Amazon Elastic Kubernetes Service (Amazon EKS) Audit Logs
- `waf_2_0` – AWS WAFv2 Logs

Example: All Security Hub findings in table `sh_findings_2_0` from us-east-1 Region

```
SELECT *
    FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
    WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
LIMIT 25
```

Database Region

When you query Security Lake data, you must include the name of the database Region from which you're querying the data. For a complete list of database Regions where Security Lake is currently available, see Amazon Security Lake endpoints.

Example: List Amazon Virtual Private Cloud activity from source IP

The following example lists all the Amazon VPC activities from the source IP *`192.0.2.1`* that were recorded after *`20230301`* (March 01, 2023), in the table *`vpc_flow_2_0`* from the *`us-west-2`* `DB_Region`.

```
SELECT *
    FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
    WHERE time_dt > TIMESTAMP '2023-03-01'
    AND src_endpoint.ip = '192.0.2.1'
    ORDER BY time_dt desc
LIMIT 25
```

Partition date

By partitioning your data, you can restrict the amount of data scanned by each query, thereby improving performance and reducing cost. Partitions work slightly different in Security Lake 2.0 compared to Security Lake 1.0. Security Lake now implements partitioning through `time_dt`, `region`, and `accountid`. Whereas, Security Lake 1.0 implemented partitioning through `eventDay`, `region`, and `accountid` parameters.

Querying `time_dt` will automatically yield the date partitions from S3, and can be queried just like any time based field in Athena.

This is an example query using the `time_dt` partition to query the logs after the time March 01, 2023:

```
SELECT *
    FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt > TIMESTAMP '2023-03-01'
AND src_endpoint.ip = '192.0.2.1'
ORDER BY time desc
LIMIT 25
```

Common values for `time_dt` include the following:

Events occurring in the last 1 year

```
WHERE time_dt > CURRENT_TIMESTAMP - INTERVAL '1' YEAR
```

Events occurring in the last 1 month

```
WHERE time_dt > CURRENT_TIMESTAMP - INTERVAL '1' MONTH
```

Events occurring in the last 30 days

```
WHERE time_dt > CURRENT_TIMESTAMP - INTERVAL '30' DAY
```

Events occurring in the last 12 hours

```
WHERE time_dt > CURRENT_TIMESTAMP - INTERVAL '12' HOUR
```

Events occurring in the last 5 minutes

```
WHERE time_dt > CURRENT_TIMESTAMP - INTERVAL '5' MINUTE
```

Events occurring between 7–14 days ago

```
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '14' DAY AND
CURRENT_TIMESTAMP - INTERVAL '7' DAY
```

Events occurring on or after a specific date

```
WHERE time_dt >= TIMESTAMP '2023-03-01'
```

Example: List of all CloudTrail activity from source IP `192.0.2.1` on or after March 1, 2023 in table `cloud_trail_mgmt_1_0`

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay >= '20230301'
    AND src_endpoint.ip = '192.0.2.1'
    ORDER BY time desc
    LIMIT 25
```

Example: List of all CloudTrail activity from source IP `192.0.2.1` in the last 30 days in table `cloud_trail_mgmt_1_0`

```
SELECT *
    FROM
 amazon_security_lake_glue_db_us_east_1.amazon_security_lake_table_us_east_1_cloud_trail_mgmt_1
    WHERE eventDay > cast(date_format(current_timestamp - INTERVAL '30' day, '%Y%m%d
%H') as varchar)
    AND src_endpoint.ip = '192.0.2.1'
    ORDER BY time desc
    LIMIT 25
```

Querying Security Lake observables

Observables is a new feature now available in Security Lake 2.0. The observable object is a pivot element that contains related information found in many places in the event. Querying observables allows users to derive high level security insights from across their data sets.

By querying specific elements within observables, you can restrict the data sets to things such as specific User names, Resource UIDs, IPs, Hashes and other IOC type information

This is an example query using the observables array to query the logs across VPC Flow and Route53 tables containing the IP value '172.01.02.03'

```
WITH a AS
    (SELECT
    time_dt,
    observable.name,
    observable.value
    FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0",
    UNNEST(observables) AS t(observable)
    WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
    AND observable.value='172.01.02.03'
    AND observable.name='src_endpoint.ip'),
b as
    (SELECT
    time_dt,
    observable.name,
    observable.value
    FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_route53_2_0",
    UNNEST(observables) AS t(observable)
    WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
    AND observable.value='172.01.02.03'
    AND observable.name='src_endpoint.ip')
SELECT * FROM a
LEFT JOIN b ON a.value=b.value and a.name=b.name
LIMIT 25
```

Example Security Lake queries for CloudTrail data

AWS CloudTrail tracks user activity and API usage in AWS services. Subscribers can query CloudTrail data to learn the following types of information:

Here are some example queries for CloudTrail data for AWS source version 2:

Unauthorized attempts against AWS services in the last 7 days

```
SELECT
    time_dt,
    api.service.name,
    api.operation,
    api.response.error,
    api.response.message,
    api.response.data,
    cloud.region,
    actor.user.uid,
    src_endpoint.ip,
    http_request.user_agent
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_cloud_trail_mgm
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND api.response.error in (
    'Client.UnauthorizedOperation',
    'Client.InvalidPermission.NotFound',
    'Client.OperationNotPermitted',
    'AccessDenied')
ORDER BY time desc
LIMIT 25
```

List of all CloudTrail activity from source IP `192.0.2.1` in the last 7 days

```
SELECT
    api.request.uid,
    time_dt,
    api.service.name,
    api.operation,
    cloud.region,
    actor.user.uid,
    src_endpoint.ip,
    http_request.user_agent
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_cloud_trail_mgm
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND src_endpoint.ip = '192.0.2.1.'
ORDER BY time desc
LIMIT 25
```

List of all IAM activity in the last 7 days

```
SELECT *
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_cloud_trail_mgm
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND api.service.name = 'iam.amazonaws.com'
```

```
ORDER BY time desc
LIMIT 25
```

Instances where the credential `AIDACKCEVSQ6C2EXAMPLE` was used in the last 7 days

```
SELECT
      actor.user.uid,
      actor.user.uid_alt,
      actor.user.account.uid,
      cloud.region
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_cloud_trail_mgm
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND actor.user.credential_uid = 'AIDACKCEVSQ6C2EXAMPLE'
LIMIT 25
```

List of failed CloudTrail records in the last 7 days

```
SELECT
      actor.user.uid,
      actor.user.uid_alt,
      actor.user.account.uid,
      cloud.region
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_cloud_trail_mgm
WHERE status='failed' and time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND
 CURRENT_TIMESTAMP
ORDER BY time DESC
LIMIT 25
```

Example queries for Route 53 resolver query logs

Amazon Route 53 resolver query logs track DNS queries made by resources within your Amazon VPC. Subscribers can query Route 53 resolver query logs to learn the following types of information:

Here are some example queries for Route 53 reesolver query logs for AWS source version 2:

List of DNS queries from CloudTrail in the last 7 days

```
SELECT
```

```
    time_dt,
    src_endpoint.instance_uid,
    src_endpoint.ip,
    src_endpoint.port,
    query.hostname,
    rcode
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_route53_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
ORDER BY time DESC
LIMIT 25
```

List of DNS queries that match `s3.amazonaws.com` in the last 7 days

```
SELECT
    time_dt,
    src_endpoint.instance_uid,
    src_endpoint.ip,
    src_endpoint.port,
    query.hostname,
    rcode,
    answers
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_route53_2_0"
WHERE query.hostname LIKE 's3.amazonaws.com.' and time_dt BETWEEN CURRENT_TIMESTAMP -
 INTERVAL '7' DAY AND CURRENT_TIMESTAMP
ORDER BY time DESC
LIMIT 25
```

List of DNS queries that didn't resolve in the last 7 days

```
SELECT
    time_dt,
    src_endpoint.instance_uid,
    src_endpoint.ip,
    src_endpoint.port,
    query.hostname,
    rcode,
    answers
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_route53_2_0"
WHERE cardinality(answers) = 0 and time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY
 AND CURRENT_TIMESTAMP
```

```
LIMIT 25
```

List of DNS queries that resolved to `192.0.2.1` in the last 7 days

```
SELECT
    time_dt,
    src_endpoint.instance_uid,
    src_endpoint.ip,
    src_endpoint.port,
    query.hostname,
    rcode,
    answer.rdata
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_route53_2_0",
UNNEST(answers) as st(answer)
WHERE answer.rdata='192.0.2.1'
AND time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
LIMIT 25
```

Example Security Lake queries for Security Hub findings

Security Hub provides you with a comprehensive view of your security state in AWS and helps you check your environment against security industry standards and best practices. Security Hub produces findings for security checks and receives findings from third-party services.

Here are some example queries for Security Hub findings for AWS source version 2:

New findings with severity greater than or equal to `MEDIUM` in the last 7 days

```
SELECT
    time_dt,
    finding_info,
    severity_id,
    status
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
    AND severity_id >= 3
    AND status = 'New'
ORDER BY time DESC
LIMIT 25
```

Duplicate findings in the last 7 days

```
SELECT
    finding_info.uid,
    MAX(time_dt) AS time,
    ARBITRARY(region) AS region,
    ARBITRARY(accountid) AS accountid,
    ARBITRARY(finding_info) AS finding,
    ARBITRARY(vulnerabilities) AS vulnerabilities
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
GROUP BY finding_info.uid
LIMIT 25
```

All non-informational findings in the last 7 days

```
SELECT
    time_dt,
    finding_info.title,
    finding_info,
    severity
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE severity != 'Informational' and time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7'
 DAY AND CURRENT_TIMESTAMP
LIMIT 25
```

Findings where the resource is an Amazon S3 bucket (no time restriction)

```
SELECT *
   FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE any_match(resources, element -> element.type = 'amzn-s3-demo-bucket')
LIMIT 25
```

Findings with a Common Vulnerability Scoring System (CVSS) score greater than 1 (no time restriction)

```
SELECT
    DISTINCT finding_info.uid
    time_dt,
    metadata,
    finding_info,
    vulnerabilities,
    resource
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
UNNEST(vulnerabilities) AS t(vulnerability),
UNNEST(vulnerability.cve.cvss) AS t(cvs)
WHERE cvs.base_score > 1.0
AND vulnerabilities is NOT NULL
LIMIT 25
```

Findings that match Common Vulnerabilities and Exposures (CVE) `CVE-0000-0000` (no time restriction)

```
SELECT *
    FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE any_match(vulnerabilities, element -> element.cve.uid = 'CVE-0000-0000')
LIMIT 25
```

Count of products that are sending findings from Security Hub in the last 7 days

```
SELECT
    metadata.product.name,
    count(*)
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
GROUP BY metadata.product.name
ORDER BY metadata.product.name DESC
LIMIT 25
```

Count of resource types in findings in the last 7 days

```
SELECT
    count(*) AS "Total",
    resource.type
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
```

```
GROUP BY resource.type
ORDER BY count(*) DESC
LIMIT 25
```

Vulnerable packages from findings in the last 7 days

```
SELECT
    vulnerabilities
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND vulnerabilities is NOT NULL
LIMIT 25
```

Findings that have changed in the last 7 days

```
SELECT
    status,
    finding_info.title,
    finding_info.created_time_dt,
    finding_info,
    finding_info.uid,
    finding_info.first_seen_time_dt,
    finding_info.last_seen_time_dt,
    finding_info.modified_time_dt
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_sh_findings_2_0
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
LIMIT 25
```

Example Security Lake queries for Amazon VPC Flow Logs

Amazon Virtual Private Cloud (Amazon VPC) provides details about IP traffic going to and from network interfaces in your VPC.

Here are some example queries for Amazon VPC Flow Logs for AWS source version 2:

Traffic in specific AWS Regions in the last 7 days

```
SELECT *
    FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
```

```
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND region in ('us-east-1','us-east-2','us-west-2')
LIMIT 25
```

List of activity from source IP `192.0.2.1` and source port 22 in the last 7 days

```
SELECT *
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND src_endpoint.ip = '192.0.2.1'
AND src_endpoint.port = 22
LIMIT 25
```

Count of distinct destination IP addresses in the last 7 days

```
SELECT
    COUNT(DISTINCT dst_endpoint.ip) AS "Total"
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
LIMIT 25
```

Traffic originating from 198.51.100.0/24 in the last 7 days

```
SELECT *
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND split_part(src_endpoint.ip,'.', 1)='198'AND split_part(src_endpoint.ip,'.', 2)='51'
LIMIT 25
```

All HTTPS traffic in the last 7 days

```
SELECT
    dst_endpoint.ip as dst,
    src_endpoint.ip as src,
    traffic.packets
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
```

```
AND dst_endpoint.port = 443
GROUP BY
    dst_endpoint.ip,
    traffic.packets,
    src_endpoint.ip
ORDER BY traffic.packets DESC
LIMIT 25
```

Order by packet count for connections destined to port 443 in the last 7 days

```
SELECT
    traffic.packets,
    dst_endpoint.ip
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND dst_endpoint.port = 443
GROUP BY
    traffic.packets,
    dst_endpoint.ip
ORDER BY traffic.packets DESC
LIMIT 25
```

All traffic between IP `192.0.2.1` and `192.0.2.2` in the last 7 days

```
SELECT
    start_time_dt,
    end_time_dt,
    src_endpoint.interface_uid,
    connection_info.direction,
    src_endpoint.ip,
    dst_endpoint.ip,
    src_endpoint.port,
    dst_endpoint.port,
    traffic.packets,
    traffic.bytes
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND(
    src_endpoint.ip = '192.0.2.1'
AND dst_endpoint.ip = '192.0.2.2')
OR (
    src_endpoint.ip = '192.0.2.2'
AND dst_endpoint.ip = '192.0.2.1')
ORDER BY start_time_dt ASC
LIMIT 25
```

All inbound traffic in the last 7 days

```
SELECT *
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND connection_info.direction = 'Inbound'
LIMIT 25
```

All outbound traffic in the last 7 days

```
SELECT *
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND connection_info.direction = 'Outbound'
LIMIT 25
```

All rejected traffic in the last 7 days

```
SELECT *
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_vpc_flow_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND action = 'Denied'
LIMIT 25
```

Example Security Lake queries for Amazon EKS audit logs

Amazon EKS logs track control plane activity provides audit and diagnostic logs directly from the Amazon EKS control plane to CloudWatch Logs in your account. These logs make it easy for you to secure and run your clusters. Subscribers can query EKS logs to learn the following types of information.

Here are some example queries for Amazon EKS audit logs for AWS source version 2:

Requests to a specific URL in the last 7 days

```
SELECT
    time_dt,
    actor.user.name,
    http_request.url.path,
    activity_name
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_eks_audit_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND activity_name = 'get'
and http_request.url.path = '/apis/coordination.k8s.io/v1/'
LIMIT 25
```

Update requests from '10.0.97.167' over the last 7 days

```
SELECT
    activity_name,
    time_dt,
    api.request,
    http_request.url.path,
    src_endpoint.ip,
    resources
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_eks_audit_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND src_endpoint.ip = '10.0.97.167'
AND activity_name = 'Update'
LIMIT 25
```

Requests and Responses associated with resource 'kube-controller-manager' over the last 7 days

```
SELECT
    activity_name,
    time_dt,
    api.request,
    api.response,
    resource.name
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_eks_audit_2_0",
UNNEST(resources) AS t(resource)
```

```
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND resource.name = 'kube-controller-manager'
LIMIT 25
```

Example Security Lake queries for AWS WAFv2 logs

AWS WAF is a web application firewall that you can use to monitor web requests that your end users send to your applications and to control access to your content.

Here are some examples queries for AWS WAFv2 logs for AWS source version 2:

Post requests from a specific source IP over the past 7 days

```
SELECT
    time_dt,
    activity_name,
    src_endpoint.ip,
    http_request.url.path,
    http_request.url.hostname,
    http_request.http_method,
    http_request.http_headers
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_waf_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND src_endpoint.ip = '100.123.123.123'
AND activity_name = 'Post'
LIMIT 25
```

Requests which matched a firewall type MANAGED_RULE_GROUP over the past 7 days

```
SELECT
    time_dt,
    activity_name,
    src_endpoint.ip,
    http_request.url.path,
    http_request.url.hostname,
    http_request.http_method,
    firewall_rule.uid,
    firewall_rule.type,
    firewall_rule.condition,
    firewall_rule.match_location,
    firewall_rule.match_details,
    firewall_rule.rate_limit
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_waf_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND firewall_rule.type = 'MANAGED_RULE_GROUP'
LIMIT 25
```

Requests which matched a REGEX in a firewall rule over the past 7 days

```
SELECT
    time_dt,
    activity_name,
    src_endpoint.ip,
    http_request.url.path,
    http_request.url.hostname,
    http_request.http_method,
    firewall_rule.uid,
    firewall_rule.type,
    firewall_rule.condition,
    firewall_rule.match_location,
    firewall_rule.match_details,
    firewall_rule.rate_limit
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_waf_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
AND firewall_rule.condition = 'REGEX'
LIMIT 25
```

Denied get requests for AWS credentials which triggered AWS WAF rule over the past 7 days

```
SELECT
    time_dt,
    activity_name,
    action,
    src_endpoint.ip,
    http_request.url.path,
    http_request.url.hostname,
    http_request.http_method,
    firewall_rule.uid,
    firewall_rule.type
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_waf_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY AND CURRENT_TIMESTAMP
```

```
AND http_request.url.path = '/.aws/credentials'
AND action = 'Denied'
LIMIT 25
```

Get requests for AWS Credentials, grouped by country over the past 7 days

```
SELECT count(*) as Total,
    src_endpoint.location.country AS Country,
    activity_name,
    action,
    src_endpoint.ip,
    http_request.url.path,
    http_request.url.hostname,
    http_request.http_method
FROM
 "amazon_security_lake_glue_db_us_east_1"."amazon_security_lake_table_us_east_1_waf_2_0"
WHERE time_dt BETWEEN CURRENT_TIMESTAMP - INTERVAL '7' DAY
    AND CURRENT_TIMESTAMP
    AND activity_name = 'Get'
    AND http_request.url.path = '/.aws/credentials'
GROUP BY src_endpoint.location.country,
    activity_name,
    action,
    src_endpoint.ip,
    http_request.url.path,
    http_request.url.hostname,
    http_request.http_method
```

Lifecycle management in Security Lake

You can customize Security Lake to store data in your preferred AWS Regions for your preferred amount of time. Lifecycle management can help you comply with different compliance requirements.

Retention management

To manage your data so that it is stored cost effectively, you can configure retention settings for the data. Because Security Lake stores your data as objects in Amazon Simple Storage Service (Amazon S3) buckets, the retention settings correspond to an Amazon S3 Lifecycle configuration. By configuring these settings, you can specify your preferred Amazon S3 storage class and the time period for S3 objects to stay in that storage class before they transition to a different storage class or expire. For more information about Amazon S3 Lifecycle configurations, see Managing your storage lifecycle in the *Amazon Simple Storage Service User Guide*.

In Security Lake, you specify retention settings at the Region level. For example, you might choose to transition all S3 objects in a specific AWS Region to the **S3 Standard-IA** storage class 30 days after they're written to the data lake. The default Amazon S3 storage class is **S3 Standard**.

⚠ **Important**

Security Lake doesn't support Amazon S3 Object Lock. When the data lake buckets are created, S3 Object Lock is disabled by default. Enabling S3 Object Lock with default retention mode interrupts the delivery of normalized log data to the data lake.

Configuring retention settings when enabling Security Lake

Follow these instructions to configure retention settings for one or more Regions when you're onboarding to Security Lake. If you don't configure retention settings, Security Lake uses the default settings for an Amazon S3 Lifecycle configuration—store the data indefinitely using the **S3 Standard** storage class.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.

2. When you reach **Step 2: Define target objective** of the onboarding workflow, choose **Add transition** under **Select storage classes**. Then choose the Amazon S3 storage class that you want to transition S3 objects to. (The unlisted, default storage class is **S3 Standard**.) Also specify a retention period (in days) for that storage class. To transition objects to another storage class after that time, choose **Add transition** and enter the settings for the subsequent storage class and retention period.
3. To specify when you want S3 objects to expire, choose **Add transition**. Then, for storage class, choose **Expire**. For retention period, enter the total number of days that you want to store objects in Amazon S3, using any storage class, after objects are created. When this time period ends, objects expire and Amazon S3 deletes them.
4. When you finish, choose **Next**.

Your changes will apply to all the Regions that you enabled Security Lake in during earlier onboarding steps.

API

To configure retention settings programmatically when you're onboarding to Security Lake, use the CreateDataLake operation of the Security Lake API. If you're using the AWS CLI, run the create-data-lake command. Specify the retention settings you want in the `lifecycleConfiguration` parameters as follows:

- For `transitions`, specify the total number of days (`days`) that you want to store S3 objects in a particular Amazon S3 storage class (`storageClass`).
- For `expiration`, specify the total number of days that you want to store objects in Amazon S3, using any storage class, after objects are created. When this time period ends, objects expire and Amazon S3 deletes them.

Security Lake applies the settings to the Region that you specify in the `region` field of the `configurations` object.

For example, the following command enables Security Lake in the `us-east-1` Region. In this Region, objects expire after 365 days, and objects transition to the `ONEZONE_IA` S3 storage class after 60 days. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake create-data-lake \
--configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
--meta-store-manager-role-arn "arn:aws:securitylake:ap-
northeast-2:123456789012:data-lake/default"
```

Updating retention settings

Follow these instructions to update retention settings for one or more Regions after enabling Security Lake.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. In the navigation pane, choose **Regions**
3. Select a Region, and then choose **Edit**.
4. In the **Select storage classes** section, enter the settings that you want. For storage class, choose the Amazon S3 storage class that you want to transition S3 objects to. (The unlisted, default storage class is **S3 Standard**.) For retention period, enter the number of days that you want to store objects in that storage class. You can specify multiple transitions.

   To also specify when you want S3 objects to expire, choose **Expire** for storage class. Then, for retention period, enter the total number of days that you want to store objects in Amazon S3, using any storage class, after objects are created. When this time period ends, objects expire and Amazon S3 deletes them.
5. When you finish, choose **Save**.

API

To update retention settings programmatically, use the UpdateDataLake operation of the Security Lake API. If you're using the AWS CLI, run the update-data-lake command. In your request, use the `lifecycleConfiguration` parameter to specify the new settings:

- To change the transition settings, use the `transitions` parameters to specify each new time period in days (`days`) that you want to store S3 objects in a particular Amazon S3 storage class (`storageClass`).
- To change the overall retention period, use the `expiration` parameter to specify the total number of days that you want to store S3 objects, using any storage class, after objects are created. When this retention period ends, objects expire and Amazon S3 deletes them.

Security Lake applies the settings to the Region that you specify in the `region` field of the `configurations` object.

The `UpdateDataLake` operation of the Security Lake API works as an "upsert" operation that performs an insert if the specified item or record does not exist, or an update if it already exists. Security Lake securely stores your data at rest using AWS encryption solutions.

Omitting the key `encryptionConfiguration` from a Region that is included in an update call that currently uses KMS will leave that Region's KMS key in place, but specifying a key will reset the key in the same region.

For example, the following AWS CLI command updates the data expiration settings and storage transition settings for the `us-east-1` Region. In this Region, objects expire after 500 days, and objects transition to the `ONEZONE_IA` S3 storage class after 30 days. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake update-data-lake \
--configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"ONEZONE_IA"}]}}]' \
--meta-store-manager-role-arn "arn:aws:securitylake:ap-
northeast-2:123456789012:data-lake/default"
```

Rollup Regions

A rollup Region consolidates data from one or more contributing Regions. This can help you comply with regional data compliance requirements.

For instructions on configuring rollup Regions, see Configuring rollup Regions in Security Lake.

Open Cybersecurity Schema Framework (OCSF) in Security Lake

What is OCSF?

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.

Security Lake automatically converts logs and events that come from natively-supported AWS services to the OCSF schema. After conversion to OCSF, Security Lake stores the data in an Amazon Simple Storage Service (Amazon S3) bucket (one bucket per AWS Region) in your AWS account. Logs and events that are written to Security Lake from custom sources must adhere to the OCSF schema and an Apache Parquet format. Subscribers can treat the logs and events as generic Parquet records or apply the OCSF schema event class to more accurately interpret the information contained in a record.

OCSF event classes

Logs and events from a given Security Lake source match a specific event class defined in OCSF. DNS Activity, SSH Activity, and Authentication are examples of event classes in OCSF. You can specify which event class a particular source matches.

OCSF source identification

OCSF uses a variety of fields to help you determine where a specific set of logs or events originated. These are the values of the relevant fields for AWS services that are natively supported as sources in Security Lake.

```
The OCSF source identification for AWS log sources (Version 1) are listed
in the following table.
```

Source	metadata.product.name	metadata.product.vendor_name	metadata.product.feature.name	class_name	metadata.version
CloudTrail Lambda Data Events	CloudTrail	AWS	Data	API Activity	1.0.0-rc.2
CloudTrail Management Events	CloudTrail	AWS	Management	API Activity, Authentication , or Account Change	1.0.0-rc.2
CloudTrail S3 Data Events	CloudTrail	AWS	Data	API Activity	1.0.0-rc.2
Route 53	Route 53	AWS	Resolver Query Logs	DNS Activity	1.0.0-rc.2
Security Hub	Security Hub	AWS	Matches Security Hub ProductName value	Security Finding	1.0.0-rc.2
VPC Flow Logs	Amazon VPC	AWS	Flowlogs	Network Activity	1.0.0-rc.2

The OCSF source identification for AWS log sources (Version 2) are listed in the following table.

Source	metadata.product.name	metadata.product.vendor_name	metadata.product.feature.name	class_name	metadata.version
CloudTrail Lambda Data Events	CloudTrail	AWS	Data	API Activity	1.1.0
CloudTrail Management Events	CloudTrail	AWS	Management	API Activity, Authentication , or Account Change	1.1.0
CloudTrail S3 Data Events	CloudTrail	AWS	Data	API Activity	1.1.0
Route 53	Route 53	AWS	Resolver Query Logs	DNS Activity	1.1.0
Security Hub	Matches AWS Security Finding Format (ASFF) ProductName value	Matches AWS Security Finding Format (ASFF) CompanyName value	Matches featureName value from ASFF ProductFields	Vulnerability Finding, Compliance Finding, or Detection Finding	1.1.0
VPC Flow Logs	Amazon VPC	AWS	Flowlogs	Network Activity	1.1.0
EKS Audit Logs	Amazon EKS	AWS	Elastic Kubernetes Service	API Activity	1.1.0

Source	metadata.product.name	metadata.product.vendor_name	metadata.product.feature.name	class_name	metadata.version
AWS WAFv2 Logs	`AWS WAF`	`AWS`	–	`HTTP Activity`	`1.1.0`

Integrations with Security Lake

Amazon Security Lake integrates with other AWS services and third-party products. Integrations can send data to Security Lake as a source or consume data in Security Lake as a subscriber. The following topics explain which AWS services and third-party products integrate with Security Lake.

Topics

AWS service integrations with Security Lake

Amazon Security Lake integrates with other AWS services. A service may either operate as a *source integration*, a *subscriber integration*, or both.

Source integrations have the following properties:

- Send data to Security Lake
- Data arrives in the Open Cybersecurity Schema Framework (OCSF) in Security Lake schema
- Data arrives in Apache Parquet format

Subscriber integrations have the following properties can read source data from Security Lake at an HTTPS endpoint or Amazon Simple Queue Service (Amazon SQS) queue, or by directly querying source data from AWS Lake Formation

The following section explains which AWS services Security Lake integrates with and how each integration works.

Integration with AWS AppFabric

Integration type: Source

AWS AppFabric is a no-code service that connects software as a service (SaaS) applications across your organization, so IT and security teams can manage and secure applications using a standard schema and central repository.

How Security Lake receives AppFabric findings

You can send AppFabric audit log data to Security Lake by selecting Amazon Kinesis Data Firehose as a destination and configuring Kinesis Data Firehose to deliver data in OCSF schema and Apache Parquet format to Security Lake.

Prerequisites

Before you can send AppFabric audit logs to Security Lake, you must output your OCSF normalized audit logs to a Kinesis Data Firehose stream. You can then configure Kinesis Data Firehose to send the output to your Security Lake Amazon S3 bucket. For more information, see Choose Amazon S3 for your destination in the *Amazon Kinesis Developer Guide*.

Send your AppFabric findings to Security Lake

To send AppFabric audit logs to Security Lake after completing the preceding prerequisite, you must enable both services and add AppFabric as a custom source in Security Lake. For instructions on adding a custom source, see Collecting data from custom sources in Security Lake.

Stop receiving AppFabric logs in Security Lake

To stop receiving AppFabric audit logs, you can use the Security Lake console, Security Lake API, or AWS CLI to delete AppFabric as a custom source. For instructions, see Deleting a custom source from Security Lake.

Integration with Amazon Detective

Integration type: Subscriber

Amazon Detective helps you analyze, investigate, and quickly identify the root cause of security findings or suspicious activities. Detective automatically collects log data from your AWS resources. It then uses machine learning, statistical analysis, and graph theory to generate visualizations that help you to conduct faster and more efficient security investigations. The Detective prebuilt data aggregations, summaries, and context help you to quickly analyze and determine the nature and extent of possible security issues.

When you integrate Security Lake and Detective, you can query the raw log data stored by Security Lake from Detective. For more information, see Integration with Amazon Security Lake.

Integration with Amazon OpenSearch Service

Integration type: Subscriber

Amazon OpenSearch Service is a managed service that makes it easy to deploy, operate, and scale OpenSearch Service clusters in the AWS Cloud. Using OpenSearch Service Ingestion to ingest data into your OpenSearch Service cluster, you can derive insights quicker for time sensitive security investigations. You can respond swiftly to security incidents, helping you protect your business critical data and systems.

OpenSearch Service dashboard

After you integrate OpenSearch Service with Security Lake, you can configure Security Lake to send security data from different sources to OpenSearch Service through serverless OpenSearch Service Ingestion. For more information on how to configure OpenSearch Service ingestion to process security data, see Generate security insights from Amazon Security Lake data using Amazon OpenSearch Service Ingestion.

After OpenSearch Service Ingestion starts writing your data into your OpenSearch Service domain. To visualize the data using the pre-built dashboards, navigate to dashboards and choose any one of the installed dashboards.

OpenSearch Service direct query

You can use OpenSearch Service direct query to analyze data in Amazon Security Lake. OpenSearch Service provides a zero-ETL integration as a way to directly query your data in Security Lake using OpenSearch SQL or OpenSearch Piped Processing Language (PPL) without incurring the friction of building ingestion pipelines or switching between analytics tools. This approach eliminates the need for data movement or duplication, allowing you to analyze your data where it rests using OpenSearch's Discover experience in OpenSearch Dashboards. When you want to switch from querying data at rest to actively monitoring with dashboards, you can build indexed views on your query results and ingest it into an OpenSearch Service index. For more information on direct queries, see Working with direct queries in the Amazon OpenSearch Service Developer Guide.

OpenSearch Service uses a OpenSearch Serverless collection to directly query the data in Security Lake and store your indexed views. To do this, you create a data source that enables you to use OpenSearch zero-ETL capabilities on Security Lake data. When you create a data source you can directly search, gain insights from, and analyze data stored in Security Lake. You can accelerate your query performance and use advanced OpenSearch analytics on select Security Lake data sets using on-demand indexing.

- For details on how to create the OpenSearch data source integration, see Creating an Amazon Security Lake data source integration in the Amazon OpenSearch Service Developer Guide.
- For details on how to configure Security Lake data source in OpenSearch, see Configuring a Security Lake data source in OpenSearch Dashboards in the Amazon OpenSearch Service Developer Guide.

Integration with Amazon QuickSight

Integration type: Subscriber

Amazon QuickSight is a cloud-scale business intelligence (BI) service that you can use to deliver easy-to-understand insights to the people who you work with, wherever they are. Amazon QuickSight connects to your data in the cloud and combines data from many different sources. Amazon QuickSight gives decision-makers the opportunity to explore and interpret information in an interactive visual environment. They have secure access to dashboards from any device on your network and from mobile devices.

Amazon QuickSight dashboard

To visualize your Amazon Security Lake data in Amazon QuickSight, to create the required AWS objects and deploy basic data sources, data sets, analysis, dashboards, and user groups to Amazon QuickSight with respect to Security Lake. For the detailed instructions, see Integration with Amazon QuickSight.

Integration with Amazon SageMaker AI

Integration type: Subscriber

Amazon SageMaker AI is a fully managed machine learning (ML) service. With Security Lake, data scientists and developers can quickly and confidently build, train, and deploy ML models into a production-ready hosted environment. It provides a UI experience for running ML workflows that makes SageMaker AI ML tools available across multiple integrated development environments (IDEs).

SageMaker AI insights

You can generate machine learning insights for Security Lake using SageMaker AI Studio. SageMaker AI Studio is a web integrated development environment (IDE) for machine learning that provides tools for data scientists to prepare, build, train, and deploy machine learning models. With this solution, you can quickly deploy a base set of Python notebooks focusing on AWS Security Hub findings in Security Lake, which can also be expanded to incorporate other AWS sources or custom data sources in Security Lake. For more details, see Generate machine learning insights for Amazon Security Lake data using Amazon SageMaker AI.

Integration with Amazon Bedrock

Amazon Bedrock is a fully managed service that makes high-performing foundation models (FMs) from leading AI startups and Amazon available for your use through a unified API. With Amazon Bedrock's serverless experience, you can get started quickly, privately customize foundation models with your own data, and easily and securely integrate and deploy them into your applications using AWS tools without having to manage any infrastructure.

Generative AI

You can use the generative AI capabilities of Amazon Bedrock and natural language input in SageMaker AI Studio to analyze data in Security Lake and work towards reducing your organization’s risk and increase your security posture. You can reduce the amount of time needed to conduct an investigation by automatically identifying the appropriate data sources, generating and invoking SQL queries, and visualizing data from your investigation. For more details see Generate AI powered insights for Amazon Security Lake using Amazon SageMaker AI Studio and Amazon Bedrock.

Integration with AWS Security Hub

Integration type: Source

AWS Security Hub provides you with a comprehensive view of your security state in AWS and helps you check your environment against security industry standards and best practices. Security Hub collects security data from across AWS accounts, services, and supported third-party partner products and helps you to analyze your security trends and identify the highest priority security issues.

When you enable Security Hub and add Security Hub findings as a source in Security Lake, Security Hub starts sending new findings and updates to existing findings to Security Lake.

How Security Lake receives Security Hub findings

In Security Hub, security issues are tracked as findings. Some findings come from issues that are detected by other AWS services or by third-party partners. Security Hub also generates its own findings by running automated and continuous security checks against rules. The rules are represented by security controls.

All findings in Security Hub use a standard JSON format called the AWS Security Finding Format (ASFF).

Security Lake receives Security Hub findings and transforms them into the Open Cybersecurity Schema Framework (OCSF) in Security Lake.

Send your Security Hub findings to Security Lake

To send Security Hub findings to Security Lake, you must enable both services and add Security Hub findings as a source in Security Lake. For instructions on adding an AWS source, see Adding an AWS service as a source.

If you want Security Hub to generate control findings and send them to Security Lake, you must enable the relevant security standards and turn on resource recording on a Regional basis in AWS Config. For more information, see Enabling and configuring AWS Config in the *AWS Security Hub User Guide*.

Stop receiving Security Hub findings in Security Lake

To stop receiving Security Hub findings, you can use the Security Hub console, Security Hub API, or AWS CLI.

See Disabling and enabling the flow of findings from an integration (console) or Disabling the flow of findings from an integration (Security Hub API, AWS CLI) in the *AWS Security Hub User Guide*.

Third-party integrations with Security Lake

Amazon Security Lake integrates with multiple third-party providers. A provider may offer a *source integration*, a *subscriber integration*, or a *service integration*. Providers may offer one or more integration types.

Source integrations have the following properties:

- Send data to Security Lake
- Data arrives in Apache Parquet format
- Data arrives in the Open Cybersecurity Schema Framework (OCSF) in Security Lake schema

Subscriber integrations have the following properties:

- Read source data from Security Lake at an HTTPS endpoint or Amazon Simple Queue Service (Amazon SQS) queue, or by directly querying source data from AWS Lake Formation
- Able to read data in Apache Parquet format
- Able to read data in OCSF schema

Service integrations can help you implement Security Lake and other AWS services in your organization. They can also provide assistance with reporting, analytics, and other use cases.

To search for a specific partner provider, see the Partner Solutions Finder. To purchase a third-party product, see the AWS Marketplace.

To request to be added as a partner integration or become a Security Lake partner, send an email to `<securitylake-partners@amazon.com>`.

If you use third-party integrations that send findings to AWS Security Hub, you can also review those findings in Security Lake if the Security Hub integration for Security Lake is enabled. For instructions on enabling the integration, see Integration with AWS Security Hub. For a list of third-party integrations that send findings to Security Hub, see Available third-party partner product integrations in the *AWS Security Hub User Guide*.

Before setting up your subscribers verify your subscriber's OCSF log support. For the latest details, review your subscriber's documentation.

Query integration

You can query the data that Security Lake stores in AWS Lake Formation databases and tables. You can also create third-party subscribers in the Security Lake console, API, or AWS Command Line Interface.

The Lake Formation data lake administrator must grant `SELECT` permissions on the relevant databases and tables to the IAM identity that queries the data. You must create a subscriber in Security Lake before querying data. For more information about how to create a subscriber with query access, see Managing query access for Security Lake subscribers.

You can configure query integration with Security Lake for the following third-party partners.

- Cribl – Search

- IBM – QRadar
- Palo Alto Networks – XSOAR
- Query.AI – Query Federated Search
- SOC Prime
- Splunk – Federated Analytics
- Tego Cyber

Accenture – MxDR

Integration type: Subscriber, Service

Accenture's MxDR integration with Security Lake offers real-time data ingestion of logs and events, managed anomaly detection, threat hunting, and security operations. This aids analytics and managed detection and response (MDR).

As a service integration, Accenture can also help you implement Security Lake in your organization.

Integration documentation

Aqua Security

Integration type: Source

Aqua Security can be added as a custom source to send audit events to Security Lake. The audit events are converted into OCSF schema and Parquet format.

Integration documentation

Barracuda – Email Protection

Integration type: Source

Barracuda Email Protection can send events to Security Lake when new phishing email attacks are detected. You can receive these events alongside other security data in your data lake.

Integration documentation

Booz Allen Hamilton

Integration type: Service

As a service integration, Booz Allen Hamilton uses a data-driven approach to cybersecurity by fusing data and analytics with the Security Lake service.

Partner link

Bosch Software and Digital Solutions – AIShield

Integration type: Source

AIShield powered by Bosch provides automated vulnerability analysis and endpoint protection for AI assets through its integration with Security Lake.

Integration documentation

ChaosSearch

Integration type: Subscriber

ChaosSearch offers multi-model data access to users with open APIs such as Elasticsearch and SQL, or with the Kibana and Superset UIs included natively. You can consume your Security Lake data in ChaosSearch without retention limits to monitor, alert, and threat hunt. This helps you face today's complex security environments and persistent threats.

Integration documentation

Cisco Security – Secure Firewall

Integration type: Source

By integrating Cisco Secure Firewall with Security Lake, you can store firewall logs in a structured and scalable manner. Cisco's eNcore client streams firewall logs from the Firewall Management Center, performs schema conversion to OCSF schema, and stores them in Security Lake.

Integration documentation

Claroty – xDome

Integration type: Source

Claroty xDome sends alerts detected within networks to Security Lake with minimal configuration. Flexible and rapid deployment options help xDome protect extended Internet of Things (XIoT) assets—consisting of IoT, IIoT, and BMS assets—within your network, while automatically detecting early indicators of threats.

Integration documentation

CMD Solutions

Integration type: Service

CMD Solutions helps businesses increase their agility by integrating security early and continuously through design, automation, and continuous assurance processes. As a service integration, CMD Solutions can help you implement Security Lake in your organization.

Partner link

Confluent – Amazon S3 Sink Connector

Integration type: Source

Confluent automatically connects, configures, and orchestrates data integrations with fully-managed, pre-built connectors. The Confluent S3 Sink Connector lets you take raw data and sink it into Security Lake at scale in native parquet format.

Integration documentation

Contrast Security

Integration type: Source

Partner product for the integration: Contrast Assess

Contrast Security Assess is an IAST tool offering real-time vulnerability detection in web apps, APIs, and microservices. Assess integrates with Security Lake to help provide centralized visibility for all your workloads.

Integration documentation

Cribl – Search

Integration type: Subscriber

You can use Cribl Search to search Security Lake data.

Integration documentation

Cribl – Stream

Integration type: Source

You can use Cribl Stream to send data from any Cribl supported third-party sources to Security Lake in OCSF schema.

Integration documentation

CrowdStrike – Falcon Data Replicator

Integration type: Source

This integration pulls data from the CrowdStrike Falcon Data Replicator on a continuous streaming basis, transforms the data into OCSF schema, and sends it to Security Lake.

Integration documentation

CrowdStrike – Next Gen SIEM

Integration type: Subscriber

Simplify ingestion of Security Lake data with the CrowdStrike Falcon Next-Gen SIEM data connector featuring native OCSF schema parsers. Falcon NG SIEM revolutionizes threat detection, investigation and response by bringing together unmatched security depth and breadth in one unified platform to stop breaches.

Integration documentation

CyberArk – Unified Identify Security Platform

Integration type: Source

CyberArk Audit Adapter, an AWS Lambda function, collects security events from CyberArk Identity Security Platform and sends the data to Security Lake in OCSF schema.

Integration documentation

Cyber Security Cloud – Cloud Fastener

Integration type: Subscriber

CloudFastener leverages Security Lake to make it easier to consolidate security data from your cloud environments.

Integration documentation

DataBahn

Integration type: Source

Centralize your security data in Security Lake using DataBahn’s Security Data Fabric.

Integration documentation (sign in to the DataBahn portal to review the documentation)

Darktrace – Cyber AI Loop

Integration type: Source

The Darktrace and Security Lake integration brings the power of Darktrace self-learning to Security Lake. Insights from Cyber AI Loop can be correlated against other data streams and elements of your organization's security stack. The integration logs Darktrace model breaches as security findings.

Integration documentation (sign in to the Darktrace portal to review the documentation)

Datadog

Integration type: Subscriber

Datadog Cloud SIEM detects real-time threats to your cloud environment, including data in Security Lake, and unifies DevOps and security teams in one platform.

Integration documentation

Deloitte – MXDR Cyber Analytics and AI Engine (CAE)

Integration type: Subscriber, Service

Deloitte MXDR CAE helps you quickly store, analyze, and visualize your standardized security data. The CAE suite of customized analytic, AI, and ML capabilities automatically provide actionable insights based on models that run against the OCSF-formatted data in Security Lake.

As a service integration, Deloitte can also help you implement Security Lake in your organization.

Integration documentation

Devo

Integration type: Subscriber

The Devo collector for AWS supports ingestion from Security Lake. This integration can help you analyze and address a variety of security use cases, such as threat detection, investigation, and incident response.

Integration documentation

DXC – SecMon

Integration type: Subscriber, Service

DXC SecMon collects security events from Security Lake and monitors them to detect and alert on potential security threats. This helps organizations gain a better understanding of their security posture and proactively identify and respond to threats.

As a service integration, DXC can also help you implement Security Lake in your organization.

Integration documentation

Eviden – AIsaac (formerly Atos)

Integration type: Subscriber

The AIsaac MDR platform consumes VPC Flow Logs ingested in OCSF schema in Security Lake and utilizes AI models for detecting threats.

Integration documentation

ExtraHop – Reveal(x) 360

Integration type: Source

You can enhance your workload and application security by integrating network data, including detections of IOCs, from ExtraHop Reveal(x) 360, to Security Lake in OCSF schema

Integration documentation

Falcosidekick

Integration type: Source

Falcosidekick collects and sends Falco events to Security Lake. This integration exports security events using the OCSF schema.

Integration documentation

Fortinet - Cloud Native Firewall

Integration type: Source

When creating FortiGate CNF instances in AWS, you can specify Amazon Security Lake as a log output destination.

Integration documentation

Gigamon – Application Metadata Intelligence

Integration type: Source

Gigamon Application Metadata Intelligence (AMI) empowers your observability, SIEM, and network performance monitoring tools with critical metadata attributes. This helps provide deeper application visibility so you can pinpoint performance bottlenecks, quality issues, and potential network security risks.

Integration documentation

Hoop Cyber

Integration type: Service

Hoop Cyber FastStart includes a data source assessment, prioritization, onboarding of data sources and helps customers query their data with existing tools and integrations offered through Security Lake.

Partner link

HTCD – AI-First Cloud Security Platform

Integration type: Subscriber

Gain instantaneous compliance automation, prioritization of security findings, and tailored patches. HTCD can query Security Lake to help you uncover threats with natural language queries and AI-driven insights.

Integration documentation

IBM – QRadar

Integration type: Subscriber

IBM Security QRadar SIEM with UAX integrates Security Lake with an analytics platform that identifies and prevents threats across hybrid clouds. This integration supports both data access and query access.

Integration documentation on consuming AWS CloudTrail logs

Integration documentation on using Amazon Athena for queries

Infosys

Integration type: Service

Infosys helps you customize your Security Lake implementation for your organizational needs and provides custom insights.

Partner link

Insbuilt

Integration type: Service

Insbuilt specializes in cloud consulting services and can help you understand how to implement Security Lake in your organization.

Partner link

Kyndryl – AIOps

Integration type: Subscriber, Service

Kyndryl integrates with Security Lake to provide interoperability of cyberdata, threat intelligence, and AI-powered analytics. As a data access subscriber, Kyndryl ingests AWS CloudTrail Management Events from Security Lake for analytics purposes.

As a service integration, Kyndryl can also help you implement Security Lake in your organization.

Integration documentation

Lacework – Polygraph

Integration type: Source

Lacework Polygraph® Data Platform integrates with Security Lake as a data source and provides security findings about vulnerabilities, misconfigurations, and known and unknown threats across your AWS environment.

Integration documentation

Laminar

Integration type: Source

Laminar sends data security events to Security Lake in OCSF schema, making them available for additional analytics use cases, such as incident response and investigation.

Integration documentation

MegazoneCloud

Integration type: Service

MegazoneCloud specializes in cloud consulting services and can help you understand how to implement Security Lake in your organization. We connect Security Lake with integrated ISV solutions to build custom tasks, and build customized insights related with customer needs.

Integration documentation

Monad

Integration type: Source

Monad automatically transforms your data into OCSF schema and sends it to your Security Lake data lake.

Integration documentation

NETSCOUT – Omnis Cyber Intelligence

Integration type: Source

By integrating with Security Lake, NETSCOUT becomes a custom source of security findings and detailed security insights into what’s happening in your enterprise, such as cyberthreats, security risks, and attack surface changes. These findings are produced in the customer account by NETSCOUT CyberStreams and Omnis Cyber Intelligence, and then sent to Security Lake in OCSF schema. The ingested data also meets other requirements and best practices for a Security Lake source, including format, schema, partitioning, and performance-related aspects.

Integration documentation

Netskope – CloudExchange

Integration type: Source

Netskope helps you strengthen your security posture by sharing security-related logs and threat information with Security Lake. Netskope findings are sent to Security Lake with a CloudExchange Plugin, which can be launched as a docker-based environment within AWS or in a local data center.

Integration documentation

New Relic ONE

Integration type: Subscriber

New Relic ONE is a Lambda-based subscriber application. It's deployed in your account, triggered by Amazon SQS, and sends data to New Relic using New Relic license keys

Integration documentation

Okta – Workforce Identity Cloud

Integration type: Source

Okta sends identity logs to Security Lake in OCSF schema through an Amazon EventBridge integration. Okta System Logs in OCSF schema will help security and data scientist teams to query security events by an open source standard. Generating standardized OCSF logs from Okta helps you perform audit activities and generate reports related to authentication, authorization, account changes, and entity changes under a consistent schema.

Integration documentation

AWS CloudFormation template to add Okta as a custom source in Security Lake

Orca – Cloud Security Platform

Integration type: Source

The Orca agentless cloud security platform for AWS integrates with Security Lake by sending Cloud Detection and Response (CDR) events in OCSF schema.

Integration documentation (sign in to the Orca portal to review the documentation)

Palo Alto Networks – Prisma Cloud

Integration type: Source

Palo Alto Networks Prisma Cloud aggregates vulnerability detection data across VMs in your cloud-native environments and sends it to Security Lake.

Integration documentation

Palo Alto Networks – XSOAR

Integration type: Suscriber

Palo Alto Networks XSOAR has built a subscriber integration with XSOAR and Security Lake.

Integration documentation

Panther

Integration type: Subscriber

Panther supports ingesting Security Lake logs for use in search and detection.

Integration documentation

Ping Identity – PingOne

Integration type: Source

PingOne sends account modification alerts to Security Lake in OCSF schema and Parquet format, allowing you to discover and act upon account changes.

Integration documentation

PwC – Fusion center

Integration type: Subscriber, Service

PwC brings knowledge and expertise to aid clients in implementing a fusion center to meet their individual needs. Built on Amazon Security Lake, a fusion center provides the ability to combine data from a variety of sources to create a centralized, near real-time view.

Integration documentation

Query.AI – Query Federated Search

Integration type: Subscriber

Query Federated Search can directly query any Security Lake table via Amazon Athena to support incident response, investigations, threat hunting, and general search across a variety of Observables, Events, and Objects in the OCSF schema.

Integration documentation

Rapid7 – InsightIDR

Integration type: Subscriber

InsightIDR, the Rapid7 SIEM/XDR solution, can ingest logs in Security Lake for threat detection and investigation of suspicious activity.

Integration documentation

RipJar – Labyrinth for Threat Investigations

Integration type: Subscriber

Labyrinth for Threat Investigations provides an enterprise-wide approach to threat exploration at scale based on data fusion, with fine-grained security, adaptable workflows, and reporting.

Integration documentation

Sailpoint

Integration type: Source

Partner product for the integration: SailPoint IdentityNow

This integration enables customers to transform event data from SailPoint IdentityNow. The integration is intended to provide an automated process to bring IdentityNow user activity and governance events into Security Lake to improve insights from security incident and event monitoring products.

Integration documentation

Securonix

Integration type: Subscriber

Securonix Next-Gen SIEM integrates with Security Lake, empowering security teams to ingest data more quickly and expand their detection and response capabilities.

Integration documentation

SentinelOne

Integration type: Subscriber

The SentinelOne Singularity™ XDR Platform extends real-time detection and response to endpoint, identity, and cloud workloads running on on-premises and public cloud infrastructure, including Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Container Service (Amazon ECS), and Amazon Elastic Kubernetes Service (Amazon EKS).

Integration documentation (sign in to the SentinelOne portal to review the documentation)

Sentra – Data Lifecyle Security Platform

Integration type: Source

After deploying the Sentra scanning infrastructure in your account, Sentra fetches findings and ingest them into your SaaS. These findings are metadata that Sentra stores and later streams to Security Lake in OCSF schema for querying.

Integration documentation

SOC Prime

Integration type: Subscriber

SOC Prime integrates with Security Lake through Amazon OpenSearch Service and Amazon Athena to facilitate smart data orchestration and threat hunting based on zero trust milestones. SOC Prime empowers security teams to increase threat visibility and investigate incidents without an overwhelming volume of alerts. You can save development time with reusable rules and queries that are automatically convertible to Athena and OpenSearch Service in the OCSF schema.

Integration documentation

Splunk

Integration type: Subscriber

The Splunk AWS Add-On for Amazon Web Services (AWS) supports ingestion from Security Lake. This integration helps you accelerate threat detection, investigation, and response by subscribing to data in OCSF schema from Security Lake.

Integration documentation

Stellar Cyber

Integration type: Subscriber

Stellar Cyber consumes logs from Security Lake and adds the records to the Stellar Cyber data lake. This connector uses OCSF schema.

Integration documentation

Sumo Logic

Integration type: Subscriber

Sumo Logic consumes data from Security Lake and provides broad visibility across AWS, on-premise, and hybrid cloud environments. Sumo Logic gives security teams comprehensive visibility, automation, and threat monitoring across all of their security tools.

Integration documentation

Swimlane – Turbine

Integration type: Subscriber

Swimlane ingests data from Security Lake in OCSF schema, and sends the data through low-code playbooks and case management to facilitate faster threat detection, investigation, and incident response.

Integration documentation (sign in to the Swimlane portal to review the documentation)

Sysdig Secure

Integration type: Source

Sysdig Secure's cloud-native application protection platform (CNAPP) sends security events to Security Lake to maximize oversight, streamline investigations, and simplify compliance.

Integration documentation

Talon

Integration type: Source

Partner product for the integration: Talon Enterprise Browser

Talon's Enterprise Browser, a secure and isolated browser-based endpoint environment, sends Talon Access, data protection, SaaS actions, and security events to Security Lake providing visibility and options to cross-correlate events for detection, forensics, and investigations.

Integration documentation (sign in to the Talon portal to review the documentation)

Tanium

Integration type: Source

Tanium Unified Cloud Endpoint Detection, Management, and Security Platform provides inventory data to Security Lake in OCSF schema.

Integration documentation

TCS

Integration type: Service

The TCS AWS Business Unit offers innovation, experience, and talent. This integration is powered by a decade of joint value creation, deep industry knowledge, technology expertise, and delivery wisdom. As a service integration, TCS can help you implement Security Lake in your organization.

Integration documentation

Tego Cyber

Integration type: Subscriber

Tego Cyber integrates with Security Lake to help you swiftly detect and investigate potential security threats. By correlating diverse threat indicators across extensive time frames and log sources, Tego Cyber uncovers hidden threats. The platform is enriched with highly contextual threat intelligence, providing precision and insight in threat detection and investigations.

Integration documentation

Tines – No-code security automation

Integration type: Subscriber

Tines No-code security automation helps you make more accurate decisions by leveraging security data centralized in Security Lake.

Integration documentation

Torq – Enterprise Security Automation Platform

Integration type: Source, Subscriber

Torq seamlessly integrates with Security Lake as both a custom source and a subscriber. Torq helps you implement enterprise-scale automation and orchestration with a simple no-code platform.

Integration documentation

Trellix – XDR

Integration type: Source, Subscriber

As an open XDR platform, Trellix XDR supports the Security Lake integration. Trellix XDR can leverage data in OCSF schema for security analytics use cases. You can also augment your Security Lake data lake with 1,000+ sources of security events in Trellix XDR. This helps you extend detection and response capabilities for your AWS environment. Ingested data is correlated with other security risks, providing you with the necessary playbooks to respond to a risk in a timely manner.

Integration documentation

Trend Micro – CloudOne

Integration type: Source

Trend Micro CloudOne Workload Security sends the following information to Security Lake from your Amazon Elastic Compute Cloud (EC2) instances:

- DNS Query activity
- File activity
- Network activity
- Process activity
- Registry Value activity
- User Account activity

Integration documentation

Uptycs – Uptycs XDR

Integration type: Source

Uptycs sends a wealth of data in OCSF schema from on-premises and cloud assets to Security Lake. The data includes behavioral threat detections from endpoints and cloud workloads, anomaly detections, policy violations, risky policies, misconfigurations, and vulnerabilities.

Integration documentation

Vectra AI – Vectra Detect for AWS

Integration type: Source

By using Vectra Detect for AWS, you can send high-fidelity alerts to Security Lake as a custom source using a dedicated AWS CloudFormation template.

Integration documentation

VMware Aria Automation for Secure Clouds

Integration type: Source

With this integration, you can detect cloud misconfigurations and send them to Security Lake for advanced analysis.

Integration documentation

Wazuh

Integration type: Subscriber

Wazuh aims to securely handle user data, provide query access for each source, and optimize querying costs.

Integration documentation

Wipro

Integration type: Source, Service

This integration allows you to collect data from the Wipro Cloud Application Risk Governance (CARG) platform to provide a unified view of your cloud applications and compliance postures across an enterprise.

As a service integration, Wipro can also help you implement Security Lake in your organization.

Integration documentation

Wiz – CNAPP

Integration type: Source

The integration between Wiz and Security Lake facilitates cloud security data collection in a single security data lake by leveraging the OCSF schema, an open source standard designed for extensible and normalized security data exchange.

Integration documentation (sign in to the Wiz portal to review the documentation)

Zscaler – Zscaler Posture Control

Integration type: Source

Zscaler Posture Control™, a cloud native application protection platform, sends security findings to Security Lake in OCSF schema.

Integration documentation

Security in Security Lake

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

Topics

Identity and access management for Security Lake

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

Topics

Audience

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

Service user – If you use the Security Lake service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Security Lake 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 Security Lake, see Troubleshooting Amazon Security Lake identity and access.

Service administrator – If you're in charge of Security Lake resources at your company, you probably have full access to Security Lake. It's your job to determine which Security Lake 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 Security Lake, see How Security Lake works with IAM.

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

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 AWS Signature Version 4 for API requests in the *IAM User Guide*.

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

AWS account root user

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

Federated identity

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

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

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

IAM users and groups

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

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

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

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. To temporarily assume an IAM role in the AWS Management Console, you can switch from a user to an IAM role (console). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Methods to assume a role in the *IAM User Guide*.

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

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

  - **Service-linked role –** A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2 –** You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

Managing access using policies

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

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

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

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

Identity-based policies

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

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

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

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

Access control lists (ACLs)

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

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

Other policy types

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

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of 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 Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) 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 Security Lake works with IAM

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

IAM features you can use with Amazon Security Lake

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

To get a high-level view of how Security Lake 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 Security Lake

Supports identity-based policies: Yes

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

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Security Lake supports identity-based policies. For more information, see Identity-based policy examples for Security Lake.

Resource-based policies within Security Lake

Supports resource-based policies: Yes

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

The Security Lake service creates resource-based policies for the Amazon S3 buckets that store your data. You don't attach these resource-based policies to your S3 buckets. Security Lake automatically creates these policies on your behalf.

An example resource is an S3 bucket with an Amazon Resource Name (ARN) of `arn:aws:s3:::aws-security-data-lake-{region}-{bucket-identifier}`. In this example, `region` is a specific AWS Region where you've enabled Security Lake, and `bucket-identifier` is a Regionally unique alphanumeric string that Security Lake assigns to the bucket. Security Lake creates the S3 bucket to store data from that Region. The resource policy defines which principals can perform actions on the bucket. Here's a sample resource-based policy (bucket policy) that Security Lake attaches to the bucket:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": {
                "AWS": "*"
            },
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::aws-security-data-lake-{region}-{bucket-identifier}/*",
                "arn:aws:s3:::aws-security-data-lake-{region}-{bucket-identifier}"
            ],
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        },
        {
            "Sid": "PutSecurityLakeObject",
            "Effect": "Allow",
            "Principal": {
                "Service": "securitylake.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::aws-security-data-lake-{region}-{bucket-identifier}/*",
                "arn:aws:s3:::aws-security-data-lake-{region}-{bucket-identifier}"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "{DA-AccountID}",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:securitylake:us-east-1:{DA-AccountID}:*"
                }
            }
        }
    ]
}
```

To learn more about resource-based policies, see Identity-based policies and resource-based policies in the *IAM User Guide*.

Policy actions for Security Lake

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.

For a list of Security Lake actions, see Actions defined by Amazon Security Lake in the *Service Authorization Reference*.

Policy actions in Security Lake use the following prefix before the action:

```
securitylake
```

For example, to grant a user permission to access information about a specific subscriber, include the `securitylake:GetSubscriber` action in the policy assigned to that user. Policy statements must include either an `Action` or `NotAction` element. Security Lake defines its own set of actions that describe tasks that you can perform with this service.

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

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

To view examples of Security Lake identity-based policies, see Identity-based policy examples for Security Lake.

Policy resources for Security Lake

Supports policy resources: Yes

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

The `Resource` JSON policy element specifies the object or objects to which the action applies. 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": "*"
```

Security Lake defines the following resource types: subscriber, and the data lake configuration for an AWS account in a particular AWS Region. You can specify these types of resources in policies by using ARNs.

For a list of Security Lake resource types and the ARN syntax for each one, see Resource types defined by Amazon Security Lake in the *Service Authorization Reference*. To learn which actions you can specify for each type of resource, see Actions defined by Amazon Security Lake in the *Service Authorization Reference*.

To view examples of Security Lake identity-based policies, see Identity-based policy examples for Security Lake.

Policy condition keys for Security Lake

Supports service-specific policy condition keys: Yes

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

The `Condition` element (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*.

For a list of Security Lake condition keys, see Condition keys for Amazon Security Lake in the *Service Authorization Reference*. To learn which actions and resources you can use a condition key with, see Actions defined by Amazon Security Lake in the *Service Authorization Reference*. For examples of policies that use condition keys, see Identity-based policy examples for Security Lake.

Access control lists (ACLs) in Security Lake

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.

Security Lake doesn't support ACLs, which means you can't attach an ACL to a Security Lake resource.

Attribute-based access control (ABAC) with Security Lake

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**.

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

You can attach tags to Security Lake resources—subscribers, and the data lake configuration for an AWS account in individual AWS Regions. You can also control access to these types of resources by providing tag information in the `Condition` element of a policy. For information about tagging Security Lake resources, see Tagging Security Lake resources. For an example of an identity-based policy that controls access to a resource based on the tags for that resource, see Identity-based policy examples for Security Lake.

Using temporary credentials with Security Lake

Supports temporary credentials: Yes

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

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

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

Security Lake supports the use of temporary credentials.

Forward access sessions for Security Lake

Supports forward access sessions (FAS): Yes

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

Some Security Lake actions require permissions for additional, dependent actions in other AWS services. For a list of these actions, see Actions defined by Amazon Security Lake in the *Service Authorization Reference*.

Service roles for Security Lake

Supports service roles: No

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

Security Lake doesn't assume or use service roles. However, related services such as Amazon EventBridge, AWS Lambda, and Amazon S3 assume service roles when you use Security Lake. To perform actions on your behalf, Security Lake uses a service-linked role.

> ⚠ **Warning**
>
> Changing the permissions for a service role may create operational issues with your use of Security Lake. Edit service roles only when Security Lake provides guidance to do so.

Service-linked roles for Security Lake

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.

Security Lake uses an IAM service-linked role named `AWSServiceRoleForAmazonSecurityLake`. The Security Lake service-linked role grants permissions to operate a security data lake service on behalf of customers. This service-linked role is an IAM role that's linked directly to Security Lake. It's predefined by Security Lake, and it includes all the permissions that Security Lake requires to call other AWS services on your behalf. Security Lake uses this service-linked role in all the AWS Regions where Security Lake is available.

For details about creating or managing the Security Lake service-linked role, see Using service-linked roles for Security Lake.

Identity-based policy examples for Security Lake

By default, users and roles don't have permission to create or modify Security Lake 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 Create IAM policies (console) in the *IAM User Guide*.

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

Topics

- Example: Allow the organization management account to designate and remove a delegated administrator
- Example: Allow users to review subscribers based on tags

Policy best practices

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

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Security Lake console

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

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

To ensure that users and roles can use the Security Lake console, create IAM policies that provide them with console access. For more information, see IAM identities in the *IAM User Guide*.

If you create a policy that allows users or roles to use the Security Lake console, ensure that the policy includes the appropriate actions for the resources that those users or roles need to access on the console. Otherwise, they won't be able to navigate to or display details about those resources on the console.

For example, to add a custom source by using the console, a user must be allowed to perform these actions:

- `glue:CreateCrawler`
- `glue:CreateDatabase`
- `glue:CreateTable`
- `glue:StartCrawlerSchedule`
- `iam:GetRole`
- `iam:PutRolePolicy`
- `iam:DeleteRolePolicy`
- `iam:PassRole`
- `lakeformation:RegisterResource`
- `lakeformation:GrantPermissions`
- `s3:ListBucket`

- `s3:PutObject`

Example: Allow users to view their own permissions

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

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Example: Allow the organization management account to designate and remove a delegated administrator

This example shows how you might create a policy that allows a user of an AWS Organizations management account to designate and remove the delegated Security Lake administrator for their organization.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "securitylake:RegisterDataLakeDelegatedAdministrator",
                "securitylake:DeregisterDataLakeDelegatedAdministrator"
            ],
            "Resource": "arn:aws:securitylake:*:*:*"
        }
    ]
}
```

Example: Allow users to review subscribers based on tags

In identity-based policies, you can use conditions to control access to Security Lake resources based on tags. This example shows how you might create a policy that allows a user to review subscribers by using the Security Lake console or the Security Lake API. However, permission is granted only if the value for the `Owner` tag for a subscriber is the user's username.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReviewSubscriberDetailsIfOwner",
            "Effect": "Allow",
            "Action": "securitylake:GetSubscriber",
            "Resource": "arn:aws:securitylake:*:*:subscriber/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
            }
        },
        {
            "Sid": "ListSubscribersIfOwner",
            "Effect": "Allow",
            "Action": "securitylake:ListSubscribers",
            "Resource": "*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

In this example, if a user who has the username `richard-roe` attempts to review the details of individual subscribers, a subscriber must be tagged `Owner=richard-roe` or `owner=richard-roe`. Otherwise, the user is denied access. The condition tag key `Owner` matches both `Owner` and `owner` because condition key names are not case sensitive. For more information about using condition keys, see IAM JSON policy elements: Condition in the *IAM User Guide*. For information about tagging Security Lake resources, see Tagging Security Lake resources.

AWS managed policies for Security Lake

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AmazonSecurityLakeMetastoreManager

Amazon Security Lake uses an AWS Lambda function to manage metadata in your data lake. Through the use of this function, Security Lake can index Amazon Simple Storage Service (Amazon S3) partitions that contain your data and data files into the AWS Glue Data Catalog tables. This managed policy contains all of the permissions for the Lambda function to index the S3 partitions and data files into the AWS Glue tables.

Permissions details

This policy includes the following permissions:

- `logs` – Allows principals to log the output of the Lambda function to Amazon CloudWatch Logs.
- `glue` – Allows principals to perform specific write actions for AWS Glue Data Catalog tables. This also allows AWS Glue crawlers to identify partitions in your data.
- `sqs` – Allows principals to perform specific read and write actions for Amazon SQS queues that send event notifications when objects are added to or updated in your data lake.
- `s3` – Allows principals to perform specific read and write actions for the Amazon S3 bucket that contains your data.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowWriteLambdaLogs",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:CreateLogGroup"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/lambda/AmazonSecurityLake*",
        "arn:aws:logs:*:*:/aws/lambda/AmazonSecurityLake*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowGlueManage",
      "Effect": "Allow",
      "Action": [
        "glue:CreatePartition",
        "glue:BatchCreatePartition",
        "glue:GetTable",
        "glue:UpdateTable"
      ],
      "Resource": [
        "arn:aws:glue:*:*:table/amazon_security_lake_glue_db*/*",
        "arn:aws:glue:*:*:database/amazon_security_lake_glue_db*",
        "arn:aws:glue:*:*:catalog"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowToReadFromSqs",
      "Effect": "Allow",
      "Action": [
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage",
        "sqs:GetQueueAttributes"
      ],
      "Resource": [
        "arn:aws:sqs:*:*:AmazonSecurityLake*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowMetaDataReadWrite",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-security-data-lake*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowMetaDataCleanup",
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-security-data-lake*/metadata/*.avro",
        "arn:aws:s3:::aws-security-data-lake*/metadata/*.metadata.json"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

AWS managed policy: AmazonSecurityLakePermissionsBoundary

Amazon Security Lake creates IAM roles for third-party custom sources to write data to the data lake and for third-party custom subscribers to consume data from the data lake, and uses this policy when creating these roles to define the boundary of their permissions. You don't need to take action to use this policy. If the data lake is encrypted with a customer managed AWS KMS key, `kms:Decrypt` and `kms:GenerateDataKey` permissions are added.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowActionsForSecurityLake",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucket",
        "s3:ListBucketVersions",
        "s3:PutObject",
        "s3:GetBucketLocation",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "sqs:ReceiveMessage",
        "sqs:ChangeMessageVisibility",
        "sqs:DeleteMessage",
        "sqs:GetQueueUrl",
        "sqs:SendMessage",
        "sqs:GetQueueAttributes",
        "sqs:ListQueues"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyActionsForSecurityLake",
      "Effect": "Deny",
      "NotAction": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucket",
        "s3:ListBucketVersions",
        "s3:PutObject",
        "s3:GetBucketLocation",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "sqs:ReceiveMessage",
        "sqs:ChangeMessageVisibility",
        "sqs:DeleteMessage",
        "sqs:GetQueueUrl",
        "sqs:SendMessage",
        "sqs:GetQueueAttributes",
        "sqs:ListQueues"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyActionsNotOnSecurityLakeBucket",
      "Effect": "Deny",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucket",
        "s3:ListBucketVersions",
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "NotResource": [
        "arn:aws:s3:::aws-security-data-lake*"
      ]
    },
    {
      "Sid": "DenyActionsNotOnSecurityLakeSQS",
      "Effect": "Deny",
      "Action": [
        "sqs:ReceiveMessage",
        "sqs:ChangeMessageVisibility",
        "sqs:DeleteMessage",
        "sqs:GetQueueUrl",
        "sqs:SendMessage",
        "sqs:GetQueueAttributes",
        "sqs:ListQueues"
      ],
      "NotResource": "arn:aws:sqs:*:*:AmazonSecurityLake*"
    },
    {
      "Sid": "DenyActionsNotOnSecurityLakeKMSS3SQS",
      "Effect": "Deny",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotLike": {
          "kms:ViaService": [
            "s3.*.amazonaws.com",
            "sqs.*.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "DenyActionsNotOnSecurityLakeKMSForS3",
      "Effect": "Deny",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "kms:EncryptionContext:aws:s3:arn": "false"
        },
        "StringNotLikeIfExists": {
          "kms:EncryptionContext:aws:s3:arn": [
            "arn:aws:s3:::aws-security-data-lake*"
          ]
        }
      }
    },
    {
      "Sid": "DenyActionsNotOnSecurityLakeKMSForS3SQS",
      "Effect": "Deny",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "kms:EncryptionContext:aws:sqs:arn": "false"
        },
        "StringNotLikeIfExists": {
          "kms:EncryptionContext:aws:sqs:arn": [
            "arn:aws:sqs:*:*:AmazonSecurityLake*"
          ]
        }
      }
    }
  ]
}
```

AWS managed policy: AmazonSecurityLakeAdministrator

You can attach the `AmazonSecurityLakeAdministrator` policy to a principal before they enable Amazon Security Lake for their account. This policy grants administrative permissions that allow a principal full access to all Security Lake actions. The principal can then onboard to Security Lake and subsequently configure sources and subscribers in Security Lake.

This policy includes the actions that Security Lake administrators can perform on other AWS services through Security Lake.

The `AmazonSecurityLakeAdministrator` policy does not support the creation of utility roles required by Security Lake to manage Amazon S3 cross-region replication, registration of new data partitions in AWS Glue, run a Glue crawler on data added to custom sources, or notify HTTPS endpoint subscribers of new data. You can create these roles ahead of time as described in Getting started with Amazon Security Lake.

In addition to the `AmazonSecurityLakeAdministrator` managed policy, Security Lake requires `lakeformation:PutDataLakeSettings` permissions for onboarding and configuration functions. `PutDataLakeSettings` allows setting an IAM principal as an administrator for all regional Lake Formation resources in the account. This role has to have `iam:CreateRole permission` as well as `AmazonSecurityLakeAdministrator` policy attached to it.

Lake Formation administrators have full access to the Lake Formation console, and control the initial data configuration and access permissions. Security Lake assigns the principal that enables Security Lake and the `AmazonSecurityLakeMetaStoreManager` role (or other specified role) as Lake Formation administrators so that they can create tables, update table schema, register new partitions, and configure permissions on tables. You must include the following permissions in the policy for the Security Lake administrator user or role:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutLakeFormationSettings",
            "Effect": "Allow",
            "Action": "lakeformation:PutDatalakeSettings",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": "securitylake.amazonaws.com"
                }
            }
        }
    ]
}
```

Permissions details

This policy includes the following permissions.

- `securitylake` – Allows principals full access to all Security Lake actions.
- `organizations` – Allows principals to retrieve information from AWS Organizations about the accounts in an organization. If an account belongs to an organization, then these permissions allow the Security Lake console to display account names and account numbers.
- `iam` – Allows principals to create service-linked roles for Security Lake, AWS Lake Formation, and Amazon EventBridge, as a required step when enabling those services. Also allows for creation and editing of policies for subscriber and custom source roles, with permissions of those roles limited to what is allowed by the `AmazonSecurityLakePermissionsBoundary` policy.
- `ram` – Allows principals to configure Lake Formation-based query access by subscribers to Security Lake sources.
- `s3`– Allows principals to create and manage Security Lake buckets, and read the contents of those buckets.
- `lambda` – Allows principals to manage the Lambda used to update AWS Glue table partitions following AWS source delivery and cross-region replication.
- `glue` – Allows principals to create and manage the Security Lake database and tables.
- `lakeformation` – Allows principals to manage Lake Formation permissions for Security Lake tables.
- `events` – Allows principals to manage rules used to notify subscribers of new data in Security Lake sources.
- `sqs` – Allows principals to create and manage Amazon SQS queues used to notify subscribers of new data in Security Lake sources.
- `kms` – Allows principals to grant access for Security Lake to write data using a customer-managed key.

- `secretsmanager` – Allows principals to manage secrets used for notifying subscribers of new data in Security Lake sources via HTTPS endpoints.

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowActionsWithAnyResource",
    "Effect": "Allow",
    "Action": [
      "securitylake:*",
      "organizations:DescribeOrganization",
      "organizations:ListDelegatedServicesForAccount",
      "organizations:ListAccounts",
      "iam:ListRoles",
      "ram:GetResourceShareAssociations"
    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowActionsWithAnyResourceViaSecurityLake",
    "Effect": "Allow",
    "Action": [
      "glue:CreateCrawler",
      "glue:StopCrawlerSchedule",
      "lambda:CreateEventSourceMapping",
      "lakeformation:GrantPermissions",
      "lakeformation:ListPermissions",
      "lakeformation:RegisterResource",
      "lakeformation:RevokePermissions",
      "lakeformation:GetDatalakeSettings",
      "events:ListConnections",
      "events:ListApiDestinations",
      "iam:GetRole",
      "iam:ListAttachedRolePolicies",
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowManagingSecurityLakeS3Buckets",
    "Effect": "Allow",
    "Action": [
      "s3:CreateBucket",
      "s3:PutBucketPolicy",
      "s3:PutBucketPublicAccessBlock",
      "s3:PutBucketNotification",
      "s3:PutBucketTagging",
      "s3:PutEncryptionConfiguration",
      "s3:PutBucketVersioning",
      "s3:PutReplicationConfiguration",
      "s3:PutLifecycleConfiguration",
      "s3:ListBucket",
      "s3:PutObject",
      "s3:GetBucketNotification"
    ],
    "Resource": "arn:aws:s3:::aws-security-data-lake*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowLambdaCreateFunction",
    "Effect": "Allow",
    "Action": [
      "lambda:CreateFunction"
    ],
    "Resource": [
      "arn:aws:lambda:*:*:function:SecurityLake_Glue_Partition_Updater_Lambda*",
      "arn:aws:lambda:*:*:function:AmazonSecurityLake*"
    ],
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowLambdaAddPermission",
    "Effect": "Allow",
    "Action": [
      "lambda:AddPermission"
    ],
    "Resource": [
      "arn:aws:lambda:*:*:function:SecurityLake_Glue_Partition_Updater_Lambda*",
      "arn:aws:lambda:*:*:function:AmazonSecurityLake*"
    ],
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      },
      "StringEquals": {
        "lambda:Principal": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowGlueActions",
    "Effect": "Allow",
    "Action": [
      "glue:CreateDatabase",
      "glue:GetDatabase",
      "glue:CreateTable",
      "glue:GetTable"
    ],
    "Resource": [
      "arn:aws:glue:*:*:catalog",
      "arn:aws:glue:*:*:database/amazon_security_lake_glue_db*",
      "arn:aws:glue:*:*:table/amazon_security_lake_glue_db*/*"
    ],
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowEventBridgeActions",
    "Effect": "Allow",
    "Action": [
      "events:PutTargets",
      "events:PutRule",
      "events:DescribeRule",
      "events:CreateApiDestination",
      "events:CreateConnection",
      "events:UpdateConnection",
      "events:UpdateApiDestination",
      "events:DeleteConnection",
      "events:DeleteApiDestination",
      "events:ListTargetsByRule",
      "events:RemoveTargets",
      "events:DeleteRule"
    ],
    "Resource": [
      "arn:aws:events:*:*:rule/AmazonSecurityLake*",
      "arn:aws:events:*:*:rule/SecurityLake*",
      "arn:aws:events:*:*:api-destination/AmazonSecurityLake*",
      "arn:aws:events:*:*:connection/AmazonSecurityLake*"
    ],
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowSQSActions",
    "Effect": "Allow",
    "Action": [
      "sqs:CreateQueue",
      "sqs:SetQueueAttributes",
      "sqs:GetQueueURL",
      "sqs:AddPermission",
      "sqs:GetQueueAttributes",
      "sqs:DeleteQueue"
    ],
    "Resource": [
      "arn:aws:sqs:*:*:SecurityLake*",
      "arn:aws:sqs:*:*:AmazonSecurityLake*"
    ],
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowKmsCmkGrantForSecurityLake",
    "Effect": "Allow",
    "Action": "kms:CreateGrant",
    "Resource": "arn:aws:kms:*:*:key/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      },
      "StringLike": {
        "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::aws-security-data-lake*"
      },
      "ForAllValues:StringEquals": {
        "kms:GrantOperations": [
          "GenerateDataKey",
          "RetireGrant",
          "Decrypt"
        ]
      }
    }
  },
  {
    "Sid": "AllowEnablingQueryBasedSubscribers",
    "Effect": "Allow",
    "Action": [
      "ram:CreateResourceShare",
      "ram:AssociateResourceShare"
    ],
    "Resource": "*",
    "Condition": {
      "StringLikeIfExists": {
        "ram:ResourceArn": [
          "arn:aws:glue:*:*:catalog",
          "arn:aws:glue:*:*:database/amazon_security_lake_glue_db*",
          "arn:aws:glue:*:*:table/amazon_security_lake_glue_db*/*"
        ]
      },
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowConfiguringQueryBasedSubscribers",
    "Effect": "Allow",
    "Action": [
      "ram:UpdateResourceShare",
      "ram:GetResourceShares",
      "ram:DisassociateResourceShare",
      "ram:DeleteResourceShare"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "ram:ResourceShareName": "LakeFormation*"
      },
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowConfiguringCredentialsForSubscriberNotification",
    "Effect": "Allow",
    "Action": [
      "secretsmanager:CreateSecret",
      "secretsmanager:GetSecretValue",
      "secretsmanager:PutSecretValue"
    ],
    "Resource": "arn:aws:secretsmanager:*:*:secret:events!connection/
AmazonSecurityLake-*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowPassRoleForUpdatingGluePartitionsSecLakeArn",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
      "arn:aws:iam::*:role/service-role/AmazonSecurityLakeMetaStoreManager",
      "arn:aws:iam::*:role/service-role/AmazonSecurityLakeMetaStoreManagerV2"
    ],
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "lambda.amazonaws.com"
      },
      "StringLike": {
        "iam:AssociatedResourceARN": "arn:aws:securitylake:*:*:data-lake/default"
      }
    }
  },
  {
    "Sid": "AllowPassRoleForUpdatingGluePartitionsLambdaArn",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
      "arn:aws:iam::*:role/service-role/AmazonSecurityLakeMetaStoreManager",
      "arn:aws:iam::*:role/service-role/AmazonSecurityLakeMetaStoreManagerV2"
    ],
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "lambda.amazonaws.com"
      },
      "StringLike": {
        "iam:AssociatedResourceARN": [
          "arn:aws:lambda:*:*:function:SecurityLake_Glue_Partition_Updater_Lambda*",
          "arn:aws:lambda:*:*:function:AmazonSecurityLake*"
        ]
      },
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowPassRoleForCrossRegionReplicationSecLakeArn",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/service-role/AmazonSecurityLakeS3ReplicationRole",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "s3.amazonaws.com"
      },
      "StringLike": {
        "iam:AssociatedResourceARN": "arn:aws:securitylake:*:*:data-lake/default"
      }
    }
  },
  {
    "Sid": "AllowPassRoleForCrossRegionReplicationS3Arn",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/service-role/AmazonSecurityLakeS3ReplicationRole",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "s3.amazonaws.com"
      },
      "StringLike": {
        "iam:AssociatedResourceARN": "arn:aws:s3:::aws-security-data-lake*"
      },
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowPassRoleForCustomSourceCrawlerSecLakeArn",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/service-role/
AmazonSecurityLakeCustomDataGlueCrawler*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "glue.amazonaws.com"
      },
      "StringLike": {
        "iam:AssociatedResourceARN": "arn:aws:securitylake:*:*:data-lake/default"
      }
    }
  },
  {
    "Sid": "AllowPassRoleForCustomSourceCrawlerGlueArn",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/service-role/
AmazonSecurityLakeCustomDataGlueCrawler*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "glue.amazonaws.com"
      },
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowPassRoleForSubscriberNotificationSecLakeArn",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/service-role/
AmazonSecurityLakeSubscriberEventBridge",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "events.amazonaws.com"
      },
      "StringLike": {
        "iam:AssociatedResourceARN": "arn:aws:securitylake:*:*:subscriber/*"
      }
    }
  },
  {
    "Sid": "AllowPassRoleForSubscriberNotificationEventsArn",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/service-role/
AmazonSecurityLakeSubscriberEventBridge",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "events.amazonaws.com"
      },
      "StringLike": {
        "iam:AssociatedResourceARN": "arn:aws:events:*:*:rule/AmazonSecurityLake*"
      },
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowOnboardingToSecurityLakeDependencies",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": [
      "arn:aws:iam::*:role/aws-service-role/securitylake.amazonaws.com/
AWSServiceRoleForSecurityLake",
      "arn:aws:iam::*:role/aws-service-role/lakeformation.amazonaws.com/
AWSServiceRoleForLakeFormationDataAccess",
      "arn:aws:iam::*:role/aws-service-role/apidestinations.events.amazonaws.com/
AWSServiceRoleForAmazonEventBridgeApiDestinations"
    ],
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": [
          "securitylake.amazonaws.com",
          "lakeformation.amazonaws.com",
          "apidestinations.events.amazonaws.com"
        ]
      }
    }
  },
  {
    "Sid": "AllowRolePolicyActionsforSubscibersandSources",
    "Effect": "Allow",
    "Action": [
      "iam:CreateRole",
      "iam:PutRolePolicy",
      "iam:DeleteRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/AmazonSecurityLake*",
    "Condition": {
      "StringEquals": {
        "iam:PermissionsBoundary": "arn:aws:iam::aws:policy/
AmazonSecurityLakePermissionsBoundary"
      },
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowRegisterS3LocationInLakeFormation",
    "Effect": "Allow",
    "Action": [
      "iam:PutRolePolicy",
      "iam:GetRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/lakeformation.amazonaws.com/
AWSServiceRoleForLakeFormationDataAccess",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "AllowIAMActionsByResource",
    "Effect": "Allow",
    "Action": [
      "iam:ListRolePolicies",
      "iam:DeleteRole"
    ],
    "Resource": "arn:aws:iam::*:role/AmazonSecurityLake*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "securitylake.amazonaws.com"
      }
    }
  },
  {
    "Sid": "S3ReadAccessToSecurityLakes",
    "Effect": "Allow",
    "Action": [
      "s3:Get*",
      "s3:List*"
    ],
    "Resource": "arn:aws:s3:::aws-security-data-lake-*"
  },
  {
    "Sid": "S3ReadAccessToSecurityLakeMetastoreObject",
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::security-lake-meta-store-manager-*"
  },
  {
    "Sid": "S3ResourcelessReadOnly",
    "Effect": "Allow",
    "Action": [
      "s3:GetAccountPublicAccessBlock",
      "s3:ListAccessPoints",
      "s3:ListAllMyBuckets"
    ],
    "Resource": "*"
  }
]
}
```

AWS managed policy: SecurityLakeServiceLinkedRole

Security Lake uses the service-linked role named `AWSServiceRoleForSecurityLake` to create and operate the security data lake.

You can't attach the `SecurityLakeServiceLinkedRole` managed policy to your IAM entities. This policy is attached to a service-linked role that permits Security Lake to perform actions on your behalf. For more information, see Service-linked role permissions for Security Lake.

AWS managed policy: SecurityLakeResourceManagementServiceRolePolicy

Security Lake uses the service-linked role named `AWSServiceRoleForSecurityLakeResourceManagement` to perform ongoing monitoring and performance improvements, which can reduce latency and costs.

You can't attach the `SecurityLakeResourceManagementServiceRolePolicy` managed policy to your IAM entities. This policy is attached to a service-linked role that permits Security Lake to perform actions on your behalf. For more information, see Service-linked role permissions for resource management.

AWS managed policy: AWSGlueServiceRole

The `AWSGlueServiceRole` managed policy invokes the AWS Glue crawler and permits AWS Glue to crawl custom source data and identify partition metadata. This metadata is necessary to create and update tables in the Data Catalog.

For more information, see Collecting data from custom sources in Security Lake.

Security Lake updates to AWS managed policies

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

Change	Description	Date
Service-linked role for Amazon Security Lake – New service-linked role	We added a new service-linked role `AWSServiceRoleForSecurityLakeResourceManagement` . This service-linked role provides permissions to Security Lake to perform ongoing monitoring and performance improvements, which can reduce latency and costs.	November 14, 2024
Service-linked role for Amazon Security Lake – Update to existing service-linked role permissions	We added AWS WAF actions to the AWS managed policy for the `SecurityLakeServiceLinkedRole` policy. The additional actions allow Security Lake to collect AWS WAF logs, when it is enabled as a log source in Security Lake.	May 22, 2024
AmazonSecurityLakePermissionsBoundary – Update to an existing policy	Security Lake added SID actions to the policy.	May 13, 2024
AmazonSecurityLakeMetastoreManager – Update to an existing policy	Security Lake updated the policy to add metadata clean up action which lets you delete the metadata in your data lake.	March 27, 2024
AmazonSecurityLakeAdministrator – Update to an existing policy	Security Lake updated the policy to allow `iam:PassR`	February 23, 2024

Change	Description	Date
	ole on the new AmazonSecurityLakeMetastoreManagerV2 role and lets Security Lake deploy or update data lake components.	
AmazonSecurityLakeMetastoreManager – New policy	Security Lake added a new managed policy that grants permissions for Security Lake to manage metadata in your data lake.	January 23, 2024
AmazonSecurityLakeAdministrator – New policy	Security Lake added a new managed policy that grants a principal full access to all Security Lake actions.	May 30, 2023
Security Lake started tracking changes	Security Lake started tracking changes for its AWS managed policies.	November 29, 2022

Using service-linked roles for Security Lake

Security Lake uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is an IAM role that's linked directly to Security Lake. It's predefined by Security Lake, and it includes all the permissions that Security Lake requires to call other AWS services on your behalf and operate the security data lake service. Security Lake uses this service-linked role in all the AWS Regions where Security Lake is available.

The service-linked role eliminates the need to manually add the necessary permissions when setting up Security Lake. Security Lake defines the permissions of this service-linked role, and unless defined otherwise, only Security Lake can assume the role. 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 must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*. You can delete a service-linked role only after you delete its related resources. This protects your 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 review the service-linked role documentation for that service.

Topics

Service-linked role permissions for Security Lake

Security Lake uses the service-linked role named `AWSServiceRoleForSecurityLake`. This service-linked role trusts the `securitylake.amazonaws.com` service to assume the role. For more information about, AWS managed policies for Amazon Security Lake, see AWS manage policies for Amazon Security Lake.

The permissions policy for the role, which is an AWS managed policy named `SecurityLakeServiceLinkedRole`, allows Security Lake to create and operate the security data lake. It also allows Security Lake to perform tasks such as the following on the specified resources:

- Use AWS Organizations actions to retrieve information about associated accounts
- Use Amazon Elastic Compute Cloud (Amazon EC2) to retrieve information about Amazon VPC Flow Logs
- Use AWS CloudTrail actions to retrieve information about the service-linked role
- Use AWS WAF actions to collect AWS WAF logs, when it is enabled as a log source in Security Lake
- Use `LogDelivery` action to create or delete an AWS WAF log delivery subscription.

The role is configured with the following permissions policy:

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "OrganizationsPolicies",
            "Effect": "Allow",
            "Action": [
                "organizations:ListAccounts",
                "organizations:DescribeOrganization"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DescribeOrgAccounts",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeAccount"
            ],
            "Resource": [
                "arn:aws:organizations::*:account/o-*/*"
            ]
        },
        {
            "Sid": "AllowManagementOfServiceLinkedChannel",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:CreateServiceLinkedChannel",
                "cloudtrail:DeleteServiceLinkedChannel",
                "cloudtrail:GetServiceLinkedChannel",
                "cloudtrail:UpdateServiceLinkedChannel"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:channel/aws-service-channel/security-
lake/*"
        },
        {
            "Sid": "AllowListServiceLinkedChannel",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:ListServiceLinkedChannels"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DescribeAnyVpc",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ListDelegatedAdmins",
            "Effect": "Allow",
            "Action": [
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "organizations:ServicePrincipal": "securitylake.amazonaws.com"
                }
            }
        },
        {
            "Sid": "AllowWafLoggingConfiguration",
            "Effect": "Allow",
            "Action": [
                "wafv2:PutLoggingConfiguration",
                "wafv2:GetLoggingConfiguration",
                "wafv2:ListLoggingConfigurations",
                "wafv2:DeleteLoggingConfiguration"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "wafv2:LogScope": "SecurityLake"
                }
            }
        },
        {
            "Sid": "AllowPutLoggingConfiguration",
            "Effect": "Allow",
            "Action": [
                "wafv2:PutLoggingConfiguration"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "wafv2:LogDestinationResource": "arn:aws:s3:::aws-waf-logs-
security-lake-*"
                }
            }
        },
        {
            "Sid": "ListWebACLs",
            "Effect": "Allow",
            "Action": [
                "wafv2:ListWebACLs"
            ],
            "Resource": "*"
        },
        {
            "Sid": "LogDelivery",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "wafv2.amazonaws.com"
                    ]
                }
            }
        }

    ]
}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Creating the Security Lake service-linked role

You don't need to manually create the `AWSServiceRoleForSecurityLake` service-linked role for Security Lake. When you enable Security Lake for your AWS account, Security Lake automatically creates the service-linked role for you.

Editing the Security Lake service-linked role

Security Lake doesn't allow you to edit the `AWSServiceRoleForSecurityLake` service-linked role. After a service-linked role is created, 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 the Security Lake service-linked role

You cannot delete the service-linked role from Security Lake. Instead, you may delete the service-linked role from the IAM console, API, or AWS CLI. For more information, see Deleting a service-linked role in the *IAM User Guide*.

Before you can delete the service-linked role, you must first confirm that the role has no active sessions and remove any resources that `AWSServiceRoleForSecurityLake` is using.

> **Note**
>
> If Security Lake is using the `AWSServiceRoleForSecurityLake` role when you try to delete the resources, the deletion might fail. If that happens, wait a few minutes and then try the operation again.

If you delete the `AWSServiceRoleForSecurityLake` service-linked role and need to create it again, you can create it again by enabling Security Lake for your account. When you enable Security Lake again, Security Lake automatically creates the service-linked role again for you.

Supported AWS Regions for the Security Lake service-linked role

Security Lake supports using the `AWSServiceRoleForSecurityLake` service-linked role in all the AWS Regions where Security Lake is available. For a list of Regions where Security Lake is currently available, see Security Lake Regions and endpoints.

Service-linked role permissions for resource management

Security Lake uses the service-linked role named `AWSServiceRoleForSecurityLakeResourceManagement` to perform ongoing monitoring and performance improvements, which can reduce latency and costs. This service-linked role trusts the `resource-management.securitylake.amazonaws.com` service to assume the role. Enabling

`AWSServiceRoleForSecurityLakeResourceManagement` will also grant it access to Lake Formation and automatically register your Security Lake managed S3 buckets with Lake Formation across all Regions for improved security.

The permissions policy for the role, which is an AWS managed policy named `SecurityLakeResourceManagementServiceRolePolicy`, allows access to manage resources created by Security Lake; including managing the metadata in your data lake. For more information about, AWS managed policies for Amazon Security Lake, see AWS managed policies for Amazon Security Lake.

This service-linked role allows Security Lake to monitor the health of the resources deployed by Security Lake (S3 Bucket, AWS Glue tables, Amazon SQS Queue, Metastore Manager (MSM) Lambda Function, and EventBridge rules) to your account. Some examples of operations that Security Lake can perform with this service-linked role are:

- Apache Iceberg manifest file compaction, which improves query performance and lowers Lambda MSM processing times and costs.
- Monitor the state of Amazon SQS to detect ingestion issues.
- Optimize cross region data replication to exclude metadata files.

Note

If you do not install the `AWSServiceRoleForSecurityLakeResourceManagement` service-linked role, Security Lake will continue to function but it's highly recommended to accept this service-linked role so Security Lake can monitor and optimize the resources in your account.

Permissions details

The role is configured with the following permissions policy:

- `events` – Allows principals to manage EventBridge rules required for log sources and log subscribers.
- `lambda` – Allows principals to manage the lambda used to update AWS Glue table partitions following AWS source delivery and cross-region replication.

- `glue` – Allows principals to perform specific write actions for AWS Glue Data Catalog tables. This also allows AWS Glue crawlers to identify partitions in your data, and allows Security Lake to manage Apache Iceberg metadata for your Apache Iceberg tables.
- `s3` – Allows principals to perform specific read and write actions on the Security Lake buckets containing log data and Glue table metadata.
- `logs` – Allows principals read access to log the output of the Lambda function to CloudWatch Logs.
- `sqs` – Allows principals to perform specific read and write actions for Amazon SQS queues that receive event notifications when objects are added to or updated in your data lake.
- `lakeformation` – Allows principals to read Lake Formation settings to monitor for misconfigurations.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadEventBridgeRules",
      "Effect": "Allow",
      "Action": [
        "events:ListRules"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "ManageSecurityLakeEventRules",
      "Effect": "Allow",
      "Action": [
        "events:PutRule"
      ],
      "Resource": "arn:aws:events:*:*:rule/AmazonSecurityLake-*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "ManageSecurityLakeLambdaConfigurations",
      "Effect": "Allow",
      "Action": [
        "lambda:GetEventSourceMapping",
        "lambda:GetFunction",
        "lambda:PutFunctionConcurrency",
        "lambda:GetProvisionedConcurrencyConfig",
        "lambda:GetFunctionConcurrency",
        "lambda:GetRuntimeManagementConfig",
        "lambda:PutProvisionedConcurrencyConfig",
        "lambda:PublishVersion",
        "lambda:DeleteFunctionConcurrency",
        "lambda:DeleteEventSourceMapping",
        "lambda:GetAlias",
        "lambda:GetPolicy",
        "lambda:GetFunctionConfiguration",
        "lambda:UpdateFunctionConfiguration"
      ],
      "Resource": [
        "arn:aws:lambda:*:*:function:SecurityLake_Glue_Partition_Updater_Lambda*",
        "arn:aws:lambda:*:*:function:AmazonSecurityLakeMetastoreManager-*-*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowListLambdaEventSourceMappings",
      "Effect": "Allow",
      "Action": [
        "lambda:ListEventSourceMappings"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowUpdateLambdaEventSourceMapping",
      "Effect": "Allow",
      "Action": [
        "lambda:UpdateEventSourceMapping"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        },
        "StringLike": {
          "lambda:FunctionArn":
"arn:aws:lambda:*:*:function:AmazonSecurityLakeMetastoreManager-*-*"
        }
      }
    },
    {
      "Sid": "AllowUpdateLambdaConfigs",
      "Effect": "Allow",
      "Action": [
        "lambda:UpdateFunctionConfiguration"
      ],
      "Resource": "arn:aws:lambda:*:*:function:AmazonSecurityLakeMetastoreManager-*-*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "ManageSecurityLakeGlueResources",
      "Effect": "Allow",
      "Action": [
        "glue:CreatePartition",
        "glue:BatchCreatePartition",
        "glue:GetTable",
        "glue:GetTables",
        "glue:UpdateTable",
        "glue:GetDatabase"
      ],
      "Resource": [
        "arn:aws:glue:*:*:table/amazon_security_lake_glue_db*/*",
        "arn:aws:glue:*:*:database/amazon_security_lake_glue_db*",
        "arn:aws:glue:*:*:catalog"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowDataLakeConfigurationManagement",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObjectAttributes",
        "s3:GetBucketNotification",
        "s3:PutBucketNotification",
        "s3:GetLifecycleConfiguration",
        "s3:PutLifecycleConfiguration",
        "s3:GetEncryptionConfiguration",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::aws-security-data-lake*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowMetaDataCompactionAndManagement",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:RestoreObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-security-data-lake*/metadata/*.avro",
        "arn:aws:s3:::aws-security-data-lake*/metadata/*.metadata.json"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "ReadSecurityLakeLambdaLogs",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogStreams",
        "logs:StartQuery",
        "logs:GetLogEvents",
        "logs:GetQueryResults",
        "logs:GetLogRecord"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/lambda/AmazonSecurityLakeMetastoreManager-*-*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "ManageSecurityLakeSQSQueue",
      "Effect": "Allow",
      "Action": [
        "sqs:StartMessageMoveTask",
        "sqs:DeleteMessage",
        "sqs:GetQueueUrl",
        "sqs:ListDeadLetterSourceQueues",
        "sqs:ChangeMessageVisibility",
        "sqs:ListMessageMoveTasks",
        "sqs:ReceiveMessage",
        "sqs:SendMessage",
        "sqs:GetQueueAttributes",
        "sqs:SetQueueAttributes"
      ],
      "Resource": [
        "arn:aws:sqs:*:*:SecurityLake_*",
        "arn:aws:sqs:*:*:AmazonSecurityLakeManager-*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    },
    {
      "Sid": "AllowDataLakeManagement",
      "Effect": "Allow",
      "Action": [
        "lakeformation:GetDataLakeSettings",
        "lakeformation:ListPermissions"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Creating the Security Lake service-linked role

You can create the `AWSServiceRoleForSecurityLakeResourceManagement` service-linked role for Security Lake using the Security Lake console or the AWS CLI.

To create the service-linked role you must grant the following permissions to your IAM user or IAM role. The IAM role must be a Lake Formation administrator in all Security Lake enabled Regions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLakeFormationActionsViaSecurityLakeConsole",
            "Effect": "Allow",
            "Action": [
                "lakeformation:GrantPermissions",
                "lakeformation:ListPermissions",
                "lakeformation:ListResources",
                "lakeformation:RegisterResource",
                "lakeformation:RevokePermissions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowIamActionsViaSecurityLakeConsole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:GetPolicyVersion",
                "iam:GetRole"
            ],
            "Resource": [
                "arn:*:iam::*:role/aws-service-role/resource-
management.securitylake.amazonaws.com/AWSServiceRoleForSecurityLakeResourceManagement",
                "arn:*:iam::aws:policy/service-role/AWSGlueServiceRole",
                "arn:*:iam::aws:policy/service-role/
AmazonSecurityLakeMetastoreManager",
                "arn:*:iam::aws:policy/aws-service-role/
SecurityLakeResourceManagementServiceRolePolicy"
            ],
            "Condition": {
                "StringLikeIfExists": {
                    "iam:AWSServiceName": [
                        "securitylake.amazonaws.com",
                        "resource-management.securitylake.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AllowGlueActionsViaConsole",
            "Effect": "Allow",
            "Action": [
                "glue:GetTables"
            ],
            "Resource": [
                "arn:*:glue:*:*:catalog",
                "arn:*:glue:*:*:database/amazon_security_lake_glue_db*",
                "arn:*:glue:*:*:table/amazon_security_lake_glue_db*/*"
            ]
        }
```

```
    ]
}
```

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Accept the new service-linked role by clicking **Enable service-linked role** in the information bar on the Summary page.

Once you've enabled the service-linked role, you won't need to repeat this process for future use of Security Lake.

CLI

To create the AWSServiceRoleForSecurityLakeResourceManagement service-linked role programatically, use the following CLI command.

```
$ aws iam create-service-linked-role
--aws-service-name resource-management.securitylake.amazonaws.com
```

When creating the AWSServiceRoleForSecurityLakeResourceManagement service-linked role using AWS CLI, you must also grant it Lake Formation table-level permissions (ALTER, DESCRIBE) to all tables on the Security Lake Glue database to manage table metadata and access data. If Glue tables in any region reference S3 buckets from previous Security Lake enablement, you must temporarily allow DATA_LOCATION_ACCESS permissions to the service-linked role to allow Security Lake to remediate this situation.

You also have to grant Lake Formation permissions to the AWSServiceRoleForSecurityLakeResourceManagement service-linked role for your account.

The following example shows how to grant the Lake Formation permissions to the service-linked role in the designated Region. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws lakeformation grant-permissions --region {region} --principal
 DataLakePrincipalIdentifier={AWSServiceRoleForSecurityLakeResourceManagement ARN} \
--permissions ALTER DESCRIBE --resource '{ "Table": { "DatabaseName":
 "amazon_security_lake_glue_db_{region}", "TableWildcard": {} } }'
```

The following example shows how the Role ARN will look like. You must edit the Role ARN to match your Region.

```
"AWS": "arn:[partition]:iam::[accountid]:role/aws-service-role/resource-management.securitylake.amazonaws.com/AWSServiceRoleForSecurityLakeResourceManagement"
```

You can also use the CreateServiceLinkedRole API call. In the request, specify the `AWSServiceName` as `resource-management.securitylake.amazonaws.com`.

After enabling the `AWSServiceRoleForSecurityLakeResourceManagement` role, if you are using AWS KMS Customer Managed Key (CMK) for encryption, you must allow the service-linked role to write encrypted objects to S3 buckets in the AWS Regions where CMK exists. In the AWS KMS console, add the following policy to the KMS key in the AWS Regions where CMK exists. For the details on how to change the KMS key policy, see Key policies in AWS KMS in the AWS Key Management Service Developer Guide.

```
{
    "Sid": "Allow SLR",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:[partition]:iam::[accountid]:role/aws-service-role/resource-
management.securitylake.amazonaws.com/AWSServiceRoleForSecurityLakeResourceManagement"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey*"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::[regional-datalake-s3-
bucket-name]"
        },
        "StringLike": {
            "kms:ViaService": "s3.[region].amazonaws.com"
        }
    }
},
```

Editing the Security Lake service-linked role

Security Lake doesn't allow you to edit the `AWSServiceRoleForSecurityLakeResourceManagement` service-linked role. After a service-linked role is created, 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 the Security Lake service-linked role

You cannot delete the service-linked role from Security Lake. Instead, you may delete the service-linked role from the IAM console, API, or AWS CLI. For more information, see Deleting a service-linked role in the *IAM User Guide*.

Before you can delete the service-linked role, you must first confirm that the role has no active sessions and remove any resources that `AWSServiceRoleForSecurityLakeResourceManagement` is using.

> **Note**
>
> If Security Lake is using the `AWSServiceRoleForSecurityLakeResourceManagement` role when you try to delete the resources, the deletion might fail. If that happens, wait a few minutes and then try the operation again.

If you delete the `AWSServiceRoleForSecurityLakeResourceManagement` service-linked role and need to create it again, you can create it again by enabling Security Lake for your account. When you enable Security Lake again, Security Lake automatically creates the service-linked role again for you.

Supported AWS Regions for the Security Lake service-linked role

Security Lake supports using the `AWSServiceRoleForSecurityLakeResourceManagement` service-linked role in all the AWS Regions where Security Lake is available. For a list of Regions where Security Lake is currently available, see Security Lake Regions and endpoints.

Data protection in Amazon Security Lake

The AWS shared responsibility model applies to data protection in Amazon Security Lake. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

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

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

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Security Lake 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

Amazon Security Lake securely stores your data at rest using AWS encryption solutions. Raw security log and event data is stored in a multi-tenant Amazon Simple Storage Service (Amazon S3) bucket in an account that Security Lake manages. Security Lake encrypts this raw data using an AWS owned key from AWS Key Management Service (AWS KMS). AWS owned keys are a collection of AWS KMS keys that an AWS service—in this case Security Lake—owns and manages for use in multiple AWS accounts.

Security Lake runs extract, transform, and load (ETL) jobs on raw log and event data. The processed data remains encrypted in the Security Lake service account.

After the ETL jobs are completed, Security Lake creates single-tenant S3 buckets in your account (one bucket for each AWS Region that you've enabled Security Lake in). Data is stored in the multi-tenant S3 bucket only temporarily until Security Lake can reliably deliver the data to the single-tenant S3 buckets. The single-tenant buckets include a resource-based policy that gives Security Lake permission to write log and event data to the buckets. To encrypt data in your S3 bucket, you can choose either an S3-managed encryption key or a customer managed key (from AWS KMS). Both options use symmetric encryption.

Using a KMS key for encryption of your data

By default, the data delivered by Security Lake to your S3 bucket is encrypted by Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3). To provide a security layer that you manage directly, you can instead use server-side encryption with AWS KMS keys (SSE-KMS) for your Security Lake data.

SSE-KMS isn't supported in the Security Lake console. To use SSE-KMS with the Security Lake API or CLI, you first create a KMS key or use an existing key. You attach a policy to the key that determines which users can use the key for encrypting and decrypting Security Lake data.

If you use a customer managed key to encrypt data that's written to your S3 bucket, you can't choose a multi-Region key. For customer managed keys, Security Lake creates a grant on your behalf by sending a `CreateGrant` request to AWS KMS. Grants in AWS KMS are used to give Security Lake access to a KMS key in a customer account.

Security Lake requires 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 `RetireGrant` requests to AWS KMS. When you make updates to your data lake, this operation enables the retirement of the grant that was added to the AWS KMS key for ETL processing.

Security Lake doesn't need `Decrypt` permissions. When authorized users of the key read Security Lake data, S3 manages the decryption, and the authorized users are able to read data in unencrypted form. However, a subscriber needs `Decrypt` permissions to consume source data. For more information about subscriber permissions, see Managing data access for Security Lake subscribers.

If you want to use an existing KMS key to encrypt Security Lake data, you must modify the key policy for the KMS key. The key policy must allow the IAM role associated with the Lake Formation data lake location to use the KMS key to decrypt the data. For instructions on how you can change the key policy for a KMS key, see Changing a key policy in the AWS Key Management Service Developer Guide.

Your KMS key can accept grant requests, allowing Security Lake to access the key, when you create a key policy or use an existing key policy with the appropriate permissions. For instructions on creating a key policy, see Creating a key policy in the *AWS Key Management Service Developer Guide*.

Attach the following key policy to your KMS key:

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleRole"}
  "Action": [
    "kms:CreateGrant",
    "kms:DescribeKey",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

Required IAM permissions when using a customer managed key

See the Getting started: Prerequisites section for an overview of IAM roles that you need to create to use Security Lake.

When you add a custom source or a subscriber, Security Lake creates IAM roles in your account. These roles are intended to be shared with other IAM identities. They permit a custom source to write data to the data lake and a subscriber to consume data from the data lake. An AWS managed policy called `AmazonSecurityLakePermissionsBoundary` sets the permission boundaries for these roles.

Encrypting Amazon SQS queues

When you create your data lake, Security Lake creates two unencrypted Amazon Simple Queue Service (Amazon SQS) queues in the delegated Security Lake administrator account. You should encrypt these queues to protect your data. The default server-side encryption (SSE) provided by Amazon Simple Queue Service isn't sufficient. You must create a customer managed key in AWS Key Management Service (AWS KMS) to encrypt the queues and the grant the Amazon S3 service principal permissions to work with the encrypted queues. For instructions on granting these permissions, see Why aren't Amazon S3 event notifications delivered to an Amazon SQS queue that uses server-side encryption? in the AWS Knowledge Center.

Since Security Lake uses AWS Lambda to support extract, transfer, and load (ETL) jobs on your data, you must also give Lambda permissions to manage messages in your Amazon SQS queues. For information, see Execution role permissions in the *AWS Lambda Developer Guide*.

Encryption in transit

Security Lake encrypts all data in transit between AWS services. Security Lake protects data in transit, as it travels to and from the service, by automatically encrypting all inter-network data using the Transport Layer Security (TLS) 1.2 encryption protocol. Direct HTTPS requests sent to the Security Lake APIs are signed by using the AWS Signature Version 4 Algorithm to establish a secure connection.

Opting out of using your data for service improvement

You can choose to opt out of having your data used to develop and improve Security Lake and other AWS security services by using the AWS Organizations opt-out policy. You can choose to opt out even if Security Lake doesn't currently collect any such data. For more information about how to opt out, see AI services opt-out policies in the *AWS Organizations User Guide*.

Presently, Security Lake does not collect any of the security data that it processes on your behalf, or security data that you upload to your security data lake created by this service. To develop and improve the Security Lake service and the functionalities of other AWS security services, Security Lake may collect such data in the future, including data that you upload from third-party data sources. We will update this page when Security Lake intends on collecting any such data and describe how this will work. You will still have an opportunity to opt out at any time.

Note

For you to use the opt-out policy, your AWS accounts must be centrally managed by AWS Organizations. If you haven't already created an organization for your AWS accounts, see Creating and managing an organization in the *AWS Organizations User Guide*.

Opting out has the following effects:

- Security Lake will delete the data that it collected and stored prior to your opt out (if any).
- After you opt out, Security Lake will no longer collect or store this data.

Compliance validation for Amazon Security Lake

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 Compliance & Governance – These solution implementation guides discuss architectural considerations and provide steps for deploying security and compliance features.
- 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.

- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- 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.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- 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 Security Lake

See the following best practices for working with Amazon Security Lake.

Grant Security Lake users minimum possible permissions

Follow the principle of least privilege by granting the minimum set of access policy permissions for your AWS Identity and Access Management (IAM) users, user groups, and roles. For example, you might allow an IAM user to view a list of log sources in Security Lake but not to create sources or subscribers. For more information, see Identity-based policy examples for Security Lake

You can also use AWS CloudTrail to track API usage in Security Lake. CloudTrail provides a record of API actions taken by a user, group, or role in Security Lake. For more information, see Logging Security Lake API calls using CloudTrail.

View the Summary page

The **Summary** page of the Security Lake console provides an overview of issue from the last 14 days that are impacting the Security Lake service and the Amazon S3 buckets in which your data is stored. You can further investigate these issues to help you mitigate possible security-related impact.

Integrate with Security Hub

Integrate Security Lake and AWS Security Hub to receive Security Hub findings in Security Lake. Security Hub generates findings from many different AWS services and third-party integrations. Receiving Security Hub findings helps you get an overview of your compliance posture and whether you're meeting AWS security best practices.

For more information, see Integration with AWS Security Hub.

Monitor for Security Lake events

You can monitor Security Lake using Amazon CloudWatch metrics. CloudWatch collects raw data from Security Lake every minute and processes it into metrics. You can set alarms that trigger notifications when metrics match specified thresholds.

For more information, see CloudWatch metrics for Amazon Security Lake.

Resilience in Amazon Security Lake

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. These Availability Zones offer you an effective way to design and operate applications and databases. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

The availability of Security Lake is tied to Region availability. Distribution across multiple Availability Zones helps the service tolerate failures in any single Availability Zone.

The availability of the Security Lake data plane is not tied to any Region availability. However, the availability of the Security Lake control plane is closely tied to US East (N. Virginia) Region availability.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

In addition to the AWS global infrastructure, Security Lake, in which data is backed by Amazon Simple Storage Service (Amazon S3); offers several features to help support your data resiliency and backup needs.

Lifecycle configuration

A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. With lifecycle configuration rules, you can tell Amazon S3 to transition objects to less expensive storage classes, archive them, or delete them. For more information, see Managing your storage lifecycle in the *Amazon S3 User Guide*.

Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. Versioning helps you recover from both unintended user actions and application failures. For more information, see Using versioning in S3 buckets in the *Amazon S3 User Guide*.

Storage classes

Amazon S3 offers a range of storage classes to choose from depending on the requirements of your workload. The S3 Standard-IA and S3 One Zone-IA storage classes are designed for data you access about once a month and need milliseconds access. The S3 Glacier Instant Retrieval storage class is designed for long-lived archive data accessed with milliseconds access that you access about once a quarter. For archive data that does not require immediate access, such as backups, you can use the S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive storage classes. For more information, see Using Amazon S3 storage classes in the *Amazon S3 User Guide*.

Infrastructure security in Amazon Security Lake

As a managed service, Amazon Security Lake is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access Security Lake through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

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

Configuration and vulnerability analysis in Security Lake

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

Monitoring Amazon Security Lake

Security Lake integrates with AWS CloudTrail, which is a service that provides a record of actions that were taken in Security Lake by a user, a role, or another AWS service. This includes actions from the Security Lake console and programmatic calls to Security Lake API operations. By using the information collected by CloudTrail, you can determine which requests were made to Security Lake. For each request, you can identify when it was made, the IP address from which it was made, who made it, and additional details. For more information, see Logging Security Lake API calls using CloudTrail.

Security Lake and Amazon CloudWatch are integrated, so you can collect, view, and analyze metrics for logs that Security Lake collects. CloudWatch metrics for your Security Lake data lake are automatically collected and pushed to CloudWatch at one-minute intervals. You can also set an alarm to send you a notification if a specified threshold is met for a Security Lake metric. For a list of all the metrics that Security Lake sends to CloudWatch, see Security Lake metrics and dimensions.

CloudWatch metrics for Amazon Security Lake

You can monitor Security Lake using Amazon CloudWatch, which collects raw data every minute 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 the data in your data lake. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met.

Topics

Security Lake metrics and dimensions

The `AWS/SecurityLake` namespace includes the following metrics.

Metric	Description
`ProcessedSize`	The volume of data from natively-supported AWS services that's currently stored in your data lake. Units: Bytes

The following dimensions are available for Security Lake metrics.

Dimension	Description
`Account`	`ProcessedSize` metric for a specific AWS account. This dimension is available only when you view the `Per-Account Source Version Metrics` on CloudWatch.
`Region`	`ProcessedSize` metric for a specific AWS Region.
`Source`	`ProcessedSize` metric for a specific AWS log source.
`SourceVersion`	`ProcessedSize` metric for a specific version of an AWS log source.

You can view metrics for specific AWS accounts (`Per-Account Source Version Metrics`) or for all accounts in an organization (`Per-Source Version Metrics`).

Viewing CloudWatch metrics for Security Lake

You can monitor metrics for Security Lake using the CloudWatch console, CloudWatch's own command line interface (CLI), or programmatically using the CloudWatch API. Choose your preferred method, and follow the steps to access Security Lake metrics.

CloudWatch console

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. On the navigation pane, choose **Metrics, All metrics**.
3. On the **Browse** tab, choose **Security Lake**.
4. Choose **Per-Account Source Version Metrics** or **Per-Source Version Metrics**.
5. Select a metric to view it in detail. You can also choose to do the following:
   - To sort the metrics, use the column heading.
   - To graph a metric, select the metric name, and choose a graphing option.
   - To filter by metric, select the metric name and then choose **Add to search**.

CloudWatch API

To access Security Lake metrics using the CloudWatch API, use the `GetMetricStatistics` action.

AWS CLI

To access Security Lake metrics using the AWS CLI, run the `get-metric-statistics` command.

For more information about monitoring using metrics, see Use Amazon CloudWatch metrics in the *Amazon CloudWatch User Guide*.

Setting CloudWatch alarms for Security Lake metrics

CloudWatch also allows you to set alarms when a threshold is met for a metric. For example, you could set an alarm for the **ProcessedSize** metric, so that you're notified when the volume of data from a specific source exceeds a specific threshold.

For instructions on setting alarms, see Using Amazon CloudWatch alarms in the *Amazon CloudWatch User Guide*.

Logging Security Lake API calls using CloudTrail

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

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Security Lake information in CloudTrail

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

For an ongoing record of events in your AWS account, including events for Security Lake, create a trail. A *trail* enables CloudTrail to deliver events as log files to an Amazon S3 bucket that you specify. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

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

Security Lake actions are logged by CloudTrail and are documented in the Security Lake API Reference. For example, calls to the `UpdateDataLake`, `ListLogSources`, and `CreateSubscriber` 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 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.

Understanding Security Lake log file entries

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 for the Security Lake `GetSubscriber` action.

```
{
   "eventVersion": "1.08",
   "userIdentity": {
      "type": "AssumedRole",
      "principalId": "AIDACKCEVSQ6C2EXAMPLE:user",
      "arn": "arn:aws:sts::123456789012:assumed-role/Admin/user",
      "accountId": "123456789012",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "sessionContext": {
         "sessionIssuer": {
            "type": "Role",
            "principalId": "AIDACKCEVSQ6C2EXAMPLE",
            "arn": "arn:aws:iam::123456789012:role/Admin",
            "accountId": "123456789012",
            "userName": "Admin"
         },
         "webIdFederationData": {
         },
         "attributes": {
            "creationDate": "2023-05-30T13:27:19Z",
            "mfaAuthenticated": "false"
         }
      }
   },
   "eventTime": "2023-05-30T17:29:17Z",
   "eventSource": "securitylake.amazonaws.com",
   "eventName": "GetSubscriber",
   "awsRegion": "us-east-1",
   "sourceIPAddress": "198.51.100.1",
   "userAgent": "console.amazonaws.com",
   "requestParameters": {
      "subscriberId": "30ed17a3-0cac-4997-a41f-f5a6bexample"
   },
   "responseElements": null,
   "requestID": "d01f0f32-9ec6-4579-af50-e9f14example",
   "eventID": "9c1bff41-0f48-4ee6-921c-ebfd8example",
   "readOnly": false,
   "eventType": "AwsApiCall",
   "managementEvent": true,
   "recipientAccountId": "123456789012",
   "eventCategory": "Management"
}
```

Tagging Security Lake resources

A *tag* is an optional label that you can define and assign to AWS resources, including certain types of Amazon Security Lake resources. Tags can help you identify, categorize, and manage resources in different ways, such as by purpose, owner, environment, or other criteria. For example, you can use tags to apply policies, allocate costs, distinguish between resources, or identify resources that support certain compliance requirements or workflows.

You can assign tags to the following types of Security Lake resources: subscribers, and the data lake configuration for your AWS account in individual AWS Regions.

Topics

Tagging fundamentals

A resource can have as many as 50 tags. Each tag consists of a required *tag key* and an optional *tag value*, both of which you define. A *tag key* is a general label that acts as a category for a more specific tag value. A *tag value* acts as a descriptor for a tag key.

For example, if you add subscribers to analyze security data from different environments (one set of subscribers for cloud data and another set for on-premises data), you might assign an `Environment` tag key to those subscribers. The associated tag value might be `Cloud` for subscribers that analyze data from AWS services, and `On-Premises` for the others.

As you define and assign tags to Amazon Security Lake resources, keep the following in mind:

- Each resource can have a maximum of 50 tags.
- For each resource, each tag key must be unique and it can have only one tag value.
- Tag keys and values are case sensitive. As a best practice, we recommend that you define a strategy for capitalizing tags and implement that strategy consistently across your resources.

- A tag key can have a maximum of 128 UTF-8 characters. A tag value can have a maximum of 256 UTF-8 characters. The characters can be letters, numbers, spaces, or the following symbols: _ . : / = + - @
- The `aws:` prefix is reserved for use by AWS. You can’t use it in any tag keys or values that you define. In addition, you can't change or remove tag keys or values that use this prefix. Tags that use this prefix don’t count against the quota of 50 tags per resource.
- Any tags that you assign are available only for your AWS account and only in the AWS Region in which you assign them.
- If you assign tags to a resource by using Security Lake, the tags are applied only to the resource that's stored directly in Security Lake in the applicable AWS Region. They aren't applied to any associated, supporting resources that Security Lake creates, uses, or maintains for you in other AWS services. For example, if you assign tags to your data lake, the tags are applied only to your data lake configuration in Security Lake for the specified Region. They aren't applied to the Amazon Simple Storage Service (Amazon S3) bucket that stores your log and event data. To also assign tags to an associated resource, you can use AWS Resource Groups or the AWS service that stores the resource—for example, Amazon S3 for an S3 bucket. Assigning tags to associated resources can help you identify supporting resources for your data lake.
- If you delete a resource, any tags that are assigned to the resource are also deleted.

For additional restrictions, tips, and best practices, see Tagging your AWS resources in the *Tagging AWS Resources User Guide*.

⚠ **Important**

Do not store confidential or other types of sensitive data in tags. Tags are accessible from many AWS services, including AWS Billing and Cost Management. They aren't intended to be used for sensitive data.

To add and manage tags for Security Lake resources, you can use the Security Lake console or the Security Lake API.

Using tags in IAM policies

After you start tagging resources, you can define tag-based, resource-level permissions in AWS Identity and Access Management (IAM) policies. By using tags in this way, you can implement granular control of which users and roles in your AWS account have permission to create and tag resources, and which users and roles have permission to add, edit, and remove tags more generally. To control access based on tags, you can use tag-related condition keys in the Condition element of IAM policies.

For example, you can create a policy that allows a user to have full access to all Amazon Security Lake resources, if the `Owner` tag for the resource specifies their username:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ModifyResourceIfOwner",
            "Effect": "Allow",
            "Action": "securitylake:*",
            "Resource": "*",
            "Condition": {
                "StringEqualsIgnoreCase": {"aws:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

If you define tag-based, resource-level permissions, the permissions take effect immediately. This means that your resources are more secure as soon as they're created, and you can quickly start enforcing the use of tags for new resources. You can also use resource-level permissions to control which tag keys and values can be associated with new and existing resources. For more information, see Controlling access to AWS resources using tags in the *IAM User Guide*.

Adding tags to Amazon Security Lake resources

To add tags to an Amazon Security Lake resource, you can use the Security Lake console or the Security Lake API.

> **Important**
>
> Adding tags to a resource can affect access to the resource. Before you add a tag to a resource, review any AWS Identity and Access Management (IAM) policies that might use tags to control access to resources.

Console

When you enable Security Lake for an AWS Region or create a subscriber, the Security Lake console provides options for adding tags to the resource—the data lake configuration for the Region or the subscriber. Follow the instructions on the console to add tags to the resource when you create the resource.

To add one or more tags to an existing resource by using the Security Lake console, follow these steps.

To add a tag to a resource

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Depending on the type of resource that you want to add a tag to, do one of the following:
   - For a data lake configuration, choose **Regions** in the navigation pane. Then, in the **Regions** table, select the Region.
   - For a subscriber, choose **Subscribers** in the navigation pane. Then, in the **My subscribers** table, select the subscriber.

     If the subscriber doesn't appear in the table, use the AWS Region selector in the upper-right corner of the page to select the Region where you created the subscriber. The table lists existing subscribers only for the current Region.
3. Choose **Edit**.
4. Expand the **Tags** section. This section lists all the tags that are currently assigned to the resource.
5. In the **Tags** section, choose **Add new tag**.
6. In the **Key** box, enter the tag key for the tag to add to the resource. Then, in the **Value** box, optionally enter a tag value for the key.

   A tag key can contain as many as 128 characters. A tag value can contain as many as 256 characters. The characters can be letters, numbers, spaces, or the following symbols: _ . : / = + - @
7. To add another tag to the resource, choose **Add new tag**, and then repeat the preceding step. You can assign as many as 50 tags to a resource.
8. When you finish adding tags, choose **Save**.

API

To create a resource and add one or more tags to it programmatically, use the appropriate `Create` operation for the type of resource that you want to create:

- **Data lake configuration** – Use the CreateDataLake operation or, if you're using the AWS Command Line Interface (AWS CLI), run the create-data-lake command.
- **Subscriber** – Use the CreateSubscriber operation or, if you're using the AWS CLI, run the create-subscriber command.

In your request, use the `tags` parameter to specify the tag key (`key`) and optional tag value (`value`) for each tag to add to the resource. The `tags` parameter specifies an array of objects. Each object specifies a tag key and its associated tag value.

To add one or more tags to an existing resource, use the TagResource operation of the Security Lake API or, if you're using the AWS CLI, run the tag-resource command. In your request, specify the Amazon Resource Name (ARN) of the resource that you want to add a tag to. Use the `tags` parameter to specify the tag key (`key`) and optional tag value (`value`) for each tag to add. As is the case for `Create` operations and commands, the `tags` parameter specifies an array of objects, one object for each tag key and its associated tag value.

For example, the following AWS CLI command adds an `Environment` tag key with a `Cloud` tag value to the specified subscriber. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake tag-resource \
--resource-arn arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab \
--tags key=Environment,value=Cloud
```

Where:

- `resource-arn` specifies the ARN of the subscriber to add a tag to.
- *`Environment`* is the tag key of the tag to add to the subscriber.
- *`Cloud`* is the tag value for the specified tag key (*`Environment`*).

In the following example, the command adds several tags to the subscriber.

```
$ aws securitylake tag-resource \
--resource-arn arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab \
--tags key=Environment,value=Cloud key=CostCenter,value=12345 key=Owner,value=jane-
doe
```

For each object in a `tags` array, both the `key` and `value` arguments are required. However, the value for the `value` argument can be an empty string. If you don’t want to associate a tag value with a tag key, don't specify a value for the `value` argument. For example, the following command adds an `Owner` tag key with no associated tag value:

```
$ aws securitylake tag-resource \
--resource-arn arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab \
--tags key=Owner,value=
```

If a tagging operation succeeds, Security Lake returns an empty HTTP 200 response. Otherwise, Security Lake returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Editing tags for Amazon Security Lake resources

To edit the tags (tag keys or tag values) for an Amazon Security Lake resource, you can use the Security Lake console or the Security Lake API.

> **Important**
>
> Editing the tags for a resource can affect access to the resource. Before you edit a tag key or value for a resource, review any AWS Identity and Access Management (IAM) policies that might use the tag to control access to resources.

Console

Follow these steps to edit a resource's tags by using the Security Lake console.

To edit the tags for a resource

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Depending on the type of resource whose tags you want to edit, do one of the following:

- For a data lake configuration, choose **Regions** in the navigation pane. Then, in the **Regions** table, select the Region.
- For a subscriber, choose **Subscribers** in the navigation pane. Then, in the **My subscribers** table, select the subscriber.

  If the subscriber doesn't appear in the table, use the AWS Region selector in the upper-right corner of the page to select the Region where you created the subscriber. The table lists existing subscribers only for the current Region.

3. Choose **Edit**.
4. Expand the **Tags** section. The **Tags** section lists all the tags that are currently assigned to the resource.
5. Do any of the following:

   - To add a tag value to an existing tag key, enter the value in the **Value** box next to the tag key.
   - To change an existing tag key, choose **Remove** next to the tag. Then choose **Add new tag**. In the **Key** box that appears, enter the new tag key. Optionally enter an associated tag value in the **Value** box.
   - To change an existing tag value, choose **X** in the **Value** box that contains the value. Then enter the new tag value in the **Value** box.
   - To remove an existing tag value, choose **X** in the **Value** box that contains the value.
   - To remove an existing tag (both the tag key and tag value), choose **Remove** next to the tag.

   A resource can have as many as 50 tags. A tag key can contain as many as 128 characters. A tag value can contain as many as 256 characters. The characters can be letters, numbers, spaces, or the following symbols: _ . : / = + - @
6. When you finish editing the tags, choose **Save**.

API

When you edit a tag for a resource programmatically, you overwrite the existing tag with new values. Therefore, the best way to edit a tag depends on whether you want to edit a tag key, a tag value, or both. To edit a tag key, remove the current tag and add a new tag.

To edit or remove only the tag value that's associated with a tag key, overwrite the existing value by using the TagResource operation of the Security Lake API. If you're using the AWS Command Line Interface (AWS CLI), run the tag-resource command. In your request, specify the Amazon Resource Name (ARN) of the resource whose tag value you want to edit or remove.

To edit a tag value, use the `tags` parameter to specify the tag key whose tag value you want to change. Also specify the new tag value for the key. For example, the following AWS CLI command changes the tag value from `Cloud` to `On-Premises` for the `Environment` tag key that's assigned to the specified subscriber. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake tag-resource \
--resource-arn arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab \
--tags key=Environment,value=On-Premises
```

Where:

- `resource-arn` specifies the ARN of the subscriber.
- *`Environment`* is the tag key that's associated with the tag value to change.
- *`On-Premises`* is the new tag value for the specified tag key (*`Environment`*).

To remove a tag value from a tag key, don’t specify a value for the `value` argument of the key in the `tags` parameter. For example:

```
$ aws securitylake tag-resource \
--resource-arn arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab \
--tags key=Owner,value=
```

If the operation succeeds, Security Lake returns an empty HTTP 200 response. Otherwise, Security Lake returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Reviewing tags for Amazon Security Lake resources

You can review the tags (both tag keys and tag values) for an Amazon Security Lake resource by using the Security Lake console or the Security Lake API.

Console

Follow these steps to review a resource's tags by using the Security Lake console.

To review the tags for a resource

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Depending on the type of resource whose tags you want to review, do one of the following:
   - For a data lake configuration, choose **Regions** in the navigation pane. In the **Regions** table, select the Region, and then choose **Edit**. Then expand the **Tags** section.
   - For a subscriber, choose **Subscribers** in the navigation pane. Then, in the **My subscribers** table, choose the subscriber's name.

     If the subscriber doesn't appear in the table, use the AWS Region selector in the upper-right corner of the page to select the Region where you created the subscriber. The table lists existing subscribers only for the current Region.

The **Tags** section lists all the tags that are currently assigned to the resource.

API

To retrieve and review the tags for an existing resource programmatically, use the ListTagsForResource operation of the Security Lake API. In your request, use the `resourceArn` parameter to specify the Amazon Resource Name (ARN) of the resource.

If you're using the AWS Command Line Interface (AWS CLI), run the list-tags-for-resource command and use the `resource-arn` parameter to specify the ARN of the resource. For example:

```
$ aws securitylake list-tags-for-resource --resource-arn arn:aws:securitylake:us-east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab
```

In the preceding example, *arn:aws:securitylake:us-east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab* is the ARN of an existing subscriber.

If the operation succeeds, Security Lake returns a `tags` array. Each object in the array specifies a tag (both the tag key and tag value) that's currently assigned to the resource. For example:

```
{
    "tags": [
        {
            "key": "Environment",
            "value": "Cloud"
        },
        {
            "key": "CostCenter",
            "value": "12345"
        },
        {
            "key": "Owner",
            "value": ""
        }
    ]
}
```

Where `Environment`, `CostCenter`, and `Owner` are the tag keys that are assigned to the resource. `Cloud` is the tag value that's associated with the `Environment` tag key. `12345` is the tag value that's associated with the `CostCenter` tag key. The `Owner` tag key doesn't have an associated tag value.

Removing tags from Amazon Security Lake resources

To remove tags from an Amazon Security Lake resource, you can use the Security Lake console or the Security Lake API.

> ⚠ **Important**
>
> Removing tags from a resource can affect access to the resource. Before you remove a tag, review any AWS Identity and Access Management (IAM) policies that might use the tag to control access to resources.

Console

Follow these steps to remove one or more tags from a resource by using the Security Lake console.

To remove a tag from a resource

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. Depending on the type of resource that you want to remove a tag from, do one of the following:
   - For a data lake configuration, choose **Regions** in the navigation pane. Then, in the **Regions** table, select the Region.
   - For a subscriber, choose **Subscribers** in the navigation pane. Then, in the **My subscribers** table, select the subscriber.

     If the subscriber doesn't appear in the table, use the AWS Region selector in the upper-right corner of the page to select the Region where you created the subscriber. The table lists existing subscribers only for the current Region.
3. Choose **Edit**.
4. Expand the **Tags** section. The **Tags** section lists all the tags that are currently assigned to the resource.
5. Do any of the following:
   - To remove only the tag value for a tag, choose **X** in the **Value** box that contains the value to remove.
   - To remove both the tag key and tag value (as a pair) for a tag, choose **Remove** next to the tag to remove.
6. To remove additional tags from the resource, repeat the preceding step for each additional tag to remove.
7. When you finish removing tags, choose **Save**.

API

To remove one or more tags from a resource programmatically, use the UntagResource operation of the Security Lake API. In your request, use the `resourceArn` parameter to specify the Amazon Resource Name (ARN) of the resource to remove a tag from. Use the `tagKeys` parameter to specify the tag key of the tag to remove. To remove multiple tags, append the `tagKeys` parameter and argument for each tag to remove, separated by an ampersand (&)—for example, `tagKeys=`*`key1`*`&tagKeys=`*`key2`*. To remove only a specific tag value (not a tag key) from a resource, edit the tag instead of removing the tag.

If you're using the AWS Command Line Interface (AWS CLI), run the untag-resource command to remove one or more tags from a resource. For the `resource-arn` parameter, specify the ARN of the resource to remove a tag from. Use the `tag-keys` parameter to specify the tag key of the tag to remove. For example, the following command removes the `Environment` tag (both the tag key and tag value) from the specified subscriber:

```
$ aws securitylake untag-resource \
--resource-arn arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab \
--tag-keys Environment
```

Where `resource-arn` specifies the ARN of the subscriber to remove a tag from, and *`Environment`* is the tag key of the tag to remove.

To remove multiple tags from a resource, add each additional tag key as an argument for the `tag-keys` parameter. For example:

```
$ aws securitylake untag-resource \
--resource-arn arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab \
--tag-keys Environment Owner
```

If the operation succeeds, Security Lake returns an empty HTTP 200 response. Otherwise, Security Lake returns an HTTP 4*xx* or 500 response that indicates why the operation failed.

Troubleshooting issues in Security Lake

If you encounter issues when working with Amazon Security Lake, use the following troubleshooting resources.

The following topics provide troubleshooting advice for errors and issues that you might encounter related to data lake status, Lake Formation, querying in Amazon Athena, AWS Organizations and IAM. If you find an issue that is not listed here, you can use the Feedback button on this page to report it.

Consult the following topics if you encounter issues while using Security Lake.

Topics

Troubleshooting data lake status

The **Issues** page of the Security Lake console shows you a summary of issues that are affecting your data lake. For example, Security Lake can't enable log collection for AWS CloudTrail management events if you haven't created a CloudTrail trail for your organization. The **Issues** page covers issues that have occurred in the last 14 days. You can see a description of each issue and the suggested remediation steps.

To programmatically access a summary of issues, you can use the ListDataLakeExceptions operation of the Security Lake API. If you're using the AWS CLI, run the list-data-lake-exceptions command. For the `regions` parameter, you can specify one or more Region codes—for example, `us-east-1` for the US East (N. Virginia) Region—to see the issues affecting those Regions. If you don't include the `regions` parameter, issues affecting all Regions are returned. For a list of Region codes, see Amazon Security Lake endpoints in the *AWS General Reference*.

For example, the following AWS CLI command lists issues that are affecting the `us-east-1` and `eu-west-3` Regions. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake list-data-lake-exceptions \
--regions "us-east-1" "eu-west-3"
```

To notify a Security Lake user about an issue or error, use the CreateDataLakeExceptionSubscription operation of the Security Lake API. The user can be notified through email, delivery to an Amazon Simple Queue Service (Amazon SQS) queue, delivery to an AWS Lambda function, or another supported protocol.

For example, the following AWS CLI command sends notifications of Security Lake exceptions to the specified account by SMS delivery. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake create-data-lake-exception-subscription \
--notification-endpoint "123456789012" \
--exception-time-to-live 30 \
--subscription-protocol "sms"
```

To view details about an exception subscription, you can use the GetDataLakeExceptionSubscription operation. To update an exception subscription, you can use the UpdateDataLakeExceptionSubscription operation. To delete an exception subscription and stop notifications, you can use the DeleteDataLakeExceptionSubscription operation.

Troubleshooting Lake Formation issues

Use the following information to help you diagnose and fix common issues that you might encounter when working with Security Lake and AWS Lake Formation databases or tables. For more Lake Formation troubleshooting topics, see the Troubleshooting section of the *AWS Lake Formation Developer Guide*.

Table not found

You may receive this error when attempting to create a subscriber.

To resolve this error, make sure that you have added sources in the Region already. If you added sources when the Security Lake service was in preview release, you must add them again before creating a subscriber. For more information on adding sources, see Source management in Security Lake.

400 AccessDenied

You may receive this error when you add a custom source and call the `CreateCustomLogSource` API.

To resolve the error, review your Lake Formation permissions. The IAM role that's calling the API should have **Create table** permissions for the Security Lake database. For more information, see Granting database permissions using the Lake Formation console and the named resource method in the *AWS Lake Formation Developer Guide*.

SYNTAX_ERROR: line 1:8: SELECT * not allowed from relation that has no columns

You may receive this error when querying a source table for the first time in Lake Formation.

To resolve the error, grant `SELECT` permission to the IAM role you are using when signed into your AWS account. For instructions on how to grant `SELECT` permission, see Granting table permissions using the Lake Formation console and the named resource method in the *AWS Lake Formation Developer Guide*.

Security Lake failed to add caller's principal ARN to Lake Formation data lake admin. Current data lake administrators may include invalid principals that no longer exist.

You may receive this error when enabling Security Lake or adding an AWS service as a log source.

To resolve the error, follow these steps:

1. Open the Lake Formation console at https://console.aws.amazon.com/lakeformation/.
2. Sign in as an administrative user.
3. In the navigation pane, under **Permissions**, choose **Administrative roles and tasks**.
4. In the **Data lake administrators** section, choose **Choose administrators**.
5. Clear principals that are labeled **Not found in IAM**, and then choose **Save**.
6. Try the Security Lake operation again.

Security Lake CreateSubscriber with Lake Formation didn't create a new RAM resource share invitation to be accepted

You may see this error if you shared resources with Lake Formation version 2 or version 3 cross-account data sharing before creating a Lake Formation subscriber in Security Lake. This is because Lake Formation version 2 and version 3 cross-account sharing optimizes the number of AWS RAM resource shares by mapping multiple cross-account permission grants with one AWS RAM resource share.

Make sure to check that the resource share name has the external ID that you specified when creating the subscriber and the resource share ARN matches the ARN in the `CreateSubscriber` response.

Troubleshooting querying in Amazon Athena

Use the following information to help you diagnose and fix common issues that you might encounter when using Athena to query objects that are stored in your Security Lake S3 bucket. For more Athena troubleshooting topics, see the Troubleshooting in Athena section of the *Amazon Athena User Guide*.

Querying isn't returning new objects in the data lake

Your Athena query may not return new objects in your data lake even when the S3 bucket for Security Lake contains those objects. This may occur if you've disabled Security Lake and then enabled it again. As a result, the AWS Glue partitions may not properly register the new objects.

To resolve the error, follow these steps:

1. Open the AWS Lambda console at https://console.aws.amazon.com/lambda/.
2. From the navigation bar, on the Regions selector, choose the Region in which Security Lake is enabled but the Athena query isn't returning results.
3. From the navigation pane, choose **Functions**, and select the function from the following list depending on the source version:

    - `Source version 1 (OCSF 1.0.0-rc.2) –` **SecurityLake_Glue_Partition_Updater_Lambda_*#region>*** function.
    - `Source version 2 (OCSF 1.1.0)–` **AmazonSecurityLakeMetastoreManager_*#region>*** function.

4. On the **Configurations** tab, choose **Triggers**.
5. Select the option next to the function, and choose **Edit**.
6. Select **Activate trigger**, and choose **Save**. This will turn the function state to **Enabled**.

Unable to access AWS Glue tables

A query access subscriber may not be able to access AWS Glue tables that contain Security Lake data.

First, ensure that you've followed the steps outlined in Setting up cross-account table sharing (subscriber step).

If the subscriber still doesn't have access, follow these steps:

1. Open the AWS Glue console at https://console.aws.amazon.com/glue/.
2. From the navigation pane, choose **Data Catalog** and **Catalog settings**.
3. Give permission to the subscriber to access the AWS Glue tables with a resource-based policy. For information about creating resource-based policies, see Resource-based policy examples for AWS Glue in the *AWS Glue Developer Guide*.

Troubleshooting Organizations issues

Use the following information to help you diagnose and fix common issues that you might encounter when working with Security Lake and AWS Organizations. For more Organizations troubleshooting topics, see the Troubleshooting section of the *AWS Organizations User Guide*.

An access denied error occurred when calling the CreateDataLake operation: Your account must be the delegated administrator account for an organization or a standalone account.

You may receive this error if you delete the organization that a delegated administrator account belonged to and then try to use that account to set up Security Lake by using the Security Lake console or the CreateDataLake API.

To resolve the error, use a delegated administrator account from a different organization or a standalone account.

Troubleshooting Amazon Security Lake identity and access

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

I am not authorized to perform an action in Security Lake

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

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`subscriber`* but does not have the fictional `SecurityLake:`*`GetSubscriber`* permissions.

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

In this case, Mateo asks his administrator to update his policies to allow him to access the *`subscriber`* information using the `SecurityLake:`*`GetSubscriber`* action.

I'm 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 Security Lake.

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

How Security Lake pricing is determined

Amazon Security Lake pricing is based on two dimensions: data ingestion and data conversion. Security Lake also works with other AWS services to store and share your data, and you may incur separate charges for these activities.

When you turn on log collection for the first time in an AWS account in any AWS Region that Security Lake supports, that account is automatically enrolled in a 15-day free trial of Security Lake. You may still incur charges from other services during the free trial.

To understand the methodology behind Security Lake pricing, watch the following video: Amazon Security Lake pricing-->

Data ingestion

These costs derive from the volume of ingested AWS CloudTrail logs and other AWS service logs and events (Amazon Route 53 resolver query logs, AWS Security Hub findings, and Amazon VPC Flow Logs).

Data conversion

These costs derive from the volume of AWS service logs and events that Security Lake normalizes to Open Cybersecurity Schema Framework (OCSF) in Security Lake schema and converts to Apache Parquet format.

Costs of related services

Here are some costs you may incur from other AWS services for storing and sharing the data in your security data lake:

- Amazon S3 – These costs derive from maintaining Amazon S3 buckets in your Security Lake account, storing your data there, and evaluating and monitoring your bucket for security and access control. For more information, see Amazon S3 pricing.
- Amazon SQS – These costs derive from creating an Amazon SQS queue for message delivery. For more information, see Amazon SQS pricing.
- Amazon EventBridge – These costs derive from Amazon EventBridge sending object notifications to subscription endpoints. For more information, see Amazon EventBridge pricing.

- AWS Glue – Monthly costs are determined by the volume of log and event data ingested from AWS services per gigabyte. Your data is stored in Amazon Simple Storage Service and standard Amazon S3 charges apply. Security Lake Lake also orchestrates other AWS services on your behalf. You will incur separate charges for AWS services used and resources set up as part of your security data lake. See pricing for AWS Glue, Amazon EventBridge, AWS Lambda, Amazon SQS, and Amazon Simple Notification Service. You are responsible for costs that you incur by querying data from Security Lake and storing query results.

Costs that a subscriber incurs by querying data from Security Lake and storing query results are the responsibility of the subscriber.

For a full list of costs and ancillary services, see Security Lake pricing.

Reviewing Security Lake usage and estimated costs

The **Usage** page of the Amazon Security Lake console lets you review your current Security Lake usage, as well as future usage and cost estimates. If you're currently participating in a 15-day free trial, your usage during the trial can help you estimate your costs for using Security Lake after your free trial ends. For an overview of Security Lake pricing, see How Security Lake pricing is determined. For detailed information and cost examples, see Amazon Security Lake Pricing.

In Security Lake, estimated usage costs are reported in US Dollars and apply only to the current AWS Region. The costs cover Security Lake usage by all accounts in your organization and include conversion to the Open Cybersecurity Schema Framework (OCSF) and Apache Parquet format. However, the predicted costs don't include costs for other services that Security Lake works with, such as Amazon Simple Storage Service (Amazon S3) and AWS Glue.

On the **Usage** page, you choose a time period for which to view usage and cost data. The default time period is the last 1 calendar day. You must have at least 1 day of Security Lake usage to see cost projections.

The top of the page shows the **Projected cost for all accounts**. This is your predicted Security Lake cost in the current AWS Region for the next 30 calendar days based on your actual usage during the selected time frame. The actual usage and predicted cost reflects all accounts in your organization.

On the remainder of the page, the usage and cost data is divided into two tables as follows:

- **Usage and cost by source** – This is your current Security Lake usage broken down by data source, as well as estimated usage and costs for the next 30 calendar days based on your actual usage during the selected time frame. The actual usage, predicted usage, and predicted cost reflect all accounts in your organization. If you select a source, a split panel opens which shows which accounts generated logs and events from that source. For each account, the split panel includes both actual usage from that source and predicted usage and costs.
- **Usage and cost by account** – This is your current Security Lake usage broken down by account, as well as estimated usage and costs for the next 30 calendar days based on your actual usage during the selected time frame. If you select an account, a split panel opens which shows the sources that contributed to that account's usage. For each contributing source, the split panel includes both actual usage and predicted usage and costs.

All supported AWS data sources appear in the preceding tables, even if you haven't added a particular source in Security Lake. We recommend adding all AWS sources if you're participating in the free trial to get cost estimates for your full set of logs and events. For instructions on adding an AWS source, see Collecting data from AWS services in Security Lake. Custom sources aren't included in usage or cost calculations.

Follow these steps to review your usage and cost data in the Security Lake console.

To review Security Lake usage and predicted costs (console)

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. By using the AWS Region selector in the upper-right corner of the page, select the Region in which you want to review your usage and costs.
3. In the navigation pane, choose **Settings** and then **Usage**.
4. Select the time period for which you want to see usage and cost data. The default is the last 1 day.
5. Select the **By data source** or **By accounts** tab to review usage and costs in detail.

Security Lake Regions and endpoints

For a list of supported Regions and service endpoints for Security Lake, see Amazon Security Lake endpoints in the *AWS General Reference*.

We recommend that you enable Security Lake in all supported AWS Regions. This allows you to use Security Lake to detect and investigate unauthorized or unusual activity even in Regions that you aren't actively using.

Disabling Security Lake

When you disable Amazon Security Lake, Security Lake stops collecting logs and events from your AWS sources. Existing Security Lake settings and the resources that were created in your AWS account are retained. In addition, the data that you stored in or published to other AWS services, such as sensitive data in AWS Lake Formation tables and AWS CloudTrail logs, remains available. Data that's stored in your Amazon Simple Storage Service (Amazon S3) bucket remains available in accordance with your Amazon S3 storage lifecycle.

Disabling Security Lake from the **Settings** page on the Security Lake console stops the collection of AWS logs and events in all AWS Regions in which Security Lake is currently enabled. You can use the **Regions** page on the console to stop log collection in specific Regions. The Security Lake API and AWS CLI also stop log collection in the Regions that you specify in your request.

If you use the integration with AWS Organizations and your account is part of an organization that centrally manages multiple Security Lake accounts, only the delegated Security Lake administrator can disable Security Lake for itself and for member accounts. However, leaving an organization stops log collection for a member account.

When you disable Security Lake for an organization, the delegated administrator designation is retained if you follow the disablement instructions provided on this page. You don't have to designate the delegated administrator again before you can re-enable Security Lake.

For custom sources, when deactivating Security Lake, you must disable each source outside of the Security Lake console. Failure to disable an integration will result in source integrations continuing to send logs into Amazon S3. Additionally, you must disable a subscriber integration or the subscriber will still be able to consume data from Security Lake. For details on how to remove the a custom source or a subscriber integration, see the respective provider's documentation.

> ⚠ **Important**
>
> You must delete AWS Glue databases before you re-enable Security Lake to ensure querying works properly.

When Security Lake is re-enabled a new data lake Amazon S3 bucket is created and data is collected in this new S3 bucket. If you had previously deleted AWS Glue tables, a new set of AWS Glue tables are created.

All the data that was collected before disabling Security Lake will stay in the old Amazon S3 bucket. If you want to query old data, you must move them to the new bucket using the Amazon S3 `Sync` command. For more details, see the Sync command in the AWS CLI Command Reference.

This topic explains how to disable Security Lake by using the Security Lake console, Security Lake API, or AWS CLI.

Console

1. Open the Security Lake console at https://console.aws.amazon.com/securitylake/.
2. In the navigation pane, under **Settings**, choose **General**.
3. Choose **Disable Security Lake**.
4. When prompted for confirmation, enter **`Disable`**, and then choose **Disable**.

API

To disable Security Lake programmatically, use the DeleteDataLake operation of the Security Lake API. If you're using the AWS CLI, run the delete-date-lake command. In your request, use the `regions` list to specify the Region code for each Region in which you want to disable Security Lake. For a list of Region codes, see Amazon Security Lake endpoints in the *AWS General Reference*.

For a Security Lake deployment utilizing AWS Organizations, only the delegated Security Lake administrator for the organization can disable Security Lake for accounts in the organization.

For example, the following AWS CLI command disables Security Lake in the `ap-northeast-1` and `eu-central-1` Regions. This example is formatted for Linux, macOS, or Unix, and it uses the backslash (\) line-continuation character to improve readability.

```
$ aws securitylake delete-data-lake \
--regions "ap-northeast-1" "eu-central-1"
```

Document history for the Amazon Security Lake User Guide

The following table describes the important changes to the documentation since the last release of Amazon Security Lake. For notification about updates to this documentation, you can subscribe to an RSS feed.

Latest documentation update: December 01, 2024

Change	Description	Date
New feature	Security Lake now supports OpenSearch Service direct query to analyze data in Security Lake. For more details, see Integration with OpenSearch Service.	December 1, 2024
New service-linked role	We added a new service-linked role `AWSServiceRoleForSecurityLakeResourceManagement`. This service-linked role provides permissions to Security Lake to perform ongoing monitoring and performance improvements, which can reduce latency and costs.	November 14, 2024
Regional availability	Security Lake is now available in the AWS GovCloud (US-East) and AWS GovCloud (US-West) AWS Regions. For a complete list of Regions where Security	June 10, 2024

	Lake is currently available, see Amazon Security Lake endpoints in the *AWS General Reference*.	
Update to existing managed policy	We added AWS WAF actions to the AWS managed policy for the `SecurityLakeServiceLinkedRole` policy. The additional actions allow Security Lake to collect AWS WAF logs, when it is enabled as a log source in Security Lake.	May 22, 2024
New AWS log source	Security Lake added AWS WAF logs as an AWS log source. AWS WAF helps you monitor web requests that end users send to applications.	May 22, 2024
Update to existing managed policy	We added SID actions to the `AmazonSecurityLakePermissionsBoundary` policy.	May 13, 2024
Update to existing managed policy	We updated the AmazonSecurityLakeMetastoreManager policy to add metadata clean up action which lets you delete the metadata in your data lake.	March 27, 2024
New source versions	Update your role permissions to ingest data from the new data source versions.	February 29, 2024

New AWS log source	Security Lake added EKS Audit Logs as an AWS log source. EKS Audit Logs help you detect potentially suspicious activities in your EKS clusters within the Amazon Elastic Kubernetes Service.	February 29, 2024
Update to existing managed policy	We updated the policy to allow `iam:PassRole` on the new `AmazonSecurityLakeMetastoreManagerV2` role and lets Security Lake deploy or update data lake components.	February 23, 2024
New managed policy	We added a new AWS managed policy, the `AmazonSecurityLakeMetastoreManager` policy. This policy grants permissions for Security Lake to manage metadata in your data lake.	January 23, 2024
Regional availability	Security Lake is now available in the following AWS Regions: Asia Pacific (Osaka), Canada (Central), Europe (Paris), and Europe (Stockholm). For a complete list of Regions where Security Lake is currently available, see Amazon Security Lake endpoints in the *AWS General Reference*.	October 26, 2023

New features	You can now edit certain settings for subscribers with query access. You can also assign tags to Security Lake resources for your AWS account.	July 20, 2023
New managed policy	Security Lake added a new AWS managed policy, the `AmazonSecurityLakeAdministrator` policy. This policy grants administrative permissions that allow a principal full access to all Security Lake actions.	May 30, 2023
General availability	Security Lake is now generally available.	May 30, 2023
New feature	Security Lake now sends metrics to Amazon CloudWatch.	May 4, 2023
Regional availability	Security Lake is now available in the following AWS Regions: Asia Pacific (Singapore), Europe (London), and South America (São Paulo).	March 22, 2023
New feature	Security Lake now creates AWS Identity and Access Management (IAM) roles on your behalf when you use the Security Lake console to enable and start using Security Lake.	February 15, 2023

Initial release	This is the initial release of the *Amazon Security Lake User Guide*.	November 29, 2022